Rumor detection with RNN

Yuchen Fang 515030910563

Abstract

Nowadays social platforms like weibo and twitter are becoming perfect places for rumor to spread. In this project I present a method to model a sequence of posts about an event and judge whether it is a rumor.The model is based on recurrent neural networks(RNN) with attention mechanism to capture long range dependence of weibo posts.

1 Introduction

With the fast development of microblogging platforms, rumors can spread fast and widely. It has become a crucial problem to detect rumor automatically. It is a natural choice to use deep learning to do this job when you have access to massive amounts of data. On microblogging platforms, the spread of rumor naturally forms a sequence structure composed of user posting and reposting. We are able to model such structure with deep learning by using rnn structure. However, the lengths of microblogging posts about an event are often large and various. In this project, I use rnn with attention structure to model sequence of posts with large length. The model can learn both the temporal and textual representations from posts compared to former method with hand-crafted features.

The experiment is mainly based on Weibo data. We have posts about 4664 events and each event contain thousands of posts. The experiment shows a better performance than previous method.

2 Methodology

In this section, I will describe the structure of our model and the data processing method we use for the experiment.

2.1 Recurrent neural network

Leveraging RNN for sequential modeling and time-series prediction has been widely applied in information retrieval systems. A basic RNN can be formalized as follows: given an input sequence $(x_1, x_2, ..., x_T)$, and for each timestamp, the model update the hidden state $(h_1, h_2, ..., h_T)$ and output $(o_1, o_2, ..., o_T)$. Figure 2.1 shows a basic unfolded structure of RNN.

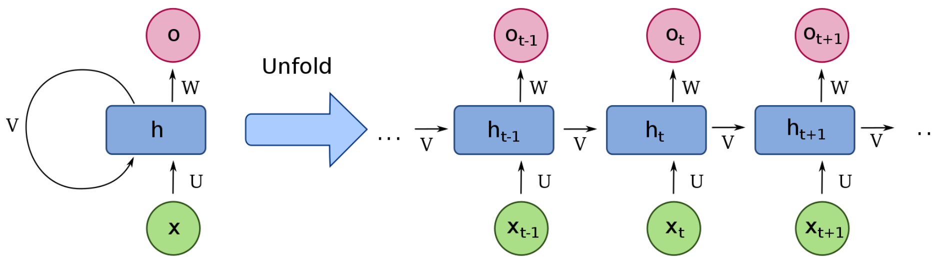

Figure 1: Basic unfolded RNN

Under the scene of rumor detecting, we take the last output o_T as the final judgment of whether an event is a rumor. In many situations, this result depend heavily on the information source, which means the early stage input of the model.In theory, RNNs are absolutely capable of handling such "long-term dependencies", however, due to reasons such as gradient vanish [2, 7]. A new RNN structure called LSTM [4] was presented to solve this problem.

2.2 LSTM

Long Short Term Memory networks(LSTM) are a special kind of RNN, capable of learning long-term dependencies. First introduced by Hochreiter & Schmidhuber (1997) [4]. In LSTM, the hidden layer in basic RNN is replaced by an LSTM cell. The structure of the cell is shown in Figure2.2

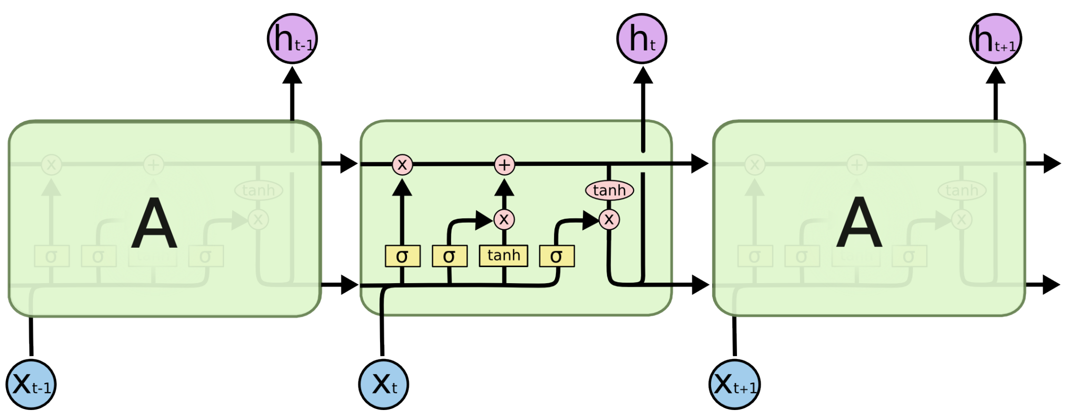

Figure 2: The structure of LSTM

It consists of three "gate" structure. The input gate determines what new information should be added to the cell state. The forget gate decide what information should be thrown away from the cell state. And the output gate decide what we are going to output. With such structure, LSTM is able to learn long-term dependencies.

2.3 Posts Segmentation

As we mentioned before, the numbers of posts about different events can be huge and varied, so we need to divide all the posts to several segmentations with similar length. We tried some methods and the algorithm described in Ma Jing et.al. [6]. The algorithm is described below.2.3

Input:

Relevant posts of one event $E_i = \{(m_{i,j}, t_{i,j})\}_{j=1}^{n}$

Reference length of RNN $\mathcal{N}$

Output:

Time intervals $I = I_1, I_2...$

$L(i) = t_{i,n_i} - t_{i,1}; l = \frac{L(i)}{\mathcal{N}}; k = 0$
while true **do**
  k++;
  $U_k \leftarrow Equipartition(L(i), l);$
  $U_0 \leftarrow \{emptyintervals\}\ subseteqU_k;$
  $U'_k \leftarrow U_k - U_0;$
  Find $\bar{U}_k \subseteq U'_k$ such that $\bar{U}_k$ contains continuous intervals that cover the longest time span;
  if $|\bar{U}_k| < \mathcal{N}$ && $|\bar{U}_k| > |\bar{U}_{k-1}|$ **then**
    $l = 0.5 * l;$
  else
    $I = \{I_1, I_2, I_3...|I_o \in \bar{U}_k\}$
    return I;
  end if
end while
return I;

