A Memory Network Information Retrieval Model for Identification of News Misinformation

Nima Ebadi, Mohsen Jozani, Kim-Kwang Raymond Choo, *Senior Member, IEEE*, and Paul Rad, *Senior Member, IEEE*

Abstract—The speed and volume at which misinformation spreads on social media have motivated efforts to automate fact-checking which begins with stance detection. For fake news stance detection, for example, many classification-based models have been proposed often with high complexity and hand-crafted features. Although these models can achieve high accuracy scores on a targeted small corpus of fake news, few are evaluated on a larger corpus of fake and conspiracy sites due to efficiency limitations and the lack of compatibility with the actual fact-checking process. In this article, we propose a practical two-stage stance detection model that is tailored to the real-life problem. Specifically, we integrate an information retrieval system with an end to end memory network model to sort articles based on their relevance to the claim and then identify the fine-grained stance of each relevant article towards its given claim. We evaluate our model on the Fake News Challenge dataset (FNC-1). The results show that the performance of our model is comparable to those of the state-of-the-art models, average weighted accuracy of 82.1, while it closely follows the real-life process of fact-checking. We also validate our model with a large dataset from a real-life fact-checking website (i.e., *Snopes.com*), and the findings demonstrate the capability of the model in distinguishing false from true news headlines.

Index Terms—Deep memory networks, stance detection, fake rumors detection/debunking, news retrieval systems, information retrieval systems, fake news detection

1 Introduction

Wide distributed networks such as Twitter and Facebook have improved democracy by enabling citizen journalism [1], while the lack of verification and the speed of information spreading on these platforms have caused a growing problem of misinformation [2]. The proliferation of false rumors on social media, for example, can negatively impact political events [1], economics [3], individual users' decision making [4], and the trustworthiness of the social cyberspace [5] (see also [6]). Therefore, it is crucial to authenticate social media claims early on and prevent misinformation from going viral.

- *Nima Ebadi is with the Department of Electrical and Computer Engineering and the Secure AI and Autonomy Lab, University of Texas at San Antonio, San Antonio, TX 78249 USA. E-mail: nima.ebadi@utsa.edu.*
- *Mohsen Jozani is with the College of Business, Louisiana State University in Shreveport, Shreveport, LA 71115 USA. E-mail: Mohsen.Jozani@lsus.edu.*
- *Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249 USA, and also with the Department of Electrical and Computer Engineering and the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249 USA. E-mail: raymond.choo@fulbrightmail.org.*
- *Paul Rad is with the the Department of Computer Science and the Secure AI and Autonomy Lab, University of Texas at San Antonio, San Antonio, TX 78249 USA, and also with the Department of Electrical and Computer Engineering and the Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249 USA. E-mail: paul.rad@utsa.edu.*

Manuscript received 8 Apr. 2019; revised 7 Dec. 2020; accepted 31 Dec. 2020. Date of publication 5 Jan. 2021; date of current version 1 Sept. 2022. (Corresponding author: Kim-Kwang Raymond Choo.) Recommended for acceptance by P. Cui. Digital Object Identifier no. 10.1109/TBDATA.2020.3048961

Fact-checking is the task of monitoring the accuracy and truthfulness of popular news claims and is often carried out by teams of professional journalists [7]. These teams attempt to fight misinformation by evaluating the unverified claims that surface the web and compiling evidence to verify or refute the claim [8]. The number of fact-checking websites has reportedly tripled since 2014, and more teams join the cause every day [9]. Nonetheless, manual fact-checking can be tedious, costly, and time-consuming. The false information can well spread around the globe and cause damage before it is discovered, reviewed, and debunked by the human fact-checkers [10]. For example, a study has reported that there is an average of 13-hour lag from the time fake content appears on the web until it is debunked [11].

Despite the growing need for automated fact-checking, full automation of the process is not yet feasible. Fact-checking is a judgment intensive task that requires an in-depth understanding of the topic and sensitivity to details, which is well beyond current implementations of artificial intelligence (AI). Even in contexts where automation yields good performance, human supervision is still necessary to minimize the potential of errors [12]. Therefore, the current focus is on developing tools that can assist humans in completing parts of the process and stance detection is the crucial first step aimed at evaluating and understanding the perspective of each news article body towards a given claim [13].

Fake News Challenge Step 1 (FNC-1)[1] was an initiative by a coalition of academics and industry experts to develop stance detection tools that could ultimately be implemented

1. http://www.fakenewschallenge.org/

by fact-checking organizations. Submissions were judged based on their prediction accuracy on the unlabeled test dataset, and the three top models were meant to become open-source and accessible to fact-checking organizations. The proposed complex and powerful machine learning models in this competition achieved high accuracy scores. However, examining the current fact-checking resources [14] suggests that the tools currently in use are still simple ranking algorithms or supervised classifiers (e.g., iCheck and ClaimBuster).

As mentioned by FNC-1 challenge organizers, the large volume of content that needs to be evaluated is a critical problem in fact-checking [15], and due to resource limitations, it may be impractical to implement complex models. In addition to conventional model performance metrics, it may be necessary to consider the big data characteristics of fact-checking content. Furthermore, to evaluate a claim, fact-checkers need to find the relevant news articles first and then identify the stance of the related articles towards the claim in question [15]. However, all the models proposed in the competition treat related and unrelated articles simultaneously and allocate an equal amount of resources regardless of the article's relevance to the claim.

Manual encoding of all the features that determine a stance label is practically impossible and such an attempt will result in a very complex model with little scalability. Therefore, featureless deep learning models are preferred for the real-life fact-checking process. Among various deep learning models, recurrent neural networks are recommended as they capture the sequential information of sentences in a set of facts/evidence which is required for the task of stance prediction [16]. Understanding the stance of an article toward a given claim often requires a careful examination of the entire article as informative pieces may appear throughout the article. For instance, an article may provide supporting and refuting evidence and take a neutral (discuss) stance at the end. However, recurrent neural network models, such as LSTMs lack well-compartmentalized, long-term memory (knowledge is encoded into a dense vector) and therefore cannot accurately remember the past evidence of a potentially long article and predict its stance [17]. Due to model limitations, the majority of prior works use truncated article texts as the input of the RNN models (e.g., [16], [18], [19]).

To address this memorization problem, memory network models were introduced by Weston *et al*. [20], [21], and have been successfully applied to various tasks of text classification, language modeling, reading comprehension, etc., yielding superior performance over alternative deep learning methods such as LSTM [22], [23], [24]. Prior works suggest that memory networks are efficient in handling large chunks of text [25], and drawing transitive inferences about the informative parts of the text utilizing different encoding in memory component [26], making them appropriate for the task of stance detection [18]. In this paper, we integrate an information retrieval (IR) system with a memory network model and propose a two-stage system that is consistent with the actual process of fact-checking. In the first stage, we use a lightweight algorithm to sort the articles based on their relevance towards a given claim. Then, for the articles that exceed the relevance threshold, we apply a more complex end-to-end memory network model coupled with the IR system to identify the article's stance. Our contributions in this paper can be summarized as the following:

- We design a practical two-stage stance detection model based on a simple IR system and a sophisticated yet featureless end-to-end memory network that matches both the process flow and efficiency requirements of fact-checking.
- Unlike the prior state of the art models that make predictions with truncated news articles, the large external component of our memory network model allows for analyzing and making decisions based on full article texts.
- We demonstrate the potential of our setup for real-life news fact-checking problems on the Snopes dataset and by comparing the performance of our model on FNC-1 against the winners of the competition.

