Investigating Rumor News Using Agreement-Aware Search

Jingbo Shang[1], Tianhang Sun[1], Jiaming Shen[1], Xingbang Liu[1],
Anja Gruenheid[2], Flip Korn[2], Adam Lelkes[2], Cong Yu[2], Jiawei Han[1]
[1]Department of Computer Science, University of Illinois Urbana-Champaign, IL, USA
[2]Google Research, New York City, NY, USA
[1]{shang7, ts7, js2, xl14, hanj}@illinois.edu [2]{anjag, flip, lelkes, congyu} @google.com

arXiv:1802.07398v1 [cs.IR] 21 Feb 2018

ABSTRACT

In recent years, rumor news has been generated by humans as well as robots in order to attract readership, influence opinions, and increase internet click revenue. Its detrimental effects have become a worldwide phenomenon, leading to confusion over facts and causing mistrust about media reports. However, evaluating the veracity of news stories can be a complex and cumbersome task, even for experts. One of the challenging problems in this context is to automatically understand different points of view, i.e., whether other news articles reporting on the same problem agree or disagree with the reference story. This can then lead to the identification of news articles that propagate false rumors (a.k.a., "fake news").

In this paper, we propose a novel agreement-aware search framework, Maester, for dealing with the problem of rumor detection. Given an *investigative question* summarizing some news story or topic, Maester will retrieve *related* articles to that question, assign and display top articles from *agree*, *disagree*, and *discuss* categories to users, and thus provide a more holistic view. Our work makes two technical observations. First, relatedness can commonly be determined by keywords and entities occurred in both questions and articles. Second, the level of agreement between the investigative question and the related news article can often be decided by a few key sentences. Accordingly, we design our approach for relatedness detection to focus on keyword/entity matching using gradient boosting trees, while leveraging recurrent neural networks and posing attentions to key sentences to infer the level of agreement. Our evaluation is based on a recently published dataset from the Fake News Challenge (FNC) "stance detection" task. Extensive experiments demonstrate up to an order of magnitude improvement of Maester over all baseline methods, including the FNC winning solution, for agreement-aware search as well as slightly improved accuracy based on the same metrics used in FNC.

KEYWORDS

Rumor News; Relatedness Classification; Agreement Detection; Key sentences; Neural Models.

1 INTRODUCTION

In order to attract readership, influence opinions, and increase internet click revenue, large amounts of rumor news have been generated and widely published in recent years. It is a serious

Figure 1: The interface of our proposed agreement-aware search framework, Maester**. Instead of a traditional ranked list of *related* articles, we propose to present 3 *agree* articles, 3 *disagree* articles, and 5 *discuss* articles respectively for a given investigative question.**

problem for the news industry as unreliable news increases mistrust of the media and can have wide-reaching implications such as impact on elections [5, 24]. According to a research poll, 64% of US adults say that rumor news has caused a "great deal of confusion" about the factual content of reported current events [3]. Therefore, tools for detecting and investigating rumor news have become an urgent necessity.

Evaluating the veracity of a news story is a complex and cumbersome task, even for trained experts [10]. The first important step in moving towards automatic rumor identification is to understand what various news organizations say with respect to a specific topic or event. Often, these topics can be phrased as *investigative questions* such as, "Did Robert Plant turn down a contract to tour with

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
KDD'18, London, UK
© 2018 Copyright held by the owner/author(s). 123-4567-24-567/08/06…$15.00
DOI: 10.475/123_4

Led Zeppelin?" For this question specifically, some news articles reported Robert Plant turned down the contract while others disputed that it was not true; yet others summarized existing evidence without passing judgment on its veracity. Thus, this question is not only investigative but also *controversial*. Identifying and ranking news articles with these diverse levels of agreement is the task that we study in this paper. Specifically, we propose a diversified search framework, Maester, which is shown in Figure 1. Given an investigative question, Maester will first retrieve *related* articles that address the target question. Each of these articles is then automatically assigned a label of either *agree*, *disagree*, or *discuss*, where *discuss* pertains to articles that merely discuss or summarize other articles reporting on the reference question without making a statement of their own with regard to veracity. Splitting the results into these three categories allows the user to (a) infer quickly whether a topic is debated, (b) get an overview of the different points of view, and (c) form a more accurate judgment about the story's veracity.

This line of work focuses on controversial questions for which traditional question answering systems do not work well. For example, given a simple fact-seeking question like "Was George Washington a U.S. president?" one should only find *agree* articles. In contrast, controversial questions lack consensus.[1]

Our methodology is based on the following observations from actual rumor news articles: (1) Relatedness can often be determined by keywords and entities in both investigative questions and articles; and (2) agreement can often be inferred from a few key sentences in the article. For example, as shown in Figure 1, all retrieved articles are related through the keywords "Robert Plant" and "Led Zeppelin". Accordingly, we propose Maester as a two-step framework, which first filters unrelated articles and then predicts their agreement status. Our relatedness classifier is based on four types of features: keyword features, entity features, word2vec features, and SVD-based features. Furthermore, we use gradient boosting trees that leverage these features. After relatedness computation, we determine the positioning of the news article by first computing the top-3 sentences in the article that are closely correlated to the investigative question. Afterwards, we use these sentences and the reference question as input for an attention-based recurrent neural network to classify agreement. These news articles are then ranked to provide users of Maester with the best quality articles first.

Based on the dataset from the Fake News Challenge[2] (FNC), extensive experiments demonstrate the significant improvements of Maester over all baseline methods, including the challenge winner's solution (i.e., an ensemble model of gradient boosting trees and a convolutional neural network), thus empirically verifying our two observations. In summary, our contributions are as follows.

- **Agreement-Aware Search Framework.** We propose and build a novel agreement-aware search framework, Maester, to burst rumor news.
- **Two-Step Model.** We make two intuitive but important modeling observations based on real-world data and formulate a two-step model accordingly.
- **Extensive Evaluation.** We conduct a thorough experimental evaluation to demonstrate the effectiveness of Maester by comparing it with the FNC winner and alternative variations. For controversial questions, Maester achieves significant improvement when ranking news articles (9.24%) and improves the weighted accuracy by 2.88% at the same time.

The remainder of this paper is organized as follows. The related work is discussed first in Section 2. The formal problem formulation and framework design are then introduced in Section 3. Section 4 covers the technical details of our proposed framework and Section 5 contains extensive experiments testing that framework on real-world data. We conclude the study in Section 6.

