

Suspicious News Detection Using Micro Blog Text

Tsubasa Tagami¹, Hiroki Ouchi², Hiroki Asano³, Kazuaki Hanawa⁴
 Kaori Uchiyama⁵, Kaito Suzuki⁶, Kentaro Inui⁷, Atsushi Komiya⁸
 Atsuo Fujimura⁹, Hitofumi Yanai¹⁰, Ryo Yamashita¹¹, Akinori Machino¹²
^{1,2,3,4,5,6,7}Tohoku University, ^{2,3,7}RIKEN, ^{8,9}SmartNews, Inc., ¹⁰FactCheck Initiative Japan
¹¹Watchdog for Accuracy inNews-reporting, Japan, ¹²Hi-Ether Japan
 {¹tagami, ³asano, ⁴hanawa, ⁷inui}@ecei.tohoku.ac.jp, ²hiroki.ouchi@riken.jp
 {⁵kaori.uchiyama.r1, ⁶kaito.suzuki.p3}@dc.tohoku.ac.jp
 {⁸atsushi.komiya, ⁹atsuo.fujimura}@smartnews.com
¹⁰yanai@fij.info, ¹¹rymst@gohoo.org, ¹²akinori.machino@hi-ether.org

Abstract

We present a new task, *suspicious news detection* using micro blog text. This task aims to support human experts to detect suspicious news articles to be verified, which is costly but a crucial step before verifying the truthfulness of the articles. Specifically, in this task, given a set of posts on SNS referring to a news article, the goal is to judge whether the article is suspicious or not. For this task, we create a publicly available dataset in Japanese and provide benchmark results by using several basic machine learning techniques. Experimental results show that our models can reduce the cost of manual fact-checking process.

1 Introduction

Fake news is a news article that is intentionally false and could mislead readers (Shu et al., 2017). The spread of fake news has a negative impact on our society and the news industry. For this reason, fake news detection and fact-checking are getting more attention.

Problematic Issue. One problematic issue of fake news detection is that human fact-checking experts cannot keep up with the amount of misinformation generated every day. Fact-checking requires advanced research techniques and is intellectually demanding. It takes about one day to fact-check a typical article and write a report to persuade readers whether it was true, false or somewhere in between (Hassan et al., 2015).

Existing Approach. As a solution to the problem, various techniques and computational models for automatic fact-checking or fake news detection have been proposed (Vlachos and Riedel, 2014; Wang, 2017; Hanselowski et al., 2018). However, in practice, current computational models for automatic fake news detection cannot be used yet now due to the performance limitation. Thus, at the present, manual or partially automatic verification is a practical solution.

Our Approach. To mitigate the problem, we aim to automate *suspicious news detection*. Specifically, we develop computational models for detecting suspicious news articles to be verified by human experts. We assume human-machine hybrid systems, in which suspicious articles are detected and sent to human experts and they verify the articles.

Our motivation of this approach is to remedy the time-consuming step to find articles to check. Journalists have to spend hours going through a variety of investigations to identify claims (or articles) they will verify (Hassan et al., 2015). By automatically detecting suspicious articles, we can expect to reduce the manual cost.

Our Task. We formalize suspicious news detection as a task. Specifically, in this task, given a set of posts on SNS that refer to a news article, the goal is to judge whether the article is suspicious or not. The reason of using posts on SNS is that some of them cast suspicion on the article and can be regarded as useful and reasonable resources for suspicious news detection.

This task distinguishes our work from previous work. In previous work, the main goal is to assess the truthfulness of a pre-defined input claim (or article). This means that it is assumed that the input claim is given in advance (Wang, 2017). As mentioned above, in real-world situations, we have to select the claims to be verified from a vast amount of texts. Thus, the automation of this procedure is desired for practical fact verification.

Our Dataset. For the task, we create a Japanese suspicious news detection dataset. On the dataset, we provide benchmark results of several models based on basic machine learning techniques. Experimental results demonstrate that the computational models can reduce about 50% manual cost of detecting suspicious news articles.

Our Contributions. To summarize, our main contributions are as follows,

- We introduce and formalize a new task, *suspicious news detection* using posts on SNS.
- We create a Japanese suspicious news detection dataset, which is publicly available.¹
- We provide benchmark results on the dataset by using several basic machine learning techniques.

