

# Memory Networks for Fake News Detection

Pham Trung Tin and Nguyen Le Minh

Japan Advanced Institute of Science and Technology  
School of Information Science  
1-1, Asahidai, Nomi, Ishikawa, Japan  
tinpt@jasit.ac.jp, nguyenml@jaist.ac.jp

**Abstract.** Fake news detection is a task of determining whether a piece of news is true or false, or in a more challenging setting, classifying news according to its credibility. In this paper, we address this problem using memory networks. While existing approaches apply deep neural networks such as Long Short-Term Memory Networks (LSTMs) along with attention mechanisms from side information, we take a further step which employs memory networks to automatically learn external information from text, and leverage both learned and side information to detect fake news. Experimental results demonstrate that our memory network outperforms the current state-of-the-art by 5.2% of accuracy on the benchmark dataset.

## 1 Introduction

Fake news is a type of news that has no basis in fact, but is presented as being factually accurate<sup>1</sup>. It may have misleading, false, imposter, manipulated, fabricated content, or satire, parody, and false connection with the intent to mislead people. As such, fake news may have economic or social impacts. In fact, President Donald Trump spoke in a TV program that “The unemployment rate may be as high as 42 percent”, while the true number was just approximately 16.4 percent<sup>2</sup>. This claim aimed to inflate the threat of unemployment, which may project a false impression and leave some people pessimistic about job opportunities in the U.S. Another example of fake news was that grapefruits could cause cancer. This unfounded allegation circulated through a number of Vietnamese newspapers in 2007, costing hundreds of billion damage since the price of grapefruits plummeted to only 10 % of the current one<sup>3</sup>. The related newspaper agencies were later charged with getting involved in spreading this fake news.

Due to its devastating implications, detecting fake news has been a hot topic recently. Most of the works make use of deep neural networks owing to their success in a wide range of natural language processing applications. Similar to

---

<sup>1</sup> [https://en.wikipedia.org/wiki/Fake\\_news](https://en.wikipedia.org/wiki/Fake_news)

<sup>2</sup> <http://www.politifact.com/truth-o-meter/statements/2015/sep/30/donald-trump/donald-trump-says-unemployment-rate-may-be-42-perc/>

<sup>3</sup> <https://tuoitre.vn/tin-don-an-buoi-bi-ung-thu-lam-thiet-hai-hang-tram-ti-216359.htm>

other tasks, deep neural networks used in detecting fake news by [1] stress on automatic representation learning for a given input. Furthermore, multiple attentions embedded in a hybrid LSTM by [2] allowed for selective emphasis on some relevant subparts of the input, which was proved to effectively boost the detection performance.

Following this trend, our work employs memory networks as a continuation of using deep learning in detecting fake news. Memory network, a kind of attention-based neural networks, can exhibit the ability to give selective focus on subregions of a given input performed by attention mechanism. Besides, they facilitate the storing of extra information in memory vectors, which is showed to be effective in many tasks such as language modeling, question answering [3].

In this paper, we proposed a memory network model for fake new detection. We found that the combination between additionally stored information and attention mechanisms improves the performance of the state-of-the-art. Our main contributions in this paper are:

- the proposal of a memory network that is able to store external information helpful for fake news detection.
- Investigation of multiple computations by reading input repetitively in a stacked memory network.
- Production of an accuracy surpassing that of the current state-of-the-art.
- Evaluation of the model for both of 6-label and 2-label classifications.

Our paper is structured as follows. We will give an overview of recent research related to fake news detection in section 2. Subsequently, a formal definition of fake news detection and background about memory networks are presented in section 3. Our proposed model is then delivered in detail in section 4, followed by experimental results and discussions section 5.

## 2 Related Work

### 2.1 Fake news detection

Study of news’ veracity started in the early 2010s, known as rumor detection. Pioneering works to detect rumor stress on data extracted from social networks due to the ease of propagating information from them. Castillo [4] took advantage of feature-based methods to assess the credibility of tweets on Twitter. Further along the line, Ma [5] extracted useful features to detect rumors. Those approaches achieved certain success, but heavily relied on feature engineering, which is expensive and time-consuming. Consequently, more recent endeavors using deep neural network were performed to get rid of the need for feature engineering. Ma [6] modeled streams of tweets as sequential data, then used Recurrent Neural Network (RNN) for predicting whether the streams were rumors or not. This approach was proved to yield better results than previous feature-based learning and effective at early rumor detection.