2 RELATED WORK

In this section, we review literature related to agreement detection of news articles, question answering, and other lines of work relevant to the problem.

Stance Detection. The natural language processing community has explored stance detection for years and have formulated it in various ways. *SemEval 2016 Task 6* defines it as determining from text whether the author is in favor of, against, or neutral towards a given target [13]. In this shared task, the text is a tweet and the target is an entity without any description. In the same line of work, researchers have explored how to decide whether a tweet or an article favors one specific entity over others [23]. However, finding agreement with respect to an investigative question is more challenging than determining the stance for specific entities, because any subtle changes in the wording may lead to a completely different interpretation of the question.

Mohammad et al. first released a dataset for tweet stance [12], and later studied sentiment and stance for tweets [14]. Other approaches to stance detection in social media include semi-supervised topic models to classify stance [28] and latent feature extraction [30]. Furthermore, stance detection has been explored in Chinese microblogs [29] and online discussion forums [21]. All of these tasks require exactly one targeted entity, however, investigative questions may contain more than one entity. Thus, these methods cannot be directly adopted for our use case.

Agreement Detection in FNC-1. In the summer of 2017, the *Fake News Challenge* (FNC) ran its first contest on agreement detection. The task of this contest was to determine agreement given pairs of headlines and news articles. The challenge provides a partially labeled dataset, denoted in the following as *FNC-1*, which is based on the *Emergent* dataset [9], and contains rumor news. The winner of the FNC-1 [15] developed an ensemble model of a tree-based model and a CNN-based model. Similar to the solution to rumor news detection proposed in this work, the tree-based model utilizes a set of handcrafted features, however, it neglects important entity features. The CNN-based model on the other hand can extract features automatically but its performance is not as good as that of the tree-based model. We use the FNC-1 dataset for our evaluation and compare Maester with the winner's solution in Section 5 thoroughly. Note that all challenge winners [15, 27, 31] in SemEval and FNC take advantage of both handcrafted and neural-network extracted features. Maester also follows the same paradigm.

[1] We recognize the sensitivity and importance of not propagating conspiracy theories (e.g., "Did 9/11 really happen?") and, for now, propose to deal with this challenge by limiting candidate results to trusted sources.

[2] http://www.fakenewschallenge.org/

Textual Entailment. Another related line of work is textual entailment, which studies whether a text entails, contradicts, or not related to a certain hypothesis [2, 18, 26]. However, entailment emphasizes the logical relation of text and hypothesis where the text is commonly only one sentence and is thus much shorter than a news article.

Question Answering. Question answering (QA) is the task of finding an article, a passage, or a sentence to answer a given question [25]. Most, if not all, of these questions have a specific and clear answer. The problem definition of QA thus differs from rumor news identification as the ground truth to the later problem is debatable. As a result, traditional QA systems struggle to address this modified problem.

Search Diversification. Search result diversification [7] has been originally proposed to deal with query ambiguity, and has been applied to improve personalized search [17] afterwards. In the same context, query reformulation [19] has been explored to retrieve more relevant articles per target, and thus diversifying the search results. In [6], the authors furthermore propose to consider the proportionality of articles instead of emphasizing diversity. However, depending on the diversity measure, articles within the same agreement group can also be diverse. Therefore, directly applying search diversification methods cannot guarantee the presence of all agreement groups. As showing multiple ranked lists for different agreement groups essentially enforces the results to be diversified, we may also apply similar techniques to optimize the overall quality of the ranked lists per agreement group.

3 PRELIMINARIES

In this section, we will first formulate the problem and then discuss our framework design and alternative models.

3.1 Problem Formulation

Given a question q, we assume that a collection of candidate articles $\mathcal{D}(q)$ is provided. There are many ways to obtain such a collection (*e.g.*, taking the top-100 articles from a collection based on BM25 scores), which is not the focus of this paper.

Definition 1 (Agreement Classes). *Given an investigative question q and an article $d \in \mathcal{D}(q)$, we define four possible classes to describe how d relates to q:*

(1) **Agree:** *The article agrees with q*
(2) **Disagree:** *The article disagrees with q*
(3) **Discuss:** *The article discusses the same question, but does not take a position w.r.t. q*
(4) **Unrelated:** *The article addresses a question other than q.*

Previously, we have noted that the key to rumor detection is to find those questions that lead to controversial discussion of a topic, i.e., on which people have more than one opinion. More formally, we use the following definition for controversial questions.

Definition 2 (Controversial Question). *When an investigative question has at least one agreeing and one disagreeing news article in $\mathcal{D}(q)$, we refer to it as a controversial question.*

For understanding controversial questions and agreement classes, consider the following example taken from the FNC that shows text snippets referencing the running example question "Did Robert Plant turn down a contract to tour with Led Zeppelin?". Here, the controversial question leads to different news articles that can be categorized according to statements made in those articles.

Example 1. *The running example showing relatedness classification and agreement detection for question "Did Robert Plant turn down a contract to tour with Led Zeppelin?"*

Question	Did Robert Plant turn down a contract to tour with Led Zeppelin?
Agree	... Led Zeppelin's Robert Plant turned down £500 MILLION to reform supergroup. ...
Disagree	... No, Robert Plant did not rip up an $800 million deal to get Led Zeppelin back together. ...
Discuss	... Robert Plant reportedly tore up an $800 million Led Zeppelin reunion deal. ...
Unrelated	... Richard Branson's Virgin Galactic is set to launch SpaceShipTwo today. ...

