From Stances' Imbalance to Their Hierarchical Representation and Detection

Conference Paper · February 2019

DOI: 10.1145/3308558.3313724

CITATIONS
22

READS
1,015

5 authors, including:

Qiang Zhang
University College London
22 PUBLICATIONS **304** CITATIONS
SEE PROFILE

Aldo Lipani
University College London
94 PUBLICATIONS **586** CITATIONS
SEE PROFILE

Emine Yilmaz
University College London
127 PUBLICATIONS **2,531** CITATIONS
SEE PROFILE

From Stances' Imbalance to Their Hierarchical Representation and Detection

Qiang Zhang
University College London
London, United Kingdom
qiang.zhang.16@ucl.ac.uk

Shangsong Liang
Sun Yat-sen University
Guangzhou, China
liangshangsong@gmail.com

Aldo Lipani
University College London
London, United Kingdom
aldo.lipani@ucl.ac.uk

Zhaochun Ren
Shandong University
Qingdao, China
zhaochun.ren@sdu.edu.cn

Emine Yilmaz
University College London
London, United Kingdom
emine.yilmaz@ucl.ac.uk

ABSTRACT

Stance detection has gained increasing interest from the research community due to its importance for fake news detection. The goal of stance detection is to categorize an overall position of a subject towards an object into one of the four classes: *agree*, *disagree*, *discuss*, and *unrelated*. One of the major problems faced by current machine learning models used for stance detection is caused by a severe class imbalance among these classes. Hence, most models fail to correctly classify instances that fall into minority classes. In this paper, we address this problem by proposing a hierarchical representation of these classes, which combines the agree, disagree, and discuss classes under a new *related* class. Further, we propose a two-layer neural network that learns from this hierarchical representation and controls the error propagation between the two layers using the Maximum Mean Discrepancy regularizer. Compared with conventional four-way classifiers, this model has two advantages: (1) the hierarchical architecture mitigates the class imbalance problem; (2) the regularization makes the model to better discern between the related and unrelated stances. An extensive experimentation demonstrates state-of-the-art accuracy performance of the proposed model for stance detection.

CCS CONCEPTS

• **Information systems** → **Information extraction**; **Sentiment analysis**.

KEYWORDS

hierarchical classifier, maximum mean discrepancy

ACM Reference Format:
Qiang Zhang, Shangsong Liang, Aldo Lipani, Zhaochun Ren, and Emine Yilmaz. 2019. From Stances' Imbalance to Their Hierarchical Representation and Detection. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3308558.3313724

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313724

1 INTRODUCTION

The quality of online news is usually less substantiated than that of traditional news services such as magazines or newspapers [1, 45, 47]. A large volume of fake news is being produced for political or economical purposes [8, 22, 41]. Fake news are those news articles that purport to be factual, but which contain misstatements of fact with intention to arouse passions, attract viewership, or deceive [25, 37, 44]. Verifying news content needs to retrieve *evidences* and determine their *stance* with respect to the *news claims*, which proposes new challenges for the conventional stance detection task [31, 36]. We specify evidence as text, e.g. web-pages and documents, that can be used to prove if news content is or is not true. Moreover, automatic stance detection has broad applications in information retrieval and text entailment [34, 42].

The task of stance detection is to identify the stance of an evidence towards a given news claim [12, 13]. Stances can be categorized into four classes: *agree*, *disagree*, *discuss* and *unrelated* [17]. Two characteristics make the stance detection task peculiar. On the one hand, news claims and evidences are often unrelated – generating a severe class imbalance problem; On the other hand, since the non-related classes are by definition related, intuitively, the identification of an evidence as related or unrelated to a news claim is semantically different from the identification of an evidence as belonging to one of the other three classes. These two characteristics suggests the natural presence of a hierarchical structure among stance classes.

Stance detection has been studied in areas of information extraction and natural language processing [11, 40]. However, previous methods tackle the task as a multiclass classification problem, neglecting the hierarchical structure in stance classes. Also, the commonly-used four-way classifiers are easily influenced by the class imbalance problem. In this paper, we address this issue by modeling the stance detection task as a two-layer neural network. The first layer aims at identifying the relatedness of the evidence, while the second layer aims at classifying, those evidences identified as related, into the other three classes: agree, disagree and discuss. Moreover, by studying various level of dependence assumptions between the two layers: (1) independent, when there is no error propagation between the two layers; (2) dependent, when the error propagation is left free, and; (3) learned, when the error propagation is controlled by Maximum Mean Discrepancy (MMD), we

show that when learned, the neural network (a) better separates the distributions of related and unrelated stances and (b) outperforms the state-of-the-art accuracy for the stance detection task.
The remainder of the paper is organized as follows: § 2 summarizes the related work; § 3 defines the stance detection task; § 4 details the proposed hierarchical classification model and the regularization term; § 5 describes the used datasets and experimental setup; § 6 is devoted to experimental results, and; § 7 concludes the paper.

2 RELATED WORK

Machine learning techniques are widely researched to tackle the stance detection task. Previous works focus on political or congressional floor debates [11, 40, 46] and online forums [2, 19, 27, 38, 39, 42]. Most of these works rely on content-based features, such as sentiment analysis and topic-specific features learned from labeled datasets for a closed set of topics.

Two methods only consider the agree, disagree and discuss classes: Bar-Haim et al. [7] split the stance detection task to three sub-tasks and propose a Contrast Classification Algorithm to distinguish agree and disagree classes; Augenstein et al. [4] build a neural network architecture based on bidirectional conditional encoding on a Tweeter dataset. A long-short term memory (LSTM) encodes the claim and another LSTM encodes the text with the encoded claim as initial states. These methods fail to consider the unrelated class.

Two other methods consider all the classes, but use two different models: Bourgonje et al. [10] use the lemmatized n-gram matching and a rule-based procedure to decide the evidence relatedness, and a three-way logistic regression classifier to distinguish among the relevant classes; Wang et al. [43] firstly develop a gradient boosted decision tree (GBDT) model [28] to determine the evidence relatedness, then another GBDT model is used to distinguish stances of the text towards the claim. These methods involve feature engineering in separate models and cannot be jointly optimized to achieve the best performance.

Other methods that also consider all the classes have been developed during the Fake News Challenge stage 1 (FNC-1) [18]. The winner team uses a 50%/50% weighted average between a GBDT model and a convolutional neural network (CNN) [5]. The second best performance is achieved by an ensemble of five multi-layer perceptrons (MLPs) where input features include bag-of-words, semantic analysis in addition to the baseline features developed by the challenge organizers [16]. Compared to the above two solutions, the third best team does not try ensemble methods. They use TF-IDF features and an MLP as a four-way classifier [33]. Zhang et al. [48] propose a ranking method to tackle the task and achieve empirical performance improvements. However, these methods all neglect the hierarchical structure among the four types of stances and suffer from class imbalance.