Detection of rumors is related to, but different from, that of fake news. While

both try to assess the credibility of news, their focused domains and data have little in common. In fact, research on rumor detection examines the trustworthiness of a group of posts related to a piece of news on Tweeter, while fake news detection works on an independent statement. Furthermore, statements to be studied in fake news detection are not only from social networks, but also from other places such as a public speech, a website, or a news advertisement, whereas posts in rumor detection are limited within social networks only.

To attract the crowd's attention towards fake news, a fake news challenge<sup>4</sup> was launched in 2017 based on the argument that support or disagreement between headline and body text are cues for debunking fake news. That year also witnessed a new direction in researching on fake news detection which focuses on political data, thanks to the introduction of Liar Dataset by Wang [1]. In that work, besides presenting a new benchmark dataset for fake news detection, the authors also proposed a hybrid architecture to solve the task. Their model made use of two components. One is a Convolutional Neural Network (CNN), which was to learn representation for text. The other was another CNN for meta-data representation learning, followed by a Long Short-Term Memory neural network (LSTM)[7]. Two kinds of representations then were passed into a fully-connected layer with softmax activation function to output the final prediction. Although being complicated with many parameters to be optimized, their model performed poorly on the test set, with only 27.4% in accuracy.

Rashkin [8] took a different perspective on detecting fake news by looking at its linguistic characteristics. They employed four types of lexical resources, that are the Linguistic Inquiry and Word Count (LIWC), subjective words with sentiment lexicon, hedging lexicon, and intensifying lexicons crawled from Wiktionary. They tried to examine lexicon's distribution in fakes news and true news so as to discover the difference between the language of true news and that of fake news. Despite substantial dependence on lexical resources, the performance on political set was even slower than [1], with only 22.0% in accuracy.

Long [2] proposed a hybrid LSTM which exploited two separate LSTMs. Word vectors were fed into the first LSTM, with topic and speaker information being two attention factors. Word vectors were again passed into the second LSTM, and speaker information was also used, but as an additional input rather than an attention factor. The two extracted vectors were then fed into a fully connected layer with softmax function to output the final prediction.

On the other hand, Volkova [9] work exclusively on data from Tweeter with the main goal is to predict if a news post is suspicious or verified, and classify it into fine-grained subsets of suspicious news - satire, hoaxes, clickbait, and propaganda. The author used linguistic neural networks with linguistic features. The insight from their work is that linguistic feature is relevant for fine-grained classification, whereas syntax and grammar features have little effect.

---

<sup>4</sup> <http://www.fakenewschallenge.org/>

## 2.2 Memory Networks

The original version of memory network using hard attention was introduced by Weston[10]. It was then adjusted by Sukhbaatar[3] with the substitute of hard attention by soft attention so that it can be trained end-to-end with less supervision required. Since then, it has been having many successful applications in a wide range of NLP tasks by virtue of its capability to store external information. In his work, Sukhbaatar demonstrates effective use of memory networks on question answering and language modeling. Das [11] exploited memory networks to perform attention between a considerable number of facts in the mixture of text and knowledge base to solve question answering task. Li [12] used memory networks to find out attitudes towards a set of entities from text.

## 3 Memory Network Model for Fake News Detection

### 3.1 Problem Definition

Suppose that we are given a training set of statements  $S = \{s_1, s_2, \dots, s_N\}$  and associated side information  $U = \{u_1, u_2, \dots, u_N\}$ , where  $N$  is the number of statements. Each  $s_i$  consists of a sequence of words  $w_1, w_2, \dots, w_n$ , while each  $u_i$  is a single or a set of side information. Our basic goal is to predict whether the statement is fake or true, or more challengingly, to classify it into a fined-grained level of truthfulness.

