Inner Attention based Recurrent Neural Networks for Answer Selection

Bingning Wang, Kang Liu, Jun Zhao
National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing, China
{bingning.wang, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

Attention based recurrent neural networks have shown advantages in representing natural language sentences (Hermann et al., 2015; Rocktäschel et al., 2015; Tan et al., 2015). Based on recurrent neural networks (RNN), external attention information was added to hidden representations to get an attentive sentence representation. Despite the improvement over non-attentive models, the attention mechanism under RNN is not well studied. In this work, we analyze the deficiency of traditional attention based RNN models quantitatively and qualitatively. Then we present three new RNN models that add attention information before RNN hidden representation, which shows advantage in representing sentence and achieves new state-of-art results in answer selection task.

1 Introduction

Answer selection (AS) is a crucial subtask of the open domain question answering (QA) problem. Given a question, the goal is to choose the answer from a set of pre-selected sentences (Heilman and Smith, 2010; Yao et al., 2013). Traditional AS models are based on lexical features such as parsing tree edit distance. Neural networks based models are proposed to represent the meaning of a sentence in a vector space and then compare the question and answer candidates in this hidden space (Wang and Nyberg, 2015; Feng et al., 2015), which have shown great success in AS. However, these models represent the question and sentence separately, which may ignore the information subject to the question when representing the answer. For example, given a candidate answer:

Michael Jordan abruptly retired from Chicago Bulls before the beginning of the 1993-94 NBA season to pursue a career in baseball.

For a question: **When did Michael Jordan retired from NBA?** we should focus on *the beginning of the 1993-94* in the sentence; however, when we were asked: **Which sports does Michael Jordan participates after his retirement from NBA?** we should pay more attention to *pursue a career in baseball*.

Recent years, attention based models are proposed in light of this purpose and have shown great success in many NLP tasks such as machine translation (Bahdanau et al., 2014; Sutskever et al., 2014), question answering (Sukhbaatar et al., 2015) and recognizing textual entailments (Rocktäschel et al., 2015). When building the representation of a sentence, some attention information is added to the hidden state. For example, in attention based recurrent neural networks models (Bahdanau et al., 2014) each time-step hidden representation is weighted by attention. Inspired by the attention mechanism, some attention-based RNN answer selection models have been proposed (Tan et al., 2015) in which the attention when computing answer representation is from question representation.

However, in the RNN architecture, at each time step a word is added and the hidden state is updated recurrently, so those hidden states near the end of the sentence are expected to capture more information[1]. Consequently, after adding the attention information to the time sequence hidden representations, the near-the-end hidden variables will be more attended due to their comparatively abundant semantic accumulation, which may result in a biased attentive weight towards the later coming words in RNN.

In this work, we analyze this attention bias

[1] so in many previous RNN-based model use the last hidden variable as the whole sentence representation

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1288–1297,
Berlin, Germany, August 7-12, 2016.

problem qualitatively and quantitatively, and then propose three new models to solve this problem. Different from previous attention based RNN models in which attention information is added after RNN computation, we add the attention before computing the sentence representation. Concretely, the first one uses the question attention to adjust word representation (i.e. word embedding) in the answer directly, and then we use RNN to model the attentive word sequence. However, this model attends a sentence word by word which may ignore the relation between words. For example, if we were asked: *what is his favorite food?* one answer candidate is: *He likes hot dog best*. *hot* or *dog* may be not relate to the question by itself, but they are informative as a whole in the context. So we propose the second model in which every word representation in answer is impacted by not only question attention but also the context representation of the word (i.e. the last hidden state). In our last model, inspired by previous work on adding gate into inner activation of RNN to control the long and short term information flow, we embed the attention to the inner activation gate of RNN to influence the computation of RNN hidden representation. In addition, inspired by recent work called Occam's Gate in which the activation of input units are penalized to be as less as possible, we add regulation to the summation of the attention weights to impose sparsity.

Overall, in this work we make three contributions: (1) We analyze the attention bias problem in traditional attention based RNN models. (2) We propose three inner attention based RNN models and achieve new state-of-the-art results in answer selection. (3) We use Occam's Razor to regulate the attention weights which shows advantage in long sentence representation.

2 Related Work

Recent years, many deep learning framework has been developed to model the text in a vector space, and then use the embedded representations in this space for machine learning tasks. There are many neural networks architectures for this representation such as convolutional neural networks(Yin et al., 2015), recursive neural networks(Socher et al., 2013) and recurrent neural networks(Mikolov et al., 2011). In this work we propose **I**nner **A**ttention based **RNN** (**IARNN**) for answer selection, and there are two main works which we are related to.

