B8IM2029

Supporting the Detection of Suspicious News through Information Extraction from Microblogs

Tsubasa Tagami

February 3, 2020

Graduate School of Information Sciences
Tohoku University

A Master's Thesis
submitted to System Information Sciences,
Graduate School of Information Science,
Tohoku University
in partial fulfillment of the requirements for the degree of
MASTER of INFORMATION SCIENCE

Tsubasa Tagami

Thesis Committee:

Professor Kentaro Inui	(Supervisor)
Professor Yoshifumi Kitamura	
Professor Kazuyuki Tanaka	
Associate Professor Jun Suzuki	(Co-supervisor)

Supporting the Detection of Suspicious News through Information Extraction from Microblogs*

Tsubasa Tagami

Abstract

False information has become a social problem and it is necessary to verify huge information such as news articles and social media sites on the Internet. In fact, there are a wide range of information to be verified, and it is impossible to verify all of them manually. For this reason, 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 to be verified 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. In addition, we developed a web application to support manual fact-checking activities. We report the results of the survey using this application in the actual worksite of fact-checking.

Keywords:

Natural Language Processing, Information Extraction, Fact-Checking

*, System Information Sciences, Graduate School of Information Sciences, Tohoku University, B8IM2029, February 3, 2020.

マイクロブログからの情報抽出に基づく疑義言説の検出支援*

田上 翼

内容梗概

国内外を問わず誤情報の拡散が社会的な問題となっており，情報の真偽を検証するフェクトチェックの必要性が急速に高まっている．しかしながら検証の対象となる情報は政治分野だけでなく多岐に渡っており，日々発信される大量の情報のすべてを人手で検証することは不可能である．また，人手の検証が必要な疑わしい情報を収集する過程は，不可欠であるがそれ自体膨大な時間を要する作業となっていることが深刻な問題となっている．そこで本研究ではマイクロブログからの情報抽出に基づいた，検証を必要とする疑わしい情報を検出するタスクを提案し，人手による検証に至る前のこの過程における負担を軽減することを目的とする．具体的には，ニュース記事や言説に対する SNS における投稿から情報を抽出し，人手による検証を必要とするかを機械学習を用いて自動的に判定する．また本研究では，タスク遂行のための公的に利用可能であるデータセットを作成し，基本的な機械学習の手法を用いた支援のためのシステムを構築した．作成したシステムを用いることで，ファクトチェックを必要とする情報を収集する作業の効率化を期待できることが確かめられた．また本システムを実際にファクトチェックを行っている現場に導入し，運用することで得られた成果について報告する．

キーワード

自然言語処理，情報抽出，ファクトチェック

*東北大学 大学院情報科学研究科 システム情報科学専攻 , B8IM2029, 2020 年 2 月 3 日.

Contents

List of Figures

List of Tables

1 Introduction

Fake news is a news article that is intentionally false and could mislead readers [1]. 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 [2].

Existing Approach. As a solution to the problem, various techniques and computational models for automatic fact-checking or fake news detection have been proposed [3, 4, 5]. However, in practice, current computational models for automatic fake news detection cannot be used yet now due to the performance limitation. Ideally, we would like to adopt full automation for verifying the contents of each article. 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 [2]. In the current situations, human experts often check the articles that are not necessary to check. 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 [4]. As mentioned above, in real-world situations, we have to select the claims to be verified from a vast amount of texts. In the context of fake news detection, it is costly to decide which article to be verified. 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. To the best of our knowledge, this is first publicly available dataset in Japanese. 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.
- We have introduced our proposed method in the actual fact-checking activities and report the fruits of our method.

[1] https://github.com/t-tagami/Suspicious-News-Detection

2 Related Work

This section describes previous studies that tackle fake news detection. In the last few years, so many works have presented various task settings, methods, and datasets for 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. We aim to clarify the similarities and differences between our work and previous works.

2.1 Task Settings of Fake News Detection

Typically, fake news detection or fact-checking is defined and solved as binary prediction [6, 7, 8] or multi-class classification [4, 9]. In this setting, given an input text x, the goal is to predict an appropriate class label $y \in \mathcal{Y}$. 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, [3]. Given an input claim x, the goal is to predict a label y from the five labels, $\mathcal{Y} = \{\textsc{True}, \textsc{MostlyTrue}, \textsc{HalfTrue}, \textsc{MostlyFalse}, \textsc{False}\}$.

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, $\mathcal{Y} = \{\textsc{Agrees}, \textsc{Disagrees}, \textsc{Discusses}, \textsc{Unrelated}\}$. A lot of studies have tackled this task and improved the computational models for it. [10, 11, 12, 13, 14, 15, 5]. A recent work [16] has extended the typical setting by integrating evidence retrieval.