Deep learning-based methods have also been applied in the FNC-1. Bajaj [6] utilizes LSTM, CNN and their variants to detect stances. Bajaj finds that an attention-augmented CNN obtains the best performance. Rakholia and Bhargava [32] analyze the effectiveness of different ways of text coding, such as independent coding, bidirectional conditional encoding and attentive readers, and conclude that the attentive reader model is the most suitable for the task. Ma et al. [23] propose a multi-task learning algorithm that jointly detect rumours and stances. However, all these methods fail to achieve high accuracy for the agree and disagree classes.

There are three major defects in all the aforementioned methods: (a) they neglect the hierarchical relationships among the four stances; (b) they suffer from the class imbalance problem, and; (c) they fail to achieve acceptable detection performance for the agree and disagree classes.

3 STANCE DETECTION TASK

The stance detection task consists in classifying the stance of an evidence towards a claim as one of the four classes: agree, disagree, discuss and unrelated. Formal definitions of these four stances are:

agree – the evidence supports the claim;
disagree – the evidence denies the claim;
discuss – the evidence does not have a position about the claim;
unrelated – the evidence is not about the claim.

4 HIERARCHICAL CLASSIFICATION

In this section, we detail our proposed two-layer neural network for stance detection. § 4.1 outlines the model. In order to better differentiate between the related and unrelated classes, we design an MMD regularization term in § 4.2. This is then integrated into the two-layer neural network loss function in § 4.3. In Figure 1, we show the architecture of our model.

4.1 Two-Layer Neural Network

Let the input space be formed by m-dimensional real vectors in a neural network, denoted as $\mathbf{v} \in \mathbb{R}^m$. The four-class label can be transformed into a one-hot vector $\mathbf{y}$. The i-dimension of $\mathbf{y}$ (y_i) is 1 when the stance is the i-element in the label set $\{agree, disagree, discuss, unrelated\}$ and 0 otherwise. The hidden layer with parameters θ_u learns to map $\mathbf{v}$ to a k-dimensional hidden representation $\mathbf{u} \in \mathbb{R}^k$:

$$\mathbf{u} = f(\mathbf{v}; \theta_u). \tag{1}$$

For the two-layer classification, the first layer decides whether the evidence is related to a claim. Hence, the first classification layer is called *the relatedness layer*. This layer is parameterized by θ_r and learns to produce a 2-dimensional normalized vector $\hat{\mathbf{r}}$ as follows:

$$\hat{\mathbf{r}} = g(\mathbf{u}; \theta_r). \tag{2}$$

Note that the Softmax function is included in g to normalize the 2-dimensional vector, so each component of the vector $\hat{\mathbf{r}}$ denotes the probability that the neural network assigns $\mathbf{v}$ to the related and unrelated classes, i.e., $p(related)$ and $p(unrelated)$.

The second layer classifies the evidence into the related classes, i.e., agree, disagree, or discuss stances. Hence, the second classification layer is called *the stance layer*. The stance layer is parameterized by θ_s and learns to produce a 3-dimensional normalized vector $\hat{\mathbf{s}}$:

$$\hat{\mathbf{s}} = h(\hat{r} \cdot (1, 0); \theta_s), \tag{3}$$

where the vector multiplication $\hat{r} \cdot (1, 0)$ extracts the first element of $\hat{\mathbf{r}}$. Note that the Softmax function is also included in h to normalize the 3-dimensional vector, so that each component of the vector $\hat{\mathbf{s}}$ denotes the conditional probability that the neural network

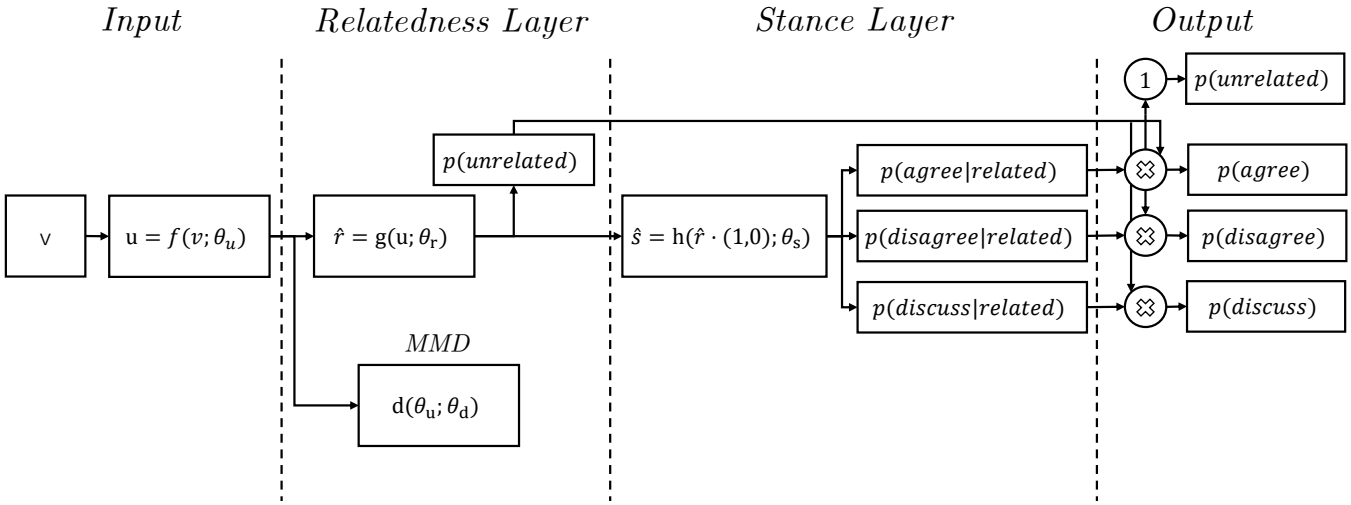


Figure 1: The architecture of our proposed two-layer neural network.

assigns $\mathbf{v}$ to agree, disagree and discuss given that $\mathbf{v}$ is related, i.e., $p(agree|related)$, $p(disagree|related)$, and $p(discuss|related)$.