2 Related Work

This section describes previous studies that tackle fake news detection. We firstly overview basic task settings of fake news detection. Then, we discuss several studies that share similar motivations with ours and deal with fake news detection on social media.

2.1 Task Settings of Fake News Detection

Typically, fake news detection or fact-checking is defined and solved as binary prediction (Pérez-Rosas et al., 2017; Volkova et al., 2017; Gilda, 2017) or multi-class classification (Wang, 2017; Hassan et al., 2017). In this setting, given an input text x , the goal is to predict an appropriate class label $y \in \mathcal{Y}$.

¹<https://github.com/t-tagami/Suspicious-News-Detection>

The input text x can be a sentence (e.g., news headline, claim or statement) or document (e.g., news article or some passage). The class labels \mathcal{Y} can be binary values or multi-class labels.

One example of this task is the one defined and introduced by the pioneering work, Vlachos and Riedel (2014). Given an input claim x , the goal is to predict a label y from the five labels, $\mathcal{Y} = \{\text{TRUE}, \text{MOSTLYTRUE}, \text{HALFTRUE}, \text{MOSTLYFALSE}, \text{FALSE}\}$.

Another example is a major shared task, Fake News Challenge. In this task, given a headline and body text of a news article, the goal is to classify the stance of the body text relative to the claim made in the headline into one of four categories, $\mathcal{Y} = \{\text{AGREES}, \text{DISAGREES}, \text{DISCUSSES}, \text{UNRELATED}\}$. A lot of studies have tackled this task and improved the computational models for it. (Thorne et al., 2017; Tacchini et al., 2017; Riedel et al., 2017; Zeng, 2017; Pfohl, 2017; Bhatt et al., 2017; Hanselowski et al., 2018).

One limitation of the mentioned settings is that the input text is predefined. In real-world situations, we have to select the text to be verified from a vast amount of texts generated every day.

Assuming such real-world situations, Hassan et al. (2017) aimed to detect important factual claims in political discourses. They collected textual speeches of U.S. presidential candidates and annotated them with one of the three labels, $\mathcal{Y} = \{\text{NON-FACTUAL SENTENCE}, \text{UNIMPORTANT FACTUAL SENTENCE}, \text{CHECK-WORTHY FACTUAL SENTENCE}\}$. There is a similarity between their work and ours. One main difference is that while they judge whether the target political speech is check-worthy or not, we judge the degree of the suspiciousness of the target article from the posts on SNS referring to the article.

2.2 Fake News Detection on Social Media

We aim to detect suspicious news using information on social media. There is a line of previous studies that share a similar motivation with our work.

Fake News Detection Using Crowd Signals

One major line of studies on fake news detection on social media leveraged crowd signals (Zhang et al., 2018; Kim et al., 2018; Castillo et al., 2011; Liu et al., 2016; Tacchini et al., 2017).

Tschiatschek et al. (2017) aimed to minimize the spread of misinformation by leveraging user’s flag activity. In some major SNS, such as Facebook and Twitter, users can flag a text (or story) as misinformation. If the story receives enough flags, it is directed to a coalition of third-party fact-checking organizations, such as Snoops² or FactCheck³. To detect suspicious news articles and stop the propagation of fake news in the network, Tschiatschek et al. (2017) used the flags as a clue. Kim et al. (2018) also aimed to stop the spread of misinformation by leveraging user’s flags.

Fake News Detection Using Textual Information

Another line of studies on fake news detection on social media effectively used textual information (Mitra and Gilbert, 2015; Wang, 2017; Tacchini et al., 2017; Pérez-Rosas et al., 2017; Long et al., 2017; Vo and Lee, 2018; Yang et al., 2018).

In particular, Volkova et al. (2017) is similar to our work. They built a computational model to judge whether a news article on social media is suspicious or verified. Also, if it is suspicious news, they classify it to one of the classes, satire, hoaxes, clickbait and propaganda. One main difference is that while the main goal of their task is to classify the input text, our goal is to detect suspicious news articles using SNS posts.

3 Tasks

Our main objective is to detect suspicious news articles to be verified. In this section, we firstly explain our motivation in Section 3.1 and our system that we assume in Section 3.2. Then, we propose and formalize the two tasks, (i) *suspicion casting post detection* in Section 3.3 and (ii) *suspicious article detection* in Section 3.4.