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, [9] 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, $\mathcal{Y} = \{\textsc{Non-Factual Sentence}, \textsc{Unimportant Factual Sentence}, \textsc{Check-Worthy Factual}$

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 referreing 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 [17, 18, 19, 20, 11].

[21] 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 Snoops[2] or FactCheck[3]. To detect suspicious news articles and stop the propagation of fake news in the network, [21] used the flags as a clue. [18] 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 [22, 4, 11, 6, 23, 24, 25]. [25] proposed a Convolutinal Neural Network model which can combine the text and image information for fake news detection. [11] seeked to judge whether each post on Facebook is hoax or not. They collected 15,550 posts (8,923 are hoaxes and 6,577 not hoaxes). The methods are Logistic regression and boolean label crowdsourcing.

In particular, [7] 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

[2]http://www.snopes.com
[3]http://www.factcheck.org

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.

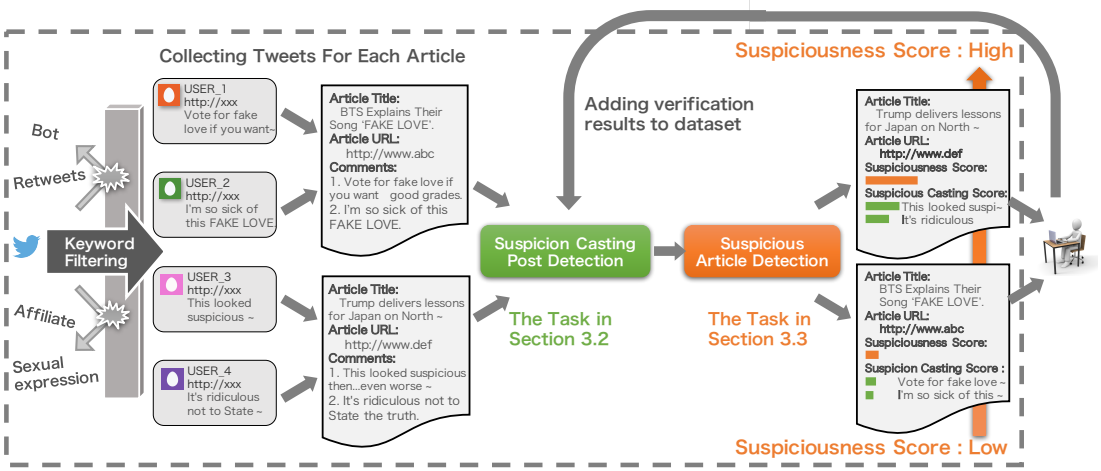

Figure 1: Overall architecture of our system.

3 Tasks

Our main objective is to detect suspicious news articles to be verified. We call such articles *suspicious articles* (SA). 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 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.

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.

[4]Established in June 2017 to support fact-checking activities and acquired an NPO corporation in January 2018. http://wanj.or.jp/

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 follwoing 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. Acctually, 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 resutls, we rank suspicous 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.

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 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 $x = (w_1, \cdots, w_T)$ 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.

$$\begin{aligned} \textsc{Input} &: x = (w_1, \cdots, w_T) \\ \textsc{Output} &: y \in \{0, 1\} \end{aligned}$$

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 gound-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.

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

$x_i^{(a)}$ 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, Recall@K. In this work, since we assume that we send articles to human fact-checking experts in order of the suspiciousness scores, Recall@K is suitable for evaluating the ability of models to properly rank the suspicious articles.

Specifically, Recall@K evaluates the propotion of the correct suspicious articles in the top-K ranked ones,

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

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,

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

$y = 1$ 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,

$$\begin{aligned} \mathcal{L}(\theta) &= -\sum_{i=1}^{N} \ell_i \ , \\ \ell_i &= \log \ \mathrm{P}_\theta(y = 1|x) + \log \left(1 - \mathrm{P}_\theta(y = 1|x)\right) \ . \end{aligned} \tag{2}$$

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,

$$\textsc{Score}(y^{(a)}) = \max_{x^{(a)} \in X^{(a)}} \mathrm{P}_\theta(y = 1|x^{(a)}) \ . \tag{3}$$

Here, the SCP probability $\mathrm{P}_\theta(y = 1|x^{(a)})$ can be calculated in the same way as Eq. 1. We determine that the article a is suspicious, i.e., $y^{(a)} = 1$, if $\textsc{Score}(y^{(a)})$ 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. We want only the posts that cast suspicion on the article. However, many posts do not mention any suspicion and just mention personal impressions on the 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 FIJ[5], 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. These noises are undesirable for analysis of tweets because it may affect prediction. 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.

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] 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.

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 [26, 27] 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 [28, 29]. The maximum depth of the tree parameter was set to 30.

(d) Random Forest (RF): A random forest classification model [30]. 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 [31, 32]. 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.

Method	Precision	Recall	F1-score
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.