Formal Problem Definition. Our goal is to declare whether a candidate news article is related to an investigative question and, if so, how it is positioned w.r.t. that question. More formally, we say that $\forall q \in \mathcal{Q}$ and $d \in \mathcal{D}(q)$, there is a label $y \in$ {unrelated, discuss, agree, disagree} that describes the relationship between q and d. Note that it is possible that, for a given reference question, any agreement class may contain multiple news articles. Therefore, we desire the output of the agreement identification step to be ranked lists per class as shown in Figure 1, with k_{agree} *agree* articles, $k_{disagree}$ *disagree* articles, and $k_{discuss}$ *discuss* articles, for example, $(k_{agree}, k_{disagree}, k_{discuss}) = (3, 3, 5)$ as shown in the running example. To measure whether an article is related or unrelated, we determine a confidence score $rel(q, d) \in [0, 1]$ where a 0 signifies that q and d are unrelated and 1 that d is highly relevant to q. For those that are related, the level of agreement can be measured with a classifier that maps an agreement score $\beta(q, d)$ to range $[-1, +1]$ with -1 indicating maximum disagreement and $+1$ indicating maximum agreement. Our models then estimate $P(y|q, d)$ for ranking, where (1) $P(y|q, d) = \beta(q, d)$ holds for agreeing articles, (2) $P(y|q, d) = -\beta(q, d)$ holds for disagreeing articles, and (3) $P(y|q, d) = rel(q, d)$ holds for discussing articles. For each $d \in \mathcal{D}(q)$, we define its agreement $\hat{y}$ as $\arg\max_y P(y|q', d)$. Thus, $\hat{y}$ and the corresponding $P(\hat{y}|q, d)$ determine the membership and ranking of an article d w.r.t. q in these three lists.

Model Training & Evaluation. To train our models, we use a training set containing labels for question q and candidate article pairs d as labelled above. After the models have been trained, they are evaluated on a separate set of questions and their candidate articles analogous to the training and verification methodology applied in the FNC. This process holds for both, classification and ranking, tasks.

3.2 Framework Overview

We structure our approach in two steps analogous to the two problems discussed above, i.e., (1) whether an article is *related* to a given question; and (2) labeling a related news article. Intuitively, the actual modeling challenges for these two problems are substantially different. We observe that content words and entity mentions in

Table 1: FNC-1 Dataset Statistics.

	Investigative Questions	Articles	Labeled Pairs	Controversial Questions	Unrelated	Discuss	Agree	Disagree
Training	1,648	1,683	49,972	260	73.13%	17.83%	7.36%	1.68%
Testing	894	904	25,413	211	72.20%	17.57%	7.49%	2.74%

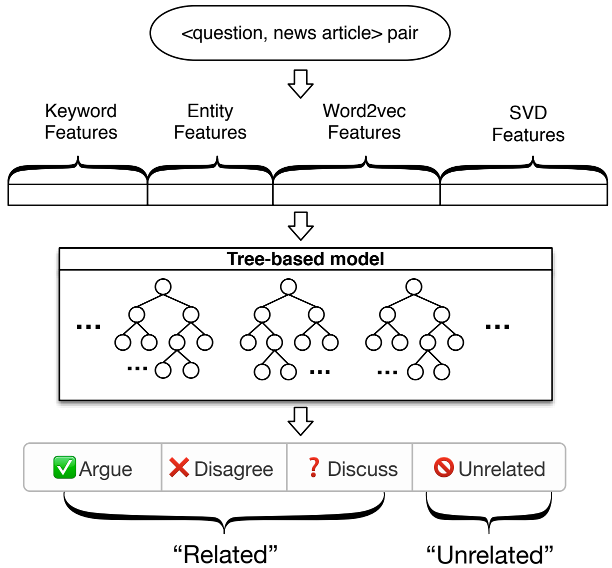


Figure 2: Tree-based Classification.

both the given question and the article may play important roles in predicting their relatedness. That is, if the article discusses the same or similar set of entities, they should be related.

Observation 1 (Relatedness: Keywords and Entities.). *Overlapping keywords and entities between the given question q and a news article d are crucial for determining their relatedness.*

In contrast, overlapping entities are weak signals for finding the level of agreement w.r.t. a question. Specifically, either an *agree* article or a *disagree* article might contain a large number of overlapping keywords and entities. We observe that for the task of agreement detection, non-entity words such as adjective, adverbs, and negation words are more important. Furthermore, inspired by many examples such as Figure 1 and the running example in Section 3, we observe that only a few sentences, referred to as *k*ey sentences, in an article will often reflect the stance w.r.t. a given question, especially for news articles. For example, from the sentence "No, Robert Plant did not rip up an $800 million deal to get Led Zeppelin back together." one can easily derive that this article disagrees with the question "Did Robert Plant turn down a contract to tour with Led Zeppelin?". Thus, we propose our second conjecture as follows.

Observation 2 (Agreement: Key Sentences.). *An article's agreement w.r.t. a given question q is largely decided based on a few key sentences. This is due to the "inverted pyramid" structure that journalists often follow when writing a news story [16].*

Finally, we observe that in practice, the distribution of agreement labels is often skewed. As shown in Table 1 for the FNC-1 dataset, the majority of labels are *unrelated* whereas *disagree* has the least number of annotations. Avoiding overemphasis of *unrelated* news articles further motivates the following two-step framework.

(1) **Relatedness Classification**. First, we merge the four stances into two categories, i.e., *related* and *unrelated*, and focus on the binary classification. Based on Observation 1, for a given question and an article, we design keyword, entity, word2vec, and SVD features based on the keywords and entity mentions. Taking these features as input, as shown in Figure 2, our tree-based model leads to a test accuracy close to 98% in our experiments, which verifies this observation empirically.
(2) **Agreement Detection**. Second, for all *related* articles, we build a 3-class classification model to estimate the agreement class. Inspired by Observation 2, for a given question and an article, we project the question and every sentence of the article into the embedding space and then choose the most similar sentences as key sentences. Afterwards, we inject these sentences into an efficient attention-based recurrent neural network model. Note that if we instead train a tree-based model using the same keyword/entity-based handcrafted features designed for relatedness classification, the performance drops significantly which is consistent with our observation.

4 METHODOLOGY

This section first introduces our feature design for the tree-based model which is used to compute relevance scores. Then, we present our recurrent neural network model with attention mechanism.

4.1 Handcrafted Features

Before explaining our features, we first make a general observation. Intuitively, the first and the last paragraphs of a news article are likely summaries of its reference topic and thus are highly relevant for feature extraction. Specifically, we not only apply the feature extraction to the news article d, but also extract features of d's first and last paragraphs for a given $\langle q, d\rangle$ pair. These three feature vectors are concatenated before being used as input for the tree-based model.

As shown in Figure 2, we design the following features for each question-article pair and categorize them into four different types: (1) keyword features, (2) entity features, (3) word2vec features, and (4) SVD features.