The rest of the paper is organized as follows: First, we review the stance detection models in the literature (see Section 2). In Section 3, we elaborate on the details of our model. Next, we propose our algorithm in Section 4. Then, we discuss the performance of our model on FNC-1 and compare our performance with the winners of the competition in Sections 5 and 6. Finally in Section 7, we discuss the performance on a real-life fact-checking dataset from *Snopes.com*, prior to concluding the paper in Section 8.

2 Related Work

Fact-checking is a complex problem and the full automation of the process is still farfetched [27]. Therefore, researchers have focused their efforts on developing AI models to automate some of the steps in this process and create tools for human fact-checkers. Stance detection is the first step which can be described as determining the position of one piece of text towards another [13]. Earlier stance detection models could use an individual's discourse and determine their *for* or *against* stance towards a certain issue [28]. These models have applications in political and online debate studies, e.g., [29], [30], [31] and [32].

The idea of using stance detection models for fact-checking was popularized by the organizers of the FNC-1 competition who extended this approach to compare article bodies to social media news claims and determine if an article *agrees*, *disagrees*, *discusses* or is *unrelated* to the claim in question [33].

2.1 Early Stance Detection Models for Fact-Checking

Along with the dataset, the FNC-1 organizers made available a simple baseline method that uses hand-crafted features, namely: global word/n-gram co-occurrence features along with polarity and refutation indicator features passed through a Gradient Boosting classifier [34]. The baseline achieves a weighted accuracy score of 79.53 percent with k-fold cross-validation on the training set (in this evaluation, the training and testing sets are split carefully to avoid bleeding of articles and headlines between the two sets).

Early models that are proposed for this competition rely heavily on hand-coded linguistic and lexical features and

commonly use deep learning approaches. The top solutions for FNC-1 challenge use an ensemble of gradient-boosted decision trees and convolutional neural networks (CNN) [35], an ensemble based on hard voting prediction between five six-layer multi-layer perceptrons (MLP) [36], or a simple multi-layer perceptron model with one hidden layer fed by text-based and similarity features, i.e., TF/TF-IDF representation features and cosine similarity [13].

2.2 Fact-Checking as a Multistage Problem

However, the end-goal of developing stance detection models is to build tools that help human fact-checkers identify and organize the arguments relevant to each claim [15]. In a real-world scenario, fact-checkers must first retrieve and sort the published news articles that are relevant to the claim they are evaluating and then identify whether those articles *agree*, *disagree* or *discuss* the claim at hand. Therefore, the stance detection task is composed of three sub-tasks with varying complexities: (1) retrieving related articles (often, relatively simple. Most of the participating models perform very well in this sub-task [18]), (2) extracting relative snippets/pieces of the articles (medium complexity); and, (3) analyzing the perspective of these pieces to the target claim (complex).

However, most prior works disregard such complexity variations and try to tackle all the sub-tasks with a single model with fixed complexity. Since an equal amount of resources are allocated to both simple and complex subtasks, either performance or efficiency must be compromised for the real-world implementation of these single-stage models. Simple models which are based on BOW representation of articles and headlines are generally efficient in handling large-scale data but fail to achieve high levels of accuracy, while high performing complex DNN based models with their numerous trainable parameters can be extremely complex and inefficient for actual implementation. Therefore, a multi-stage model using weak learner and strong learner combination is recommended to address the stance detection task both efficiently and effectively.

2.3 The Importance of Sequential Processing and Memory Component

More recent works highlight the importance of sequential processing for the stance detection task and recommend the use of recurrent neural networks [16]. Inspired by [37], Hanselowski *et al.* [16] propose a feature-rich stacked LSTM model with GloVe embeddings [38] that outperforms the early models in predicting the minority classes. Moreover, influenced by [39], Mrowca, and Wang [19] implement a conditional bidirectional LSTM model to tackle the task of stance detection. They combine the hand-coded features of the FNC-1 baseline with the output of the bidirectional LSTM and achieve decent results in comparison to other state-of-the-art models. However, most of these complex RNN based solutions make predictions with truncated text inputs [16], [18], [19] as efficiency issues along with vanishing gradient problems make it impossible to fit entire article texts into these models. Therefore, information loss occurs as key arguments that determine the stance against a claim can be spread across a given article. For instance, a discussing article may explain the evidence that supports a certain claim in the first few paragraphs and take a neutral stance towards the end. An RNN model that uses truncated text can mistakenly label the above article's stance as 'agree'. Therefore, long-term, well-compartmentalized memory is necessary to analyze information in different paragraphs of the article and identify the key information that determines the article's stance against the claim [18]. However, the capability of RNN models such as LSTMS in modeling complex dependencies is limited as they encode historical knowledge into a dense hidden vector and fail to remember the evidence when the data sequences are lengthy as is the case in news articles [17]. Memory networks are designed to handle large chunks of text using a large, long-term memory component [20], [21]. Memory networks are particularly appropriate for stance detection because various inference strategies can be customized over their large, well-compartmentalized memory component [18]. A key component that is missing in other variants of DL models used for sequential analysis.

2.4 Memory Networks

Initially proposed by [20], memory networks are a class of deep learning models that incorporate a long-term memory component with a trainable inference mechanism on top, i.e., using the supervised learning approach, the model learns how to effectively read and write into the memory component [25]. Memory networks are proven to be efficient for the various tasks of sequence tagging, text classification, aspect level sentiment analysis, and reading comprehension, to name a few, which require transitive, flexible reasoning over a potentially large memory component [24], [40], [41], [42].

Memory networks generally have four components that interact with the memory [18]: (1) input component encodes the input sequence to its distributed vector representation space [26]; (2) generalization component modifies the state of the memory based on the input; (3) output component outputs a value in the internal feature representation space based on the input and the memory; and, (4) response component transforms the output of the memory network to the human-readable symbol, e.g., strings of words or sentences. These components can be trained and various learning models can be utilized for each of them. If these components are neural networks, the model is called a memory neural network (MemNN) model [21].

In this research, we leverage end-to-end memory neural networks (MemN2N), proposed by [21], and customize it for stance detection. In comparison to the general framework, MemN2Ns require significantly less supervision, thereby more suitable for the process of automatic stance detection.

3 Proposed System

As shown in Fig. 1, our proposed system consists of two subsystems, namely: a) an information retrieval (IR) system, and b) an end-to-end memory network (MemN2N) which works in tandem with the IR system both sequentially and through an attention layer.