3.1 Motivative Situation

One example of fake news detection or fact-checking in the real-world situations is the activity of Watchdog for Accuracy in News-reporting, Japan (WANJ)⁴, Nonprofit Organization (NPO) in Japan. They verify news articles following the three manual steps.

²<http://www.snopes.com>

³<http://www.factcheck.org>

⁴<http://wanj.or.jp/>

1. Collect the posts on SNS that refer to news articles and select only the posts that cast suspicion on the articles.
2. Select suspicious articles to be verified by taking into account the content of each collected post and the importance of the articles.
3. Verify the content of each article, and if necessary, report the investigation result.

In the first step, they collect and select only the SNS posts that cast suspicion on news articles. We call them *suspicion casting posts* (SCP). Based on the selected SCP, in the second and third steps, the articles to be verified are selected, and the contents are actually verified by some human experts.

All these steps are time-consuming and intellectually demanding. Although full automation of them is ideal, it is not realistic at present due to the low performance for fact verification. Thus, in this work, we aim to realize partial automation to support human fact-checking experts.

What We Want to Do. We aim to automate suspicious article detection by leveraging SCP information. It is costly to collect only SCP from a vast amount of SNS posts generated every day. Not only time-consuming, it is sometimes challenging for computational models to tell SCP from others. Consider the following two posts.

- (a) この記事は誤報では？千代田区も路上喫煙はダメで過料が科されているはずです！
This article denotes misinformation, doesn’t it? If you had smoked on the street, you should have been fined in Chiyoda Ward!
- (b) 本当に信じられない。嘘であって欲しい。言葉が見つからないけどご冥福をお祈りします！
I really can not believe it. I wish it were a lie. I’m lost for words, but I’ll send my prayers!

While the post (a) casts suspicion on the article, the post (b) just mentions personal impression on it. Actually, only a few of the total SCP candidates are true SCP, which means that SCP detection is a heavy burden to human experts.

We develop computational models for SCP detection, and by using the results, we rank suspicious articles. We assume that the suspicious articles are sent to and verified by human experts in order of suspiciousness scores. In the following subsection, we describe the system that we assume.

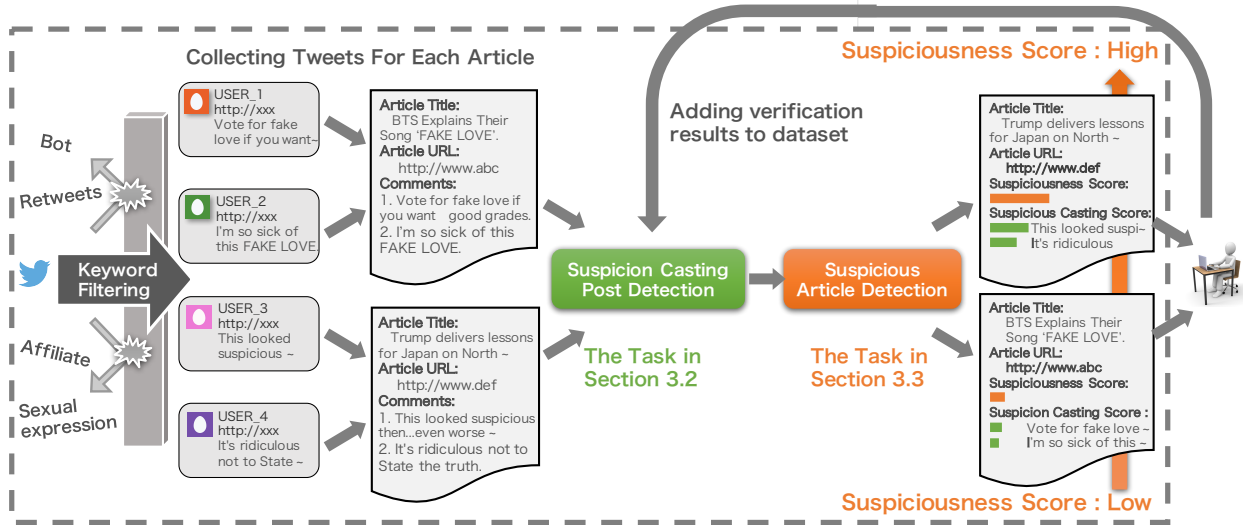


Figure 1: Overall architecture of our system.