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 [33]. As the features for these four models, we used unigram and bigram word features. Also, we implemented the LSTM model by using Keras [34]. As the features for the LSTM model, we used word embeddings trained on 4.5M tweets using Word2Wec CBOW model [35, 36]. 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.

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 compititive 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 [37, 38], 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. This suggests that a simple logistic regression is more suitable method for this prediction task than other methods that using unigram and bigram features. Furthermore, this result shows that using complex LSTM method with word representations in vector space is not

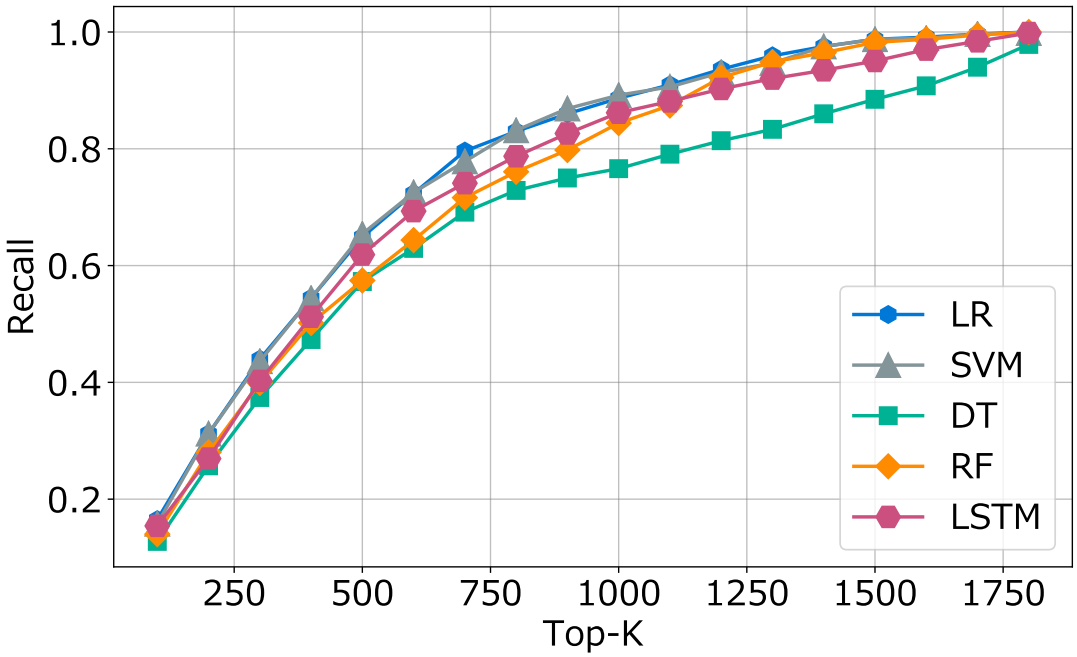

Figure 2: Recall@K in suspicious article detection.

suitable for this task with the current dataset.

6.3 Results for Suspicious Article Detection

Table 3 indicates the result for suspicious article detection. Similarily to the results in SCP detection, the logistic regression, SVM and LSTM models achieved higher scores than the other two models. One possible explanation is that because we defined an article that requires verifying as pointed out article about it's reliability by even one tweet. This result suggests that high-recall models such as LSTM tend to find the verification-required articles more effectively than high-precision models. This suggest that a simple logistic regression and svm classification models are the most suitable method to find the verification-required articles using tweets that refer to.

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%

Method	Precision	Recall	F1-score
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.

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 suspicios articles.

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 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.

