Multi-Source Multi-Class Fake News Detection

Hamid Karimi, Proteek Chandan Roy, Sari Saba-Sadiya, and Jiliang Tang
Department of Computer Science and Engineering, Michigan State University
{karimiha,royprote,sadiyasa,tangjili}@msu.edu

Abstract

Fake news spreading through media outlets poses a real threat to the trustworthiness of information and detecting fake news has attracted increasing attention in recent years. Fake news is typically written intentionally to mislead readers, which determines that fake news detection merely based on news content is tremendously challenging. Meanwhile, fake news could contain true evidence to mock true news and presents different degrees of fakeness, which further exacerbates the detection difficulty. On the other hand, the spread of fake news produces various types of data from different perspectives. These multiple sources provide rich contextual information about fake news and offer unprecedented opportunities for advanced fake news detection. In this paper, we study fake news detection with different degrees of fakeness by integrating multiple sources. In particular, we introduce approaches to combine information from multiple sources and to discriminate between different degrees of fakeness, and propose a **M**ulti-source **M**ulti-class **F**ake news **D**etection framework MMFD, which combines automated feature extraction, multi-source fusion and automated degrees of fakeness detection into a coherent and interpretable model. Experimental results on the real-world data demonstrate the effectiveness of the proposed framework and extensive experiments are further conducted to understand the working of the proposed framework.

1 Introduction

Given its negative impacts, fake news has been identified as a global threat (Webb et al., 2016). Detecting fake news has become increasingly important and can benefit individuals and even our society in many aspects. First, people will be well-informed about events and news and their political and social activities will not be misguided. Second, despite a few recent initiatives by some social media providers like Facebook, there is no systematic fake news detection by social media platforms. Third, identifying fake news is one step toward targeting financial incentives encouraging the spreaders running their "business". For instance, Google attempts to stop providing its ad services to fake news websites (Wingfiled et al., 2016). Fourth, people would not lose their trust in web and social media.

However, fake news detection is naturally challenging especially in the era of social media. First, fake news is usually written intentionally to mislead its readers and the content of fake news is rather diverse in terms of length, topics, and styles (Shu et al., 2017a). For example, fake news in social media is short, informal and is often related to newly emerging and time-critical events. Thus, since we have not gained enough insights into the nature of fake news, hand-crafted features based on the news content are generally not sufficient (Ruchansky et al., 2017). Second, to mock true news, fake news could mix false statements with true ones. For example, fake news can cite true evidence to support a non-factual claim (Shu et al., 2017a). Hence, fake news has different degrees of *fakeness* such as half-true, false, etc. However, the majority of existing algorithms consider fake news detection as a binary classification in the form of the *true/false* dichotomy. Considering degrees of fakeness adds further difficulty on fake news detection.

Proceedings of the 27th International Conference on Computational Linguistics, pages 1546–1557
Santa Fe, New Mexico, USA, August 20-26, 2018.

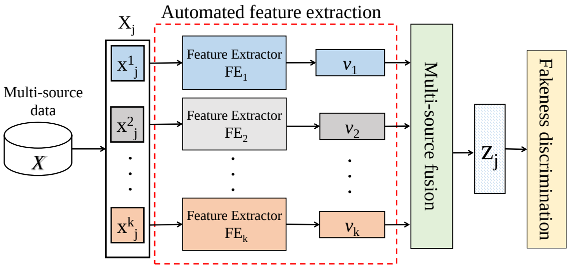

Figure 1: An overview of the proposed model for fake news detection (MMFD)

On the other hand, recent advances in mobile techniques and the popularity of emerging media (e.g., social media) enable the widespread of fake news and this process produces a large amount of data, which allows us to collect information about fake news from various perspectives such as the origin of fake news, the fake news writers and their historical data, etc. Such data provides rich contextual information beyond the news content. For example, a writer who created plenty of fake news is likely to create more fake news (Del Vicario et al., 2016); while news from authoritative organizations such as governments is less likely to be false. Thus, the availability of multiple sources related to fake news has the great potential to help fake news detection.

In this paper, we study the problem of multi-class fake news detection with multiple sources. In particular, we aim to answer two major research questions – (1) how to effectively combine information from multiple sources for fake news detection and (2) how to discriminate between degrees of fakeness mathematically. Our solutions to these two questions result in the proposed Multi-source Multi-class Fake news Detection (MMFD) framework. Our main contributions are summarized as a) we introduce an automated and interpretable way to integrate information from multiple sources; b) we provide a principled approach to discriminate between degrees of fakeness mathematically; c) the proposed framework MMFD coherently incorporates automated feature extraction, multi-source fusion and degrees of fakeness discrimination in an end-to-end way; and d) we conduct experiments on real-world data to demonstrate the effectiveness of the proposed framework.