We define the classification loss by the Kullback-Leibler (KL) divergence [21], which measures the difference between the network outputs and labels:

$$l^r(\theta_u, \theta_r) := \mathrm{KL}(\mathbf{r}\|\hat{\mathbf{r}}), \tag{4}$$

where $\mathbf{r}$ is the ground-truth relatedness of the input data. $\mathbf{r}$ is computed from a label $\mathbf{y}$ as follows:

$$\mathbf{r} = (\mathbb{1}(\mathbf{y} \neq \mathbf{e}_4), \mathbb{1}(\mathbf{y} = \mathbf{e}_4)), \tag{5}$$

where $\mathbb{1}$ is the indicator function, $\mathbf{e}_4$ is a 4-dimensional one-hot vector with fourth element equal to 1. When $\mathbf{y} = \mathbf{e}_4$ is verified, it indicates that the label belongs to the unrelated class. Similarly, the stance classification loss can be defined as:

$$l^s(\theta_u, \theta_r, \theta_s) := \mathrm{KL}(\mathbf{s}\|\hat{\mathbf{s}}), \tag{6}$$

where $\mathbf{s}$ is the ground-truth stance of the input data. $\mathbf{s}$ is computed from a label $\mathbf{y}$ as follows:

$$\mathbf{s} = (\mathbb{1}(\mathbf{y} = \mathbf{e}_1), \mathbb{1}(\mathbf{y} = \mathbf{e}_2), \mathbb{1}(\mathbf{y} = \mathbf{e}_3)), \tag{7}$$

where $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ are 4-dimensional one-hot vectors with first, second, and third elements equal to 1. When $\mathbf{y} = \mathbf{e}_1$ is verified, it indicates that the label belongs to the agree class, when $\mathbf{y} = \mathbf{e}_2$ is verified, it indicates that the label belongs to the disagree class, and when $\mathbf{y} = \mathbf{e}_3$ is verified, it indicates that the label belongs to the discuss class.

Finally, we now define the loss function for the two-layer neural network as the linear combination between the loss function of the relatedness layer (l^r) and the loss function of the stance layer (l^s):

$$l^c(\theta_u, \theta_r, \theta_s) := l^r(\theta_u, \theta_r) + \alpha \cdot l^s(\theta_u, \theta_r, \theta_s), \tag{8}$$

where α leverages the importance of the two classification layers.

4.2 Maximum Mean Discrepancy

The classification of related/unrelated stances is a different task from that of agree/disagree/discuss stances. Therefore, data representations from the relatedness layer and the stance layer can be seen as samples drawn from two different distributions. In order to measure distribution discrepancy between these two layers, we employ the Maximum Mean Discrepancy (MMD) [9] as a regularization term. The MMD does not involve density estimation and thus is a non-parametric way of measuring the difference between distributions. MMD has achieved success in face recognition and image annotation [15].

MMD is defined as follows:

Definition 4.1. Maximum Mean Discrepancy [9]: "Let p and q be two Borel probability distributions over a space $\mathcal{X}$ and let X and Z be sets with independent identically distributed samples drawn from p and q. The MMD is defined by a class Ψ of map functions ψ: $\mathcal{X} \to \mathcal{H}$ as:

$$MMD(p, q, \Psi) = \sup_{\psi \in \Psi} (\mathbf{E}_p[\psi(x)] - \mathbf{E}_q[\psi(z)]). \tag{9}$$

Here, x and z are samples from X and Z."

In other words, the MMD equation defines the largest possible distance between two expectations over the set of function Ψ. Moreover, "when $\mathcal{H}$ is the reproducing kernel Hilbert space (RKHS) [3], this means that for all $x \in \mathcal{X}$, the linear point evaluation function mapping $\psi \to \psi(x)$ exists and is continuous. When Ψ is the unit ball in a universal RKHS, it is guaranteed that $MMD(p, q, \Psi)$ will detect any discrepancy between p and q [9, 35]."

Let p denote the distribution for the first layer samples (unrelated hidden representations) in our model, with sample set $\mathcal{U}^1 = \{\mathbf{u}_1^1, \ldots, \mathbf{u}_{n_1}^1\}$ and according to Eq. (1) their generating set $\mathcal{V}^1 = \{\mathbf{v}_1^1, \ldots, \mathbf{v}_{n_1}^1\}$. And, q denotes the distribution for the second

layer samples (agree, disagree and discuss hidden representations), with sample set $\mathcal{U}^2 = \{\mathbf{u}_1^2, \ldots, \mathbf{u}_{n_2}^2\}$ and according to Eq. (1) their generating set $\mathcal{V}^2 = \{\mathbf{v}_1^2, \ldots, \mathbf{v}_{n_2}^2\}$. n_1 and n_2 are the number of samples in $\mathcal{U}^1$ and $\mathcal{U}^2$. Thus we have $\mathcal{X} = \mathcal{R}^k$ and $\mathcal{H} = \mathcal{R}^j$ with $\psi(x) = \theta_d x$, where θ_d is a $j \times k$ matrix in *the projection layer*. k and j are the space dimensions. According to Eq. (1), the hidden representation $\mathbf{u}$ is parameterized by θ_u, thus the empirical expression of MMD is parameterized by θ_u and θ_d:

$$d(\theta_u, \theta_d) = \frac{1}{n_1}\sum_{i=1}^{n_1} \theta_d \mathbf{u}_i^1 - \frac{1}{n_2}\sum_{i=n_1+1}^{n_1+n_2} \theta_d \mathbf{u}_i^2 \tag{10}$$

$$= \frac{1}{n_1}\sum_{i=1}^{n_1} \theta_d f(\mathbf{v}_i^1; \theta_u) - \frac{1}{n_2}\sum_{i=n_1+1}^{n_1+n_2} \theta_d f(\mathbf{v}_i^2; \theta_u). \tag{11}$$

By constantly changing *the projection layer* parameterized by θ_d, we find the maximum expectation difference between the representations of the two classification layers.

4.3 Optimization

The more different two distributions are, the larger the MMD is. Hence, in order to make the distributions easier to be distinguished, a larger MMD regularization term is preferred, and we treat the regularization term as an extra goal besides classification. We integrate the two-layer classification loss (see Eq. (8)) and the MMD regularization term (see Eq. (10)) into a single objective function (L). Specifically, we add these two sub-goals with a hyperparameter β as follows:

$$L(\theta_u, \theta_r, \theta_s, \theta_d) = l^c(\theta_u, \theta_r, \theta_s) - \beta \cdot d(\theta_u, \theta_d), \tag{12}$$

where β leverages the importance of the regularization. The larger the MMD regularization term is, the easier is for the classifier to distinguish between the related and unrelated stances. Thus, the sign of the regularization term is negative.