3.2 Human-Machine Hybrid System

Our system integrates computational models with human fact-checking experts. Figure 1 illustrates the overall architecture of our system. This system consists of the five components.

1. **Filtering Component:** To collect and filter the posts on SNS referring to news articles.
2. **Arranging Component:** To arrange and put together the posts referring to the same article.
3. **Scoring Component:** To detect the posts that cast suspicion on the article and score the suspiciousness.
4. **Ranking Component:** To rank the articles based on the suspiciousness scores of each post.
5. **Verification Component:** To verify the articles by human experts.

For the third component, we build a scoring model by tackling a binary prediction task, SCP detection in Section 3.3. In this task, given a post, the goal is to judge whether the post is SCP or not. For the fourth component, we score and rank articles based on the SCP. We define a ranking task for it, suspicious article detection in Section 3.4. In the following subsections, we describe the task settings in detail.

3.3 Suspicion Casting Post Detection

As the example posts in Section 3.1 show, one challenge of detecting suspicion casting posts (SCP) is

that a lot of posts referring to an article do not cast suspicion and just mention personal impression on the article. Thus, a key to detecting SCP is how to capture linguistic expressions related to the truthfulness of articles.

Formal Setting

Given a post $x = (w_1, \dots, w_T)$ that consists of T words and refers to an article $a \in \mathcal{A}$, the goal is to judge whether the post casts suspicion on the article or not.

$$\begin{aligned} \text{INPUT} : x &= (w_1, \dots, w_T) \\ \text{OUTPUT} : y &\in \{0, 1\} \end{aligned}$$

y is a binary value, i.e., 1 represents that the post x is SCP and 0 otherwise.

Evaluation

To evaluate the performance for this task, we use precision, recall and F1 scores. If the prediction \hat{y} matches with the ground-truth y , we regarded it as correct.

3.4 Suspicious Article Detection

Formal Setting

Given an article a and $N^{(a)}$ posts referring to the article $X^{(a)} = \{x_i^{(a)}\}_{i=0}^{N^{(a)}}$, the goal is to judge whether

the article is suspicious or not.

$$\begin{aligned} \text{INPUT} : X^{(a)} &= \{x_i^{(a)}\}_{i=0}^{N^{(a)}} \\ \text{OUTPUT} : y^{(a)} &\in \{0, 1\} \end{aligned}$$

$x_i^{(a)}$ is each post, and $y^{(a)}$ is a binary value, i.e., 1 represents the article is suspicious and 0 otherwise.

Evaluation

Not only precision, recall and F1 scores, we evaluate the performance using a ranking criterion, $\text{Recall@}K$. In this work, since we assume that we send articles to human fact-checking experts in order of the suspiciousness scores, $\text{Recall@}K$ is suitable for evaluating the ability of models to properly rank the suspicious articles.

Specifically, $\text{Recall@}K$ evaluates the proportion of the correct suspicious articles in the top- K ranked ones,

$$\text{Recall@}K = \frac{1}{|T|} \sum_{1 \leq i \leq K} b_i ,$$

where T is the number of the total articles in the test set, and b_i is a binary value, i.e., 1 if the i -th ranked article is suspicious and 0 otherwise.

4 Methods

This section describes our methods for the two tasks formalized in the previous section.

Suspicion Casting Post Prediction

For SCP detection, we can simply predict y based on a binary prediction approach,

$$\text{P}_\theta(y = 1|x) = f_\theta(x) . \quad (1)$$

$y = 1$ represents the post x is SCP and 0 otherwise. Function f_θ with the parameters θ can be arbitrarily defined. In this paper, as the function f_θ , we use several models described in Section 6.1.