2.1 Attention based Models

Many recent works show that attention techniques can improve the performance of machine learning models (Mnih et al., 2014; Zheng et al., 2015). In attention based models, one representation is built with attention (or supervision) from other representation. Weston et al (2014) propose a neural networks based model called Memory Networks which uses an external memory to store the knowledge and the memory are read and written on the fly with respect to the attention, and these attentive memory are combined for inference. Since then, many variants have been proposed to solve question answering problems (Sukhbaatar et al., 2015; Kumar et al., 2015). Hermann (2015) and many other researchers (Tan et al., 2015; Rocktäschel et al., 2015) try to introduce the attention mechanism into the LSTM-RNN architecture. RNN models the input sequence word-by-word and updates its hidden variable recurrently. Compared with CNN, RNN is more capable of exploiting long-distance sequential information. In attention based RNN models, after computing each time step hidden representation, attention information is added to weight each hidden representation, then the hidden states are combined with respect to that weight to obtain the sentence (or document) representation. Commonly there are two ways to get attention from source sentence, either by the whole sentence representation (which they call *attentive*) or word by word attention (called *impatient*).

2.2 Answer Selection

Answer selection is a sub-task of QA and many other tasks such as machine comprehension. Given a question and a set of candidate sentences, one should choose the best sentence from a candidate sentence set that can answer the question. Previous works usually stuck in employing feature engineering, linguistic tools, or external resources. For example, Yih et al. (2013) use semantic features from WordNet to enhance lexical features. Wang and Manning (2007) try to compare the question and answer sentence by their syntactical matching in parse trees. Heilman and Smith (Heilman and Smith, 2010) try to fulfill the matching using minimal edit sequences between their dependency parse trees. Severyn and Moschitti (2013) automate the extraction of discriminative tree-edit features over parsing trees.

While these methods show effectiveness, they might suffer from the availability of additional resources and errors of many NLP tools such as dependency parsing. Recently there are many works use deep learning architecture to represent the question and answer in a same hidden space, and then the task can be converted into a classification or learning-to-rank problem (Feng et al., 2015; Wang and Nyberg, 2015). With the development of attention mechanism, Tan et.al(2015) propose an attention-based RNN models which introduce question attention to answer representation.

3 Traditional Attention based RNN Models and Their Deficiency

The attention-based models introduce the attention information into the representation process. In answer selection, given a question $Q = \{q_1, q_2, q_3, ..., q_n\}$ where q_i is i-th word, n is the question length, we can compute its representation in RNN architecture as follows:

$$\begin{aligned} \mathbf{X} &= \mathbf{D}[q_1, q_2, ..., q_n] \\ \mathbf{h}_t &= \sigma(\mathbf{W}_{ih}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h) \\ \mathbf{y}_t &= \sigma(\mathbf{W}_{ho}\mathbf{h}_t + \mathbf{b}_o) \end{aligned} \tag{1}$$

where $\mathbf{D}$ is an embedding matrix that projects word to its embedding space in R^d; $\mathbf{W}_{ih}$, $\mathbf{W}_{hh}$, $\mathbf{W}_{ho}$ are weight matrices and $\mathbf{b}_h$, $\mathbf{b}_o$ are bias vectors; σ is active function such as $tanh$. Usually we can ignore the output variables and use the hidden variables. After recurrent process, the last hidden variable $\mathbf{h}_n$ or all hidden states average $\frac{1}{n}\sum_{t=1}^{n}\mathbf{h}_t$ is adopted as the question representation $\mathbf{r}_q$.

When modeling the candidate answer sentence with length m:$S = \{s_1, s_2, s_3, ..., s_m\}$ in attention based RNN model,

instead of using the last hidden state or average hidden states, we use attentive hidden states that are weighted by $\mathbf{r}_q$:

$$\begin{aligned} \mathbf{H}_a &= [\mathbf{h}_a(1), \mathbf{h}_a(2), ..., \mathbf{h}_a(m)] \\ s_t &\propto f_{attention}(\mathbf{r}_q, \mathbf{h}_a(t)) \\ \tilde{\mathbf{h}}_a(t) &= \mathbf{h}_a(t)s_t \\ \mathbf{r}_a &= \sum_{t=1}^{m}\tilde{\mathbf{h}}_a(t) \end{aligned} \tag{2}$$

where $\mathbf{h}_a(t)$ is hidden state of the answer at time t. In many previous work (Hermann et al., 2015; Rocktäschel et al., 2015; Tan et al., 2015), the attention function $f_{attention}$ was computed as:

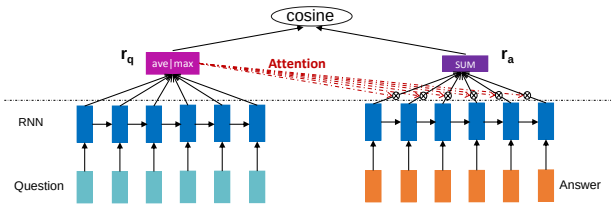

Figure 1: Traditional attention based RNN answer selection model. Dark blue rectangles represent hidden virable, $\otimes$ means gate opperation.

$$\begin{aligned} \mathbf{m}(t) &= tanh(\mathbf{W}_{hm}\mathbf{h}_a(t) + \mathbf{W}_{qm}\mathbf{r}_q) \\ f_{attention}(\mathbf{r}_q, \mathbf{h}_a(t)) &= exp(\mathbf{w}_{ms}^T\mathbf{m}(t)) \end{aligned} \tag{3}$$

$\mathbf{W}_{hm}$ and $\mathbf{W}_{qm}$ are attentive weight matrices and $\mathbf{w}_{ms}$ is attentive weight vector. So we can expect that the candidate answer sentence representation $\mathbf{r}_a$ may be represented in a question-guided way: when its hidden state $\mathbf{h}_a(t)$ is irrelevant to the question (determined by attention weight s_t), it will take less part in the final representation; but when this hidden state is relavent to the question, it will contribute more in representing $\mathbf{r}_a$. We call this type of attention based RNN model OARNN which stands for *Outer Attention based RNN* models because this kind of model adds attention information outside the RNN hidden representation computing process. An illustration of traditional attention-based RNN model is in Figure 1.

However, we know in the RNN architecture, the input words are processed in time sequence and the hidden states are updated recurrently, so the current hidden state $\mathbf{h}_t$ is supposed to contain all the information up to time t, when we add question attention information, aiming at finding the useful part of the sentence, these near-the-end hidden states are prone to be selected because they contains much more information about the whole sentence. In other word, if the question pays attention to the hidden states at time t , then it should also pay attention to those hidden states after t (i.e $\{\mathbf{h}_{t'}|t' > t\}$) as they contain the information at least as much as $\mathbf{h}_t$, but in answer selection for a specific candidate answer, the useful parts to answer the question may be located anywhere in a sentence, so the attention should also distribute uniformly around the sentence. Traditional attention-based RNN models under *attention after representation* mechanism may cause the attention to bias towards the later coming hidden states. We will analyze this attention bias problem quantita-

tively in the experiments.

4 Inner Attention based Recurrent Neural Networks

In order to solve the attention bias problem, we propose an intuition:

Attention before representation

Instead of adding attention information after encoding the answer by RNN, we add attention before computing the RNN hidden representations. Based on this intuition, we propose three inner attention based RNN models detailed below.

4.1 IARNN-WORD

As attention mechanism aims at finding useful part of a sentence, the first model applies the above intuition directly. Instead of using the original answer words to the RNN model, we weight the words representation according to question attention as follows:

$$\begin{aligned} \alpha_t &= \sigma(\mathbf{r}_q^T \mathbf{M}_{qi} \mathbf{x}_t) \\ \tilde{\mathbf{x}}_t &= \alpha_t * \mathbf{x}_t \end{aligned} \tag{4}$$

where $\mathbf{M}_{qi}$ is an attention matrix to transform a question representaion into the word embedding space. Then we use the dot value to determine the question attention strength, σ is sigmoid function to normalize the weight α_t between 0 and 1.

The above attention process can be understood as sentence distillation where the input words are distilled (or filtered) by question attention. Then, we can represent the whole sentence based on this distilled input using traditional RNN model. In this work, we use GRU instead of LSTM as building block for RNN because it has shown advantages in many tasks and has comparatively less parameter(Jozefowicz et al., 2015) which is formulated as follows:

$$\begin{aligned} \mathbf{z}_t &= \sigma(\mathbf{W}_{xz}\tilde{\mathbf{x}}_t + \mathbf{W}_{hz}\mathbf{h}_{t-1}) \\ \mathbf{f}_t &= \sigma(\mathbf{W}_{xf}\tilde{\mathbf{x}}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1}) \\ \tilde{\mathbf{h}}_t &= tanh(\mathbf{W}_{xh}\tilde{\mathbf{x}}_t + \mathbf{W}_{hh}(\mathbf{f}_t \odot \mathbf{h}_{t-1})) \\ \mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \end{aligned} \tag{5}$$

where $\mathbf{W}_{xz}, \mathbf{W}_{hz}, \mathbf{W}_{xf}, \mathbf{W}_{hh}, \mathbf{W}_{xh}$ are weight matrices and $\odot$ stands for element-wise multiplication. Finally, we get candidate answer representation by average pooling all the hidden state $\mathbf{h}_t$. we call this model IARNN-WORD as the attention is paid to the original input words. This model is shown in Figure 2.