The optimization involves the minimization of the classification loss L with respect to θ_u, θ_r, θ_s, and θ_d as follows:

$$\min_{\theta_u, \theta_r, \theta_s, \theta_d} L(\theta_u, \theta_r, \theta_s, \theta_d). \tag{13}$$

Optimizing the model consists of two sub-goals. On the one hand, we want to maximize the distribution discrepancy between the two classification layers. On the other hand, we want to minimize the classification loss of both layers. Both of these two sub-goals involve the feature layer parameter θ_u update, but in opposite update directions. The optimization process will not stop until a saddle point (the feature layer parameters can be well applied in both sub-goals) is reached. Algorithm 1 shows the parameter update process, which is based on the mini-batch gradient descent algorithm.

4.4 Prediction

Given as input a feature vector $\mathbf{v}$, the classifier outputs the following probabilities: $p(unrelated)$, $p(agree|related)$, $p(disagree|related)$, and $p(discuss|related)$. However, these last 3 probabilities are not comparable with the first one. To make them comparable we derive

Algorithm 1: Parameter update process based on the mini-batch gradient descent algorithm.

input : Sample mini-batch $\{\mathbf{v}_i, \mathbf{r}_i, \mathbf{s}_i\}_{i=1}^n$, mini-batch size n, hyperparameters α, β, and μ
output: θ_u, θ_r, θ_s, θ_d

begin
  Initialize θ_u, θ_r, θ_s, θ_d;
  repeat
    /* forward propagation */
    $l^r, l^s \leftarrow 0$;
    for *i from* 1 *to n* **do**
      $\mathbf{u}_i \leftarrow f(\mathbf{v}_i; \theta_u)$;
      $\hat{\mathbf{r}}_i \leftarrow g(\mathbf{u}_i; \theta_r)$;
      $l_i^r \leftarrow \mathrm{KL}(\mathbf{r}_i \| \hat{\mathbf{r}}_i)$;
      $l^r \leftarrow l^r + l_i^r$;
      if $\mathbf{r}_i \cdot (1, 0) = 1$ **then**
        /* classify related */
        $\hat{\mathbf{s}}_i \leftarrow h(\hat{\mathbf{r}}_i \cdot (1, 0); \theta_s)$;
        $l_i^s \leftarrow \mathrm{KL}(\mathbf{s}_i \| \hat{\mathbf{s}}_i)$;
      else
        /* unrelated */
        $l_i^s = 0$;
      $l^s \leftarrow l^s + l_i^s$;
    $d = \mathrm{MMD}(\{\mathbf{u}_i, \mathbf{r}_i\}_{i=1}^n; \theta_d)$;
    /* backward propagation */
    $\theta_s \leftarrow \theta_s - \mu \cdot \alpha \cdot \frac{\partial l^s}{\partial \theta_s}$;
    $\theta_r \leftarrow \theta_r - \mu \cdot (\frac{\partial l^r}{\partial \theta_r} + \alpha \cdot \frac{\partial l^s}{\partial \theta_r})$;
    $\theta_d \leftarrow \theta_d + \mu \cdot \beta \cdot \frac{\partial d}{\partial \theta_d}$;
    $\theta_u \leftarrow \theta_u - \mu \cdot (\frac{\partial l^r}{\partial \theta_u} + \alpha \cdot \frac{\partial l^s}{\partial \theta_u} - \beta \cdot \frac{\partial d}{\partial \theta_u})$;
  until θ_u, θ_r, θ_s, θ_d *converge*;

$p(agree)$, $p(disagree)$ and $p(discuss)$. By observing that the class agree is assumed as related, thus $p(agree, related) = p(agree)$, we derive that:

$$\begin{aligned} p(agree) &= p(agree, related) \\ &= p(agree|related) \times p(related) \\ &= p(agree|related) \times (1 - p(unrelated)). \end{aligned} \tag{14}$$

Similarly, for the other two classes we derive that:

$$\begin{aligned} p(disagree) &= p(disagree|related) \times (1 - p(unrelated)), \\ p(discuss) &= p(discuss|related) \times (1 - p(unrelated)). \end{aligned} \tag{15}$$

Thereby, the model actual output $\hat{\mathbf{y}}$ is:

$$\hat{\mathbf{y}} = (p(agree), p(disagree), p(discuss), p(unrelated)), \tag{16}$$

where the class with the highest probability corresponds to the predicted stance.

5 EXPERIMENTAL SETUP

We start this section by presenting the datasets and evaluation measures relevant to the stance detection task. Then, we describe the features used by our model and the model parameterization. Finally, we present the baselines. The software used to run the experiments of this paper is available on the website of the first author.

5.1 Datasets

Experiments are conducted on two publicly available datasets: the *Emergent* dataset[1] [14] and the *FNC-1* dataset[2]. In these two datasets, a claim consists of a news article headline and an evidence of a news article content. These datasets are split into train and test subsets; see Table 1 for statistics about the splits.

The FNC-1 dataset consist of 75,385 instances. Each instance in the dataset is a pair claim-evidence labeled as one of the four stances: agree, disagree, discuss and unrelated. The ratio of training data over testing data in the FNC-1 dataset is ∼2:1. Every class accounts for a similar percentage in the train and test subsets. The unrelated stances are the majority (over 70%) in both subsets, while the disagree stances are less than 3%. The agree and discuss stances are less than 20% and 10%.

The Emergent dataset is similar to the FNC-1 dataset, however it contains only agree, disagree and discuss stances. Hence, it needs to be augmented with unrelated stances. Similarly to how the FNC-1 dataset unrelated stances have been labeled, we manually labeled unrelated stances by pairing a claim with an unrelated evidence, i.e., paired with another claim. Moreover, to make the class distributions less imbalanced, we make the ratio of related stances and unrelated ones ∼1:1. The augmented Emergent dataset contains 4,071 training labels and 1,024 testing labels with a ratio of ∼4:1. Class distributions between train and test subsets are similar.

Compared to the FNC-1 dataset, the class distributions of the augmented Emergent dataset is more balanced. The percentage of unrelated stances is about 50%, whereas the percentages of agree and disagree stances are about 24% and 8%. Both datasets have similar percentages of the discuss stances.

5.2 Evaluation Measure

In line with the FNC-1 challenge, the evaluation is based on a weighted two-level scoring system based on the *accuracy* measure. This evaluation measure, called *relative score*, evaluates a model by splitting the stance detection task into two sub-tasks, related/unrelated and agree/disagree/discuss classification sub-tasks. To the former sub-task is given a 25% weight. This is done because this sub-task is considered to be easier than the latter sub-task to which is given a 75% weight.

We report the evaluation measures: relative score, accuracy, and accuracy on a per class basis.

5.3 Feature Extraction

To represent claims and evidences we use a bag-of-words approach. For each claim and evidence we generate a TF-IDF vector, and for each pair claim-evidence we compute their cosine similarity. We also include the FNC-1 official features into the input feature vector.