### 3.2 Single layer Memory Networks for fake news detection

**Input memory representation:** An embedding matrix  $A \in R^{v \times d}$  is used to transform words  $\{w_i\}$  in a statement into memory vectors  $\{m_i\}$ , where  $v$  and  $d$  are the vocabulary size and embedding size respectively. The associated side information, except for credit history which is already in form of a vector, is also converted into a vector  $u$  using another embedding matrix  $B \in R^{v' \times d'}$ . Unlike the original version of end-to-end memory networks proposed by [3] which computes dot product of  $w_i$  and  $u$  to find the relevance between them, we employ a different approach by doing aggregation since this allows for difference, hence flexibility in dimensions of embedding matrices  $A$  and  $B$ .

$$score(m_i, u) = v^T \tanh(W_m m_i + W_u u + b) \quad (1)$$

Where  $W_m \in R^{d \times a}$ ,  $W_u \in R^{v' \times a}$ , and  $v \in R^a$  with  $a$  is the dimension of attention vector.

Then, softmax function is applied to calculate vector  $p$ , normalized matching of  $w_i$  and  $u$ , also interpreted as a probability vector.

$$p_i = softmax(score(m_i, u)) \quad (2)$$

**Output memory representation:** Each  $w_i$  is converted into a vector  $c_i$  using another embedding matrix C. The output vector  $o$  is a weighted sum of  $c_i$  by probability vector from the input memory:

$$o = \sum_i p_i c_i \quad (3)$$

**Generating prediction:** since we allow for the difference in dimension of A and C, instead of simply doing summation between  $o$  and  $u$  we compute as follows.

$$h = \text{relu}(W_o o + W'_u u + b) \quad (4)$$

Where  $W_o \in R^{d' \times d'}$  and  $W_u \in R^{d' \times d'}$ , and **relu** is rectifier function.

A fully-connected layer ( $F$ ) is then applied, followed by a softmax layer to generate the final prediction.

$$\hat{y} = \text{softmax}(F(h)) \quad (5)$$

Cross-entropy is then used as the objective function.

$$L = \sum_i \sum_j y_j \log(\hat{y}_j) \quad (6)$$

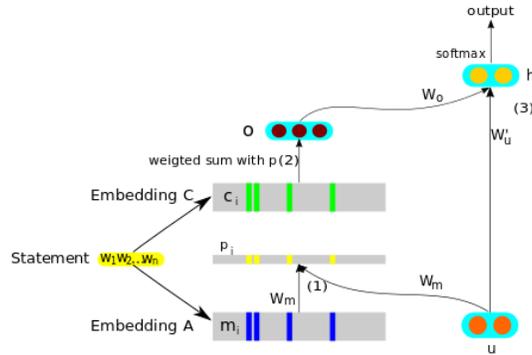


Fig. 1: Single layer memory networks for fake news detection

Despite having the similar structure, our proposed memory network is different from end-to-end memory networks by [3] in the receiving input and the way memory vectors are formed. Concretely, the input in [3] is a set of sentences, each of which is transformed into a memory vector using a weighting scheme, while ours is a set of words, each of which is converted into a memory vector directly by looking into an embedding matrix. As such, what we try to learn is external, different representations of words.

### 3.3 Multiple Layer Memory Networks for fake news detection

we extend the model by stacking multiple layers such that output of equation (4) at layer  $k$  will be the input  $u^{k+1}$  in the next layer.

$$u^{k+1} = \text{relu}(W_o o^k + W'_u u^k + b)$$

Where  $W_o$ ,  $W'_u$  are weight matrices shared or distinct across layers.  $W_m$  and  $W_u$  presented in the previous section are also shared or varying through different layers. The rest of the network will be the same of that in single layer one.