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 "misin-

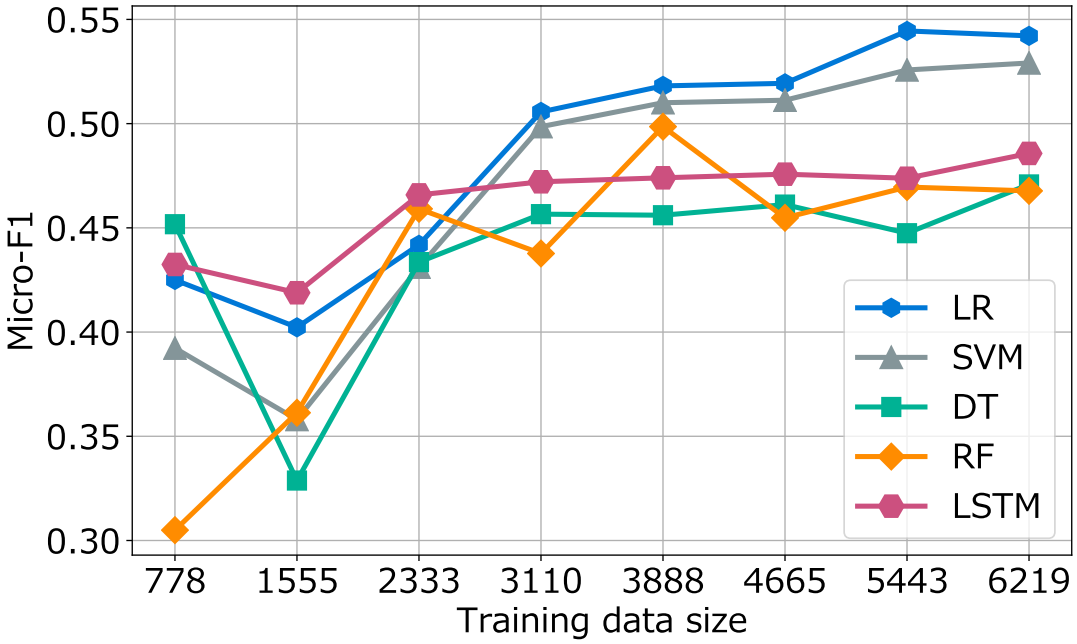

Figure 3: Performance curves of each model according to the size of the training set.

formation." 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.

	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 commited 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.

Figure 4: Web application interface.

7 Application

This section provides examples of applying our proposed methods to the real world fact-checking activities. To make effective use of our methods, we developed a web application with a logistic regression model described in Section 6. The reason for using this model is that fast to train, good classification performance and possible to interpret the coefficients. This application aims to suggest suspicious articles to the user, and the interface is as shown in Figure 4. It provides users with both articles that are predicted to be suspicious and posts on SNS that cast suspicion on the articles at the same time. When users search for suspicious articles with this application, they can easily label each article as SA or not and each post as SCP or not. By adding these labeled data to the training

data for machine learning, further improvement in classification performance can be expected as describe in Section 6.4.

FactCheck Initiative Japan (FIJ)[6], a joint researcher, is actually utilizing this application for daily fact-checking activities nearly two years. They usually publish about 3 to 4 fact-checking results per week for the collected suspicious articles. In addition to these daily activities, they conducted fact-checking projects during the elections that drew public attention. To give an example of those election, Okinawa gubernatorial election in 2018 and Japanese House of Councillor selection in 2019. Here we describe the results of these two fact-checking projects and analyses.

7.1 Okinawa Gubernatorial Election, 2018

The 12th Okinawa gubernatorial election was held from 1 September to 3 October 2018 to choose the next Governor. A huge amount of false information that distracts voters was spread on the internet and newspapers during the election period. As an example of false information, a certain web news site had reported that one of the candidates was using cannabis and lying about his career. And then such information was quickly and widely spread on social media sites such as Twitter and Facebook throughout the election period. For this reason, the candidate was obliged to issue a statement that such rumors were groundless false information.

To investigate whether the information spreading in society is based on facts or not and to share accurate information, FIJ conducted a fact-checking project[7] for the first time in Japan. The Ryukyu Shimpo[8], a media member of the Japan Newspaper Association, participated in this project, and 26 members including employees and reporters participated as support members. Throughout this project, our application has picked up about 100 suspicious articles per day. We selected articles that are likely to have a social impact, and finally they con-

[6]FIJ was founded in June 2017 by academics, journalists, a lawyer, and a tech company to encourage and support journalists, media outlets, and others to fact-check widespread questionable information.

[7]https://archive.fij.info/wordpress/project/okinawa2018/outline

[8]https://ryukyushimpo.jp/

ducted a manual fact-checking on 94 suspicious articles. Looking at the details of providers of these articles, 23 of them are politicians and candidates, 32 are journalists and media professionals and 39 are general public. As a result of fact-checking, we found that 14 of them are false or misleading information. This suggests that the application helps to manual fact-checking activities and our proposed method is effective for detecting suspicious articles.

Here we describe the detail of false information detected through our application. To give an example, a newspaper company published an article that criticizing a particular candidate and it was widely spread among voters. The article has shown that when the candidate was the mayor, he was elected as a pledge for free school lunch, but the price increased as a result. The day after this article was published, our application suggested that the article is suspicious based on the following posts.

(a) 佐喜真候補が「当選したら給食費を値上げした」というのもミスリードです。両陣営、虚偽の内容を含む宣伝が出てきてますね
It is also misleading information that the candidate Sakima raised the school lunch fee after winning the election. There is propaganda that contains false information about both sides.

(b) 佐喜真さんに対して給食費が値上がりしたとフェイクが流れてるね。実際は保護者負担半額になっていますよ。
It is false information that the school lunch fee has risen since the candidate Sakima become the mayor. As a matter of fact, the burden on parents was reduced by half.

As a result of fact-checking the article collected through our application, it turned out that the article is false information. Through this fact-checking project, we could obtain a number of such suspicious articles. For this reason, we could confirm that our developed application contributed to the fact-checking activities.