The final set of features include:

- TF-IDF vectors of claims;
- TF-IDF vectors of evidences;
- Cosine similarity (CosSim) between the claim vector and the evidence;
- Ratio of word overlap (WordLap) between the claim and the evidence;
- An Indicator whether a claim has refuting words (RefWord);
- The polarity (Pol) of the claim and the evidence;
- The number of overlapping n-grams (NGrams) for $n \in \{2, 3, 4, 5, 6\}$ between the claim and the evidence.

For the TF-IDF vectors, we only use the top 2,000 most frequent terms except stop-words. All of these features are concatenated to form the input feature vector $\mathbf{v}$.

5.4 Experimental Setting

The following hyperparameters have been set via a five-cross validation on the train subsets:

- The dimension k of hidden representations is set to 100;
- The dimension j of the MMD is set to 10;
- The activation function used in the hidden layers is set to ReLu;
- The parameters α are set to 1.5 and 1.3 for the Emergent and FNC-1 datasets.
- The parameter β is set to 0.001;

We include a L2 regularization term [29] for the MLP weight parameters in the final loss function to mitigate overfitting. Dropout is also used to mitigate overfitting with rate set to 0.6. We train in mini-batches of size 64 over the entire train subset. Note that the gradient steps in Algorithm 1 can easily be alternated with a more powerful optimizer such as the Adam optimizer [20]. Early stopping is applied when the classification loss on the validation subset does not get smaller for three continuous iterations. The whole model is implemented with TensorFlow.

5.5 Baselines

We compare our model against the methods mentioned in Section 2. These methods are detailed in the following. Among them we distinguish between methods that use the same features as ours and methods that learn their representations. We start with the latter type, we call these *representation learning-based baselines*:

Bidirectional LSTM (BiLSTM). Augenstein et al. [4] build a neural network architecture based on bidirectional LSTM on a Tweeter dataset. A LSTM encodes the claim, and another LSTM encodes the evidence with the encoded claim set as initial states. The 100-d GloVe word embedding is used as input [30];

Attentive CNN (AtCNN). Bajaj [6] builds an attention-augmented CNN. The claim and the evidence are input to a convolutional neural network to obtain hidden representations, and the attention mechanism is employed to locate the most influential words or phases on the final results;

[1] https://github.com/willferreira/mscproject.
[2] https://github.com/FakeNewsChallenge/fnc-1.

Table 1: Statistics of the datasets.

Subset	Stance	Emergent		FNC-1	
		Number	Percentage	Number	Percentage
Training	agree	992	24.37	3,678	7.36
	disagree	303	7.44	840	1.68
	discuss	776	19.06	8,909	17.83
	unrelated	2,000	49.13	36,545	73.13
		4,071		49,972	
Testing	agree	246	24.02	1,903	7.49
	disagree	91	8.89	697	2.74
	discuss	776	19.06	4,464	17.57
	unrelated	500	48.83	18,349	72.20
		1,024		25,413	

Memory Network (MN). Mohtarami et al. [26] develop an end-to-end memory network for stance detection. The network operates at the paragraph level and integrates convolutional and recurrent neural networks, as well as a similarity matrix as part of the overall architecture;

Ranbking Model (RM). Zhang et al. [48] build a ranking method to tackle the stance detection and achieve empirical performance improvements. A ranking loss function is proposed to replace Softmax and maximize the representation difference between four classes of stance.

We now review the second type of baselines: those methods that use the same features as our method, we call these *feature engineering-based baselines*:

Official Baseline (OB). This is the FNC-1 official baseline that uses one gradient boosting decision trees model for four-way classification;

Logistic Regression (LR). Bourgonje et al. [10] use n-gram matching and a rule-based procedure to decide relatedness, and three-way logistic regression to distinguish among the related classes;

Gradient Boosted Decision Trees (GBDT). Wang et al. [43] develop two GBDT models, one to determine the relatedness of an evidence to a claim, and another to distinguish among the related classes;

Multi-Layer Perception (MLP). This model [33] achieved the third best performance in FNC-1. It extracts TF-IDF and cosine similarity between claims and evidences as input features, and uses a MLP as the four-class classifier.

6 RESULTS AND DISCUSSION

In this section, we start by analyzing the dependency assumption. Then, we compare and contrast our model against the baselines. Next, we provide a sensitivity analysis of the hyperparameters. We conclude with an impact analysis of the features used by the model.

6.1 Dependency Assumption

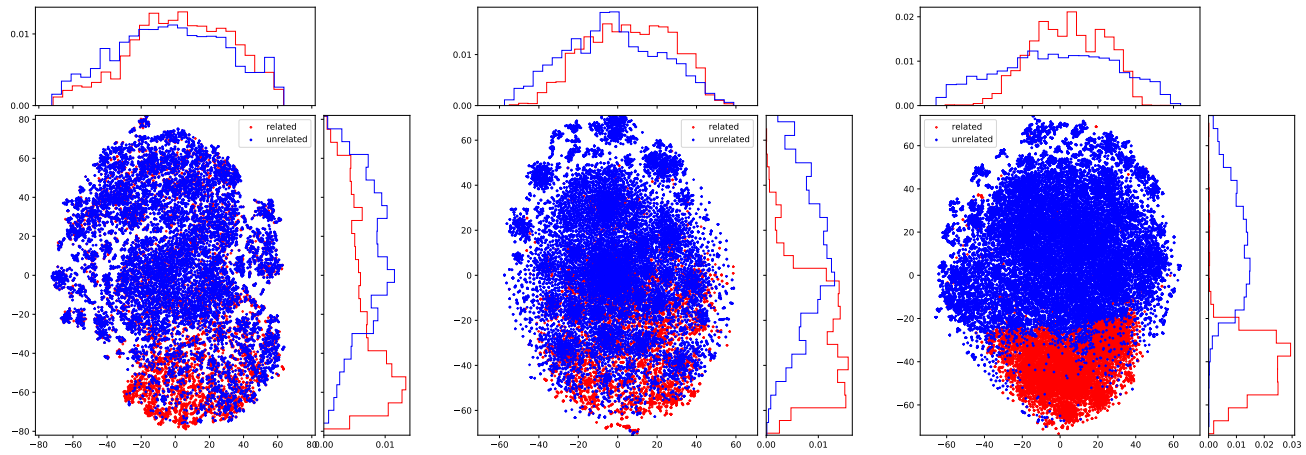
In Figure 2 we show the effect of the 3 dependency assumptions by visualizing the learned representations using a t-SNE projection [24]. We observe that when the classifiers are assumed independent, i.e., the classification is performed in cascade — no error is propagated from the second layer to the first during training — then the learned representation well separates the unrelated class from the unrelated ones. When the classifiers are assumed dependent, i.e., the two classifiers are trained together — the error is left free to propagate from the second layer to the first — then the learned representation is not very well separated. However, when the dependence assumption of the two classifiers is learned via the MMD regularization, i.e., the two classifiers are trained together with the error propagation controlled by the regularizer, then the learned representation is again well separated like in the first case. Well-separated representations suggest a greater discriminative power of the model — the unrelated and related classes are almost linearly separable.