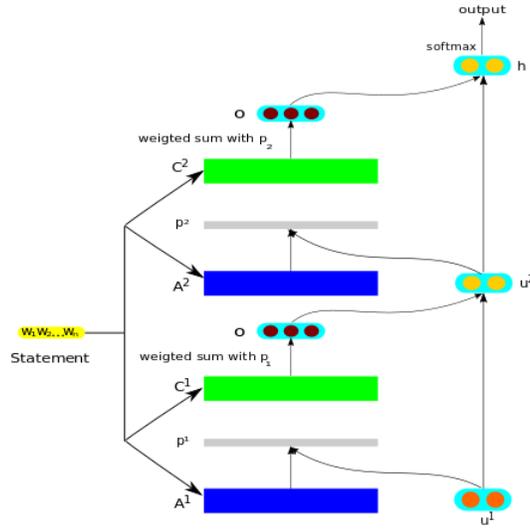


Fig. 2: Two layer memory networks for fake news detection

## 4 Experimental Results and Discussion

### 4.1 Dataset and Preprocessing

We evaluate our model using LIAR dataset by Wang[1]. The dataset includes 12,836 examples divided into separate train, validation, and test set by a ratio of 8:1:1. Each example encompasses a statement and a number of side information, that are topic, speaker name, title, party, affiliation, job, speech location and credit history. Credit history holds an account about the numbers of statements a speaker has made in pants-fire, false, barely-true, half-true and mostly-true categories. In Table 2, we show the distribution of six labels over training, development, and test set.

From the statistics, it can be noticed that the dataset is somewhat imbalanced in that the numbers of pain-fire, barely-true, and true are significantly fewer

Table 1: Distribution of six classes

	<b>pants-fire</b>	<b>false</b>	<b>barely-true</b>	<b>half-true</b>	<b>mostly-true</b>	<b>true</b>	<b>total</b>
Train	842	1,998	1,657	2,123	1,966	1,683	10,269
Valid	116	263	237	248	251	169	1,284
Test	92	250	214	267	249	211	1,283

than those of false, half-true, and mostly-true. For this reason, we use precision-macro, recall-macro, and f1-macro as evaluation metrics along with accuracy in the evaluation step.

Table 2: Examples of statements and side information in the dataset

<b>ID</b>	<b>Statement</b>	<b>Speaker</b>	<b>Credit history</b>	<b>Label</b>
1	Says he won the second debate with Hillary Clinton in a landslide in every poll.	Donald Trump	(63,114,51,37,61)	pants-fire
2	Each year, 18,000 people die in America because they don't have health care.	Hillary Clinton	(40,29,69,76,7)	true
3	Suzanne Bonamici supports a plan that will cut choice for Medicare Advantage seniors.	Rob Cornilles	(1,1,3,1,1)	half-true

The data is processed as follows. Each statement is tokenized using NLTK<sup>5</sup>, then stopwords and punctuation are removed. All money characters are converted into one token, so are percentage and number. For credit history, it is normalized to be a 5-dimensional vector, each value ranges within 0 and 1. Specifically, vector (70, 71, 160, 163, 9) is converted into (0.281, 0.285, 0.642, 0.654, 0.036). Other types of side information are processed to become one token.

## 4.2 Experiments

We compare our memory network (MM) against the following baselines:

**CNN-WangP**: a hybrid CNN using side information by [1]. In their model, one CNN is used to capture text representation, and CNN-LSTM is used for side information representation learning. Their CNN based on CNN for text by Kim[13]

**LSTM-L**: a hybrid LSTM using two LSTMs by [2]; one takes as input a statement and a type of associated side information, while the other takes as

<sup>5</sup> <http://www.nltk.org/>

input a statement only, but with topic and speaker information as attention components. The performance of baseline models was displayed in Table 3. For our model, with 50-dimensional vectors were used for word embeddings. We strictly turned all hyperparameters on dev set and observe the best result for the dimension of side information and attention vector are 32. The dropout keep probability was applied to 0.8 at the fully connected layer. Batch size was set to 64. We used Adadelta[14] as the optimizer with learning rate of 0.25.

Table 3: Accuracy of baseline models for 6-label classification (%)

Method	Dev	Test
Majority	20.4	24.7
CNN-WangP	24.7	27.0
LSTM-L	40.7	41.5

Table 4: Single Layer Memory network (MM) using one type of side information for 6-label classification(%). Sp, tp, jb, st, lc, pa, and ch stand for speaker name, topic, speaker job, state, location, party, and credit history respectively.