To train the model parameters θ , we use the binary cross-entropy loss function,

$$\mathcal{L}(\theta) = - \sum_{i=1}^N \ell_i , \quad (2)$$

$$\ell_i = \log \text{P}_\theta(y = 1|x) + \log (1 - \text{P}_\theta(y = 1|x)) .$$

Suspicious Article Prediction

For suspicious articles detection, we predict $y^{(a)}$ based on the SCP prediction score of each post. We firstly score each of the posts $x^{(a)} \in X^{(a)}$ referring to the article a . Then we use the highest score among them as the score of $y^{(a)}$. Specifically, we calculate the score of $y^{(a)}$ as follows,

$$\text{SCORE}(y^{(a)}) = \max_{x^{(a)} \in X^{(a)}} \text{P}_\theta(y = 1|x^{(a)}) . \quad (3)$$

Here, the SCP probability $\text{P}_\theta(y = 1|x^{(a)})$ can be calculated in the same way as Eq. 1. We determine that the article a is suspicious, i.e., $y^{(a)} = 1$, if $\text{SCORE}(y^{(a)})$ is greater than 0.5. The parameters θ are optimized by using the same loss function as the one for SCP prediction (Eq. 2).

5 Datasets

This section describes the procedure of our dataset creation. We created the two datasets, the one for suspicion casting post (SCP) detection and the other for suspicious article (SA) detection. Note that these two datasets are independent sets of posts, which means that they do not share the same posts with each other. In the following subsections, we explain the procedures in detail.

5.1 Dataset for Suspicion Casting Post Detection

First, we collected the posts on Twitter including the URL of a news article. Of these posts, we left only the posts that have the potential to cast suspicion by using specific keywords, such as *misinformation*, *fabrication* and *untrue*. In this work, we adopted the list of the keywords that is actually used for fact-checking by FIJ⁵, the third-party fact-checking organization in Japan. If the post contains any key words in the list, we regarded it as a candidate post and added it to the dataset.

Second, we preprocessed the collected posts. We want to leave only the comment part of a post except for some noises, such as hashtags, mentions and title of news articles. Thus, we removed the article title, URL and hashtags from posts. As a result, we obtained only the comment part other than noise from the original post.

⁵<http://fij.info/>

Suspicion Casting Post Dataset	
# Samples (pos / neg)	7,775 (1,036 / 6,739)
Avg. Length of Comments	56.6
Suspicious Article Dataset	
# Samples (pos / neg)	1,836 (564 / 1,272)
Avg. Length of Comments	60.4
Avg. Tweets / Article	2.75

Table 1: Statistics of our datasets. “pos” and “neg” denotes the number of positive (i.e. suspicious casting posts or suspicious articles) and negative samples, respectively.

Finally, to each collected post, we annotated 1 if the post casts suspicion and -1 otherwise. For example, the post (a) in Section 3.1 is annotated as 1 because it casts suspicion on the article. By contrast, the post (b) is annotated as -1 because it is regarded as the one that just mentions personal impression. The upper part of Table 1 indicates the statistics of this dataset. The number of samples are 7,775, in which 1,036 are positive and 6,739 are negative samples.

5.2 Dataset for Suspicious Article Detection

First, we collected a set of the posts referring to the same article (URL). Second, we preprocessed and annotated the posts in the same way as in the SCP dataset creation. Finally, we annotated 1 to the article if a set of posts referring to the article includes at least one SCP post, and -1 otherwise. The value 1 means that the article is suspicious and to be verified by human experts, and -1 is not. The lower part of Table 1 indicates the statistics of this dataset. The number of samples are 1,836, in which 564 are positive and 1,272 are negative samples.

6 Experiments

This section provides the benchmark results on our datasets. Since our datasets have imbalanced class distributions, we used stratified 5-fold cross-validation to keep the distributions between true and false labels consistent in the train, development and test sets.

6.1 Experimental Setup

Models

We built and used the five models based on the following machine learning techniques.

- (a) **Logistic Regression (LR)**: An L1 regularized logistic regression classification model. The hyper-parameter C , representing inverse of regularization strength, was set to 20.
- (b) **SVM**: A support vector machine classification model (Cortes and Vapnik, 1995; Chang and Lin, 2011) using the radial basis function kernel (RBF). The penalty parameter C for the error term was set to 3000.
- (c) **Decision Tree (DT)**: A decision tree classification model (Quinlan, 1986; Quinlan and Rivest, 1989). The maximum depth of the tree parameter was set to 30.
- (d) **Random Forest (RF)**: A random forest classification model (Breiman, 2001). The maximum depth of the tree parameter was set to 15. The number of features used for prediction was set to 300. The number of trees in the forest was set to 90.
- (e) **LSTM**: A Long Short Term Memory (LSTM) network based classification model (Hochreiter and Schmidhuber, 1997; Gers et al., 2000). Every tweet is represented as a sequence of word vectors and fed to the LSTM layer whose hidden units was set to 200. Then the averaged hidden unit vector is fed to the output layer with softmax activation function. The hyperparameters of this model are described in more detail in Table A in the Appendix Section.