The rest of the paper is organized in the following manner. In Section 2, we formally define the multi-source multi-class fake news detection problem. Section 3 describes the proposed approach for fake news detection. Section 4 evaluates the proposed method and presents the experimental results and discussions. In Section 5, we present the related work. Section 6 concludes the paper and sheds light onto the future directions.

2 Problem Definition

In this section, we introduce the mathematical notations and formally define the multi-source multi-class fake news detection problem. We follow the previous work (Allcott and Gentzkow, 2017; Shu et al., 2017b) and define fake news as follows.

Definition. A news item is called *fake* if its content is verified to be false and *true* otherwise.

Let $\mathcal{X} = \{X_1, X_2, \cdots, X_n\}$ denote a multi-source dataset containing n news items. Each new item $j \in [1, n]$ contains k sources of data and is denoted as $X_j = \{x_j^1, x_j^2, \cdots x_j^k\}$. Additionally, let $Y = \{y_1, y_2, \cdots, y_n\}$ denote a set of class labels associated with news items of dataset $\mathcal{X}$. Each class label $y_i \in Y$ takes a label from the label set $L = \{l_1, l_2, \cdots, l_m\}$ where m denotes the number of recognized degrees of fakeness in our framework and $l_j \in L$ is a particular degree of fakeness e.g., *half-true*. With the aforementioned notations and definitions, the problem of multi-source multi-class fake news detection is formally defined as follows:

Given the multi-source dataset $\mathcal{X}$ and its corresponding multi-class labels Y, we aim to learn the model $\mathcal{M}$ mapping $\mathcal{X}$ to Y, which can automatically predicts the degrees of fakeness for unlabeled news.

3 The Proposed Framework

Multi-source multi-class fake news detection faces three challenges. First, hand-crafted features based on news content are not very efficient for fake news, which calls for an automated feature extraction approach from multiple sources. Second, multiple sources of fake news data offer complementary information and combining the multiple sources while delivering an interpretable solution is another challenge. Third, degrees of fakeness offer a better understanding of fake news; however, fake news with different degrees may not be easily separable. In this work, we propose a framework which can tackle these challenges simultaneously. Figure 1 demonstrates an overview of the proposed **M**ulti-source **M**ulti-class **F**ake news **D**etection framework (MMFD). MMFD incorporates three coherent components into an end-to-end way – automated feature extraction (Section 3.1), interpretable multi-source fusion (Section 3.2), and fakeness discrimination (Section 3.3). In the following, we detail each component, followed by presenting the training procedure of the proposed model in Section 3.4.

3.1 Automated feature extraction

Since hand-crafted features (Gupta et al., 2014; Yang et al., 2012; Khurana and Intelligentie, 2017) are not very effective, automated feature extraction is more desired. Thus, we employ a deep neural network model to extract powerful features from each source inspired by the promising performance of deep learning in representation learning (Bengio et al., 2013). Different types of sources may need different deep network architectures. Since most of the sources in our dataset are textual sources (see Section 4), we propose a method to automatically extract features from a textual source.

The textual data can contain indicative information revealing the nature of fake news. Inspired by recent advancements in deep neural networks for modeling the text, we propose a deep model to extract features from textual sources based the CNN (convolutional Neural Network) (LeCun et al., 2015) and the LSTM (Long Short-Term Memory) network (Hochreiter and Schmidhuber, 1997). A CNN extracts local patters from a text similar to n-grams features (Kalchbrenner et al., 2014). Then, we apply an LSTM on top of the features extracted from the CNN aiming at capturing the temporal dependencies in the entire text.

Suppose a textual source contains x words. To apply the CNN model, we represent the text by an input matrix of word embeddings denoted as $W \in \mathbb{R}^{x \times e}$ where e is the dimension of the word embedding. More specifically, $w_j \in W$ is a e dimensional vector representing j-th word of the text and populates j-th row of matrix W. This matrix is produced from a word representation method such as word2vec (Mikolov et al., 2013). Then, convolution operations on W are performed M iterations (e.g., $\mathsf{M} = 2$ in Figure 2). At each iteration $m \in [1, \mathsf{M}]$, a filter (a weight matrix) $f^m \in \mathbb{R}^{l^m \times e}$ is convolved with W where l^m is the length of the filter. Convolution of f^m with the input matrix W produces feature maps $\mathbf{p}^m \in \mathbb{R}^{x-l^m+1}$. Each entry $p_j \in \mathbf{p}^m$ $(1 \leq j \leq x - l^m + 1)$ is generated as follows:

$$p_j = \mathcal{G}(r_j \odot f^m + b) \tag{1}$$

where $r_j = \{w_j, w_{j+1}, \cdots, w_{j+l^m-1}\}$ is a window of l^m consecutive words in W (a region of m words), $\mathcal{G}$ is a non-linear activation function such as sigmoid, $\odot$ denotes element-wise product operation, and $b \in \mathbb{R}$ is a bias term. Following the common way in CNNs, we repeat the process described above n_m times producing the sequence $U^m = [\mathbf{p}_1^m || \mathbf{p}_2^m || \cdots || \mathbf{p}_{n_m}^m]$ where $||$ denotes the vector concatenation operator (one can see U^1 and U^2 in Figure 2 shown as colorful column vectors). Rows of U^m represent the **local patterns** extracted from a text via the CNNs. In order to find the **global patterns** throughout the entire text, we feed rows of U^m to an LSTM network as depicted in Figure 2. The output of the LSTM network is considered as the hidden output of its last unit and is denoted as s^m having size q i.e., $|s^m| = q$. Then, the final feature vector of the proposed CNN-LSTM model is generated by passing the concatenation of all last hidden outputs through a Fully Connected Network (denoted as FCN in Figure 2) as follows:

$$v = \mathcal{G}([s^1 || s^2 || \cdots || s^{\mathsf{M}}]^T \times A + b) \tag{2}$$

where v denotes the extracted feature vector of a textual source, $A \in \mathbb{R}^{(\mathsf{M} \times q) \times d}$ is a weight matrix, and d is the desired final feature vector size. Note that $[s^1 || s^2 || \cdots || s^{\mathsf{M}}]$ is flattened into a vector of size $\mathsf{M} \times q$.

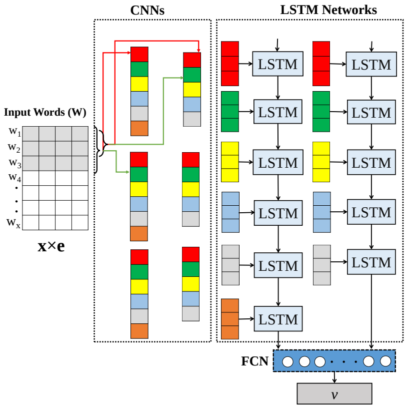

Figure 2: An illustration of the proposed deep model for fake news textual source feature extraction.

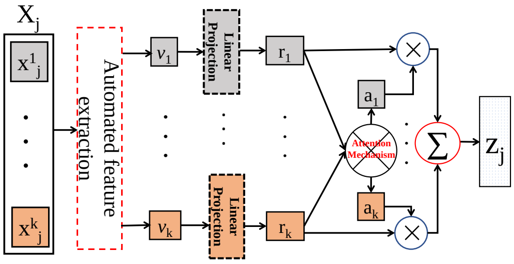

Figure 3: An illustration of the proposed interpretable multi-source fusion.

3.2 Interpretable multi-source fusion

The interpretable multi-source fusion component aims to combine features from different sources. A naive way of doing this is concatenating the feature vectors extracted by automated feature extraction component i.e., v_1 to v_k in Figure 1, into a single vector. This scheme considers all sources equally. However, different sources may have substantially different powers to detect fake news. Hence, when combining multiple sources, we propose to consider their contributions via an attention mechanism as shown in Figure 3. The attention mechanisms have achieved great success in many applications such as language translation (Bahdanau et al., 2014), image captioning (Xu et al., 2015), etc. Next, we present the details of the interpretable multi-source fusion component

The set of source-specific features of a news item is extracted by the automated feature extraction component as described in Section 3.1. Then suppose each source's extracted feature vector is denoted by $v_i \in \mathbb{R}^{d_i}$ $(1 \leq i \leq k)$ and has the dimension d_i. Now, to ensure feature vectors from different sources all have the same dimension h, we map each v_i to the vector r_i via a linear projection as follows:

$$r_i = v_i^T \times W_i + b_i, \quad \forall 1 \leq i \leq k \tag{3}$$

where $W_i \in \mathbb{R}^{d_i \times h}$ is a weight matrix, and $b_i \in \mathbb{R}^h$ is a bias vector. Then, the proposed attention method is carried out as follows:

$$z_j = \mathcal{G}(\sum_{i=1}^{k} a_i r_i), \quad \forall 1 \leq j \leq n \tag{4}$$

where z_j denotes the final feature vector of j-th news item and the scalar a_i is the attention score associated with i-th source i.e., the contribution of i-th source at z_j. Typically the attention scores are represented by a probability distribution. Therefore, each attention score a_i is normalized by the softmax function as follows.

$$a_i = \frac{e^{u_i}}{\sum_{l=1}^{k} e^{u_l}}, \quad \forall 1 \leq i \leq k \tag{5}$$

where u_i is a real value number denoting attention score of source i and is calculated according to Eq. 6:

$$u_i = w^T tanh(r_i), \quad \forall 1 \leq i \leq k \tag{6}$$

where $w \in \mathbb{R}^h$ is a weight matrix.