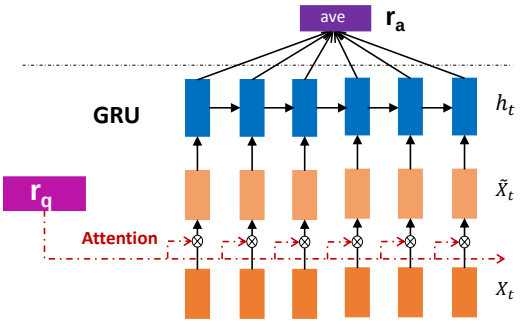

Figure 2: IARNN-WORD architecture. $\mathbf{r}_q$ is question representation.

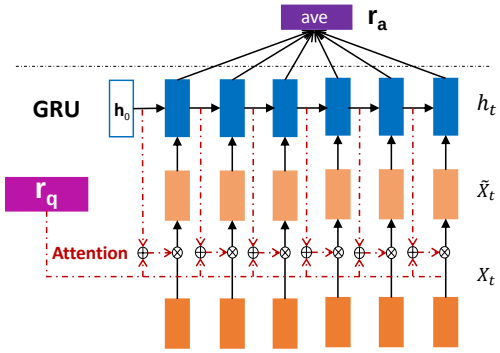

Figure 3: IARNN-CONTEXT architecture for building candidate answer sentence representation. $\mathbf{h}_0$ is added for completeness.

4.2 IARNN-CONTEXT

IABRNN-WORD attend input word embedding directly. However, the answer sentence may consist of consecutive words that are related to the question, and a word may be irrelevant to question by itself but relevant in the context of answer sentence.

So the above word by word attention mechanism may not capture the relationship between multiple words. In order to import contextual information into attention process, we modify the attention weights in Equation 4 with additional context information:

$$\begin{aligned} \mathbf{w}_C(t) &= \mathbf{M}_{hc}\mathbf{h}_{t-1} + \mathbf{M}_{qc}\mathbf{r}_q \\ \alpha_C^t &= \sigma(\mathbf{w}_C^T(t)\mathbf{x}_t) \\ \tilde{\mathbf{x}}_t &= \alpha_C^t * \mathbf{x}_t \end{aligned} \tag{6}$$

where we use $\mathbf{h}_{t-1}$ as context, $\mathbf{M}_{hc}$ and $\mathbf{M}_{qc}$ are attention weight matrices, $\mathbf{w}_C(t)$ is the attention representation which consists of both question and word context information. This additional context attention endows our model to capture relevant part in longer text span. We show this model in Figure 3.

4.3 IARNN-GATE

Inspired by the previous work of LSTM (Hochreiter and Schmidhuber, 1997) on solving the gradient exploding problem in RNN and recent work on building distributed word representation with topic information(Ghosh et al., 2016), instead of adding attention information to the original input, we can apply attention deeper to the GRU inner activation (i.e $\mathbf{z}_t$ and $\mathbf{f}_t$). Because these inner activation units control the flow of the information within the hidden stage and enables information to pass long distance in a sentence, we add attention information to these active gates to influence the hidden representation as follows:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{W}_{xz}\mathbf{x}_t + \mathbf{W}_{hz}\mathbf{h}_{t-1} + \mathbf{M}_{qz}\mathbf{r}_q) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{M}_{qf}\mathbf{r}_q) \\
\tilde{\mathbf{h}}_t &= tanh(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}(\mathbf{f}_t \odot \mathbf{h}_{t-1})) \\
\mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t
\end{aligned} \tag{7}$$

where $\mathbf{M}_{qz}$ and $\mathbf{M}_{hz}$ are attention weight matrices. In this way, the update and forget units in GRU can focus on not only long and short term memory but also the attention information from the question. The architecture is shown in Figure 4.

4.4 IARNN-OCCAM

In answer selection, the answer sentence may only contain small number of words that are related to the question. In IARNN-WORD and IARNN-CONTEXT, we calculate each word attention weight without considering total weights. Similar with Raiman(2015) who adds regulation to the input gate, we punish the summation of the attention weights to enforce sparsity. This is an application of Occam's Razor: *Among the whole words set, we choose those with fewest number that can represent the sentence*. However, assigning a pre-defined hyper-parameter for this regulation[2] is not an ideal way because it punishes all question attention weights with same strength. For different questions there may be different number of snippets in candidate answer that are required. For example, when the question type is *When* or *Who*, answer sentence may only contains a little relavant words so we should impose more sparsity on the summation of the attention. But when the question type is *Why* or *How*, there may be much more words on the sentence that are relevant to the question so we should set the regulation value small accordingly. In this work, this attention regulation is added as follows: for the specific question Q_i and its representation $\mathbf{r}_q^i$, we use a vector $\mathbf{w}_{qp}$ to project it into scalar value n_p^i, and then we add it into the original objective J_i as follows:

$$\begin{aligned}
n_p^i &= \max\{\mathbf{w}_{qp}^T \mathbf{r}_q^i, \lambda_q\} \\
J_i^* &= J_i + n_p^i \sum_{t=1}^{mc} \alpha_t^i
\end{aligned} \tag{8}$$

where α_t^i is attention weights in Equation 4 and Equation 6. λ_q is a small positive hyper-parameter. It needs to mention that we do not regulate IARNN-GATE because the attention has been embedded to gate activation.

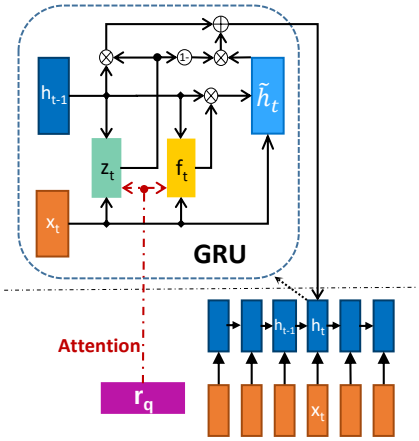

Figure 4: IABRNN-GATE architecture. We show one time step GRU inner state process within the blue dotted line.

5 Experiments

5.1 Quantify Traditional Attention based Model Bias Problem

In order to quantify the outer attention based RNN model's attention bias problem in Section 3, we build an outer attention based model similar with Tan (2015). First of all, for the question we build its representation by averaging its hidden states in LSTM, then we build the candidate answer sentence representation in an attentive way introduced in Section 3. Next we use the cosine similarity to compare question and answer representation similarity. Finally, we adopt max-margin hinge loss as objective:

$$\begin{aligned}
L = max\{0, M - cosine(\mathbf{r}_q, \mathbf{r}_{a_+}) \\
+ cosine(\mathbf{r}_q, \mathbf{r}_{a_-})\}
\end{aligned} \tag{9}$$

[2]For example, in many machine learning problem the original objective sometimes followed with a L_1 or L_2 regulation with hyper-parameter $\lambda 1$ or $\lambda 2$ to control the tradeoff between the original objective J and the sparsity criterion:$J^* = J + (\lambda 1 | \lambda 2) \sum(L_1 | L_2 norm)$

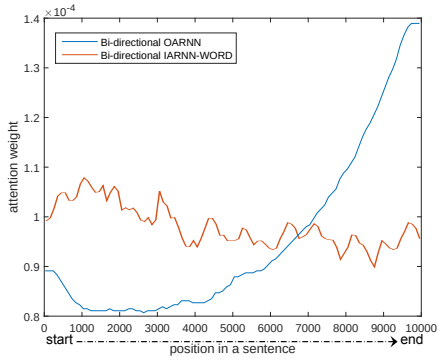

Figure 5: One directional OARNN attention distribution, the horizontal axis is position of word in a sentence that has been normalized from 1 to 10000.

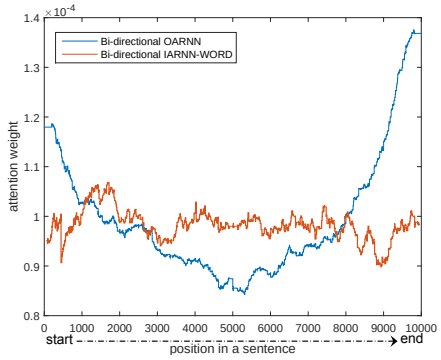

Figure 6: Bi-directional OARNN attention distribution, the horizontal axis is the postion of the word in a sentence that has been normalized from 1 to 10000.

where a_+ is ground truth answer candidate and a_- stands for negative one, the scalar M is a predefined margin. When training result saturates after 50 epoches, we get the attention weight distribution (i.e. s_q in Equation 2). The experiment is conducted on two answer selection datasets: WikiQA (Yang et al., 2015) and TrecQA (Wang et al., 2007). The normalized attention weights is reported in Figure 5.

However, the above model use only forward LSTM to build hidden state representation, the attention bias problem may attribute to the biased answer distribution: the useful part of the answer to the question sometimes may located at the end of the sentence. So we try OARNN in bidirectional architecture, where the forward LSTM and backward LSTM are concatenated for hidden representation, The bidirectional attention based LSTM attention distribution is shown in Figure 6.