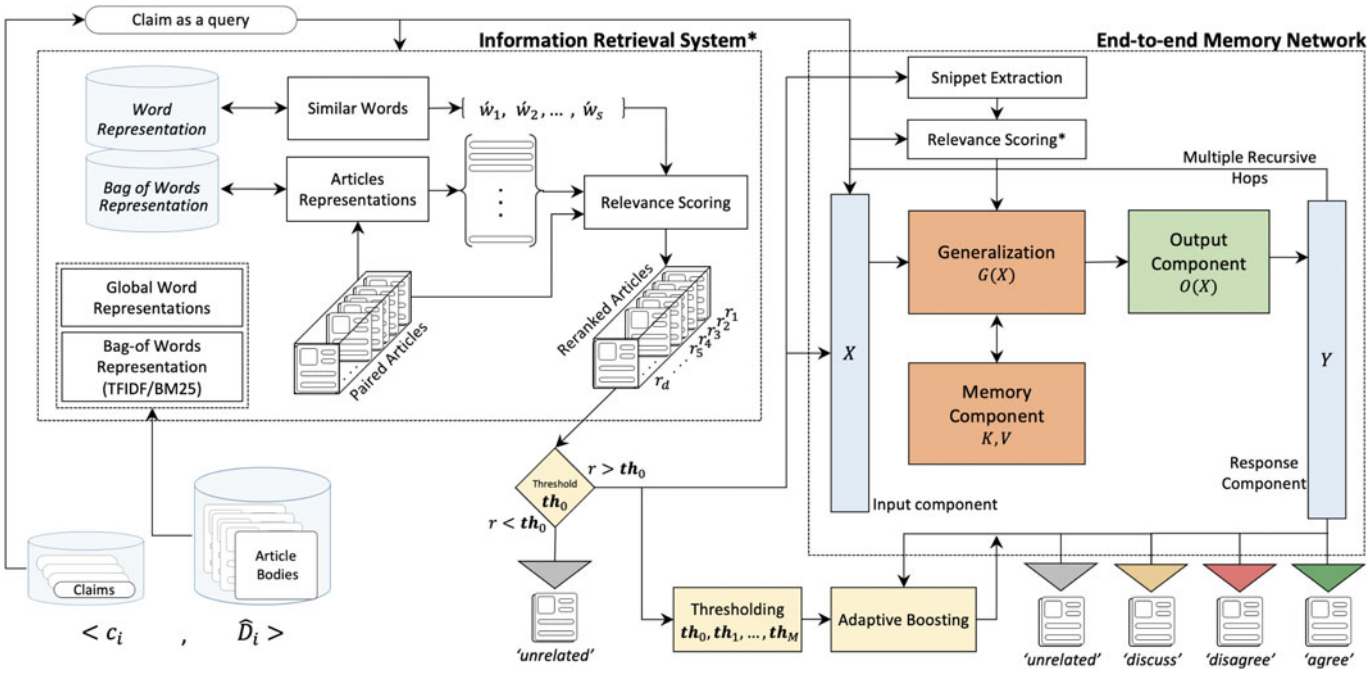

Fig. 1. Our proposed model consists of an Information Retrieval (IR) system and an end-to-end memory network (MemN2N). The IR system scores the relevancy of paired articles, and through a thresholding process, the unrelated articles are filtered out. Then, potentially related articles along with their relevancy scores are passed to the first hop of the memory network (an adaptive boosting process to filter "hard-to-distinguish" unrelated articles). Finally, through multiple hops of analysis, the end-to-end memory network predicts the label for the remaining articles.

In this section, we explain our two-stage process model. First, we present a simple while effective IR system which computes the relevance of each claim to its paired article. Then, we introduce a sophisticated end-to-end memory network (with multiple hops and state-of-the-art sentence similarity measures) integrated with the IR system to predict the fine-grained stance label of each claim towards its paired article. In what follows, we will explain in detail how we design and integrate these two systems.

3.1 Overview of Stance Detection Task: Input, Output and Problem Formulation

As mentioned in Section 2, stance detection is the task of detecting the relative perspective of one piece of article towards a given claim. In the context of fake news detection, the goal is to investigate a set of articles that are potentially related to a specific claim and determine the reaction of each article to the claim at hand [43].

In this regard, the input is a corpus of claims paired with potentially related body articles:

$$D = \{(c_i, \hat{D}_i)\}_{i=1}^{N_D}, \tag{1}$$

where c_i denotes every claim and $\hat{D}_i$ refers to the set of paired body articles $\{(d_{ij})\}_{j=1}^{N_{\hat{D}_i}}$.

Formally, the goal of stance detection task is to design a classifier f that learns to map every input claim-article pairs (c_i, d_{ij}) to one of the four following classes:

- *Unrelated* : The article and the claim discuss different topics.
- *Agree* : The article agrees with the claim.
- *Disagree* : The article disagrees with the claim.
- *Discuss* : The article merely discusses the claim, taking a neutral position.

i.e., $f : (c_i, d_{ij}) \rightarrow y$, where $y \in$ {*agree*, *disagree*, *discuss*, *unrelated*}.

3.2 Information Retrieval System (STAGE(I))

As mention in Section 2 and [15], a crucial step to stance detection is to retrieve relevant pieces of evidence to a given claim. Therefore, we first design a simple IR system to roughly determine the relevancy of the articles to a given claim. The purpose of this basic estimation is to filter out the claim-article pairs which are clearly unrelated and also detect unrelated pieces of evidence within an article as a weak learner for more fine-grained stance classification.

To build the IR system, we initially extract bag-of-word (BoW) representations of the entire corpus using both TF-IDF and BM25 ranking functions [44]. Then for every claim-article pair the relevance score is computed based on the summation of the BoW weights of the similar words between the claim and its paired article.

In this regard, for every word in the claim, we extract a list of similar words using cosine similarity and Global Vector representation of words (GloVe) pre-trained word embeddings [38].[2] Next, the claim-article relevance score is computed by summing the weights of the similar words of the claim found in the article adjusted by their cosine similarity scores, formulated as follows:

$$r_d = \sum_i \sum_j r'_{w_j, w_i} \cdot \mathbf{wf}_{w_j, d}. \tag{2}$$

2. For the vocabularies that are out of the scope of the pre-trained GloVe, we use Mittens [49] on a domain-specified corpus to extend GloVe; as to map new words with similar word embeddings to existing GloVe embeddings. We also pass the GloVe to the next stage of our model.

Which r' is the cosine similarity between the most similar words w_j to every word w_i of the claim. $\mathbf{wf}_{d,w_j}$ refers to the TF-IDF or BM25 weight of word w_j in article d.

NOTE: As mentioned earlier, the reason for implementing the IR system is to lighten the load on our main, sophisticated stance detection model without compromising the predictive power. Therefore, we seek to maximize recall (in detecting unrelated articles) while maintaining precision at 100 percent. In other words, the goal is to filter out as many unrelated articles as possible without misclassifying and therefore losing any of the related ones.

3.3 End-to-End Memory Network (STAGE(II))

As for the sophisticated stance detection section of our methodology, we design a supervised deep learning model based on end-to-end memory networks (MemN2N), and add customized attention layers along with an inference mechanism coupled with the IR system from the first stage. Next, we elaborate on the architecture of the implemented model as well as the details of the inference mechanism used to generate the stance output.

3.3.1 Input Component

First, we map the input claim-article pair to a sequence of sentence embeddings as internal feature representation of the memory component. In this regard, we parse each body article into sentences and further tokenizes them into words. Since the claims in the FNC-1 dataset are single sentences, we only word tokenize them. Initially, the vector embedding of every word is calculated using a trainable word embedding matrix, initialized with GloVe from Section 3.2. Next, we map the word vector embedding to v^s, a new vector embedding for sentences using the sentence embedding scheme of smooth inverse frequency (SIF) [50]:

$$x_i = \frac{1}{|s_i|} \sum_j \frac{a}{a + N(w_j)} \cdot \mathbf{W}^I(w_j). \tag{3}$$

Where $s_i = (w_1, w_2, \ldots, w_m)$ is a unique sentence in the dataset, w_j refers to the jth word in the sentence, $\mathbf{W}^I$ is the trainable word embedding matrix[3] with embedding size d. $N(w_j)$ is the global word frequency of w_j, and a is a small correction factor.[4] Additionally a matrix is formed from all the unique sentences in FNC-1, and the first singular vector u is removed from each sentence:

$$x_i{}^I = x_i - uu^T x_i. \tag{4}$$

Where $x_i{}^I$ refers to the input sentence representation of the unique sentence s_i in FNC-1 dataset.