Implementation Details

Parameters of these models were set by using cross-validation on the development set. We used the default settings for unspecified hyper-parameters.

We implemented the LR, SVM, DT and RF models using scikit-learn (Pedregosa et al., 2011). As the features for these four models, we used unigram and bigram word features. Also, we implemented the LSTM model by using Keras (Chollet and others, 2015). As the features for the LSTM model, we used word embeddings trained on 4.5M tweets using Word2Vec CBOW model (Mikolov et al., 2013a; Mikolov et al., 2013b). The vocabulary size of the embeddings is about 80,000. The hyper-parameters used for Word2Vec are shown in Table A in the Appendix Section.

Method	Precision	Recall	Micro-F1
Logistic Regression	0.61	0.51	0.56
SVM	0.61	0.49	0.55
Decision Tree	0.45	0.54	0.49
Random Forest	0.62	0.37	0.46
LSTM	0.48	0.61	0.54

Table 2: Results for suspicion casting post detection.

Method	Precision	Recall	Micro-F1
Logistic Regression	0.74	0.61	0.67
SVM	0.75	0.60	0.67
Decision Tree	0.61	0.60	0.61
Random Forest	0.70	0.51	0.59
LSTM	0.60	0.74	0.66

Table 3: Results for suspicious article detection.

6.2 Results for Suspicion Casting Post Detection

Table 2 indicates the results for suspicion casting post detection on the test set. Overall, the logistic regression, SVM and LSTM models yielded higher F1 scores than those of the decision tree and random forest models, and achieved competitive performance with each other. While some previous studies reported that LSTM-based model work better than other discrete feature based models in text classification tasks similar to ours (Tang et al., 2015; Lee and Dernoncourt, 2016), our LSTM-based model yielded almost the same F1 scores as those of logistic regression and SVM models. One possible explanation for it is that while LSTM requires larger size of training samples, our dataset is relatively small.

6.3 Results for Suspicious Article Detection

Table 3 indicates the result for suspicious article detection. Similarly to the results in SCP detection, the logistic regression, SVM and LSTM models achieved higher scores than the other two models.

Figure 2 shows the Recall@ K curve for each model. Most of the models achieved 80% recall at the top 750 ranked articles, which corresponds to 40% of the total articles. This means that by checking the top 40% ranked articles, we can collect 80% suspicious articles to be verified. Thus, our models can efficiently reduce the manual cost of selecting suspicious articles.

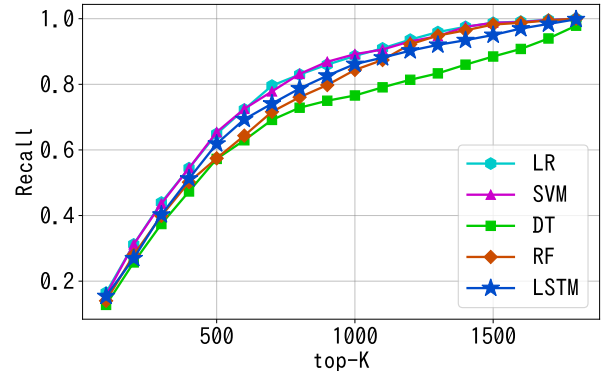
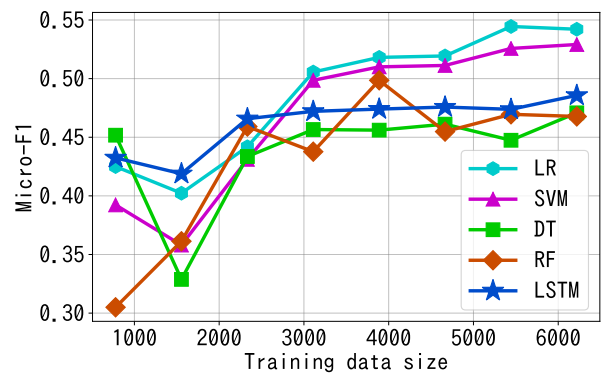
Figure 2: Recall@ K in suspicious article detection.