7.2 Japanese House of Councillor Selection, 2019

The 25th Japanese House of Councillor selection was held from July 4 to 21 to elect 124 of 245 members of House of Councillors. Also in this election, the

spread of false information was confirmed by fact-checking activities. FIJ also conducted a fact-checking project[9] to prevent the spread of false information and share correct information. Throughout the duration of this project, we collected 72 suspicious information that needs to be verified manually. Looking at the provider of this suspicious information, 19 of them are politicians, 13 of them are the media, 9 are famous persons and 31 are general public. As a result of fact-checking, we found that 10 of them are false or misleading information. Although this election was a national election, there was less false information spread than the Okinawa gubernatorial election.

Here we explain the false information that was characteristic of this election. Prime Minister Shinzo Abe mentioned the profits of pension reserves on the political broadcast, and said the Liberal Democratic Party has increased profits 10 times that of the Democratic Party. This statement was quickly spread on multiple social media sites, but at the same time, our application suggested that this is suspicious information based on the following post.

(a) 「10 倍」と強調するために都合のいい数字をわざわざつまみ食いしている。誇張というより嘘でしょう。
In order to emphasize "10 times", he mentions only favorable statistical data. It's a lie rather than an exaggeration.

(a) 民主党時代より運用益は増えていることは間違いないが、「10 倍」ではない。
There is no doubt that investment profits have increased since the Democratic Party, but it is not "10 times".

As a result of fact-checking, it was turned out that his statement was groundless false information. As you can see from this example, we found that our application is effective even for politician remarks that do not reported as a news article. Like this example, we confirmed through this project that we were also able to collect false information from the contents of the candidate's street speeches that were not in the article.

[9]https://fij.info/archives/category/factchecks/sangiin2019

8 Further Experiments

We have been operating the web application described in section 7 on the actual daily fact-checking activities from January 2018 to the present. In daily activities, we check in order from articles with a high probability of suspicious articles. This application collects more than 10,000 articles per day. However, we can only check and label the top 100 articles at best due to lack of manpower. Table 5 shows the number of labeled data collected each month by this daily activity. It shows large variations in the number of data collected every month. The reason is not there are few articles suggested by the application, but because there is a lack of human resources to utilize the application. This section provides the results of experiments using this collected data.

8.1 Evaluating the Effectiveness of Data Expansion

In Section 6.4, we showed that an improvement in classification performance can be expected as the data increases. We have expanded the training data using the application, so we investigate whether the classification performance increases or not using these collected data. However, the collected data is only articles with a high probability of being suspicious, so this is not suitable for test data to measure classification performance. To address this problem, we created a new test data by randomly sampling articles published in September 2019 and labeling the articles and posts that make mention of it. The statistics of this test data are shown in Table 6. As you can see from this table, the suspicious articles are only about 2% of the total articles, and suspicion casting post is only about 5% of the total posts.

We added the collected data for each month to the training data, and analyzed how the classification performance in suspicion casting post detection has changed. Figure 5 shows the performance curve in suspicion casting post detection. We can observe that the precision, recall and F1 scores got improved as the number of training data increased. This result shows that it is effective in classification to retrain the model using manually labeled data. The reason for the slight decrease in performance from December 2018 to January 2019 is probably due to a major change in application specifications. Until December 2018,