3.3.2 Memory and Generalization Component

As depicted in Fig. 2, the MemN2N part of our model is composed of two memory components of keys K, and values V, each of which is fed by the feature representations of the body article as a sequence of SIF sentence embeddings from the input component, i.e., $x_i{}^I$ from Eq. (4).

Furthermore, each slot of each of the memory components, i.e., k_i and v_i, also gets appended with an additional, distinctive sentence embedding of the corresponding sentence from the article, following the sentence embedding strategy of the original implementation [21]. In other words, $k_i = [x_i{}^K; x_i{}^I]$ and $v_i = [x_i{}^V; x_i{}^I]$ which $x_i{}^K$ is computed through the following equation ($x_i{}^V$ is also computed similarly):

$$x_i{}^K = \sum_j l_j \cdot \mathbf{W}^K(w_j) + \mathbf{T}^K(i), \forall i. \tag{5}$$

Where $x_i{}^K$ is the the additional sentence embedding of the ith sentence of the article for keys memory component. w_j refers to the jth word in the sentence, $\mathbf{W}^K$ is the trainable word embedding matrix for keys (with embedding size d). l_j is the jth column of the positional encoding matrix for words, and $\mathbf{T}^K(i)$ is the ith row of the positional encoding matrix for sentences to make use of the order of words within a sentence, and sentences within a body article respectively. l is a fixed matrix, whereas $\mathbf{T}^K$ is trainable (see [21] for more details). The equations for the values memory component V are quite identical–the trainable embedding matrix is $\mathbf{W}^V$ (with the same embedding size d) and positional encoding matrix for sentences is $\mathbf{T}^V$.

In every memory slot, $x_i{}^K$ and $x_i{}^K$ indicate the distributed representation of the sentences locally, i.e., within the input article. $x_i{}^I$ denotes the globally distributed representation of the sentences across the whole FNC-1 dataset. In the training session, the SIF's word embedding $\mathbf{W}^I$ gets updated every iteration,[5] similar to all trainable parameters including $\mathbf{W}^K$ and $\mathbf{W}^V$; however, the singular vector u of Eq. (4), gets updated every epoch, once the word embedding is updated for the whole dataset.

Finally, the claim as a query is also embedded using the same sentence embedding mechanism of K:

$$q = [x_c{}^Q; x_c{}^I] \text{ , where } x_c{}^Q = \sum_j l_j \cdot \mathbf{W}^Q(w_j). \tag{6}$$

Where c is the claim as a query, and $x_c{}^Q$ is its sentence representation that we embed using the word embedding matrix of the keys (i.e., $\mathbf{W}^Q = \mathbf{W}^K$), and $x_c{}^I$ is the input's SIF sentence representation of the claim.

3.3.3 Output Component

The model produces the output vector based on an attention function over the claims as a *query*, and the *keys* and *values* memory components [47]. The output is a weighted sum of the values memory component V. The corresponding weights, however, are computed with local and global sentence similarity measures along with the IR system's relevance score at the paragraph level.

The local and global sentence similarity measures are computed between the claim as a query and the keys K

3. $\mathbf{W}^I(w_j)$ is the row of $\mathbf{W}^I$ which refers to the word embedding of w_j.
4. 10^3 in our case
5. both sentence embedding schemes use the same initial word embedding.

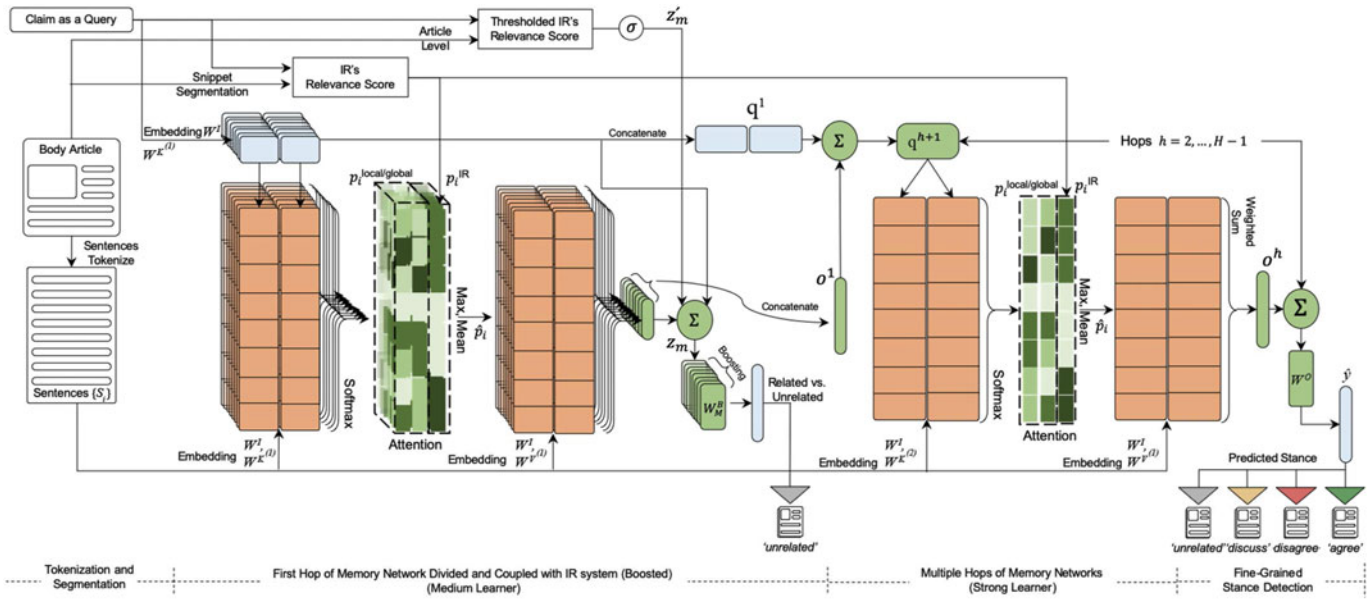

Fig. 2. The architecture of the second stage of our model is based on multi-hop end-to-end memory networks. The first hop is broken down to M small single-hop MemN2N with an embedding size of d/M. Each small MemN2N is coupled with a thresholded IR system followed by a nonlinear function as weak learners to a boosted classifier– a medium learner to detect "hard-to-distinguish" unrelated articles (hard to the IR system). Next hops of MemN2N perform more fine-grained stance classification.

using their corresponding local and global representations. This is mathematically formulated as follows:

$$p_i^{\text{local}} = \text{softmax}\left(\frac{{x_c}^Q \cdot {x_i}^K}{|{x_c}^Q||{x_i}^K|}\right), \ \forall i \tag{7}$$

$$p_i^{\text{global}} = \text{softmax}\left(\frac{{x_c}^I \cdot {x_i}^I}{|{x_c}^I||{x_i}^I|}\right), \forall i, \tag{8}$$

p_i^{local} and p_i^{global} are the local and global attention weights corresponding to the ith memory slot (i.e., ith sentence in the body article). Also $\text{Softmax}(\cdot_i) = e^{\cdot_i} / \sum_j (e^{\cdot_j})$.

Additionally, we use the IR system to compute another semantic similarity vector and detect unrelated pieces of evidence. To do so, every article is segmented into paragraphs, as text snippets, which can be a potential piece of evidence for the overall stance–a paragraph represents a coherent argument about one or more inter-related topics. The relevance score of the IR system is assigned to the corresponding sentences in the paragraph p_i^{IR}. The main purpose of this attention weight is to filter out the most unrelated evidence at the paragraph level.