Figure 3: Performance curves of each model according to the size of the training set.

6.4 Analysis

Performance Curve

To better understand the models and benchmark results, we analyzed how the performance changes according to the size of the training set. Figure 3 shows the performance curve of each model. An overall tendency we observed is that the micro-F1 scores got improved as the number of training data increased. This result suggests that there is room for performance improvements by increasing the training data size. As an interesting future direction, we plan to increase the data size by crowdsourcing.

Error Examples

To shed light on the tendency of what post is difficult to predict in SCP detection, we analyze the predicted results. Table 6.4 shows the examples of the predictions.

	Tweet	Answer	Prediction
(1)	これは全くの誤報、増えたのは単純労働に従事する技能実習生と留学生だろう This is completely misinformation because what has increased is the number of technical intern and exchange students for manual labor.	+1	+1
(2)	とうとうニュースソースきちちゃったの... 誤報であって欲しかった At last, the news source has got clear... I wished it had been misinformation	-1	+1
(3)	反体制派の一部に戦争犯罪があったのはかねて報道されていた通りであり、その点で記述が間違いではないのですが、戦争犯罪のレベルは天地の差があり、このタイトルはミスリード As it have been reported for a long time, the description that a part of the dissidents committed a war crime is not wrong, but since the level of the war crime was so different from the reported one, this title can mislead readers.	+1	-1

Table 4: Analysis on model predictions. The column "Answer" denotes the correct labels, and the column "Prediction" denotes the model predictions.

The post of example (1) points out that the article is misinformation. All the models correctly predicted that this post is an SCP one (+1). We observed that if posts contain some key phrases, such as "misinformation" and "false," the models tend to predict that they are SCP.

By contrast, all the models made wrong predictions on the post of example (2). Like the post of example (1), this post also contains a key phrase "misinformation." However, this post is not an SCP one (-1) because it just expresses the user's desire by the phrase "I wished it had been misinformation." It is difficult for the basic models to correctly capture the meaning of the sentence-level structure.

Similar tendencies were observed in other examples. The post of example (3) is an SCP because it denotes the title of the article can mislead readers, but all the models wrongly judged it is not an SCP. While this post points out that the title of the article can mislead, the post also partially acknowledges the truthfulness of the content of the article by the phrase "the description ... is not wrong." This could lead to the wrong predictions. Since the models mainly used word-level features, it is difficult for them to properly capture sentence-level meanings.

7 Conclusion

To support human fact-checking activity, we have tackled the automation of suspicious news detection.

Summary. To detect suspicious articles to be verified, this paper has formalized and tackled two

tasks, *suspicion casting post detection* and *suspicious article detection*. For these tasks, we have created the first publicly available dataset. On the dataset, we have provided benchmark results using several basic machine learning techniques. The experimental results have demonstrated that we can cover most of the suspicious articles by checking only the top ranked 40% of the total articles.

Future Direction. One of our future directions is to use more sophisticated models for our tasks. Since our main objective of this work is to provide benchmark results on the datasets, we did not use complex models. To develop systems that work well in real-world situations, it is an interesting future research to propose better models and integrate them into the systems.

Another future direction is to increase the dataset size. As the analysis in Section 6.4 suggests, there is room for performance improvements by using more training samples. We plan to increase the dataset by leveraging human experts' feedback. In our system, human experts verify each predicted suspicious article. In this process, we can ask the experts to correct the model predictions if they are wrong, and can add the articles and their corrected annotations (labels) to the dataset.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP15H01702. We thank the anonymous reviewers for many helpful advices and comments.