We should emphasize that attention scores a_i $(1 \leq i \leq k)$ are learned along with other parts of the model in an end-to-end manner where they are optimized to capture the informativeness of the different sources at the fake news detection task. As a result, capturing contributions of different sources reflected at scalar values a_i ($1 \leq i \leq k$) makes the proposed framework somehow interpretable. This is due to the fact that now we can identify the roles of different sources in the determining the veracity of a news item as will be verified in Section 4.4.

3.3 Fakeness discrimination

Many of the existing fake news detection approaches look into the problem from a binary perspective. However, in practice, a piece of news might be a mixture of factual and false statements. Therefore, it is crucial to classify fake news into multiple classes reflecting degrees of fakeness. Nevertheless, multi-class fake news detection (i.e., fakeness discrimination) is challenging. The classifier needs to offer a better discriminative power because the boundary between two classes becomes more and more intertwined as the number of classes increases. To handle this challenge, we propose **M**ulti-class **D**iscriminative **F**unction (MDF) and is described in the following.

Classification accuracy on unknown test cases mostly depends on how we map our input data into a new feature space. In general, we want to group similar samples together and dissimilar ones far apart. To achieve this, we can learn the 'centers' of the classes in the feature space. If the centers are too close to one another, their samples might overlap resulting in less discriminative feature learning. Additionally, samples should be close to their class centers. This guarantees that samples belonging to the same class will be mapped into the same cluster (Wen et al., 2016). In general, the two aforementioned goals can be regarded as optimizing the precision and recall of the clusters in the learned feature space. Therefore, we include two terms in MDF: one which *pushes* the centers by a margin referred as *inter-class* term and one which *pulls* the samples toward their centers referred as *intra-class* term. The proposed inter-class term helps to learn discriminative features. Next, we introduce the MDF mathematically.

We present the MDF in a generalized non-convex form. The interpretable multi-source fusion component yields output feature vector of each news item (z_j in Eq. 4). For convenience, let $\mathcal{D}$ denote a dataset that contains features of all news items in $\mathcal{X}$ (i.e., $\mathcal{D} = \{z_1, z_2, \cdots, z_n\}$) and $\mathcal{D}_i$ denote all samples belonging to class i where $1 \leq i \leq m$. We calculate the centers of the different classes in the output feature space by the following equation.

$$c_i = \frac{1}{|\mathcal{D}_i|} \sum_{z_j \in \mathcal{D}_i} z_j, \forall i \in \mathcal{C} \tag{7}$$

During the optimization process, centers are not fixed anymore, as both weights and centers are getting updated. Also, the centers and feature outputs are normalized as $||c_j||_2 = 1$ and $||z_j||_2 = 1$, and therefore they reside on a unit hyper-sphere. Eq. 8 shows the formulation of MDF.

$$\begin{aligned} \varepsilon^{intra} &= \frac{1}{|\mathcal{D}|} \sum_{\forall \mathcal{D}_i \in D} \sum_{z_j \in \mathcal{D}_i} \|z_j - c_i\|_2^2 \\ \varepsilon^{inter} &= \frac{1}{\binom{m}{2}} \sum_{o=1}^{m} \sum_{r=1, r \neq o}^{m} max(0, \alpha - \|c_o - c_r\|_2^2) \\ MDF &= \beta_1 \varepsilon^{intra} + \beta_2 \varepsilon^{inter} \end{aligned} \tag{8}$$

where ε^{intra} term is the intra-class term that tries to minimize a sample's distance to its class center, and ε^{inter} is the inter-class term that is responsible for enforcing margin α between centers of every two classes. Two hyperparameters β_1 and β_2 control the intra-class and inter-class terms, respectively.

3.4 Training procedure

In this subsection, we describe the training procedure of MMFD as shown in Algorithm 1. At each iteration of the algorithm, a mini-batch of samples is selected for training (line 2). The features of each source are extracted by the automated feature extraction component described in Section 3.1 (line 5). The extracted features are re-weighted according to the proposed attention mechanism described in Section 3.2 (line 8). We combine MDF (Eq. 8) and cross entropy as the model loss function and their contributions