With the maximum and average of the local and global attention weights masked with p_i^{IR}, a single attention weight is computed for every memory slot:

$$\hat{p}_i = \text{softmax}((\max(p_i^{\text{local}}, p_i^{\text{global}}) + \text{mean}(p_i^{\text{local}}, p_i^{\text{global}})) \cdot p_i^{\text{IR}}). \tag{9}$$

Next, using the resultant attention weights $\hat{p}_i$, the output o is generated as a weighted summation of the representations of the values v_i:

$$o = \text{Attention}(q, Q, V) = \sum_i \hat{p}_i \cdot v_i, o \in \mathbb{R}^{2d \times 1}, \tag{10}$$

where $\text{Attention}(q, K, V)$ refers to all the procedure of computing the attention weights $\hat{p}_i$ from query q and Keys K and multiplying to the corresponding values from V.

3.3.4 Response Component

Single-Hop Setting. Finally, the response mechanism converts the output vector to a vector of size 4 corresponding to the stance labels: *agree*, *disagree*, *discuss*, and *unrelated*.

$$\hat{y} = \text{softmax}(\mathbf{W}^O(q + o)), \tag{11}$$

$\mathbf{W}^O$ is a trainable matrix $\in \mathbb{R}^{4 \times 2d}$ which converts the summation of the claim and output vectors $q + o$ to the predicted stance label using a softmax operation.

Multi-Hop Setting. To predict the stance accurately, our MemN2N may need to read each article several times. Therefore, the above operations must be conducted within a multi-hop model and the following iterations:

$$q^{(h+1)} = q^h + o^h, \text{for } h = 1, 2, \ldots, H \tag{12}$$

$$\hat{y} = \text{softmax}(\mathbf{W}^O(q^H + o^H)). \tag{13}$$

Where h refers to the number of the hop. H is the total number of hops. The other operations and parameters stay the same, e.g., $\mathbf{W}^{K^{(1)}} = \mathbf{W}^{K^{(2)}} = \ldots = \mathbf{W}^{K^{(H)}}$ and $\mathbf{W}^{V^{(1)}} = \mathbf{W}^{V^{(2)}} = \ldots = \mathbf{W}^{V^{(H)}}$.

3.4 IR + MemN2N (Boosted)

For every claim as a query in MemN2N, we only consider a subset of all the articles for two reasons. i) It may not be computationally efficient to run a neural network model on the full dataset for the simple task of identifying unrelated pairs. Although it is a complex task to determine whether an article agrees, disagrees, or discusses a certain claim, understanding whether the article and the claim are related is far less

complicated [18]. ii) For every query there are far more negative samples, i.e., unrelated articles, than positive ones–more than 73 percent of claim-article pairs in the FNC-1 dataset are unrelated. We achieve down-sampling of the negative samples by considering only a subset of samples that passed the first stage of our model. Specifically, each claim is, first, fed to the IR system (weak learner) as a query and the articles in the corpus are ranked based on their relevance score. Next, we feed the articles that exceed a determined threshold to the main stance detection module, multi-hop memory network (strong learner), which performs more complex operations to detect the final stance label, vanilla version of our model (*IR + MemN2N*).

Detecting some of the unrelated articles may require a more sophisticated approach than an IR system. Yet, allocating the full processing resources is not reasonable. Therefore, we devise a medium learner to detect such "hard-to-distinguish" unrelated articles by integrating the IR system and the first hop of the memory network. In this regard, we use adaptive boosting algorithms to boost the performance by improving generalization and reducing bias to the distribution of the input data.

As for the weak learners of our boosting algorithm, we divide the first hop of the memory network into M similar operations (as those of a single hop) performed in parallel. Instead of a single hop with $2d$-dimensional embeddings (query and memory components), we linearly project the input M times with different, learned linear projections to $2d/M$ dimensions–the overall computational complexity is similar to that of a single hop with full dimensionality. On each of these projected versions of queries and memory components we next perform the similar operations of a single-hop memory network which yields M output vectors of size $2d/M$, and M final responses. The outputs are once again concatenated and passed to the next hops for more fine-grained stance classification. The M final responses are, however, used as the weak learners' response to our boosting algorithm.

We train M weak learners using the adaptive boosting algorithm from Beygelzimer et al. [48]. Every weak learner is trained with weighted samples from training dataset to detect unrelated from related samples (weak u/r classifiers). After training every weak learner, weights are updated based on the learner's overall accuracy and whether the sample is predicted correctly or not (see [48] for more details).

For every weak learner, we also use two base learners: i) the IR system's thresholded relevance score passed through a non-linear function; $z_m^{'} = \sigma(r - \mathbf{th}_m)$; and ii) one of the divisions of the memory network which takes $z_m^{'}$ as a bias factor; $z_m = \sigma(\mathbf{W}_m^B(q_m^1 + o_m^1) + z_m^{'})$. To train every weak learner, we initially train z_m (i.e., set $\mathbf{th}_m$) to maximize accuracy and the weights are updated accordingly, then the memory network division z_m is trained to classify the updated inputs.

The final "medium" learner is formed using a linear combination of the weak learners (boosted u/r classifier).

During the main training session, for every batch, we first train our medium learners using weighted samples and update their weights along with the weights of the first hop of the model. Then, we train the whole multi-hop MemN2N to detect finer-grained stances, using uniformly weighted samples. All trainable weights of the model are updated including those of the first-hop.

4 Analysis of Algorithm and Time Complexity

4.1 Algorithm Flow

First, we perform an initial analysis of the data and extract various bag-of-words representation features that will be used to compute the relevance score of the claim-article pairs. Next, based on the score, articles can either be labeled "unrelated" or be passed to the memory network stage.

Algorithm 1. Stage I: the IR System Either Labels Articles "Unrelated" or Passes Them to the Second Stage.

procedure STANCE-DETECTION D, Θ
  for $(c_i, \hat{D}_i)$ **in** D **do**
    for d **in** $\hat{D}_i$ **do**
      $r_d = \sum_i \sum_j r'_{w_j,w_i} \cdot \mathbf{wf}_{w_j,d}$ ▷ STAGE(I), Section 3.2
      if $r_d \leq \mathbf{th}_0$ **then**
        return *'unrelated'*
      else
        return STAGE(II)$((c_i, d), \Theta)$

Algorithm 1 summarizes the first stage of our stance detection model. It details the procedure through which the IR system computes the claim-article relevance score, labels the highly unrelated articles, and feeds the rest to the second stage.