Analysis: As is shown in Figure 5 and 6, for one-directional OARNN, as we move from beginning to the end in a sentence, the question attention gains continuously; when we use bidirectional OARNN, the hidden representations near two ends of a sentence get more attention. This is consistent with our assumption that for a hidden representation in RNN, the closer to the end of a sentence, the more attention it should drawn from question. But the relevant part may be located anywhere in a answer. As a result, when the sample size is large enough[3], the attention weight should be unformly distributed. The traditional *attention after representation* style RNN may suffer from the biased attention problem. Our IARNN models are free from this problem and distribute nearly uniform (orange line) in a sentence.

5.2 IARNN evaluation

Common Setup: We use the off-the-shelf 100-dimension word embeddings from word2vec[4], and initiate all weights and attention matrices by fixing their largest singular values to 1 (Pascanu et al., 2013). IARNN-OCCAM base regulation hyper-parameter λ_q is set to 0.05, we add L_2 penalty with a coefficient of 10^{-5} . Dropout (Srivastava et al., 2014) is further applied to every parameters with probability 30%. We use Adadelta(Zeiler, 2012) with $\rho = 0.90$ to update parameters.

We choose three datasets for evaluation: InsuranceQA, WikiQA and TREC-QA. These datasets contain questions from different domains. Table 1 presents some statistics about these datasets. We adopt a max-margin hinge loss as training objective. The results are reported in terms of MAP and MRR in WikiQA and TREC-QA and accuracy in InsuranceQA.

We use bidirectional GRU for all models. We share the GRU parameter between question and answer which has shown significant improvement on performance and convergency rate (Tan et al., 2015; Feng et al., 2015).

There are two common baseline systems for above three datasets:

- **GRU**: A non-attentive GRU-RNN that models the question and answer separately.
- **OARNN**: Outer attention-based RNN models (OARNN) with GRU which is detailed in Section 5.1.

WikiQA (Yang et al., 2015) is a recently released open-domain question answering

[3] 10000 for WikiQA and 5000 for TrecQA in experiment.
[4] https://code.google.com/archive/p/word2vec/

Dataset(train / test / dev)	InsuranceQA	WikiQA	TREC-QA
# of questions	12887 / 1800x2 /1000	873 / 243 / 126	78 / 68 / 65
# of sentences	24981(ALL)	20360 / 6165 / 2733	5919 / 1442 / 1117
Ave length of question	7.16	7.16 / 7.26 / 7.23	11.39 / 8.63 / 8.00
Ave length of sentence	49.5	25.29 / 24.59 / 24.59	30.39 / 25.61 / 24.9

Table 1: The statistics of three answer selection datasets. For the TREC-QA, we use the cleaned dataset that has been edit by human. For WikiQA and TREC-QA we remove all the questions that has no right or wrong answers.

System	**MAP**	**MRR**
(Yang et al., 2015)	0.652	0.6652
(Yin et al., 2015)	0.6921	0.7108
(Santos et al., 2016)	0.6886	0.6957
GRU	0.6581	0.6691
OARNN	0.6881	0.7013
IARNN-word	0.7098	0.7234
IARNN-Occam(word)	0.7121	0.7318
IARNN-context	0.7182	0.7339
IARNN-Occam(context)	**0.7341**	**0.7418**
IARNN-Gate	0.7258	0.7394

Table 2: Performances on WikiQA

System	Dev	Test1	Test2
(Feng et al., 2015)	65.4	65.3	61.0
(Santos et al., 2016)	66.8	67.8	60.3
GRU	59.4	53.2	58.1
OARNN	65.4	66.1	60.2
IARNN-word	67.2125	67.0651	61.5896
IARNN-Occam(word)	69.9130	69.5923	63.7317
IARNN-context	67.1025	66.7211	63.0656
IARNN-Occam(context)	69.1125	68.8651	**65.1396**
IARNN-Gate	**69.9812**	**70.1128**	62.7965

Table 3: Experiment result in InsuranceQA, (Feng et al., 2015) is a CNN architecture without attention mechanism.

System	**MAP**	**MRR**
(Wang and Nyberg, 2015) †	0.7134	0.7913
(Wang and Ittycheriah, 2015) †	0.7460	0.8200
(Santos et al., 2016) †	**0.7530**	**0.8511**
GRU	0.6487	0.6991
OARNN	0.6887	0.7491
IARNN-word	0.7098	0.7757
IARNN-Occam(word)	0.7162	0.7916
IARNN-context	0.7232	0.8069
IARNN-Occam(context)	0.7272	0.8191
IARNN-Gate	0.7369	0.8208

Table 4: Result of different systems in Trec-QA.(Wang and Ittycheriah, 2015) propose a question similarity model to extract features from word alignment between two questions which is suitable to FAQ based QA. It needs to mention that the system marked with † are learned on TREC-QA original full training data.