Algorithm 1: Training procedure of MMFD.

```
Data: Sample $\mathcal{X} = \{X_1, \cdots, X_n\}$ and labels $Y = \{y_1, \cdots, y_n\}$
Input: Learning rate: $\eta$; weight of MDF : $\lambda$; and hyperparameters of MDF: $\beta_1, \beta_2$ and $\alpha$
Output: Model $\mathcal{M}$ and parameters $\theta = \{\theta_1, \cdots, \theta_p\}$
1  while Not convergent do
2      Select mini-batch $\mathcal{B} \subseteq \mathcal{X}$
3      foreach $X_j \in \mathcal{B}$ do
4          foreach $x_j^i \in X_j$ do
5              $v_i = FE_i(x_j^i)$ /* see Figure 1 */
6              $r_i = v_i^T \times W_i + b_i$ /* see Figure 3 */
7          end
8          $z_j = \mathcal{G}(\sum_{i=1}^{k} a_i r_i)$ /* see Figure 3 */
9          $p(z_j) = Softmax(z_j)$
10     end
11     $CrossEntropy = -\sum_{z_j} p(y_j) \times log(p(z_j))$
12     $J = \lambda \times CrossEntropy + (1 - \lambda)(\beta_1 \varepsilon^{intra} + \beta_2 \varepsilon^{inter})$ /* refer to Eq. 8 for $\varepsilon$ terms */
13     Backpropagate the error to get $\nabla_\theta J(\theta)$
14     $\forall \theta_i \in \theta, \theta_i = \theta_i - \eta \nabla_{\theta_i} J(\theta)$
15 end
16 if condition = TRUE then
17     $CalculateCenters(\mathcal{X}; \mathcal{M})$
18 end
19 return $\mathcal{M}$, $\{\theta_1, \cdots, \theta_p\}$
```

are controlled by the hyperparameter λ (line 12). Algorithm 1 backpropagates the error to compute gradients of the loss function with respect to each parameter of the model. Following the common way, we use Stochastic Gradient Descent (SGD) to update the parameters (line 14). Finally, once the training converges, the optimized parameters are returned, which can be used for prediction.

4 Experiment

In this section, we empirically evaluate the proposed framework on a real-world fake news dataset. We demonstrate the effectiveness of the approach by conducting a set of experiments. Particularly, we seek to answer the following research questions.

- *Q1*. How does the proposed framework perform on fake news detection?
- *Q2*. Can the proposed framework handle multiple sources effectively?
- *Q3*. How does each component of the proposed framework affect the performance?

We first present the experimental settings. Then, the experiments and their results are conducted to answer the aforementioned questions. Finally, we present a case study to qualitatively demonstrate the effectiveness of the proposed approach.

4.1 Experimental settings

In this work, we use the dataset published in (Wang, 2017) known as LIAR. LIAR is one of the largest publicly available fake news datasets. It has been constructed from a collection of statements investigated by experts in *politifact.com*, which is a well-known fact-checking service. The author did split the dataset into three sets: train, test, and development. His sample selection for sets is random where the train, test, and development sets contain, respectively, 80%, 10%, and 10% of the entire dataset. We use the same split as Wang (Wang, 2017). LIAR contains three sources of data. We supplement it and add another source. In the following, we describe the sources used to evaluate the proposed framework.

- **Statements.** The statements are short sentence(s), mostly from American politicians covering different topics. The statements have been manually classified into six classes including *True*, *Mostly-True*, *Half-True*, *Barely-True*, *False*, and *Pants-on-Fire*. We use the architecture presented in Section 3.1 (Figure 2) to model the statements.

- **Metadata.** The metadata is a textual context about the statements or their speakers if known. This includes the name of a statement's speaker, his/her job title, his/her political party affiliation, the U.S state associated with a speaker, and the venue/location that a statement has been made. When a speaker is unknown, the related fields are blank. We combine all metadata fields and use the architecture presented in Section 3.1 (Figure 2) to model it.

- **History.** The history is a non-textual source. It is a 5-dimensional vector representing a speaker's count of statements on five classes, namely *Mostly-True*, *Half-True*, *Barely-True*, *False*, and *Pants-on-Fire*. The history is modeled by a single-layer fully connected network with 10 hidden units.

- **Report.** In addition to the sources described above, we supplement the dataset by adding the verdict reports generated by experts in *politifact.com*. The reports are longer than the statements and metadata. We remove the class labels from the reports. Similar to other textual sources, the architecture presented in Section 3.1 (Figure 2) is utilized to model the reports.

We use the train set to train the model. For the textual sources, we populate word embeddings (see Figure 2) from the Google word2vec embeddings trained on roughly 100 billion words from Google News (Mikolov et al., 2013). In all experiments, we utilize the development set to tune the hyperparameters, which have been done carefully with grid search over ranges of different values. We utilize the mini-batch 32, apply the dropout rate 60%, and use Adam optimizer (Kingma and Ba, 2014) to apply the gradient descent with the learning rate of 0.001. For center computation in MDF (Eq. 7), we gradually increase the center computation period starting from 20 optimization steps to 160 steps as the training proceeds. The hyperparameters β_1, β_2, and α are set to 0.6, 0.4, and 0.3, respectively. The number of hidden units in LSTM network is set to 200. The test set is used to evaluate the effectiveness of the model and the performance reported in this paper is based on the evaluation of the test set. For the performance metric, we use accuracy since the dataset is fairly balanced and also consistent with Wang (Wang, 2017), we found that f-measure performs similarly as accuracy.