Algorithm 2. Stage II: Multi-Hop MemN2N Model Predicts the Final Stance for Every Article Passed From the IR System

procedure STAGE(II)(c_i, d), Θ
  $q^1, K, V \leftarrow c_i, d$ ▷ sentence embedding, Section 3.3.2
  $\{q_m^1, K_m, V_m\}_{m=1,\ldots,M} \leftarrow q^1, K, V$ ▷ dividing to M sets. Section 3.4
  for m **in** number of divisions M **do**
    $o_m^1 = \text{ATTENTION}(q_m^1, K_m, V_m)$ ▷ Section 3.3.3
    $z_m = \sigma(\mathbf{W}_m^B(q_m^1 + o_m^1) + \sigma(r - \mathbf{th}_m))$
  end for
  $\hat{r} = \sum_{m=1}^{M} \alpha_m z_m$ ▷ boosted u/r classifier, Section 3.4
  if $\hat{r} \leq \hat{\mathbf{th}}$ **then**
    return *'unrelated'*
    continue
  else
    $o^1 = \text{concatenate}(\{o_1^1, \ldots, o_M^1\})$
  end if
  for $h = 2 : H$ **do** ▷ H is the number of hops. Section 3.3.4
    $q^h = q^{(h-1)} + o^{(h-1)}$
    $o^h = \text{ATTENTION}(q^h, K, V)$ ▷ Section 3.3.3
  end for
  $\hat{y} = \text{softmax}\ (\mathbf{W}^O(q^H + o^H))$
  return $\arg\max_i(\hat{y}_i)$
    ▷ $i \in$ {*'unrelated'*, *'agree'*, *'disagree'*, *'discuss'*}

Algorithm 2 describes the second stage. A multi-hop MemN2N that outputs the fine-grained stance for the claim-article pairs that pass the first stage. During the first hop, M weak learners are combined to detect the remaining unrelated articles. Then, the outputs of the weak learners are concatenated and are used to perform multiple hops of

TABLE 1
Time Complexity Analysis, Comparing Our Proposed Model With the Baselines

Model	Complexity of Individual Components	Overall Complexity
UCL	$1\times$ MLP: $O(n_w + \mathcal{V}_{\text{BoW}} \cdot d)$	$O(\mathcal{V}_{\text{BoW}} \cdot d)$
Athene	$5\times$ MLP: $O(2n_w + 5\mathcal{V}_{\text{BoW}} \cdot d + d^2 \cdot L_{mlp})$	$O(\mathcal{V}_{\text{BoW}} \cdot d + d^2 \cdot L_{mlp})$
SOLAT	CNN: $O(w_k \cdot n_w \cdot d^2 \cdot L_{cnn})$ + MLP: $O(d^2 L_{mlp})$ + GBDT: $O(M_T \cdot n_w \cdot L_{gbdt})$	$O(n_w \cdot d^2 \cdot (L_{cnn} + L_{mlp}))$
sMemN2N	CNN: $O(w_k \cdot n_w \cdot d^2 \cdot L_{cnn})$ + RNN: $O(4n_w \cdot d^2 \cdot L_{rnn})$ + MemN2N: $O(2n_p \cdot d \cdot H)$	$O(n_w \cdot d^2 \cdot (L_{cnn} + L_{rnn}))$
IR + MemN2N	IR: $O(n_w + \mathcal{V}_{\text{BoW}})$ + MemN2N$_1$: $O(2n_s \cdot (d/M) \cdot M)$ + MemN2N$_{2-H}$: $O(2n_s \cdot d \cdot (H-1))$	$O(n_s \cdot d \cdot H)^*$

*w_k is the window kernel size for convolutional neural networks. M_T is the number of trees for the gradient boosting decision tree. * is the worst case where the input is not detected as* unrelated *and has to go through all stages of our model.*

operations for the final stance prediction. Θ denotes all the trainable parameters and hyper-parameters of our model, e.g., $\mathbf{W}^{I,K,V,O,B}$, $\mathbf{T}^{K,V}$, $\mathbf{th}_m$ etc.

4.2 Time Complexity Analysis

In this section, we present a time complexity analysis for our stance detection model, comparing it to those of the baselines. Table 1 shows the asymptotic growth of the overall models as well as their individual components with respect to model dimensions d; vocabulary size for the various bag of words representations $\mathcal{V}_{\text{BoW}}$, number of words, sentences and paragraphs in each pair n_w, n_s, n_p; and number of layers/hops L/H. The overall time complexity of our model is relatively less than the baselines in terms of model dimensions.

Besides, RNN and/or CNN-based models grow by $O(d^2)$ and therefore, the number of words can cause them serious scale-up challenges. However, our proposed model grows by the number of sentences rather than words, making it more efficient to apply to lengthier articles and larger corpora (since the model requires more dimensions to encode more articles and topics).

The run time for our proposed model is defined as:

$$T(N_D) = N_D T^{\text{IR}} + \left(\frac{N_{\hat{D}}}{\rho_{u/r}} \cdot N_D\right) T^{\text{MemN2N}}. \tag{14}$$

Where $T(N_D)$ is the total run time for a corpus size of N_D, $N_{\hat{D}}$ refers to the average number of paired articles with every claim, and $\rho_{u/r}$ shows the proportion of unrelated to related articles. $T^{\text{IR}}, T^{\text{MemN2N}}$ are the run time for the IR and MemN2N stages.

The IR + MemN2N complexity shown in Table 1 is based on a worst-case scenario where the IR component is unable to detect the input as *unrelated* and therefore the data has to pass through all the stages of the model. However, the rise in the number of daily published news stories increases the size and the topics inside a stance detection corpus [49], [50]. Therefore, we expect the dataset to skew even further towards the unrelated articles as the dataset gets larger resulting in a greater $\rho_{u/r}$ ratio in Eq. (14). We believe our two-stage model is more efficient and practical to scale up for real-world stance detection tasks.

5 Experimental Evaluations

5.1 FNC-1 Dataset

To evaluate our system, we use the Fake News Challenge - stage 1 dataset (FNC-1) that is launched by a non-profit organization to explore and detect the relative perspective of two pieces of text (a claim and a body article) as a means to help fact-checkers. The challenge is derived from a digital journalism project called Emergent [9] which attempts to address the task of fake rumors debunking.

The dataset contains 1,648 distinct claims about 300 topics that are paired with relative news articles. Every claim-article pair has a stance label annotated by professional journalists that indicates the stance of the article towards the paired claim. The stance labels are from three classes of *agree*, *disagree* and *discuss* (see Table 3). Every claim is paired with 5-20 news articles. Furthermore, FNC-1 organizers generate an additional stance class of *unrelated* by randomly matching claims and articles belonging to different topics. Finally, the overall dataset includes 75,385 claim-article pairs along with their stance labels. Stance labels are biased towards the unrelated class, while agree, disagree, and discuss classes are far less represented. The descriptive statistics are shown in Table 2. To avoid data bleeding between the topics, claims, and articles of training and testing sets, claim-article pairs regarding 200 topics are set aside for training, while remaining pairs of 100 topics are reserved for testing [16]. To avoid outsourcing, FNC-1 organizers provide additional 266 claim-article pairs in the testing set. The length of claims varies from 10 to 220 words, while the lengths of the articles are between 25 to 5000 words.

5.2 Evaluation Metrics

We present multiple evaluation measures to understand and compare our model performance against the state-of-the-art

TABLE 2
The Distribution of Stance Classes in the FNC-1 Train and Test Sets

Dataset	Unrelated	Discuss	Agree	Disagree
Train	36,545	8,909	3,678	840
(N = 49,972)	(73%)	(18%)	(7%)	(2%)
Test	18,348	4,464	1,903	697
(N = 25,413)	(72%)	(18%)	(7%)	(3%)

The four classes are imbalanced with about 75 percent unrelated and 25 percent of the three related stances.

TABLE 3
Example of a Claim With its Paired Body Articles From the FNC-1 Dataset Along With Their Respective Stance Labels

Claim: Justin Bieber saves man from bear attack.	
Stance	Body Article
Agree	A man fishing in northern Russia was attacked by a bear. But the bear fled when the men's cellphone rang. The ringtone was Bieber's song "Baby."
Disagree	According to Google Translate, the original Russian version said the bear was scared away when Igor Vorozhbitsyn's phone began speaking out the current time. So, yes, the phone apparently scared off the bear mid-mauling. But no Bieber.
Discuss	A Russian fisherman says a Justin Bieber ringtone on his phone scared away a bear that was mauling him near his favorite fishing spot, according to video from Newsy.
Unrelated	An airline passenger headed to Dallas was removed from a plane at La Guardia Airport on Christmas Eve because he raged after workers wished him a Merry Christmas, The New York Post reports.

stance detection models. The official metric for the FNC-1 is a two-step weighted accuracy scoring system that is designed to value the correct detection of *agree*, *disagree* and *discuss* labels over unrelated label by first assigning a score of 0.25 for identifying related or unrelated claim-article pairs and another 0.75 for determining the stance of the related pairs. Then, by dividing the overall score by the number of tested samples, the score is normalized. However, because the dataset is imbalanced, such a weighting system tends to overestimate the accuracy of models that perform better in the overrepresented class, i.e., *discuss* [16].