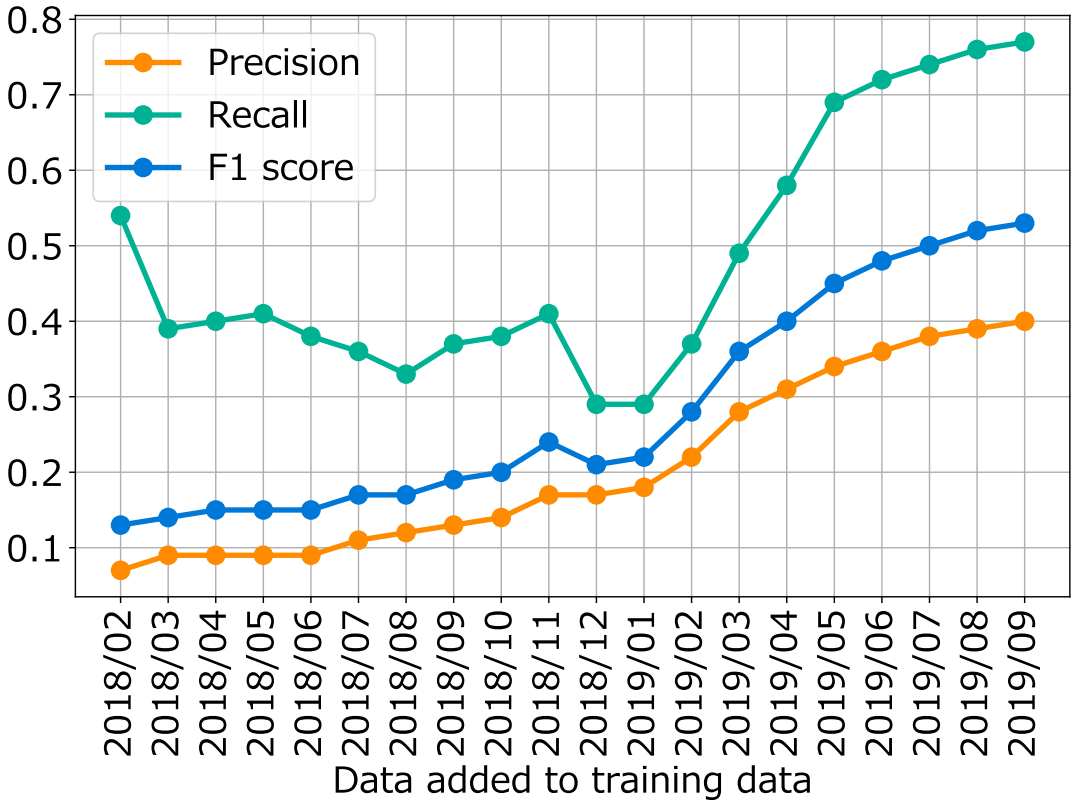

Figure 5: Performance curve in suspicion casting post detection.

we collected only suspicious articles from newspaper companies, but we started to collect suspicious articles from the general public the following month.

8.2 Evaluating the Effectiveness of Article Metadata

For suspicious article detection, we predict whether an article is suspicious using only the suspicion casting post prediction score described in Section 4. Moreover, we consider only the highest suspicion casting post score among posts as mentioned in Equation 3. This indicates that our method is unable to take into account the article provider or the public attention to the article. We are facing a big problem that articles with low importance provided by the general public are ranked higher in our application. To tackle this problem and boost classification

performance, we used three metadata of the article for suspicious article detection. The first one is the number of posts that mention the article. If this number is large, the public attention to the article is considered high. The second one is the number of the verified account that refers to the article. A verified account means an account of a specified real individual or organization, and post from such an account is believed to have a significant impact. The last one is the importance of the provider of the article. We consider newspapers with a national circulation are the most important, local newspapers are the second most important, and the others are the least important. In addition to these metadata, we use the distribution of suspicion casting post scores of posts referring to articles as a feature for machine learning.

Our goal is to measure the impact of using metadata on the performance of suspicious article detection. We use a logistic regression classifier to interpret the relationship between the response variable and explanatory variables. The training data we will use is collected data using the application, and the test data is the data created for evaluation in Section 8.1. For this experiment, we use precision and recall at k as metrics to give insight into the impact on classification performance. This is because when we use the application, we can only check the top of the ranking due to lack of manpower, so precision and recall at the top of the ranking is important.

Figure 6 shows the precision and recall at k of a method using only the top of the suspicion casting post score and a method of logistic regression using metadata. We can see that when the value of k is small, metadata of an article can increase precision and recall. This suggests that using metadata for prediction is suitable for scoring articles for fact-checking activities. In addition, as a result of examining the relationship between the response variable and explanatory variables, the number of verified accounts is the most influential metadata for prediction.

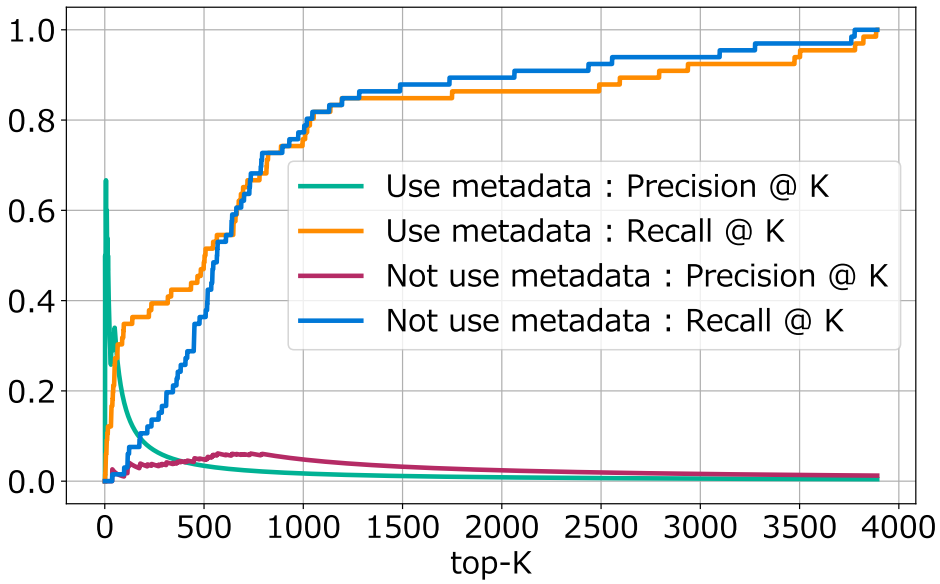

Figure 6: Precision, Recall@K in suspicious article detection.