4.2 Performance comparison

To answer questions *Q1* and *Q2*, we compare our approach, MMFD, with a set of representative baselines:

- Basic-SVM. For this baseline, we extract a set of features from sources. For the textual sources, we extract Bag-of-Words, bigrams, and 3-grams, and simply combine them with the history. Since the number of features is large, we apply PCA to reduce the dimension to 300. Finally, we train a SVM model (Support Vector Machine) on the extracted features. We use a grid search on the performance of the development set to tune the SVM classifier hyperparameters.

- Basic-RandomForests. This baseline is similar to Basic-SVM except that Random Forests is employed as the classifier.

- Basic-NN. In addition to the employed traditional classifiers, i.e., SVM and Random Forests, we also use a NN (Neural Network) as the baseline. The input to NN is the same with other two classifiers. We use a fully connected network with one layer. Again, the hyperparameters are set using the development set. Basic-NN further refines the initial features.

- Wang (Wang, 2017). Wang (Wang, 2017) developed a model based on CNN and BLSTM (Bi-directional LSTM) for fake news detection on LIAR dataset.

- Random. Random includes randomly selecting the class of a test sample.

- Majority. In this method, each test news item is labeled with a class having the largest number of samples i.e., the *Half-True* class.

Table 1: Performance comparison. S1: Statement, S2: Metadata, S3: History, S4: Report

Sources	Method	Accuracy (%)
–	Random	17.4
–	Majority	20.8
S1	Basic-SVM	20.12
	Basic-RandomForests	23.19
	Basic-NN	21.73
	Wang (Wang, 2017)	27.00
	MMFD	**29.06**
S1+S2+S3	Basic-SVM	25.07
	Basic-RandomForests	27.63
	Basic-NN	28.16
	Wang (Wang, 2017)	27.04
	MMFD	**34.77**
S1+S2+S3+S4	Basic-SVM	29.98
	Basic-RandomForests	27.01
	Basic-NN	29.12
	Wang (Wang, 2017)	N/A
	MMFD	**38.81**

The comparison results on different combinations of sources have been shown in Table 1. As mentioned before, the original dataset (Wang, 2017) contains only the sources “S1: Statement”, “S2: Metadata” and “S3: History” and we enrich the dataset by adding the fourth source “S4: Report”. Thus, we only report combinations with news content (i.e., S1), all sources in the original dataset (i.e., S1+S2+S3) and all sources in the enriched dataset (i.e., S1+S2+S3+S4). Note that the performance of Wang (Wang, 2017) is not available on the enriched dataset. We make the following observations from these results:

- MMDF outperforms Wang (Wang, 2017) when only using the news content (e.g., S1). Both methods are based on CNN + LSTM. However, the proposed framework also has the model component to discriminate between multiple classes. This observation supports the importance of the proposed discriminative function i.e., MDF.

- By incorporating more sources, the performance of all methods tends to increase. Compared to baselines, the proposed framework MMDF enjoys more performance improvement with more sources. These observations suggest that the proposed framework can effectively combine multiple sources.

- The proposed framework always obtains the best performance with all settings. In the following subsection, we further investigate how each model component contributes to the performance improvement.

To sum up, the proposed framework outperforms representative baselines and it is also effective in integrating information from multiple sources.

4.3 MMFD component analysis

In this subsection, we conduct several experiments to study the impact of model components on the performance of the proposed framework to answer *Q3*. The experiment are as follow:

- $MMFD_{PCA}$. In this experiment, we remove feature extractors in MMDF (FEs in Figure 1). Instead, we apply PCA (with dimension 100) on each input of the textual sources. We keep the history and the interpretable multi-source fusion and the fakeness discrimination components unchanged.

Table 2: Impact of model components on the proposed framework.

Investigated component	Setting	Accuracy (%)
Automated feature extraction	$MMDF_{PCA}$	26.06
	$MMDF_{NN}$	30.95
Multi-source fusion	$MMDF_{Concat}$	28.69
	$MMDF_{EQ}$	32.08
Fakeness discrimination	$MMDF_{CE}$	31.11
	$MMDF_{CL}$	35.17
–	MMFD	**38.81**

Table 3: A case study demonstrating the interpretability of the proposed framework.

Source	Content	Attention score
Statement	”Virtually every person across this country has seen premiums going up and up and up due to Obamacare”	0.17
Metadata	Ted Cruz, Senator Texas, Republican a conversation with reporters	0.09
History	[36 33 15 19 8]	0.43
Report	The full report is available from the URL[1]	0.31

- $MMFD_{NN}$. For this configuration, we perform the same feature extraction as $MMFD_{PCA}$ except that instead of PCA, we extract features from a NN.