The last three rows of Tables 2 and 3 show the performance of our model on the two test subsets for each one of the three assumptions: independent, dependent, and learned. Looking at the accuracy of the unrelated class, we observe that the accuracy is greater when the learned representations are well-separated, as in the independent and learned cases. Furthermore, looking at all the other scores, we observe that the learned assumption outperforms both the independent and dependent assumptions in all other cases, demonstrating that learning together both, relatedness and stance of the evidences towards claims, is beneficial to the stance detection task.

6.2 Overall Performance

In Tables 2 and 3 we compare our model against the state-of-the-art models. Our model achieves the best stance detection performance for the relative score on both datasets. The model achieves 89.30% on the augmented Emergent test subset and 88.15% on the FNC-1 test subset.

By comparing with four-way classification baselines (OB, MLP, BiLSTM, AtCNN, MN and RM) we demonstrate the advantage of separating the relatedness detection from the stance detection. We observe that these classifiers perform poorly on the disagree class, which is caused by the large percentage difference between the minority disagree class and the majority unrelated class. Further, the more imbalanced the evaluation dataset becomes, the worse performance the four-way classifiers achieve on the minority disagree class.



(a) Independent (b) Dependent (c) Learned

Figure 2: t-SNE visualization of the hidden representations on the training data. The hidden representations of model trained (a) with separated layers (b) together but without regularization, and (c) with MMD regularization.

Table 2: Performance comparison of our model against the State-of-the-Art models on the augmented Emergent dataset.

Model	Accuracy (%)				Relative Score (%)
	agree	disagree	discuss	unrelated	
Feature Engineering-Based Baselines					
OB	33.56	23.44	70.23	84.00	74.86
LR (Bourgonje et al.)	66.73	40.51	78.33	78.00	83.45
GBDT (Wang et al.)	80.62	50.42	83.52	88.00	87.53
MLP (Riedel et al.)	58.53	23.64	79.05	95.00	85.43
RM (Zhang et al.)	64.56	40.42	**85.45**	96.00	87.69
Representation Learning-Based Baselines					
BiLSTM (Augenstein et al.)	43.21	12.57	78.55	96.00	81.37
AtCNN (Bajaj)	44.78	14.60	72.44	**97.00**	83.56
MN (Mohtarami et al.)	54.64	40.05	72.10	89.00	85.92
Our Models					
Independent	74.54	45.32	82.59	95.49	86.33
Dependent	63.54	44.68	68.35	95.00	86.72
Learned	**82.52**	**69.05**	84.30	**97.00**	**89.30**

By comparing with baselines that separate the relatedness detection from the stance detection (LR and GBDT) we demonstrate the superiority of a single end-to-end model. LR and GBDT are better on the disagree class, although their overall performance is worse than our model.

In Figure 3 we show the confusion matrix of our model. Here we observe the detection performance on a per class basis. For the related/unrelated classification, we correctly classify 97.00% and 99.53% unrelated instances on the augmented Emergent and the FNC-1 test subsets. We can see that there is some misclassification between the agree and unrelated classes, and between the discuss and unrelated classes. The misclassification of the disagree class accounts for the largest error of the unrelated instances.

Our model achieves an accuracy of 69.05% and 72.35% for the disagree class on the Emergent and the FNC-1 test subsets. The classification accuracy is largely improved compared to the state-of-the-art. Some misclassification error exists between agree and disagree. However, our model can distinguish between the discuss and the disagree with few errors. While the number of discuss cases is the largest and the number of disagree instances is the smallest, our model does not mistake disagree instances as discuss ones, i.e., the model has learned the core representation difference between these two classes. Due to ambiguous expressions, misclassification between agree and discuss is the cause of most errors between these classes, which leads to a slightly worse accuracy for the discuss class on the Emergent (84.30%) and FNC-1 (77.49%) test subsets.

Table 3: Performance comparison of our model against the State-of-the-Art models on the FNC-1 dataset.

Model	Accuracy (%)				Relative Score (%)
	agree	disagree	discuss	unrelated	
Feature Engineering-Based Baselines					
OB	10.51	1.00	79.66	97.98	75.20
LR (Bourgonje et al.)	67.42	31.61	75.23	95.36	80.63
GBDT (Wang et al.)	**82.93**	69.82	33.52	95.42	86.72
MLP (Riedel et al.)	44.04	6.60	81.38	97.90	81.72
RM (Zhang et al.)	64.90	27.26	**84.41**	99.12	86.66
Representation Learning-Based Baselines					
BiLSTM (Augenstein et al.)	35.96	0.94	80.33	98.54	78.70
AtCNN (Bajaj)	38.67	8.24	70.63	91.25	75.77
MN (Mohtarami et al.)	16.92	60.22	81.27	95.50	79.92
Our Models					
Independent	72.41	37.90	68.23	97.43	83.47
Dependent	61.34	42.93	59.38	99.05	85.32
Learned	80.61	**72.35**	77.49	**99.53**	**88.15**