Keyword Features. To compute keyword features, we first prune stopwords in questions and articles. Second, we compute the keyword overlap between the question and the news article. Imagine the question q and the article d as two multi-sets of words. Then, we can calculate the intersection between them as follows.

$$|q \cap d|_{\text{raw}} = \sum_{w \in q} \min\{\mathit{freq}(w, q), \mathit{freq}(w, d)\}$$

Here, $\mathit{freq}(w, q)$ and $\mathit{freq}(w, d)$ are the counts of words in the question q and the article d, respectively. Intuitively, a bigger overlap implies a higher relevance. However, we also observe that different keywords may have different importance. Therefore, we define the weighted intersection as

$$|q \cap d|_{\text{weighted}} = \sum_{w \in q} \min\{f(w,q), f(w,d)\} * idf(w)$$

where the inverted document frequency (idf) of word w is denoted as $idf(w)$. Using the *idf* score of a word automatically scales down the importance of popular words. Furthermore, to make sure the computed scores are comparable across different questions, we normalize them to $[0,1]$ by dividing $|q \cap q|_{\text{raw}}$ and $|q \cap q|_{\text{weighted}}$.

Entity Features. We apply the spaCy[3] toolkit to extract named entities from questions and articles. As both question and news article may contain multiple entities, we model them using the bag-of-entities representation. Analogous to the keyword features above, we can then compute their intersection and IDF-weighted intersection.

word2vec Features. So far, all features are based on exact word matching. To enable semantic matching and partially solve synonym problem, we apply word embedding techniques. We utilize pre-trained word2vec 300-dimension vectors and use the average vector to build vector representations for each question and news article. The word vectors are trained on a Google News corpus with 100 billion words and a vocabulary size of 3 million words.

SVD Features. In our chosen setup with an investigative question and a news article, the question text is usually short while the article text could be long. Thus, it is difficult to find a unified topic model that works well for both. As an approximation, we use PCA analysis [8] to determine the topics. More specifically, we first get the TF-IDF weighted bag-of-words representations of all articles after which we apply SVD decomposition to get the principal components. Finally, we project all questions and articles onto these components to get dense feature vectors. We further compute similarity based on these dense feature vectors, which indicates whether the news articles is related to the headline or not.

Comparison with the FNC winner's model. The winning model of the FNC has a tree-based model using similar features to the model we present here for relevance computation. Specifically, they also use TF-IDF, word2vec, and SVD features. However, as described in Observation 1, we observe that *entity features* play an important role in establishing relevance. Our experimental results confirm such observation and shows that *entity features* are more important than *word2vec features*. Finally, we note that *sentiment features* that have been explored in alternative models are not effective in practice thus we do not leverage them in our model. See Section 5.5 for details.

4.2 RNN+attention Architecture

Although handcrafted features work well for relevance classification, they cannot capture more subtle expressions that indicate agreement or disagreement. However, recent advances on neural networks provide an automatic, high-quality way for this type of feature extraction. We explain the convolutional neural network (CNN) used in the FNC winner's implementation next and then introduce Maester's solution based on recurrent neural networks (RNNs).

[3]http://spacy.io/

FNC Winner's CNN-based model. The FNC winner leverages a CNN-based neural network[4] to handle agreement classification. Given a question and an article, it first concatenates word embeddings for the question and the article respectively to form two matrices. After five 1-D convolutional layers, the question and the article representations are merged into three dense fully connected layers. The convolution operator only focuses on the word dimension and will not affect the embedding dimension. In the end, a soft-max layer is adopted for the 4-class classification, *i.e.*, *unrelated*, *discuss*, *agree*, and *disagree*.

We observe that this model is not very robust because (1) the convolutional layer is sensitive to the absolute position of words, and (2) there are many redundant words in the article, which are essentially noise for the agreement detection task. However, as this model is combined with a tree-based model, the authors presumably add this component to prevent its counterpart's tendency to overfit. In fact, we observe that the tree-based model works better for relevance classification if it is not combined with the CNN-based model which we discuss in detail in Section 5.

Our RNN+Attention Model. Recent research has shown that recurrent neural networks (RNNs), such as long-short term memory (LSTM) networks, often perform better than CNN in understanding text. For example, a tree-based RNN achieved state-of-the-art performance for sentiment analysis as discussed in [22]. Furthermore, related work has shown that LSTM structures can outperform CNN structures in sequence labeling tasks [11].

While there are many variations of LSTM, we use the following one for the rumor detection problem. Suppose the input sequence is $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$, where $\mathbf{x}_k \in \mathbb{R}^l$ is the vector representation of the k-th element. At each position k, there is a set of internal vectors, including an input gate $\mathbf{i}_k$, a forget gate $\mathbf{f}_k$, an output gate $\mathbf{o}_k$, and a memory cell $\mathbf{c}_k$. All these vectors together are used to generate a hidden state $\mathbf{h}_k \in \mathbb{R}^d$ as

$$\begin{aligned}
\mathbf{i}_k &= \sigma(\mathbf{W}^i \mathbf{x}_k + \mathbf{V}^i \mathbf{h}_{k-1} + \mathbf{b}^i) \\
\mathbf{f}_k &= \sigma(\mathbf{W}^f \mathbf{x}_k + \mathbf{V}^f \mathbf{h}_{k-1} + \mathbf{b}^f) \\
\mathbf{o}_k &= \sigma(\mathbf{W}^o \mathbf{x}_k + \mathbf{V}^o \mathbf{h}_{k-1} + \mathbf{b}^o) \\
\mathbf{c}_k &= \mathbf{f}_k \odot \mathbf{c}_{k-1} + \mathbf{i}_k \odot \tanh(\mathbf{W}^c \mathbf{x}_k + \mathbf{V}^c \mathbf{h}_{k-1} + \mathbf{b}^c) \\
\mathbf{h}_k &= \mathbf{o}_k \odot \tanh(\mathbf{c}_k)
\end{aligned}$$

where σ is the sigmoid function, $\odot$ is the element-wise multiplication of two vectors, and all $\mathbf{W}^* \in \mathbb{R}^{d \times l}$, $V^* \in \mathbb{R}^{d \times d}$, and $b^* \in \mathbb{R}^d$ are parameters to be learned.