- $MMFD_{Concat}$. In this setting, we keep the automated feature extraction and the fakeness discrimination components. However, we replace the interpretable multi-source fusion component by concatenating extracted features.

- $MMFD_{EQ}$. Similar to $MMFD_{Concat}$, in this setting, we keep the automated feature extraction and the fakeness discrimination components. However, we replace the interpretable multi-source fusion component by explicitly considering all the sources equally important. In other words, in Eq. 4 all a_is are set to the same value.

- $MMDF_{CE}$. In this setting, we evaluate the effectiveness of the fakeness discrimination component. We keep two other components and remove MDF from the loss function of the model and just keep Cross Entropy (CE).

- $MMDF_{CE}$. Center Loss (CL) was proposed in (Wen et al., 2016). This function penalizes cross entropy by an intra-class sparsity. In other words, it just tries to move samples toward their class centers. In this setting, we use Center Loss and cross entropy.

The results of component analysis with all sources in the enriched dataset are presented in Table 2. It can be observed that:

- Automated feature extraction effectively extracts features. Neither $MMFD_{FCN}$ nor $MMFD_{PCA}$ can extract informative features compared to the automated feature extraction component.

- Hand-crafted features are ineffective. Features in $MMFD_{FCN}$ and $MMFD_{PCA}$ are basic linguistic features which are not effective as much as the automated ones from deep networks.

- The interpretable multi-source fusion component integrates features effectively. Simple concatenation methods such as $MMFD_{Concat}$ and $MMFD_{EQ}$ fail to distinguish different sources. The reason for better performance of $MMFD_{EQ}$ is because the linear projection presented in Eq. 3.

- MMFD can discriminate different classes effectively. Neither $MMDF_{CE}$ nor $MMDF_{CL}$ can effectively discriminate different classes. The reason for better accuracy of $MMDF_{CL}$ is due to the extra penalty added to the cross entropy.

In conclusion, the three major model components including automated feature extraction, multi-source fusion, and fakeness discrimination can help boosting the performance of the proposed framework in detecting fake news.

[1] http://www.politifact.com/truth-o-meter/statements/2013/oct/17/ted-cruz/sen-ted-cruz-says-premiums-have-gone-virtually-eve/

4.4 A case study

As mentioned before, the interpretable multi-source fusion components equips the proposed framework with a sort of interpretability. To investigate the interpretability power of the framework , we present a case study demonstrated in Table 3. In this case study, we randomly select a sample news item from the test set. The ground truth label of this news item is *False* and the model correctly predicts it as *False*. Then, we show the attention score of each source. As shown in Table 3, the highest score is given to the history source. This seems reasonable as the history of the speaker, Ted Cruz, regarding the number of previous false statements is quite high. Moreover, the report attains the second highest attention score. This seems reasonable as well because the report includes contextual detail about a news item. In addition, it is quite hard to predict the fakeness of news from the short news statement and the writer's profile; hence low scores are given to these two sources.

5 Related Work

There have been substantial number of works studying fake news and misinformation detection in recent years. One well-known approach is taking advantage of linguistic features. In their seminal work (DePaulo et al., 2003), DePaulo et al. shed light on cues of fake stories from physiological point of view. They pointed out that fake stories contain an unusual language. Li et al. (Li et al.,) discovered that linguistic features such as sentiments and singular pronouns are informative in online spam reviews. In (Qazvinian et al., 2011), the authors used unigram, bigram and Part-of-Speech (POS) features of tweets for rumor detection. Martinez-Romo and Araujo (Martinez-Romo and Araujo, 2013) used discrepancy and lack of semantic relation between the language of spam tweets and that of the websites redirected by those tweets. Kumar et al. (Kumar et al., 2016) showed that Wikipedia hoaxes tend to contain more words than genuine articles. Also, Linguistic Inquiry and Word Count (LIWC) (Pennebaker et al., 2015) has been employed to investigate the role of individual words in a document for deception detection (Ott et al., 2011). For a review of other linguistic features, you can refer to survey papers (Conroy et al., 2015; Heydari et al., 2015).

Despite the success of aforementioned works, we still lack a comprehensive and optimized set of features for fake news. This is even more stressed for short statements as they offer little to fake news detection. Hence, deep neural networks as automatic feature extractors have gained attention for fake news detection (Ruchansky et al., 2017). Closely related to our work is (Wang, 2017). Wang (Wang, 2017) introduced the dataset used for our evaluation. Further, he developed a combination of CNN and LSTM for modeling fake news detection. Our model is substantially different from that of Wang (Wang, 2017) because, 1) we provide an interpretable model component to combine multiple sources, (2) we propose a new discriminative function, MDF, to discriminate degrees of fakeness, (3) we supplemented the dataset and modeled the verdict reports.