This algorithm divide the posts into segmentations with same time intervals and the number of segmentations is around $\mathcal{N}$

2.4 Vectorization

I used term frequency–inverse document frequency(TFIDF) value to vectorize the paragraphs I got after segmentation. It is a modified word bagging method which takes inverse document frequecy into consideration.The TFIDF value of an word in a document is

$$TFIDF(w, d) = tf(w, d) \times idf(w, D)$$

where

$$tf(w, d) = 0.5 + 0.5 \cdot \frac{f_{w,d}}{max\{f_{t',d} : t' \in d\}}$$

$$itf(w, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

I took the 5000 words with max TFIDF values as the dictionary, so every segmentation is vectorized into a 5000-dimension vector.

2.5 Attention

Attention mechanism is widely used with CNN to deal with image recognition problems [3,5,8]. Attention is recently found useful in NLP missions such as machine translation [1] combined with seq2seq RNN models. RNN with attention mechanism is able to discover longer dependencies compaired to basic LSTM. I used a relatively easy attention structure in my model.

$$c_t = tanh(h_t W_c + x_T U_c)$$
$$a = softmax(c)$$
$$\bar{h} = \sum_{t=1}^{T} a_t * h_t$$
$$o = \sigma(\bar{h} W_o + b_o)$$

Where h_t is the hidden layer value at timestamp t, x_T is the last input, which can be seen as an "conclusion" of the whole sequence.

The final model structure is shown in Figure2.5.

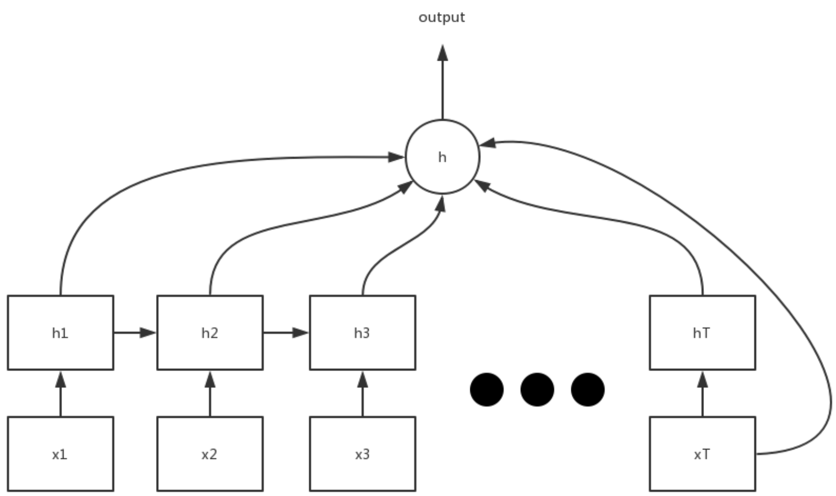

Figure 3: The structure of our model

3 Experiments and results

3.1 Data collection

We use the dataset offered in Ma Jing et.al. [6]. It is crawled from Weibo. Basic description is listed in Table1

Table 1: The dataset of Weibo

Statistic	Weibo
Users #	2,746,818
Posts #	3,805,656
Events #	4664
Rumors #	2313
Non-Rumors #	2351
Avg. time length/event	2,460.7 hours
Avg. # of posts/event	816
Max # of posts/event	59,318
Min # of posts/event	0

3.2 Experiment result

Our Baselines are Simple LSTM and multilayer LSTM. The experiment result is shown in Table2

Table 2: Rumor detection results

Model	Accuracy	AUC	F1
Simple LSTM	0.9048	0.9640	0.9024
Multilayer LSTM	0.9119	0.9668	0.9084
RNN with attention	0.9238	0.9728	0.9223

As we can see, RNN with attention get 2% higher accuracy than Simple LSTM and 1% higher than Multilayer LSTM. Considering the accuracy of baselines are over 90%, the increase is significant. For the AUC metric, the relatively high result shows that the result our model get is more robust than the baselines.

4 Conclusion

In this project, I added a modified attention structure to RNN to detect rumor in social medias and reached a state-of-the-art result in experiment.The following are the future improvements:

1. Experiment on more large dataset may be more convincing
2. Further study can be done on attention model. The learned attention value can tell us some information between specific words and rumor. Such information may help social networks improve their keyword detection technology.

Finally, I'd like to thank professor Xingbing Wang and Luoyi Fu for the class Mobile Network. I learned a lot in this class thanks to their guidance.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. pages 1–15, 2014.

[2] Y Bengio, P Simard, and P Frasconi. Learning Long Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[3] Misha Denil, Loris Bazzani, Hugo Larochelle, and Nando de Freitas. Learning where to attend with deep architectures for image tracking. *Neural Computation*, 24(8):2151–2184, 2012.

[4] Sepp Hochreiter and Jj Urgen Schmidhuber. LONG SHORT-TERM MEMORY. *Neural Computation*, 9(8):1735–1780, 1997.

[5] Hugo Larochelle and Geoffrey Hinton. Learning to combine foveal glimpses with a third-order Boltzmann machine. *Nips-2010*, pages 1243–1251, 2010.

[6] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-fai Wong, and Meeyoung Cha. Detecting Rumors from Microblogs with Recurrent Neural Networks. pages 3818–3824, 2015.

[7] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training Recurrent Neural Networks.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. 2017.