However, directly applying RNNs to model long articles is challenging. In order to capture and memorize useful information, RNNs require a bigger state size for the longer texts, and thus decrease the efficiency. Fortunately, based on Observation 2, it is possible to reduce long news articles to a few key sentences with only minimal loss of output quality. To obtain these sentences, we leverage word embeddings again. Considering the limited training data and the model simplicity, we define the sentence embedding as the average of its pre-trained word embeddings. Specifically, we utilize the pre-trained Glove 300-dimension vectors and skip the stopwords when computing the average vector. Since questions usually consist of one or two sentences, we apply the same approach

[4]https://github.com/Cisco-Talos/fnc-1/tree/master/deep_learning_model

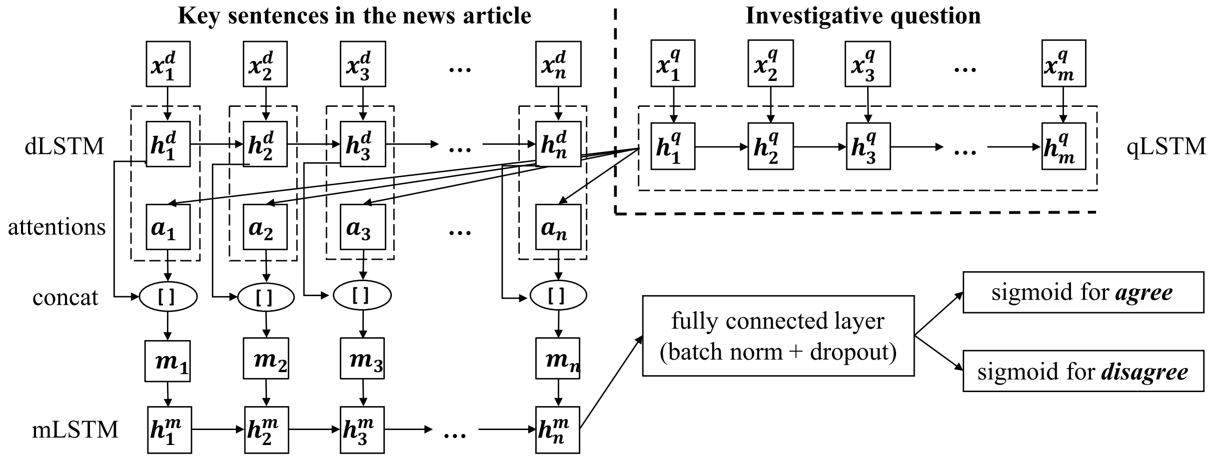


Figure 3: The architecture of our proposed RNN+attention Model.

for them. We then evaluate the cosine similarity between the given question and all sentences in a news article. The sentences with the highest similarities to the question are the key sentences which then replace the news article text. The sentences are organized in their relative similarity order. In the following, we assume a default number of key sentences k of 3. The effect of different values for k will be discussed in Section 5.

We follow Wang et al. [26] to build a neural attention model, as shown in Figure 3. Formally, we have two sequences $\mathbf{X}^q = \{\mathbf{x}_1^q, \mathbf{x}_2^q, \ldots, \mathbf{x}_m^q\}$ and $\mathbf{X}^d = \{\mathbf{x}_1^d, \mathbf{x}_2^d, \ldots, \mathbf{x}_n^d\}$, where m is the length of the question and n is the number of tokens in the selected sentences, and each $\mathbf{x}$ is an embedding vector of the corresponding word. We build three LSTMs in total: qLSTM processes $\mathbf{X}^q$ and generates its hidden states $\mathbf{h}_j^q$; dLSTM reads $\mathbf{X}^d$ and outputs hidden states $\mathbf{h}_k^d$; and mLSTM models the matching between the question and the article and produces hidden states $\mathbf{h}_k^m$ which we discuss in detail later.

Next, we generate the attention vectors $\mathbf{a}_k (1 \le k \le n)$ as follows.

$$\mathbf{a}_k = \sum_{j=1}^{m} \alpha_{kj} \mathbf{h}_j^d \tag{1}$$

Here, α_{kj} is an attention weight that encodes the degree to which $\mathbf{x}_k^d$ in the article is aligned with $\mathbf{x}_j^q$ in the question.

The attention weight α_{kj} is generated as

$$\alpha_{kj} = \frac{exp(e_{kj})}{\sum_{j'} exp(e_{kj'})} \tag{2}$$

$$e_{kj} = \mathbf{w}^e \cdot tanh(\mathbf{W}^q \mathbf{h}_j^q + \mathbf{W}^d \mathbf{h}_k^d + \mathbf{W}^m \mathbf{h}_{k-1}^m) \tag{3}$$

where $\cdot$ is the dot product between two vectors and the vector $\mathbf{w}^e \in \mathbb{R}^d$ as well as all matrices $\mathbf{W}^* \in \mathbb{R}^{d \times d}$ are the parameters to be learned.

The input of mLSTM, $\mathbf{m}_k$, is the concatenation of $\mathbf{h}_k^d$, which is the hidden state for the k-th token in the article, and $\mathbf{a}_k$, which is its attention weighted version. Thus, mLSTM will 'remember' important matching results, and 'forget' non-essential ones.

To predict the agreement class of a news article, we use $\mathbf{h}_N^m$, i.e., the last hidden state of mLSTM. Instead of using a soft-max layer for 3-class classification, we choose to use two separate sigmoid modules for *agree* and *disagree*, which make the predicted scores comparable across different articles.

Furthermore, we use an agreement score $\beta(q, d) \in [-1, +1]$ with -1 indicating maximum disagreement and $+1$ indicating maximum agreement. When $\text{score}_{\text{agree}}$ is larger than $\text{score}_{\text{disagree}}$, we let $\beta(q, d)$ be a positive score of $\text{score}_{\text{agree}}$. Otherwise, we set $\beta(q, d)$ as a negative score of $-\text{score}_{\text{disagree}}$. Based on $\beta(q, d)$, we can define $P(y|q, d)$ accordingly as described in Section 3.