Month		# Samples (pos / neg)	
		Suspicious Article	Suspicion Casting Post
2018	Jan.	91 (5 / 86)	201 (8 / 193)
	Feb.	2,675 (48 / 2,627)	3,029 (49 / 2,980)
	Mar.	275 (11 / 264)	295 (16 / 279)
	Apr.	28 (0 / 28)	52 (0 / 52)
	May.	144 (1 / 143)	182 (5 / 177)
	Jun.	626 (73 / 553)	2,432 (135 / 2,297)
	Jul.	729 (63 / 666)	1,190 (117 / 1,073)
	Aug.	672 (64 / 608)	621 (86 / 535)
	Sep.	3,310 (439 / 2,871)	4,668 (392 / 4,276)
	Oct.	1,167 (44 / 1,123)	3,155 (111 / 3,044)
	Nov.	1,236 (142 / 1,094)	5,588 (270 / 5,318)
	Dec.	828 (139 / 689)	1,673 (307 / 1,366)
2019	Jan.	37 (4 / 33)	688 (244 / 444)
	Feb.	464 (186 / 278)	811 (307 / 504)
	Mar.	261 (79 / 182)	452 (234 / 218)
	Apr.	278 (41 / 237)	373 (269 / 104)
	May.	607 (59 / 548)	775 (636 / 139)
	Jun.	1,838 (116 / 1,722)	2,611 (907 / 1,704)
	Jul.	2,129 (166 / 1,963)	2,749 (264 / 2,485)
	Aug.	1,162 (137 / 1,025)	688 (642 / 46)
	Sep.	1,223 (80 / 1,143)	1,418 (1,230 / 188)

Table 5: Number of labeled data that we were collected by daily activity per month using our application. "pos" and "neg" denotes the number of positive (i.e. suspicious casting posts or suspicious articles) and negative samples, respectively.

Suspicion Casting Post	
# Samples (pos / neg)	5,166 (290 / 4,876)
Suspicious Article	
# Samples (pos / neg)	3,891 (66 / 3,825)

Table 6: Statistics of test set. "pos" and "neg" denotes the number of positive (i.e. suspicious casting posts or suspicious articles) and negative samples, respectively.

9 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. Furthermore, we confirmed that our method is effective in actual fact-checking activities.

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. Also, to further improve the models, other types of features, such as inter-user relations and external knowledge, are worth trying to use.

Also, the error analysis show that some expressions, such as personal impression, are difficult for models to tell from suspicion-casting ones. To deal with such confusing expressions, it can be a potential approach to define and add more fine-grained labels, such as "impression" and "irony," to the datasets. We can expect that training on them allows models to distinguish confusing expressions. By training models on the dataset with such labels, we can expect that they can distinguish the labels.

Acknowledgements

I am deeply grateful to Dr. Kentaro Inui and Dr. Jun Suzuki for able guidance and generous support. Many thanks also to Dr. Hiroki Ouchi and Kazuaki Hanawa for insightful comments and constructive suggestions during our many discussions. Discussions with my academic colleagues in our laboratory have been quite illuminating. I would also like to appreciate the support received through the joint research undertaken with Atsushi Komiya, Ryo Yamashita and Hitofumi Yanai.

References

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

[2] Naeemul Hassan, Bill Adair, James T Hamilton, Chengkai Li, Mark Tremayne, Jun Yang, and Cong Yu. The quest to automate fact-checking. *world*, 2015.

[3] Andreas Vlachos and Sebastian Riedel. Fact checking: Task definition and dataset construction. In *LTCSS@ACL*, 2014.

[4] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of ACL*, pages 422–426, 2017.

[5] Andreas Hanselowski, S. AvineshP.V., Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, and Iryna Gurevych. A retrospective analysis of the fake news challenge stance detection task. *CoRR*, abs/1806.05180, 2018.

[6] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. *CoRR*, abs/1708.07104, 2017.

[7] Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Oken Hodas. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In *ACL*, 2017.

[8] S. Jerril Gilda. Evaluating machine learning algorithms for fake news detection. *2017 IEEE 15th Student Conference on Research and Development (SCOReD)*, pages 110–115, 2017.

[9] Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. Toward automated fact-checking : Detecting check-worthy factual claims by claimbuster. 2017.

[10] James Thorne, Mingjie Chen, Giorgos Myrianthous Jiashu Pu, Xiaoxuan Wang, and Andreas Vlachos. Fake news detection using stacked ensemble of classifiers. 2017.