6 Conclusion

In this study, we investigated the challenging task of multi-source multi-class fake news detection. The task faces several challenges – how to incorporate multiple sources and how to discriminate degrees of fakeness. To address these challenges, we proposed a coherent and interpretable framework MMFD, which incorporates automated feature extraction, multi-source fusion and fakeness discrimination. Through extensive experiments, we demonstrated that our model can effectively distinguish different degrees of the fakeness of news. In fakeness discrimination, we treated all classes the same. Thus, we would like to incorporate class differences by enforcing larger margin between certain classes. By doing so, several classes can be merged easily if needed, allowing for a less fine grained but possibly more precise detection. Another possible research direction is to incorporate more sources such as temporal information, social networks and user interactions.

Acknowledgments

This work is supported by the National Science Foundation (NSF) under grant number IIS-1714741 and IIS-1715940. We would like to thank Nazanin Donyapour for her help and support.

References

[Allcott and Gentzkow2017] Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36.

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

[Bengio et al.2013] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

[Conroy et al.2015] Niall J Conroy, Victoria L Rubin, and Yimin Chen. 2015. Automatic deception detection: methods for finding fake news. In *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*, page 82. American Society for Information Science.

[Del Vicario et al.2016] Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H Eugene Stanley, and Walter Quattrociocchi. 2016. The spreading of misinformation online. *Proceedings of the National Academy of Sciences*, 113(3):554–559.

[DePaulo et al.2003] Bella M DePaulo, James J Lindsay, Brian E Malone, Laura Muhlenbruck, Kelly Charlton, and Harris Cooper. 2003. Cues to deception. *Psychological Bulletin*, 129(1):74.

[Gupta et al.2014] Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. 2014. Tweetcred: Real-time credibility assessment of content on twitter. In *International Conference on Social Informatics*, pages 228–243. Springer.

[Heydari et al.2015] Atefeh Heydari, Mohammad ali Tavakoli, Naomie Salim, and Zahra Heydari. 2015. Detection of review spam: A survey. *Expert Systems with Applications*, 42(7):3634–3642.

[Hochreiter and Schmidhuber1997] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Kalchbrenner et al.2014] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

[Khurana and Intelligentie2017] Urja Khurana and Bachelor Opleiding Kunstmatige Intelligentie. 2017. The linguistic features of fake news headlines and statements.

[Kingma and Ba2014] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kumar et al.2016] Srijan Kumar, Robert West, and Jure Leskovec. 2016. Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes. In *Proceedings of the 25th international conference on World Wide Web*, pages 591–602. International World Wide Web Conferences Steering Committee.

[LeCun et al.2015] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.

[Li et al.] Jiwei Li, Myle Ott, Claire Cardie, and Eduard H Hovy. Towards a general rule for identifying deceptive opinion spam.

[Martinez-Romo and Araujo2013] Juan Martinez-Romo and Lourdes Araujo. 2013. Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Systems with Applications*, 40(8):2992–3000.

[Mikolov et al.2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Ott et al.2011] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 309–319. Association for Computational Linguistics.

[Pennebaker et al.2015] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. The development and psychometric properties of liwc2015. Technical report.

[Qazvinian et al.2011] Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599. Association for Computational Linguistics.

[Ruchansky et al.2017] Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. Csi: A hybrid deep model for fake news. *arXiv preprint arXiv:1703.06959*.

[Shu et al.2017a] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017a. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36.

[Shu et al.2017b] Kai Shu, Suhang Wang, and Huan Liu. 2017b. Exploiting tri-relationship for fake news detection. *arXiv preprint arXiv:1712.07709*.

[Wang2017] William Yang Wang. 2017. ”liar, liar pants on fire”: A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.

[Webb et al.2016] Helena Webb, Marina Jirotka, Bernd Carsten Stahl, William Housley, Adam Edwards, Matthew Williams, Rob Procter, Omer Rana, and Pete Burnap. 2016. Digital wildfires: hyper-connectivity, havoc and a global ethos to govern social media. *ACM SIGCAS Computers and Society*, 45(3):193–201.

[Wen et al.2016] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. 2016. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515. Springer.

[Wingfiled et al.2016] Nick Wingfiled, Mike Isaac, and Benner Kate. 2016. Google and facebook take aim at fake news sites, November. [Online; posted Nov. 14, 2016].

[Xu et al.2015] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057.

[Yang et al.2012] Lei Yang, Tao Sun, Ming Zhang, and Qiaozhu Mei. 2012. We know what@ you# tag: does the dual role affect hashtag adoption? In *Proceedings of the 21st international conference on World Wide Web*, pages 261–270. ACM.