dataset in which all answers are collected from Wikipedia. In addition to the original (question,positive,negative) triplets, we randomly select a bunch of negative answer candidates from answer sentence pool and finally we get a relatively abundant 50,298 triplets. We use cosine similarity to compare the question and candidate answer sentence. The hidden variable's length is set to 165 and batch size is set to 1. We use *sigmoid* as GRU inner active function, we keep word embedding fixed during training. Margin M was set to 0.15 which is tuned in the development set. We adopt three additional baseline systems applied to WikiQA: (1) A bigram CNN models with average pooling(Yang et al., 2015). (2) An attention-based CNN model which uses an interactive attention matrix for both question and answer(Yin et al., 2015) [5] (3) An attention based CNN models which builds the attention matrix after sentence representation(Santos et al., 2016). The result is shown in Table 2.

InsuranceQA (Feng et al., 2015) is a domain specific answer selection dataset in which all questions is related to insurance. Its vocabulary size is comparatively small (22,353), we set the batch size to 16 and the hidden variable size to 145, hinge loss margin M is adjusted to 0.12 by evaluation behavior. Word embeddings are also learned during training. We adopt the Geometric mean of Euclidean and Sigmoid Dot (GESD) proposed in (Feng et al., 2015) to measure the similarity between two representations:

$$GESD(x,y) = \frac{1}{1+||x-y||} \times \frac{1}{1+exp(-\gamma(xy^T+c))} \quad (10)$$

which shows advantage over cosine similarity in experiments.

We report accuracy instead of MAP/MRR because one question only has one right answers in InsuranceQA. The result is shown in Table 3.

TREC-QA was created by Wang et al.(2007) based on Text REtrieval Conference (TREC) QA track (8-13) data. The size of hidden variable was set to 80, M was set to 0.1. This dataset is comparatively small so we set word embedding vector

[5] In their experiment some extra linguistic features was also added for better performance.

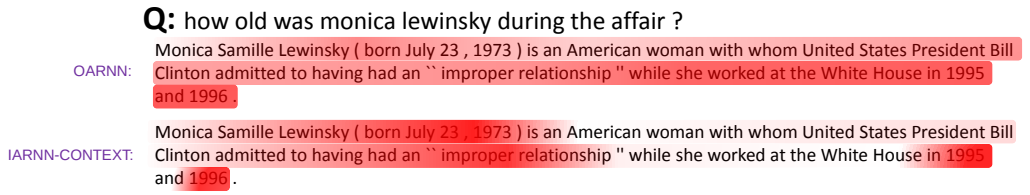

Figure 7: An example demonstrates the advantage of IARNN in capturing the informed part of a sentence compared with OARNN.

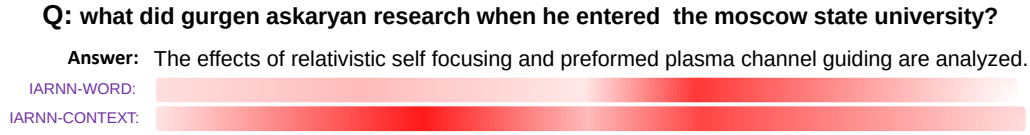

Figure 8: An example illustrates the IARNN-CONTEXT could attend the consecutive words in a sentence.

size to 50 and update it during training. It needs to mention that we do not use the original TREC-QA training data but the smaller one which has been edited by human. The result is shown in Table 4.

6 Result and Analysis

We can see from the result tables that the attention based RNN models achieve better results than the non-attention RNN models (GRU). OARNN and IARNN beat the non-attentive GRU in every datasets by a large margin, which proves the importance of attention mechanism in representing answer sentence in AS. For the non-attentive models, the fixed width of the hidden vectors is a bottleneck for interactive information flow, so the informative part of the question could only propagate through the similarity score which is blurred for the answer representation to be properly learned. But in attention based models, the question attention information is introduced to influence the answer sentence representation explicitly, in this way we can improve sentence representation for the specific target (or topic (Ghosh et al., 2016)).

The inner attention RNN models outperform outer attention model in three datasets, this is corresponds to our intuition that the bias attention problem in OARNN may cause a biasd sentence representation. An example of the attention heatmap is shown in Figure7. To answer the question, we should focus on *"born July 23 , 1973"* which is located at the beginning of the sentence. But in OARNN, the attention is biases towards the last few last words in the answer. In IARNN-CONTEXT, the attention is paid to the relevant part and thus results in a more relevant representation.