4.3 Online Pipeline

Once an investigative question q and its candidate collection $\mathcal{D}(q)$ arrive for processing, Maester will first apply the tree-based model to compute the relatedness score $rel(q, d)$ for each article $d \in \mathcal{D}$. Then, for the articles with $rel(q, d) \ge 0.5$, Maester will leverage the attention-based RNN to determine the agreement classes for each relevant news article. We will thus compute the agreement $\hat{y}$ based on $P(y|q, d)$. Note that at this stage, $P(y = \text{discuss}|q, d) = rel(q, d) \ge 0.5$. Therefore, if we finally get $\hat{y}$ as *agree* or *disagree*, its probability will be more than 0.5. The *agree* and *disagree* articles will be ranked based on the absolute values of $\beta(q, d)$, while *discuss* articles will be ranked by their $rel(q, d)$ scores.

5 EXPERIMENTS

5.1 Dataset

We use the recently published dataset, FNC-1[5], from the Fake News Challenge. FNC-1 was designed as a fake news detection dataset and it contains 75,385 labeled headline and article pairs. The labels are analogous to the agreement classes that we consider, namely *agree*, *disagree*, *discuss*, and *unrelated*. Each headline in the dataset is phrased as a statement. Note that our techniques hold for statements as well as investigative questions. In fact, we observe that investigative questions are most commonly rephrased statements. Detailed statistics of the dataset can be found in Table 1.

Note furthermore that the topics mentioned in the questions and articles in the training and testing set are significantly different.

[5] https://github.com/FakeNewsChallenge/fnc-1

Consequently, this setting is challenging and even harder than a real-world setup where partial overlap can often be assumed.

5.2 Evaluation Metrics

This evaluation focuses mainly on ranking accuracy. Since some of the questions in this dataset are not controversial, we split the result into two parts. In addition, we use the FNC metrics, i.e., relatedness accuracy and weighted accuracy, to study the performance of two components separately and compare them to existing approaches.

NDCG@K and Average NDCG. Because we are presenting three ranked lists to the user, we utilize the average normalized discounted cumulative gain, $NDCG@K$, for evaluation where K is a small number like 3 or 5. The gain is then defined as follows. In the ranked list of label *agree*, only *agree* articles will receive a score of 1, while other articles will get a zero score. Articles in the *disagree* list are treated analogously. In the *discuss* list, all *related* (*agree*, *disagree*, and *discuss*) articles will get a score of 1, while *unrelated* articles will receive a zero score. The discounted cumulative gain is calculated as $DCG@K = gain_1 + \sum_{i=2}^{K} \frac{gain_i}{\log_2(i)}$. The $NDCG@K$ is then computed as a normalization by the best possible $DCG@K$. If the ideal $DCG@K$ is 0 for any of the lists, we will skip it.

Considering the numbers of articles from each class displayed in our proposed interface (*i.e.*, Figure 1), we evaluate $NDCG@3$ for both *agree* and *disagree* classes, and $NDCG@5$ for the *discuss* class. To conduct an overall comparison, we adopt the average NDCG score of these three classes, denoted as **Avg NDCG**.

Relatedness Accuracy. In the relatedness accuracy, we consider only two classes: *related vs. unrelated.* The score is then calculated by articles (not) matching their underlying class correctly.

Weighted Accuracy. This is the official metric for FNC-1: For a question and an article, if the model successfully predicts the *related*/*unrelated* label, it receives a score of 0.25. For a question and a *related* article, if the model successfully predicts *agree*, *disagree*, or *discuss*, it receives a score of 0.75. The final score is then normalized by the maximum possible score.

5.3 Experimental Setting

All experiments are conducted on a single machine equipped with an Intel Xeon processor E5-2650@2.2GHz and a NVIDIA GeForce GTX 1080. In Maester, the tree-based model is implemented in XGBoost [4] and the RNN+attention model is implemented using Tensorflow [1]. The source code is available in the author's Github[6].

Our Model, Maester**.** By default, the number of key sentences, k, is set to 3, and the number of training epochs is set to 10. For further details on the parameters, please refer to the study on parameter sensitivities in Section 5.7. As our models, and thus the results, contain some randomness, we run all experiments multiple times and report the average performance.

FNC-1 Winner. As we discussed before, the FNC-1 winner's solution is an ensemble of a tree-based and a convolutional neural network (CNN) models. This combined model is able to detect the relatedness of the article effectively, primarily due to their effective tree-based model with human designed features like TF-IDF weighted keywords. However, it is limited in detecting the actual *agree* or *disagree* label of articles. Since the dataset is imbalanced, most of the related articles are labelled *discuss* and *disagree* labels are rare. Thus, the winner's solution will aggressively classify most of articles as *discuss* and the rest as *agree*, in order to achieve a high overall accuracy. In the following, we use **FNC Winner (Tree)** and **FNC Winner (CNN)** to denote the tree-based model and the CNN model in FNC-1 winner's original solution respectively. We report the best performance for FNC-1 Winner during the competition.

[6]https://github.com/shangjingbo1226/Maester

Table 2: Ranking performance of the agreement-aware search framework.

Method	Agree NDCG@3	Disagree NDCG@3	Discuss NDCG@5	Avg NDCG
All Questions				
FNC-1 Winner	**51.71%**	2.31%	61.51%	38.51%
Maester	48.11%	**20.38%**	**68.20%**	**45.56%**
Controversial Questions				
FNC-1 Winner	**43.75%**	2.58%	50.84%	32.39%
Maester	40.88%	**19.13%**	**64.89%**	**41.63%**

Alternative Models. As an alternative to our two-step framework, we also considered more straightforward models that have been applied in similar use cases before. The first of these is **bag-of-words**. It is unsuitable for our use case as language is evolving and there may be different vocabulary present in the application than in the training data. However, combining bag-of-words with some feature selection techniques leads to some interesting keywords that signal different types of agreement. For example, we observe that "reportedly" is a strong signal for *discuss.* We tried incorporating keyword lists based on the bag-of-words model in our own framework, however, improvements were negligible. Another type of models that is widely adopted when learning to match questions and articles is **matrix factorization** [20]. In our experiments, we observed that this technique has worse and unstable performance for this particular problem. Again, this is caused by the fact that not all words appearing in the application or test dataset are covered in the training data. For example, the weighted accuracy of the bag-of-words model is only 77.64%. The weighted accuracy of the matrix factorization approach is similar. Therefore, they are not included in this evaluation.