To address this shortcoming, apart from the weighted accuracy score, we calculate the F_1 scores for each class and also consider the Macro-F_1 score which is the F_1 averaged across all the four classes.

5.3 Ablation Analysis

In this section, we elaborate on the performance of the different parts and combinations of our two-stage model on the training dataset. As for the IR system, we use both the TF-IDF and BM25 bag-of-words representation features and apply the ranking function with/without the related words of the fine-tuned GloVe. Table 4 a depicts the results of their performances–*n*-gram matching is also implemented as a baseline for relevance scoring. The analysis shows the superiority of BM25 over TF-IDF and suggests the use of word representation. The best performing IR system is coupled with the first hop of the memory network and shows significant improvement, specifically in the F_1 score.

As for the MemN2N, we implement various models with different hyper-parameters that are tuned using a grid search algorithm [51]. In this section we provide results from the two best performing versions; i) *v1*: includes 2 hops of memory networks with d=200 and memory size of 150; and *v2*: includes 3 hops of memory networks with d=200 and memory size of 200. We also compare the performance of the model once we feed in the entire dataset (i.e., including the *unrelated* samples), as well as a memory network model without pre-trained GloVe. We combine these memory network models with the best performing IR systems, as the vanilla two-stage model (IR + MemN2N), and the one coupled with the adaptive boosting algorithm of Section 3.4 (IR + MemN2N (boosted)). Table 4 b shows the ablation analysis for the MemN2N stage of our model. The utilization of GloVe and the IR system significantly improves performance, especially through the adaptive boosting algorithm.

6 Comparison With Prior Works

As mentioned in Section 2, the FNC-1 organizers proposed a baseline method which is evaluated on the training set using a 10-fold cross-validation evaluation scheme. However, due to repetitive headlines and articles presented in the dataset, data bleeding occurs in such evaluation. Therefore, the organizers also made available a testing dataset which addresses the data bleeding issue. We utilize the same testing set for evaluation and comparison purposes.

To compare our stance detection models with the state-of-the-art ones, we use the open source code of the winners of the FNC-1, [13], [16] and [52] in addition to the baseline. We further implement a two-stage version of the state-of-the-art models using our best performing IR system. All the baselines are also fine-tuned using the training set, similar to our models.

TABLE 4
(a) Shows the Ablation Analysis for Our Information Retrieval System

Model	Accuracy	F1-Score	Model	Agree	Disagree	Discuss	Unrelated
n-gram Matching	78.9%	78.2%	MemN2N (w/o GloVe)	43.6%	3.3%	77.2%	94.7%
IR (BM25)	82.2%	84.1%	MemN2N (v1)	37.2%	5.1%	80.2%	94.1%
IR TF-IDF	81.9%	82.1%	MemN2N (v2)	32.5%	7.3%	**88.8%**	95.2%
IR (TF-IDF + GloVe)	**92.9%**	**91.5%**	IR + MemN2N (v1)	**58.4%**	9.9%	74.7%	98.5%
IR (BM25 + GloVe)	**98.9%**	**95.9%**	IR + MemN2N (v2)	54.3%	**22.1%**	81.2%	**99.0%**
IR (BM25 + GloVe) + MemN2N (H1)	**99.9%**	**99.6%**	IR + MemN2N (boosted)	**57.1%**	**21.6%**	85.9%	**99.9%**
(a)						(b)	

IR + MemN2N (H1) and BM25 + GloVe systems perform better than the competitors. (b) shows the ablation analysis for the sophisticated stage of the model.

TABLE 5
The Comparison of One and Two-Stage Implementation of Our System Against the Competition in Terms of Performance and Training Cost

Model	Stance Score	F1-Score					Training Costs (Peta FLOPS)
		Macro	Agree	Disagree	Discuss	Unrelated	
All Unrelated	39.4	21.0	-	-	-	83.7	-
All Discuss	43.9	7.6	-	-	30.5	-	-
FNC Baseline	79.5	26.7	19.1	1.1	70.0	97.0	-
SOLAT in the SWEN	**82.1**	58.0	**52.9**	3.1	**77.0**	**98.9**	2,765.8
Athene (UKP Lab)	82.0	**60.0**	48.7	**15.1**	**78.0**	98.1	2,073.6
UCL Machine Reading	81.8	56.8	46.9	8.3	74.7	97.4	1,036.8
IR + SOLAT in the SWEN	**82.1**	59.1	**53.3**	5.4	76.8	**98.8**	794.9
IR + Athene (UKP Lab)	**82.1**	**60.9**	50.4	**17.2**	76.8	98.6	656.6
IR + UCL Machine Reading	81.7	56.2	46.8	5.8	75.3	97.1	**276.5**
MemN2N	81.4	53.3	31.2	6.8	**78.2**	95.1	709.8
IR + MemN2N	81.9	**60.3**	49.6	**17.6**	75.1	**98.9**	**181.1**
IR + MemN2N (Boosted)	**82.1**	**61.1**	**50.3**	**19.9**	**77.1**	**99.1**	**245.6**

We use the evaluation metrics mentioned in Section 5.2, namely stance score (weighted accuracy), Macro-F_1 score, and F_1 score for each stance label. Since the distribution of the FNC-1 data is highly imbalanced, inspired by [18], we also compare the performance of our model with two classifiers which assign a default stance label to all the test data points; we do this for the two common classes of *unrelated* and *discuss*.

To evaluate the efficiency of our stance detection model during the procedure, we also present the training cost of each model. Following [47], we have formulated the training cost as the number of floating points used during the training procedure estimated as the training time multiplied by the number and the single-point floating capacity of the utilized GPUs. Table 5 summarizes the performance of our system on the FNC-1 dataset and compares its performance and training cost to those of the baselines. The MemN2N performs better than the baseline without using any hand-coded features, and at a significantly smaller training cost than other baselines. It performs very well on detecting *discuss* articles and achieves the highest F_1 score which we believe is due to capturing valuable text snippets and not truncating the articles.

Integrating the memory network with the IR system both as an initial filtering stage and a linear attention layer over the memory component significantly improves the overall performance by a 0.5 stance score. In detecting the unrelated headline-article pairs, it performs very well, given the fact that it uses an unsupervised algorithm (the IR algorithm) to detect a great portion of the unrelated samples. It also decreases the training costs by 65.3 to 74.5 percent.

This is while integrating other state-of-the-art models with the IR system–albeit improving their training cost–only slightly improves their performances, maximum improvement is a 0.1 stance score for the model of [16]. In some cases, the performance may even decrease such as the F_1 score of [52] in unrelated and discuss classes, even stance score of [13] decreases by 0.1. The reason is that the other models may not be compatible with the first stage of the IR system (weak learner). As mentioned in Section 2 These models are not sufficiently flexible to learn various sub-tasks involved in the stance detection problem. This issue is further amplified when the model is remained with "hard-to-distinguish" classes of unrelated-related and fine-grained sub-categories of agree, disagree, and discuss. They either focus more on the classification of the articles that are very similar and lose their ability in detecting other stance classes or vice versa. While as for memory networks, utilizing a well-compartmentalized memory component combined with a flexible inference mechanism facilitates the stage-wise reasoning for the various sub-tasks of stance detection. As such the overall performance of IR + MemN2N is significantly higher than that of the stand-alone MemN2N; however, as shown in Table 5, the F_1 score for the class of *discuss* drops by 3.1.