	MM+sp		MM+tp		MM+jb		MM+st		MM+lc		MM+pa		MM+ch	
	Dev	Test	Dev	Test										
Acc	25.2	25.4	24.7	25.7	25.3	24.2	26.1	24.9	25.5	24.6	27.0	24.2	<b>47.4</b>	<b>44.2</b>
Pre	16.8	22.8	22.4	21.5	17.9	17.1	21.0	20.5	20.7	21.6	13.5	32.1	<b>54.7</b>	<b>53.7</b>
Rec	20.6	20.7	21.9	22.2	21.3	20.5	22.7	21.4	21.6	20.9	22.7	20.4	<b>44.3</b>	<b>43.5</b>
F1	15.9	16.6	19.3	19.8	16.4	15.8	20.0	19.0	17.9	17.7	16.8	15.5	<b>43.1</b>	<b>42.1</b>

It is indicated from Table 3 and Table 4 that credit history, is the most informative factor in detecting fake news, which is consistent with the finding by Long [2]. However, our proposed memory network using credit history only (MM+ch) already outperformed hybrid LSTM incorporating all side information with attentions by [2] by 6.7% and 2.7% accuracy score on dev and test set respectively. The low values of precision, recall, and f1 scores when using side information other than credit history were because our model failed to give correct predictions for paint-fire, barely-true, and true. This could be explained by imbalance nature of the dataset since the numbers of examples in these categories are fewer than the others. Comparisons between our model and the baselines regarding precision, recall, and f1 are impossible because those values are unavailable from those baseline models.

Results from Table 5 also confirm the dominant contribution of credit history, when all evaluation scores rose by more than 10%. Speaker name (sp) and party (pa) information came in the second place on dev and test set.

Table 5: Single Layer Memory networks (MM) using one type of side information for 2-label classification (%)

	MM-sp		MM-tp		MM-jb		MM-st		MM-lc		MM-pa		MM-ch	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Acc	59.4	<u>64.1</u>	62.6	62.4	60.2	63.8	61.8	62.8	62.7	62.3	61.6	62.6	<b>73.8</b>	<b>74.4</b>
Pre	59.2	<u>63.3</u>	63.0	61.5	60.2	62.9	61.9	61.9	63.0	61.3	61.8	61.6	<b>74.0</b>	<b>74.0</b>
Rec	59.1	<u>63.2</u>	62.1	60.3	59.9	62.6	61.3	61.3	62.3	60.6	61.1	61.0	<b>73.6</b>	<b>73.7</b>
F1	59.0	<u>63.2</u>	61.7	60.2	59.8	62.7	61.0	61.3	62.0	60.7	60.9	61.0	<b>73.6</b>	<b>73.8</b>

Table 6: Single Layer Memory networks with combined side information for 6-class setting (%)

	MM+ch		MM+ch+sp		MM+ch+jb		MM+ch+st		MM+ch+pa	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Acc	47.4	44.2	47.9	45.4	48.4	44.8	<b>49.1</b>	45.9	<b>49.1</b>	<b>46.4</b>
Pre	54.7	53.7	53.0	51.5	53.0	51.0	56.5	52.4	<b>57.1</b>	<b>55.1</b>
Rec	44.3	43.5	46.1	45.3	46.6	44.7	47.2	45.5	<b>47.2</b>	<b>46.0</b>
F1	43.1	42.1	46.7	45.6	47.3	44.6	<b>48.0</b>	45.5	47.7	<b>45.8</b>

From Table 6, it is noticed that incorporating more side information boosts the performance. Specifically, when combining credit history with party information, accuracy scores increased by 1.7% and 2.2% on dev and test set respectively. Likewise, precision scores went up by 2.4% on dev set and 1.4% on test set. Recall scores witnessed the same trend with the rise of 2.9% on dev set and 2.5% on test set. F1 score was also improved by 4.6% on dev set and 3.7% on test set. State and party information perform somewhat better than speaker and job information. Overall, MM+ch+pa produced accuracies higher than that of the state-of-the-art, which are by 8.4% and 5.2% on dev and test set respectively. Since the addition of more side information (eg. MM+ch+pa+st) did not raise the performance further, we omit the results for such cases. Similarly, in 2-label classification, it seems that credit history delivers the most needed information that adding others (eg. MM+ch+st or MM+ch+pa) did not help. Therefore, results for those cases are not presented either. Speaking of multiple computations, stacking our memory networks with more layers does not further improve the performance in our task. Table 7 shows that the performance of using two layer memory network was lower than that of using one layer on all evaluation metrics when using only credit history (MM2+ch). However, in case of using both credit and party information, our two layer (MM2+ch+pa) and three layer memory networks (MM3+ch+pa) nearly reached the single layer one (MM+ch+pa). This demonstrated the relevance of external learned information.