[11] Eugenio Tacchini, Gabriele Ballarin, Marco L. Della Vedova, Stefano Moret, and Luca de Alfaro. Some like it hoax: Automated fake news detection in social networks. *CoRR*, abs/1704.07506, 2017.

[12] Benjamin Riedel, Isabelle Augenstein, Georgios P. Spithourakis, and Sebastian Riedel. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *CoRR*, abs/1707.03264, 2017.

[13] Qi Zeng. Neural stance detectors for fake news challenge. 2017.

[14] Stephen R. Pfohl. Stance detection for the fake news challenge with attention and conditional encoding. 2017.

[15] Gaurav Bhatt, Aman Sharma, Shivam Sharma, Ankush Nagpal, Balasubramanian Raman, and Ankush Mittal. On the benefit of combining neural, statistical and external features for fake news identification. *CoRR*, abs/1712.03935, 2017.

[16] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and verification. In *NAACL-HLT*, 2018.

[17] Jiawei Zhang, Limeng Cui, Yanjie Fu, and Fisher B. Gouza. Fake news detection with deep diffusive network model. *CoRR*, abs/1805.08751, 2018.

[18] Jooyeon Kim, Behzad Tabibian, Alice Oh, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. Leveraging the crowd to detect and reduce the spread of fake news and misinformation. In *WSDM*, 2018.

[19] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *WWW*, 2011.

[20] Qiang Liu, Shu Wu, Feng Yu, Liang Wang, and Tieniu Tan. Ice: Information credibility evaluation on social media via representation learning. *CoRR*, abs/1609.09226, 2016.

[21] Sebastian Tschiatschek, Adish Singla, Manuel Gomez-Rodriguez, Arpit Merchant, and Andreas Krause. Detecting fake news in social networks via crowdsourcing. *CoRR*, abs/1711.09025, 2017.

[22] Tanushree Mitra and Eric Gilbert. Credbank: A large-scale social media corpus with associated credibility annotations. In *ICWSM*, 2015.

[23] Yunfei Long, Qin Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang. Fake news detection through multi-perspective speaker profiles. In *IJCNLP*, 2017.

[24] Nguyen Vo and Kyumin Lee. The rise of guardians: Fact-checking url recommendation to combat fake news. In *SIGIR*, 2018.

[25] Yang Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, and Philip S. Yu. Ti-cnn: Convolutional neural networks for fake news detection. *CoRR*, abs/1806.00749, 2018.

[26] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[27] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM TIST*, 2:27:1–27:27, 2011.

[28] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[29] J. Ross Quinlan and Ronald L. Rivest. Inferring decision trees using the minimum description length principle. *Inf. Comput.*, 80:227–248, 1989.

[30] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[31] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 8:1735–80, 1997.

[32] Felix A. Gers, Jürgen Schmidhuber, and Fred A. Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12 10:2451–71, 2000.

[33] 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. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[34] François Chollet et al. Keras. `https://keras.io`, 2015.

[35] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[36] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[37] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*, 2015.

[38] Ji Young Lee and Franck Dernoncourt. Sequential short-text classification with recurrent and convolutional neural networks. In *HLT-NAACL*, 2016.

[39] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

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 7: Hyper-parameters for Word2Vec training.

Hyper-parameter	Values
Embedding size	300
Batch size	100
Max epoch	50
Optimizer	Adam [39]
Adam α	{0.002, 0.9, 0.009}

Table 8: Hyper-parameters for the LSTM model.

List of Publications

Awards

- 平成 29 年度 情報処理学会東北支部学生奨励賞.

International Conferences Papers

- Tsubasa Tagami, Hiroki Ouchi, Hiroki Asano, Kazuaki Hanawa, Kaori Uchiyama, Kaito Suzuki, Kentaro Inui, Atsushi Komiya, Atsuo Fujimura, Ryo Yamashita, Hitofumi Yanai and Akinori Machino. Suspicious News Detection Using Micro Blog Text, In The 32nd Pacific Asia Conference on Language, Information and Computation (PACLIC 32), pages 648-656, December 2018.

Other Publications

- 田上翼, 浅野広樹 (東北大), 乾健太郎 (東北大/理研 AIP), 楊井人文, 山下亮 (日本報道検証機構), 小宮篤史, 藤村厚夫 (スマートニュース), 町野明徳 (フリー). ファクトチェックを必要とするニュース記事の探索の支援. 言語処理学会第 24 回年次大会 (NLP2018), March 2018.

- 内山香, 鈴木海渡, 田上翼, 塙一昇 (東北大), 乾健太郎 (東北大/理研 AIP), 楊井人文, 山下亮 (日本報道検証機構), 小宮篤史, 藤村厚夫 (スマートニュース), 町野明徳 (フリー). ファクトチェックのための要検証記事探索の支援. 人工知能学会全国大会 (第 32 回), June 2018.