References

- Gaurav Bhatt, Aman Sharma, Shivam Sharma, Ankush Nagpal, Balasubramanian Raman, and Ankush Mittal. 2017. On the benefit of combining neural, statistical and external features for fake news identification. *CoRR*, abs/1712.03935.
- Leo Breiman. 2001. Random forests. *Machine Learning*, 45:5–32.
- Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *WWW*.
- Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: A library for support vector machines. *ACM TIST*, 2:27:1–27:27.
- François Chollet et al. 2015. Keras. <https://keras.io>.
- Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20:273–297.
- Felix A. Gers, Jürgen Schmidhuber, and Fred A. Cummins. 2000. Learning to forget: Continual prediction with lstm. *Neural computation*, 12 10:2451–71.
- S. Jerril Gilda. 2017. Evaluating machine learning algorithms for fake news detection. *2017 IEEE 15th Student Conference on Research and Development (SCORED)*, pages 110–115.
- Andreas Hanselowski, S. AvineshP.V., Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, and Iryna Gurevych. 2018. A retrospective analysis of the fake news challenge stance detection task. *CoRR*, abs/1806.05180.
- Naeemul Hassan, Bill Adair, James T Hamilton, Chengkai Li, Mark Tremayne, Jun Yang, and Cong Yu. 2015. The quest to automate fact-checking. *world*.
- Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017. Toward automated fact-checking : Detecting check-worthy factual claims by claimbuster.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9 8:1735–80.
- Jooyeon Kim, Behzad Tabibian, Alice Oh, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. 2018. Leveraging the crowd to detect and reduce the spread of fake news and misinformation. In *WSDM*.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In *HLT-NAACL*.
- Qiang Liu, Shu Wu, Feng Yu, Liang Wang, and Tieniu Tan. 2016. Ice: Information credibility evaluation on social media via representation learning. *CoRR*, abs/1609.09226.
- Yunfei Long, Qin Lu, Rong Xiang, Minglei Li, and Churen Huang. 2017. Fake news detection through multi-perspective speaker profiles. In *IJCNLP*.
- Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *NIPS*.
- Tanushree Mitra and Eric Gilbert. 2015. Credbank: A large-scale social media corpus with associated credibility annotations. In *ICWSM*.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2017. Automatic detection of fake news. *CoRR*, abs/1708.07104.
- Stephen R. Pfohl. 2017. Stance detection for the fake news challenge with attention and conditional encoding.
- J. Ross Quinlan and Ronald L. Rivest. 1989. Inferring decision trees using the minimum description length principle. *Inf. Comput.*, 80:227–248.
- J. Ross Quinlan. 1986. Induction of decision trees. *Machine Learning*, 1:81–106.
- Benjamin Riedel, Isabelle Augenstein, Georgios P. Spithourakis, and Sebastian Riedel. 2017. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *CoRR*, abs/1707.03264.
- Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36.
- Eugenio Tacchini, Gabriele Ballarin, Marco L. Della Vedova, Stefano Moret, and Luca de Alfaro. 2017. Some like it hoax: Automated fake news detection in social networks. *CoRR*, abs/1704.07506.
- Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*.
- James Thorne, Mingjie Chen, Giorgos Myrionthous Jishu Pu, Xiaoxuan Wang, and Andreas Vlachos. 2017. Fake news detection using stacked ensemble of classifiers.
- Sebastian Tschiatschek, Adish Singla, Manuel Gomez-Rodriguez, Arpit Merchant, and Andreas Krause. 2017. Detecting fake news in social networks via crowdsourcing. *CoRR*, abs/1711.09025.

- Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *LTCSS@ACL*.
- Nguyen Vo and Kyumin Lee. 2018. The rise of guardians: Fact-checking url recommendation to combat fake news. In *SIGIR*.
- Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Oken Hodas. 2017. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In *ACL*.
- William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of ACL*, pages 422–426.
- Yang Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, and Philip S. Yu. 2018. Ti-cnn: Convolutional neural networks for fake news detection. *CoRR*, abs/1806.00749.
- Qi Zeng. 2017. Neural stance detectors for fake news challenge.
- Jiawei Zhang, Limeng Cui, Yanjie Fu, and Fisher B. Gouza. 2018. Fake news detection with deep diffusive network model. *CoRR*, abs/1805.08751.

Appendix

A Hyper-Parameters

Hyper-parameter	Values
Embedding size	300
Window size	7
Minimum count	20
Subsampling frequency	0.00001
Negative samples size	5
Epochs to train	5

Table 5: Hyper-parameters for Word2Vec training.

Hyper-parameter	Values
Embedding size	300
Batch size	100
Max epoch	50
Optimizer	Adam (Kingma and Ba, 2014)
Adam α	{0.002, 0.9, 0.009}

Table 6: Hyper-parameters for the LSTM model.