Attention Based Recurrent Neural Networks for Online Advertising

Shuangfei Zhai [†], Keng-hao Chang [††], Ruofei Zhang [††], Zhongfei Zhang [†]
[†]Binghamton University, Binghamton, NY, USA
[††]Microsoft, Sunnyvale, CA, USA
[†]{szhai2,zhongfei}@binghamton.edu
[††]{kenchan,bzhang}@microsoft.com

ABSTRACT

We investigate the use of recurrent neural networks (RNNs) in the context of online advertising, where we use RNNs to map both query and ads to real valued vectors. In addition, we propose an attention network that assigns scores to different word locations according to their intent importance. The vector output is computed by a weighted sum of the vectors at each word. We perform end-to-end training of both the RNN and attention network under the guidance of user click logs. We show that the attention network improves the quality of learned vector representations evaluated by AUC on a manually labeled dataset. Moreover, we show that keywords extracted according to the attention scores are easy to interpret and significantly outperform the state-of-the-art query intent extraction methods.

Categories and Subject Descriptors

H.3.3 [**Information Storage And Retrieval**]: Information Search and Retrieval

Keywords

Online Advertising; Deep Learning; RNN; Attention

1. INTRODUCTION

In this paper, we address two important problems in sponsored search: 1) compact representations of queries and ads, 2) identifying keywords from queries and ads. The two problems are of central importance to matching ads to user search intent for a search engine. To this end, we propose a machine learning model which consists of three parts: 1) a recurrent neural network (RNN) mapping a sequence of words to a sequence of vectors, 2) an attention based pooling layer which compresses a sequence to a single vector, 3) utilizing user click logs to match queries to ads. Our model considers the local context of each word, and dynamically summarizes a sequence to a vector by putting large weights on important words, which enables it to outperform existing methods.

2. MODEL

2.1 Recurrent Neural Networks

RNNs belong to a family of neural networks that model sequential data. Given the current input $x_t \in R^d$ and history hidden state $h_{t-1} \in R^k$, an RNN computes the current hidden state $h_t \in R^k$ as follows:

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h), \tag{1}$$

where σ is a nonlinear function applied element-wise to a vector; $W_{xh} \in R^{k\times d}$, $W_{hh} \in R^{k\times k}$, $b_h \in R^k$ are the model parameters. It is also possible to build a bidirectional RNN (BRNN) by concatenating the output of a forward RNN $\overrightarrow{h}_t$ and that of a backward RNN $\overleftarrow{h}_t$: $h_t = concatenate(\overrightarrow{h}_t, \overleftarrow{h}_t)$.

2.2 Attention Based Pooling

Given a sequence $\{x_1, ..., x_T\}$, the output of an RNN is another sequence of vectors $\{h_1, ..., h_T\}$. As text sequences naturally come with variable lengths, it is desirable to further compress the sequence into a fixed-length vector. In machine learning, this step is often referred to as (global) pooling. Commonly used approaches include mean pooling: $\frac{1}{T}\sum_{t=1}^{T} h_t$, last pooling (using the last hidden state): h_T, and element-wise max pooling: $\max_{t=1}^{T} h_t$. However, these methods are static and unable to make distinction between interesting and uninteresting words. To this end, we accordingly propose an attention based pooling as follows:

$$h = \sum_{t=1}^{T} a_t h_t, \ a_t = \frac{\exp(s(h_t;\theta))}{\sum_{t=1}^{T}\exp(s(h_t;\theta))}. \tag{2}$$

Here the single vector output for a sequence of vectors is computed as a weighted sum of all the steps. To compute the weight a_t, which we call *attention*, we first pass h_t through a function $s(\cdot;\theta)$ then normalize $s(h_t;\theta)$ across the whole sequence with the softmax function. In this way, with a properly chosen $s(\cdot;\theta)$, one is able to assign higher attention to words that are interesting w.r.t. a certain task, and ignore those uninteresting ones. In this work, $s(\cdot;\theta)$ is implemented as another neural network, whose parameters are jointly learned with those of the RNN. In the sequel, we refer to $s(\cdot;\theta)$ as the attention network.

2.3 Loss Function

We follow [1] to use user click logs as implicit supervision. The idea is that we should learn representations for queries and ads such that prediction of user click behavior is easy. Let (q, d^+) be a clicked (query, ad) pair, and $\{d_1^-, ..., d_n^-\}$ be

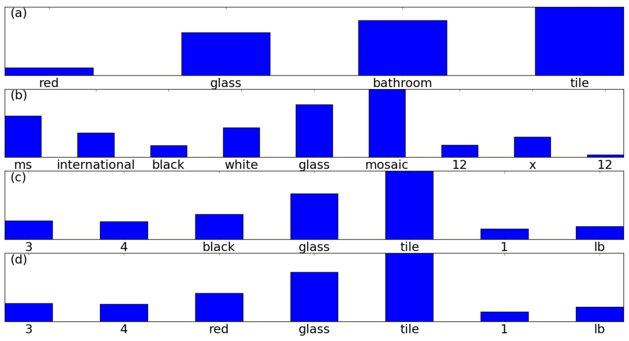

Figure 1: Attention score visualization of a query (a) and 3 top ranked ads (b)(c)(d).