Table 7: Single layer (MM) and two layer (MM2) and three layer memory network(MM3) for 6-label classification in comparison.

	MM+ch		MM+ch+pa		MM2+ch		MM2+ch+pa		MM3+ch+pa	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Acc	47.3	44.2	<b>49.1</b>	<b>46.7</b>	40.5	38.7	48.8	<u>46.2</u>	<b>49.1</b>	45.9
Pre	50.2	49.7	<b>56.7</b>	<b>55.4</b>	31.6	26.2	54.9	53.5	54.1	52.6
Rec	44.6	43.6	47.3	<b>46.3</b>	34.4	33.2	47.3	<u>46.1</u>	<b>47.4</b>	45.9
F1	44.0	42.0	47.7	<b>46.1</b>	29.8	28.6	47.6	45.7	<b>47.7</b>	<u>45.8</u>

### 4.3 Discussion

From our experiments, we observe several phenomena.

**It is difficult to distinguish pants-fire from false.** For 6-label classification, our model predicts correctly 48 out of 92 instances with *pants-fire* label, but misclassifies 25 other instances as *false* label, which accounts for 27% of wrong predictions. The reason may come from the language use, especially strong determiners such as *every* or *any*. For example, in the statement “*Says he won the second debate with Hillary Clinton in a landslide in every poll.*” Justification from POLITIFACT.COM tells us that not only did Trump not win by a landslide in any of the polls, he didn’t win any of the polls<sup>6</sup>. It is obvious that the statement is false, but it is rated as pants-fire to stress the over exaggeration of the lie. However, our model is unable to recognize that emphasis. Similarly, in the statement “*This town (Wilmington, Ohio) hasn’t taken any money from the government,. They don’t want any money from the government*”, the mention of *any* is so subtle that our model fails to classify it as pants-fire, but as false instead.

**Likewise, separating true from half-true and mostly-true is challenging.** In 2-label setting, our model successfully predicts only 34 out of total 211 examples to be true, and misclassifies 65 and 49 others to be mostly-true and true respectively. We observe that reference to numbers, percentages, and money is exploited as a factor to mix false opinion into a true story. In particular, the statement “*However, it took \$19.5 million in Oregon Lottery funds for the Port of Newport to eventually land the new NOAA Marine Operations Center-Pacific*” in the training set in which an amount of money also appears is annotated as half-true since the statement mixes true number of \$19.5 million and the misleading place where the money went to. On the other hand, our model misclassified statements 3, 4, 5, and 6 in Table 7 which share the same pattern of money or number reference.

Figure 3 illustrates attention weights the proposed memory model generates for statement 1, 3, 2, and 4 in Table 9. We realize that our model gives focus on

<sup>6</sup> <http://www.politifact.com/wisconsin/statements/2016/oct/12/donald-trump/donald-trumps-ridiculous-claim-all-polls-show-he-w/>

Table 8: Examples of correct prediction on 2-label, but wrong in 6-label classification.

ID	Statement	For 6-label		For 2-label	
		Predict	Truth	Predict	Truth
1	Says he won the second debate with Hillary Clinton in a landslide in <i>every</i> poll.	false	pants-fire	false	false
2	This town (Wilmington, Ohio) hasn't taken <i>any</i> money from the government. They dont want <i>any</i> money from the government.	false	pants-fire	false	false
3	The Fed created \$1.2 trillion out of nothing, gave it to banks, and some of them foreign banks, so that they could stabilize their operations.	mostly-true	true	true	true
4	Texas families have kept more than \$10 billion in their family budgets since we successfully fought to restore Texas sales tax deduction a decade ago.	mostly-true	true	true	true
5	Says the unemployment rate for college graduates is 4.4 percent and over 10 percent for noncollege-educated.	half-true	true	true	true
6	Each year, 18,000 people die in America because they don't have health care.	mostly-true	true	true	true

Table 9: Examples of correct predictions in both 6-label and 2-label setting.