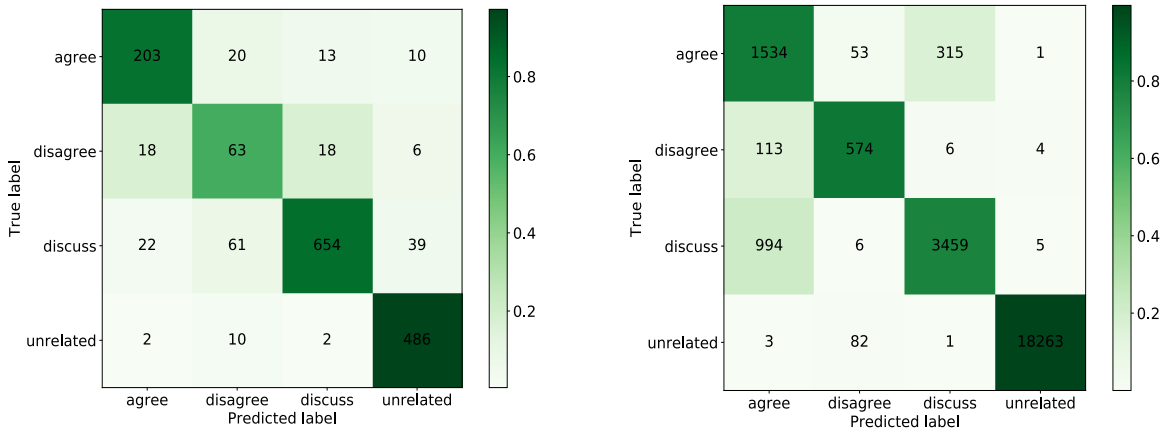


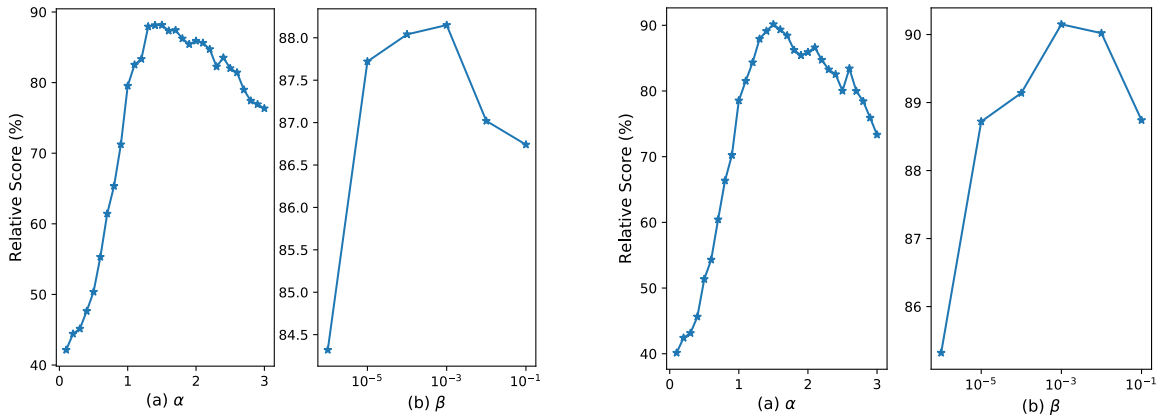
Figure 3: The confusion matrices of our model for the augmented Emergent (on the left) and FNC-1 (on the right) datasets.

Two reasons account for the improved empirical performance observed on our model. On the one hand, the mitigation of the class imbalance problem. Contrary to the four-way classifiers that directly compare the disagree and unrelated instances, the hierarchical model avoids the direct comparison of this minority disagree class (which is less than 2% in the FNC-1 dataset) with the majority unrelated one (which is more than 70% in the FNC-1 dataset). On the other hand, the MMD term that maximizes the discrepancy between the unrelated class and the aggregated related classes. Since the agree, disagree and discuss belong to the same class, the related class, the MMD regularization promotes the emergence of features that are useful to separate the class pairs: agree with unrelated, disagree with unrelated, and discuss with unrelated.

6.3 Hyperparameters Sensitivity

In this subsection we discuss the sensitivity to the hyperparameters of our model. The most influential hyperparameters for the proposed model are α and β. The former controls the relative importance of classification layers. The latter leverages the regularization.

In Figures 4(a) and 4(b) we show how the performance of the model changes when varying α and β for the augmented Emergent and FNC-1 test subsets. α is searched between 0.1 and 3.0 with steps of 0.1, and β is searched in $\{0, 0.1, 0.01, 0.001, 0.0001, 0.00001\}$. For α, we observe that the performance of the model improves quickly as α increases and peaks at 1.5 and 1.3 for the FNC-1 and augmented Emergent datasets, then the performance experiences a slight decrease when α is increased. We hypothesize that the optimal α is related to the class balance between the unrelated class and the related ones. The more unbalanced the dataset is towards the unrelated class, larger is the optimal α. For β, we observe that the performance is the highest when β is set to 0.001. This happens for both augmented Emergent and FNC-1 test subsets. These optimal values of α and β observed on the test subsets are equal to the one found when training the model.



(a) Emergent (b) FNC-1

Figure 4: Sensitivity of the trained model when varying the parameters α and β on the test subset of the augmented Emergent (on the left) and FNC-1 (on the right) datasets.

Table 4: Performance of our model with different feature sets on the FNC-1 dataset. "/" denotes no feature set is removed.

Removed Feature Set	Accuracy (%)			
	agree	disagree	discuss	unrelated
CosSim	71.53	85.08	78.76	69.37
WordLap	67.43	80.49	77.31	77.89
RefWord	74.43	64.37	77.03	97.49
Pol	60.49	67.93	80.92	98.79
NGrams	74.27	75.73	87.82	84.52
/	80.61	82.35	77.49	99.53

6.4 Feature Analysis

In this subsection we evaluate and discuss the importance of each feature towards the final prediction. To examine the influence of each feature on the final performance, we do a leave-one feature set-out approach and record the classification accuracy on the stance detection task. The following analysis is only based on the FNC-1 dataset. Similar results are observed on the augmented Emergent dataset.

In Table 4 we show the results of this analysis. We observe that removing the CosSim feature leads to a large decrease in accuracy for the unrelated class. Similarly, the use of WordLap has a positive effect for the agree class, and it also contributes to the unrelated class. The RefWord and Pol features help for the classes agree and disagree, while removing the NGram feature leads to an increase on the discuss class, i.e., the NGram feature causes confusion between the discuss and the other classes.

7 CONCLUSION

In this paper, we studied the problem of stance detection: the classification of the stance of an evidence towards a claim into one of the four classes: agree, disagree, discuss and unrelated.

We proposed a hierarchical representation of the stance classes, where the classes agree, disagree and discuss are combined together into a class referred as the related class. The main idea here is to divide a concept into sub-concepts that are organized in a hierarchical structure, and design constraints between sub-concepts in order to make the model parameter optimization more sensible. The primary advantage of this hierarchical representation is that it is useful to overcome the class imbalance problem.

This hierarchical representation has inspired the proposed two-layer neural network to tackle the stance detection task. The first layer performs a related-unrelated classification, while the second layer performs a more fine-grained classification among the related classes. Furthermore, we have empirically demonstrated that (1) it is advantageous to learn these two classification tasks together, and (2) the dependency between these two layers can be learned through a MMD regularization term, which measures the representation discrepancy between the two layers. Experiments on two publicly available datasets have shown that our model is able to outperform the state-of-the-art stance detection methods.

As future work we consider the enriching of the proposed model as follows. First, integrating a credibility evaluation of information sources as features. Second, improving the explainability of the model by showing which words or phrases are the most influential in predicting the stance via attention mechanisms.