5.4 Ranking Evaluation

We evaluate the results as three ranked lists because we consider rumor detection as our target application. As mentioned before, we strongly believe that agreeing and disagreeing news articles can provide users with a more holistic view of the available data, thus, improving rumor detection. Results of our ranking experiments with Maester are shown in Table 2. Compared to FNC-1 Winner, Maester's Avg NDCG is much higher, independent whether all types or only controversial questions are evaluated. Specifically, we observe an absolute ranking improvement of 9.24% for controversial questions and an improvement of 7.05% over all questions.

Moreover, Maester's NDCG score in the *disagree* class is astonishing. Compared to the FNC-1 Winner's results, we show an improvement of 10x. The reason is that FNC-1 Winner aggressively

Table 3: Accuracy of relatedness classification.

Method	All Questions	Controversial Questions
FNC-1 Winner	96.96%	96.25%
FNC-1 Winner (Tree)	97.70%	97.30%
FNC-1 Winner (CNN)	76.96%	70.01%
Maester	**97.87%**	**97.54%**

Table 4: Relative importance of each feature type in Maester**.**

Feature	Keyword	Entity	word2vec	SVD
Importance	29.68%	22.50%	13.75%	34.07%

predicts articles as *agree* and *discuss* where very few articles are categorized as *disagree*. Such biased prediction can still harvest a high weighted accuracy, but it gets punished when evaluating ranking performance. The improvement on the NDCG score in the *discuss* class is also noticeable, while the NDCG score in the *agree* class is slightly lower than the reference score but is still comparable.

5.5 FNC metric: Relatedness Accuracy

Next, we focus on the relatedness classification, as shown in Table 3. Interestingly, FNC-1 winner (Tree) achieves better performance than FNC-1 Winner, independent of the question specifics. Therefore, we argue that the effectiveness of FNC-1 Winner when it comes to relatedness classification is mainly due to its tree-based model.

Comparatively, Maester always shows the best performance which demonstrates the importance of the added entity features compared to previously utilized sentiment features which tend to be noisy. An accuracy over 97% demonstrate that Maester's tree-based model built upon handcrafted features is precise enough to predict whether a document is related or not.

To compare the significance of different features, we calculate the relative feature importance for each feature type using the built-in function in XGBoost [4], as shown in Table 4. Here, we can see that the combined importance of keyword features and entity features is significant, i.e., 52.18%. Therefore, Observation 1 is empirically verified with this experiment.

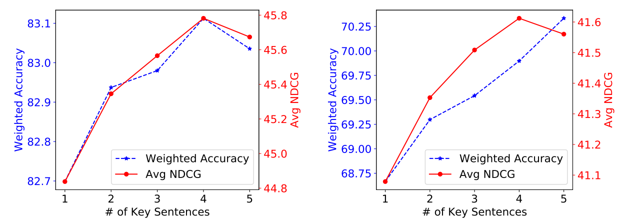
5.6 FNC metric: Agreement Accuracy

The task of agreement classification is more challenging than relatedness computation as it requires the model to distinguish articles in a more subtle way. The official metric (i.e., weighted accuracy) in FNC-1 is used for the evaluation. Table 5 presents the results. We find that Maester outperforms FNC-1 winner where the absolute improvement of accuracy is 0.96% and 2.88% on all questions and controversial questions respectively. Considering that FNC-1 winner has won the FNC by a margin of 0.05%, these improvements are significant.

In fact, recall that Maester relies only on the top-3 key sentences from the article, whereas FNC-1 Winner considers all sentences in the article. These results reflect that using only three key sentences can still capture enough information to detect agreement. Therefore, Observation 2 is verified empirically.

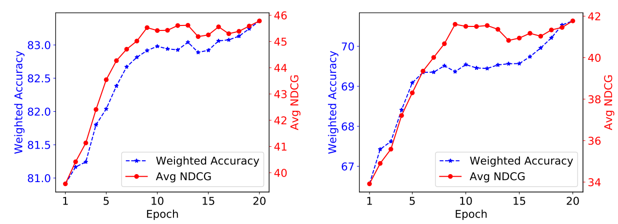
Table 5: Weighted accuracy of agreement detection. Note that FNC-1 winner **wins the challenge by an advantage of** 0.05%**.**

Method	All Questions	Controversial Questions
FNC-1 Winner	82.02%	66.66%
Maester	**82.98%**	**69.54%**



(a) On all questions. (b) On controversial questions.

Figure 4: How many key sentences are enough?



(a) On all questions. (b) On controversial questions.

Figure 5: Convergence study on test data.

5.7 Parameter Sensitivities

Here, we study the parameter sensitivities for the two major parameters in Maester: (1) the number of key sentences, k and (2) the number of epochs needed for model convergence.

As shown in Figure 4 only knowing the top sentence of an article already provides good quality results. When more key sentences are available, the weighted accuracy on controversial questions grows constantly, while the ranking performance drops a little when $k = 5$ is reached. This implies that more sentences disclose more information, however, a few key sentences are enough for good ranking quality, which further supports Observation 2.

Second, we studied the convergence of the RNN+attention model in Maester in Figure 5. The results show that the result quality, measured with either weighted accuracy or Avg NDCG, stabilizes after 10 epochs. This is a promising time span for early stops and savings on training time.

5.8 Efficiency Evaluation

Building the whole model needs less than 1 hour, including the tree-based model and the RNN+attention model. However, in a real-world application, online serving time is more important than offline model training. With that in mind, we observe that Maester can process a pair of question and article within about 5.86 ms. Specifically, in our setup Maester spends about 0.16 seconds on average to present the final results to the user analogous to the results shown in Figure 1.

Table 6: Case Study: Key sentences as determined by Maester **for agreement detection.**

Question	Is it true that a woman pays $20,000 for third breast to make herself LESS attractive to men?
	Our Selected Top-3 Similar Sentences
An *agree* article	1. No, you do not need to adjust your sets, you are actually looking at a woman with three breasts. 2. Jasmine added: I got it because I wanted to make myself unattractive to men. 3. She denies that she had the extra breast put on to get fame and fortune.
A *disagree* article	1. Did a woman claiming to have a third breast play a hoax on us? 2. A top plastic surgeon, Mr Nilesh Sojitra, also cast doubt over the surgery after claiming no reasonable doctor would perform the operation. 3. Snopes.com came up with a number of intriguing arguments that could indicate Jasmine Tridevil did not actually pay $20,000 for an extra breast.