ID	Statement	For 6-label		For 2-label	
		Predict	Truth	Predict	Truth
1	Says Thom Tillis gives tax breaks to yacht and jet owners.	pants-fire	pants-fire	false	false
2	Says John McCain has done nothing to help the vets.	pants-fire	pants-fire	false	false
3	The United States has a low voter turnout rate.	true	true	true	true
4	Says he would be first CPA to serve as Texas comptroller.	true	true	true	true

proper nouns (Thom in Fig. 3a and McCain Fig. 3c) and verb *give*. Therefore, it seems that dealing with name entities and verbs is promising for this task. Language use is also a cue for detecting fake news. In fact, Fig. 3b shows that adjective *low* is given strong attention, while Fig. 3c reveals that the attention is put largely on model verb *would*.

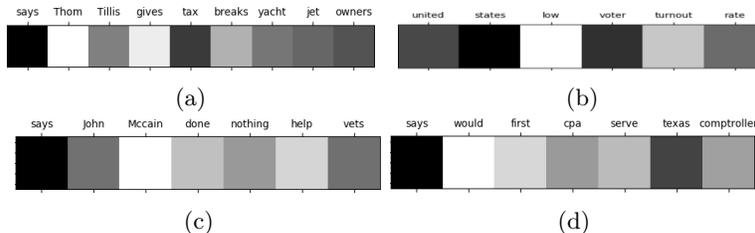


Fig. 3: Visualizations of attention weights. (a), (b), (c), (d) represent statements of 1,3,2,4 in Table 9 respectively

## 5 Conclusion

This paper proposes a memory network model for fake news detection. Our memory network takes advantage of attention mechanisms to focus on the most relevant subparts of a given input as well as storing external information by the network itself. Our experiment results show that the additionally stored information is helpful for the task. Moreover, dealing with fine-grained labels is difficult as neighboring labels are so confusing to be recognized correctly. The performance largely comes from credit history, and is improved more with the presence of other side information. Overall, our model outperforms the current state-of-the-art by 8.4% and 5.2% on dev and test set respectively.

## References

1. Wang, W.Y.: “liar, liar pants on fire”: A new benchmark dataset for fake news detection. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Vancouver, Canada, Association for Computational Linguistics (2017) 422–426
2. Long, Y., Lu, Q., Xiang, R., Li, M., Huang, C.R.: Fake news detection through multi-perspective speaker profiles. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers). Volume 2. (2017) 252–256
3. Sukhbaatar, S., szlam, a., Weston, J., Fergus, R.: End-to-end memory networks. In Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., eds.: Advances

- in Neural Information Processing Systems 28. Curran Associates, Inc. (2015) 2440–2448
4. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on twitter. In: Proceedings of the 20th international conference on World wide web, ACM (2011) 675–684
  5. Ma, J., Gao, W., Wei, Z., Lu, Y., Wong, K.F.: Detect rumors using time series of social context information on microblogging websites. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM (2015) 1751–1754
  6. Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B.J., Wong, K.F., Cha, M.: Detecting rumors from microblogs with recurrent neural networks. In: IJCAI. (2016) 3818–3824
  7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural computation* **9** (1997) 1735–1780
  8. Rashkin, H., Choi, E., Jang, J.Y., Volkova, S., Choi, Y.: Truth of varying shades: Analyzing language in fake news and political fact-checking. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. (2017) 2921–2927
  9. Volkova, S., Shaffer, K., Jang, J.Y., Hodas, N.: Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Volume 2. (2017) 647–653
  10. Weston, J., Chopra, S., Bordes, A.: Memory networks. *arXiv preprint arXiv:1410.3916* (2014)
  11. Das, R., Zaheer, M., Reddy, S., McCallum, A.: Question answering on knowledge bases and text using universal schema and memory networks. *arXiv preprint arXiv:1704.08384* (2017)
  12. Li, C., Guo, X., Mei, Q.: Deep memory networks for attitude identification. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, ACM (2017) 671–680
  13. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics (2014) 1746–1751
  14. Zeiler, M.D.: Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* (2012)