ACKNOWLEDGMENTS

This project was funded by the EPSRC Fellowship titled "Task Based Information Retrieval", grant reference number EP/P024289/1. We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

[1] Hunt Allcott and Matthew Gentzkow. 2017. *Social media and fake news in the 2016 election.* Technical Report. National Bureau of Economic Research.

[2] Pranav Anand, Marilyn Walker, Rob Abbott, Jean E Fox Tree, Robeson Bowmani, and Michael Minor. 2011. Cats rule and dogs drool!: Classifying stance in online debate. In *Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis*. Association for Computational Linguistics, 1–9.

[3] Nachman Aronszajn. 1950. Theory of reproducing kernels. *Transactions of the American mathematical society* 68, 3 (1950), 337–404.

[4] Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance Detection with Bidirectional Conditional Encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 876–885. https://doi.org/10.18653/v1/D16-1084

[5] Sean Baird, Sibley Doug, and Yuxi Pan. 2017. Talos Targets Disinformation with Fake News Challenge Victory. (2017). https://blog.talosintelligence.com/2017/06/talos-fake-news-challenge.html

[6] Samir Bajaj. 2017. "The Pope Has a New Baby!" Fake News Detection Using Deep Learning. (2017).

[7] Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017. Stance classification of context-dependent claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Vol. 1. 251–261.

[8] Adam J Berinsky. 2017. Rumors and health care reform: experiments in political misinformation. *British Journal of Political Science* 47, 2 (2017), 241–262.

[9] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. 2006. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics* 22, 14 (2006), e49–e57.

[10] Peter Bourgonje, Julian Moreno Schneider, and Georg Rehm. 2017. From clickbait to fake news detection: an approach based on detecting the stance of headlines to articles. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*. 84–89.

[11] Clinton Burfoot, Steven Bird, and Timothy Baldwin. 2011. Collective classification of congressional floor-debate transcripts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 1506–1515.

[12] Sophie Chesney, Maria Liakata, Massimo Poesio, and Matthew Purver. 2017. Incongruent headlines: Yet another way to mislead your readers. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*. 56–61.

[13] Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. 2017. Stance classification with target-specific neural attention networks. International Joint Conferences on Artificial Intelligence.

[14] William Ferreira and Andreas Vlachos. 2016. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. ACL.

[15] Bo Geng, Dacheng Tao, and Chao Xu. 2011. DAML: Domain adaptation metric learning. *IEEE Transactions on Image Processing* 20, 10 (2011), 2980–2989.

[16] Andreas Hanselowski, PVS Avinesh, Benjamin Schiller, and Felix Caspelherr. 2017. *Description of the system developed by team athene in the FNC-1.* Technical Report. Technical report.

[17] Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M Meyer, and Iryna Gurevych. 2018. A Retrospective Analysis of the Fake News Challenge Stance Detection Task. *arXiv preprint arXiv:1806.05180* (2018).

[18] Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, and Iryna Gurevych. 2018. A Retrospective Analysis of the Fake News Challenge Stance-Detection Task. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, 1859–1874. http://aclweb.org/anthology/C18-1158

[19] Kazi Saidul Hasan and Vincent Ng. 2013. Stance classification of ideological debates: Data, models, features, and constraints. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. 1348–1356.

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

[21] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[22] Srijan Kumar and Neil Shah. 2018. False information on web and social media: A survey. *arXiv preprint arXiv:1804.08559* (2018).

[23] Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Detect Rumor and Stance Jointly by Neural Multi-task Learning. In *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 585–593.

[24] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[25] Todor Mihaylov, Georgi Georgiev, and Preslav Nakov. 2015. Finding opinion manipulation trolls in news community forums. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*. 310–314.

[26] Mitra Mohtarami, Ramy Baly, James Glass, Preslav Nakov, Lluís Màrquez, and Alessandro Moschitti. 2018. Automatic Stance Detection Using End-to-End Memory Networks. *arXiv preprint arXiv:1804.07581* (2018).

[27] Akiko Murakami and Rudy Raymond. 2010. Support or oppose?: classifying positions in online debates from reply activities and opinion expressions. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, 869–875.

[28] Nasser M Nasrabadi. 2007. Pattern recognition and machine learning. *Journal of electronic imaging* 16, 4 (2007), 049901.

[29] Arnold Neumaier. 1998. Solving ill-conditioned and singular linear systems: A tutorial on regularization. *SIAM review* 40, 3 (1998), 636–666.

[30] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[31] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2017. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 1003–1012.

[32] Neel Rakholia and Shruti Bhargava. 2017. "Is it true?" – Deep Learning for Stance Detection in News. (2017).

[33] 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. *arXiv preprint arXiv:1707.03264* (2017).

[34] Sebastian Ruder, John Glover, Afshin Mehrabani, and Parsa Ghaffari. 2018. 360 Stance Detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. 31–35.

[35] Bernhard Schölkopf, Koji Tsuda, and Jean-Philippe Vert. 2004. *Kernel methods in computational biology*. MIT press.

[36] Prashant Shiralkar, Alessandro Flammini, Filippo Menczer, and Giovanni Luca Ciampaglia. 2017. Finding streams in knowledge graphs to support fact checking. In *Data Mining (ICDM), 2017 IEEE International Conference on*. IEEE, 859–864.

[37] 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 (2017), 22–36.

[38] Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing stances in online debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, 226–234.

[39] Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. Association for Computational Linguistics, 116–124.

[40] Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Proceedings of the 2006 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 327–335.

[41] Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science* 359, 6380 (2018), 1146–1151.

[42] Marilyn A Walker, Pranav Anand, Robert Abbott, and Ricky Grant. 2012. Stance classification using dialogic properties of persuasion. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*. Association for Computational Linguistics, 592–596.

[43] Xuezhi Wang, Cong Yu, Simon Baumgartner, and Flip Korn. 2018. Relevant Document Discovery for Fact-Checking Articles. In *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 525–533.

[44] Jen Weedon, William Nuland, and Alex Stamos. 2017. Information operations and Facebook. *version* 1 (2017), 27.

[45] Houping Xiao et al. 2018. *Multi-sourced Information Trustworthiness Analysis: Applications and Theory*. Ph.D. Dissertation. State University of New York at Buffalo.

[46] Ainur Yessenalina, Yisong Yue, and Claire Cardie. 2010. Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1046–1056.

[47] Qiang Zhang, Aldo Lipani, Shangsong Liang, and Emine Yilmaz. 2019. Reply-Aided Detection of Misinformation via Bayesian Deep Learning. In *Companion Proceedings of the The Web Conference 2019*. ACM Press.

[48] Qiang Zhang, Emine Yilmaz, and Shangsong Liang. 2018. Ranking-based Method for News Stance Detection. In *Companion Proceedings of the The Web Conference 2018*. ACM Press.