5.9 Case Study

To study the results returned by Maester and to compare them against previous techniques, we first take a look at the controversial question "*Is it true that Woman pays $20,000 for third breast to make herself LESS attractive to men?*". For this question, Maester achieves 100% NCDG@3 in both *agree* and *disagree* ranked lists, while the FNC-1 winner has a ranking score of 29.82% and 0% respectively. We randomly pick two articles from the *agree* and *disagree* classes and show the top-3 similar sentences selected by Maester in Table 6. From these results, we observe that the chosen sentences are essential for agreement classification, which highlights our conclusions of Observation 2.

6 CONCLUSION & FUTURE WORK

In this paper, we studied the problem of rumor detection using an agreement-aware article search. We developed an agreement-aware search framework that is designed to provide users with a holistic view of an investigative question, for which the ground truth is not certain. Based on two intuitive but important observations, we designed a two-step model consisting of a tree-based model based on handcrafted features and an attention-based recurrent neural network model focusing on only a few key sentences. Our experimental results and case studies not only demonstrate the effectiveness of our model, but also verify both observations empirically.

There are many related problems and follow-up work that should be explored in the future. In the context of rumor detection, we propose using statements, here in the form of controversial questions, to further the understanding of a topic. However, it remains unclear how to derive such statements. Another line of interesting follow-up work is to allow not only a limited set of labels but to enable additional entity-driven options. For example, given the question "Who is the best basketball player in history?" many people will say "Michael Jordan" but there are others who will mention names such as "Kobe Bryant" and "Lebron James".

REFERENCES

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *OSDI* (2016), vol. 16, pp. 265–283.
[2] Androutsopoulos, I., and Malakasiotis, P. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research 38* (2010), 135–187.
[3] Barthel, M., Mitchell, A., and Holcomb, J. Many americans believe fake news is sowing confusion. *Pew Research Center 15* (2016).
[4] Chen, T., and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (2016), ACM, pp. 785–794.
[5] Connolly, K., Chrisafis, A., McPherson, P., Kirchgaessner, S., Haas, B., Phillips, D., Hunt, E., and Safi, M. Fake news: an insidious trend that's fast becoming a global problem. *The Guardian 2* (2016).
[6] Dang, V., and Croft, W. B. Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval* (2012), ACM, pp. 65–74.
[7] Drosou, M., and Pitoura, E. Search result diversification. *ACM SIGMOD Record 39*, 1 (2010), 41–47.
[8] Dunteman, G. H. *Principal components analysis*. No. 69. Sage, 1989.
[9] Ferreira, W., and Vlachos, A. Emergent: a novel data-set for stance classification. In *HLT-NAACL* (2016).
[10] Ghulati, D. Introducing factmataâĂŤâĂŤâĂŁartificial intelligence for political fact-checking. https://medium.com/factmata/introducing-factmata-artificial-intelligence-for-political-fact-checking-db8acdbf4cf1, Dec 2016. Accessed: 2018-01-16.
[11] Liu, L., Shang, J., Xu, F., Ren, X., Gui, H., Peng, J., and Han, J. Empower sequence labeling with task-aware neural language model. *arXiv preprint arXiv:1709.04109* (2017).
[12] Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X.-D., and Cherry, C. A dataset for detecting stance in tweets. In *LREC* (2016).
[13] Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X.-D., and Cherry, C. Semeval-2016 task 6: Detecting stance in tweets. In *SemEval@NAACL-HLT* (2016).
[14] Mohammad, S. M., Sobhani, P., and Kiritchenko, S. Stance and sentiment in tweets. *Special Section of the ACM Transactions on Internet Technology on Argumentation in Social Media 17*, 3 (2017).
[15] Pan, Y., Sibley, D., and Baird, S. Fake news challenge - team solat in the swen. https://github.com/Cisco-Talos/fnc-1, June 2017. Accessed: 2017-12-27.
[16] Po¨ ttker, H. News and its communicative quality: The inverted pyramidâĂŤwhen and why did it appear? *Journalism Studies 4*, 4 (2003), 501–511.
[17] Radlinski, F., and Dumais, S. Improving personalized web search using result diversification. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (2006), ACM, pp. 691–692.
[18] Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiskỳ, T., and Blunsom, P. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664* (2015).
[19] Santos, R. L., Macdonald, C., and Ounis, I. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web* (2010), ACM, pp. 881–890.
[20] Shang, J., Chen, T., Li, H., Lu, Z., and Yu, Y. A parallel and efficient algorithm for learning to match. In *Data Mining (ICDM), 2014 IEEE International Conference on* (2014), IEEE, pp. 971–976.
[21] Skeppstedt, M., Kerren, A., and Stede, M. Automatic detection of stance towards vaccination in online discussion forums. In *Proceedings of the International Workshop on Digital Disease Detection using Social Media 2017 (DDDSM-2017)* (2017), pp. 1–8.
[22] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (2013), pp. 1631–1642.
[23] Somasundaran, S., and Wiebe, J. 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* (2009), Association for Computational Linguistics, pp. 226–234.
[24] Tavernise, S. As fake news spreads lies, more readers shrug at the truth. *The New York Times* (2016).
[25] Voorhees, E. M., et al. The trec-8 question answering track report. In *Trec* (1999), vol. 99, pp. 77–82.
[26] Wang, S., and Jiang, J. Learning natural language inference with lstm. *arXiv preprint arXiv:1512.08849* (2015).
[27] Wei, W., Zhang, X., Liu, X., Chen, W., and Wang, T. pkudblab at semeval-2016 task 6: A specific convolutional neural network system for effective stance detection. In *SemEval@ NAACL-HLT* (2016), pp. 384–388.
[28] Xu, K., Bi, S., and Qi, G. Semi-supervised stance-topic model for stance classification on social media. In *Joint International Semantic Technology Conference* (2017), Springer, pp. 199–214.
[29] Xu, R., Zhou, Y., Wu, D., Gui, L., Du, J., and Xue, Y. Overview of nlpcc shared task 4: Stance detection in chinese microblogs. In *NLPCC/ICCPOL* (2016).
[30] Xu, X., Hu, F., Du, P., Wang, J., and Li, L. Efficient stance detection with latent feature. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data* (2017), Springer, pp. 21–30.
[31] Zarrella, G., and Marsh, A. Mitre at semeval-2016 task 6: Transfer learning for stance detection. *arXiv preprint arXiv:1606.03784* (2016).