The attention with context information could also improves the result, we can see that IARNN-CONTEXT and IARNN-GATE outperform IARNN-WORD in three experiments. IARNN-WORD may ignore the importance of some words because it attends answer word by word, for example in Figure8, the specific word *self* or *focusing* may not be related to the question by itself, but their combination and the previous word *relativistic* is very informative for answering the question. In IARNN-CONTEXT we add attention information dynamically in RNN process, thus it could capture the relationship between word and its context.

In general, we can see from table3-5 that the IARNN-GATE outperforms IARNN-CONTEXT and IARNN-WORD. In IARNN-WORD and IARNN-CONTEXT, the attention is added to impact each word representation, but the recurrent process of updating RNN hidden state representations are not influenced. IARNN-GATE embeds the attention into RNN inner activation, the *attentive activation gate* are more capable of controlling the attention information in RNN. This enlights an important future work: we could add attention information as an individual activation gate, and use this additional gate to control attention information flow in RNN. The regulation of the attention weights (Occam's attention) could also improve the representation. We also conduct an experiment on WikiQA (training process) to measure the Occam's attention regulation on different type of questions. We use rules to classify question into 6 types

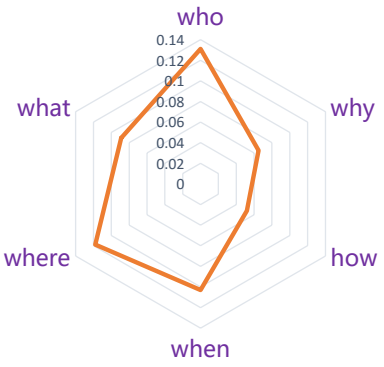

Figure 9: The Occam's attention regulation on different types of question.

(i.e. who,why,how,when,where,what), and each of them has the same number of samples to avoid data imbalance. We report the Occam'm regulation (n_p^i in Equation.8) in Figure 9. As we can see from the radar graph, *who* and *where* are regulized severely compared with other types of question, this is correspond to their comparetively less information in the answer candidate to answer the question. This emphasize that different types question should impose different amount of regulation on its candidate answers. The experiment result on three AS datasets shows that the improvement of Occam's attention is significant in WikiQA and insuranceQA. Because most of the sentence are relatively long in these two datasets, and the longer the sentence, the more noise it may contain, so we should punish the summation of the attention weights to remove some irrelevant parts. Our question-specific Occam's attention punishes the summation of attention and thus achieves a better result for both IARNN-WORD and IARNN-CONTEXT.

7 Conclusion and Future Work

In this work we present some variants of traditional attention-based RNN models with GRU. The key idea is *attention before representation*. We analyze the deficiency of traditional outer attention-based RNN models qualitatively and quantitatively. We propose three models where attention is embedded into representation process. Occam's Razor is further implemented to this attention for better representation. Our results on answer selection demonstrate that the inner attention outperforms the outer attention in RNN. Our models can be further extended to other NLP tasks such as recognizing textual entailments where attention mechanism is important for sentence representation. In the future we plan to apply our *inner-attention* intuition to other neural networks such as CNN or multi-layer perceptron.

Acknowledgments

The work was supported by the Natural Science Foundation of China (No.61533018), the National High Technology Development 863 Program of China (No.2015AA015405) and the National Natural Science Foundation of China (No.61272332). And this research work was also supported by Google through focused research awards program.

References

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

Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.

Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. 2016. Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*.

Michael Heilman and Noah A Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1684–1692.

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

Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. 2015. An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2342–2350.

Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask me anything: Dynamic memory networks for natural language processing. *arXiv preprint arXiv:1506.07285*.

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Honza Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE.

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. 2014. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, pages 2204–2212.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1310–1318.

Jonathan Raiman and Szymon Sidor. 2015. Occam's gates. *arXiv preprint arXiv:1506.08251*.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

Cicero dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *arXiv preprint arXiv:1602.03609*.

Aliaksei Severyn and Alessandro Moschitti. 2013. Automatic feature engineering for answer selection and extraction. In *EMNLP*, pages 458–467.

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

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, pages 2431–2439.

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.

Ming Tan, Bing Xiang, and Bowen Zhou. 2015. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.

Zhiguo Wang and Abraham Ittycheriah. 2015. Faq-based question answering via word alignment. *arXiv preprint arXiv:1507.02628*.

Di Wang and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in question answering. *ACL, July*.

Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *EMNLP-CoNLL*, volume 7, pages 22–32.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Citeseer.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *HLT-NAACL*, pages 858–867. Citeseer.

Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *Proceedings of ACL*.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193*.

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

Yin Zheng, Richard S Zemel, Yu-Jin Zhang, and Hugo Larochelle. 2015. A neural autoregressive approach to attention-based recognition. *International Journal of Computer Vision*, 113(1):67–79.