a set of ads that q has not clicked on. We denote $h_q(\cdot)$ and $h_d(\cdot)$ as the encoding function according to Equation 2 for query and ad sequence, respectively. We then formulate our loss function as follows:

$$- \sum_{(q,d^+)} \log \frac{\exp(S(q,d^+))}{\exp(S(q,d^+)) + \sum_{i=1}^{n} \exp(S(q,d_i^-))} \tag{3}$$
$$s.t.\ S(q,d) = h_q(q)^T h_d(d).$$

In Equation 3, for each clicked (query, ad) pair (q, d^+), we randomly sample n ads $\{d_1^-, ..., d_n^-\}$ that are not clicked by q; we then form a pseudo $n+1$ way classification problem where d^+ is considered the positive class, and all the rest are negative. The probability is computed according to the inner product of the vector representation of the query with all those of the ads. We then directly apply gradient descent to minimize Equation 3 w.r.t. the parameters of the RNN and the attention network.

3. EXPERIMENTS

3.1 Training Dataset

We sample user click logs from a commercial "product ads" search engine as training data, where product-centered ads are presented to user queries with product intent. A total of 15 million clicks are sampled from a month long of click logs, which ends up with 6.4 million distinct user queries and 5.1 million distinct clicked ads.

3.2 Evaluation of Learned Representations

We use a fully labeled test set to evaluate the learned representations of queries and ads. The test set contains about 915 thousands of (query, ad) pairs sampled from the early selection stage of a commercial ads search engine, with each pair manually labeled by human judges from the label set {bad, fair, good, excellent} for query-ad relevancy. We use AUC of ROC as the metrics for evaluation, by considering the bad label as the negative class and the rest labels as the positive class (fair, good and excellent).

For comparison, we choose the encoding methods from 1) Bag of Words (BoW) with linear projection 2) RNN 3) BRNN and try max, last and attention pooling, respectively [1]. For all the models, we set the dimensionality of projection

[1]All the models are implemented with Theano and trained on an Nvidia Tesla K20 GPU.

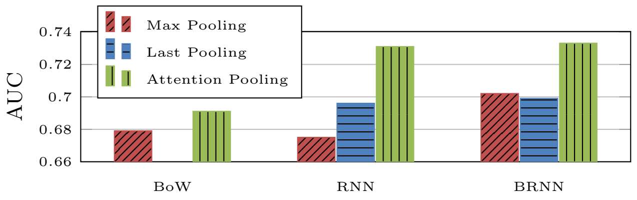

Figure 2: AUC evaluated on the test set of di erent models, BRNN + attention achieves the best result.

Table 1: The comparison of our method (attention) with PartialDrop (PD) and SmartIntent (SI).

Method	#Good	%Good	Method	#Good	%Good
Ours	228	0.275	Ours	212	0.286
PD	192	0.222	SI	170	0.225

to 400. We summarize the AUC of all the combinations in Figure 2. We clearly see that attention outperforms others on all encoding architectures. Also, BRNN works best as the encoder, followed by RNN; all encoders that consider time sequence outperform Bag of Words (BoW). We also visualize the attention scores of a query and three ads given by BRNN + attention in Figure 1.

3.3 Evaluation of Attention Scores

We now evaluate the quality of the attention scores. Given a query, we extract the words that are assigned high attention scores and compose them as a sub-query. This step is also referred to as query rewriting. We ask human judges to label the relationship of the (query, rewrite) pairs into five categories: {same, superset, subset, overlap, disjoint}. We consider a rewrite of good quality when the intent space of the original query is the same as or a superset of it, and of bad quality otherwise. We sample 1000 queries with product intent and compare with two existing algorithms, namely Smart Intent [2] and Partial Drop [3]. For each sub-query that PartialDrop or SmartIntent generates, we generate an attention-score based counterpart: a sub-query with the same length as the baseline and with words having the highest attention scores from a BRNN. We summarize the results in Table 1. We see that our method significantly outperforms both the two baselines, yielding much more good quality rewrites.

4. CONCLUSIONS

We have proposed an attention based RNN for modeling queries and ads data in the context of sponsored search. We show that our proposed model is able to learn effective vector representations as well as extract good quality sub-queries at the same time.

5. REFERENCES

[1] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. In *CIKM 2013*.
[2] P. Liu, J. Azimi, and R. Zhang. Contextual query intent extraction for paid search selectio. In *WWW 2015*.
[3] K. T. Maxwell and W. B. Croft. Compact query term selection using topically related text. In *SIGIR 2013*.