The training cost for all two-stage models is improved in comparison to their single-stage versions, which we believe is due to more efficient coverage of related articles. On average, models achieve their optimum Macro-F_1 score on 61 percent fewer number of iterations. However, our models (even the boosted version) is still superior in terms of training costs which shows that it is not only faster at inference time (as discussed in Section 4, by the ratio $rho_{u/r}$), but also is more efficient during the training time.

Finally, the last row of the table shows the performance of our two-stage model coupled with an adaptive boosting algorithm, IR + MemN2N (Boosted). As expected, it outperforms IR + MemN2N due to better segregation of stance detection sub-tasks and a more advanced integration mechanism. It also doesn't require much more training cost than the vanilla version since it divides the encoding dimensions by the exact boosting number M (see Section 3.4).

Although our boosted model is featureless and significantly less complex, its performance is on par with the two-stage implementation of the top-performing solutions, while it has a significantly lower training cost. Our proposed model outperforms all the other models across all classes except [52] and [16] which perform better in detecting agree and discuss labels, respectively; however, our model achieves higher performance in terms of Macro-F_1 score.

7 Implications on Snopes Dataset

To examine the potential of our proposed model in real-life scenarios, we collect data from Snopes.com which is among the most popular fact-checking websites in the United States. For each claim presented on Snopes.com, the origin of the story

TABLE 6
Examples of Claims and Ratings on Snopes.com Along With Their Source Articles, Arguments, and Detected Stances

Claim: The skeletal remains of Joyce Carol Vincent were found in her home, with the television on, years after her death. **Rating:** *True*	**Confirming Report:** The TV and heating were still on when housing officers discovered the body of Joyce Vincent, 40, in her living room. **Stance:** *Agree*
	Source Article: The skeleton of a woman was discovered in her flat nearly three years after she is believed to have died, it emerged today. **Stance:** *Agree*
Claim: President Trump instructed his acting secretary of state to nullify oaths taken on the Quran. **Rating:** *False*	**Debunking Report:** The CNN anchor Jake Tapper responded: "You don't actually have to swear on a Christian Bible. You can swear on anything, really. I don't know if you knew that." - **Stance:** *Disagree*
	Source Article: The Constantinople Clause says that Christianity must be observed for oaths 'lest they have no meaning.' This isn't about church and state. This is about assigning Christian morals and ideals where appropriate in our Christian society. If you can't take an oath on a bible, you can't serve this country. Period." - **Stance:** *Agree*

along with the fact-checking team's decision (credibility rating) and the supporting or refuting arguments are provided.

7.1 Snopes Dataset

In this section, we focus only on the claims that were rated *true* or *false*. We collect the original articles from which the claims had propagated, and the argument snippets provided by the Snopes team. We pair each claim with every original article and argument snippet. Our final dataset consists of 7,478 claim-article pairs on which we run our two-stage model. 1,622 pairs are removed by the IR system since their relevance scores were zero and the remaining 5,856 are fed into the memory network model. Table 6 shows examples of these claim-article pairs.

The existence of true/false credibility labels enables us to examine the predicted stance output for true and false claims separately. We use the predicted probability of each claim-article pair belonging to stance classes of *agree*, *disagree* and *discuss*–instead of one-hot encoded version–and compare the mean of these probabilities across the true and false groups. The mean comparisons between the two groups are shown in Table 8.

As shown in Table 8, at $\alpha = 0.05$ level of significance, we have enough evidence to reject the null hypotheses and state that the mean probability score of predicting $agree$ stance for the true group is significantly higher than the false group while the mean score for $disagree$ and $discuss$ labels are significantly lower. The comparison between the mean score of the two groups is shown in Fig. 3. It is important to note that the articles examined in our dataset are the argument snippets provided by journalists to prove or reject a certain claim and therefore, it is natural to expect more supporting articles for a true claim and more disagreeing articles for a false one. These findings show that our model is capable of distinguishing between the true and false labels based on the detected stance of claim-article pairs.

TABLE 7
The Snopes Dataset Consists of $N_D = 4,720$ True/False Claims Verified by Expert Journalists, and $N_{\hat{D}} = 5674$ Articles Corresponding to These Claims

	Claims		Source Articles	
Dataset	True	False	True	False
Size	848	3872	1037	4637
(%)	(18%)	(82%)	(18%)	(82%)

8 Conclusion and Future Work

In this research, we propose a novel stance detection model to address the problem put forth by the FNC-1 organizers. Inspired by the real-life process of stance detection, our two-stage model combines a simple IR system with a sophisticated MemN2N. Examining the FNC-1 dataset and reviewing the prior works reveals that the instances of the $unrelated$ class are overrepresented and most models perform well in detecting them [18]. Considering the difficulty of identifying the three stance classes, we design an IR system that can easily filter out the majority of unrelated claim-article pairs (more than 70 percent) with minimal loss of related instances. Afterward, the more complex task of identifying the remaining unrelated pairs along with detecting the stance of the related ones is carried out by a MemN2N model that operates without any handcrafted features. We evaluate the performance of our model in terms of weighted accuracy and Macro-F_1 scores and the results suggest that our two-stage model performs on par with the top-performing models of the FNC-1.

TABLE 8
At the $\alpha = 0.05$ Level of Significance we Have Enough Evidence to Reject the Null Hypotheses and State That the Mean Probability Score of Predicting $agree$ Stance for the True Group is Significantly Higher Than the False Group While the Mean Score for $disagree$ and $discuss$ Labels are Significantly Lower

	Rating	N	Mean	Std. Deviation	Std. Error Mean	Mean Difference
Agree	False	433	0.2954	0.07367	0.00354	-0.46479 ***
	True	93	0.7602	0.05184	0.00538	
Disagree	False	235	0.5779	0.03921	0.00256	0.23561 ***
	True	26	0.3423	0.04311	0.00845	
Discuss	False	1107	0.5102	0.03067	0.00092	0.03435 ***
	True	336	0.4758	0.03989	0.00218	

***$p < 0.001$

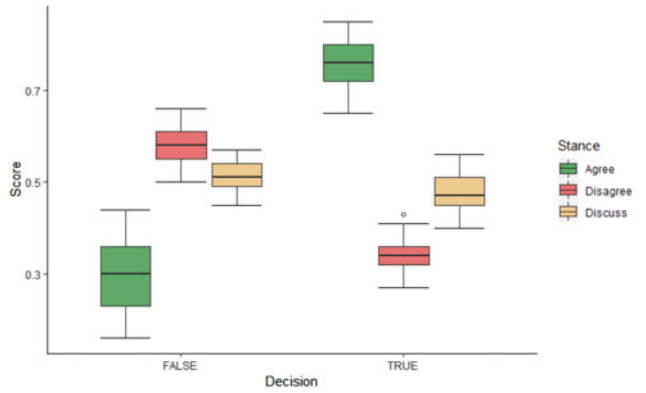

Fig. 3. The average predicted probability scores for the stance labels of *agree*, *disagree*, and *discuss* between the true and false articles.

Our study has a number of limitations. First, the proportion of unrelated instances is a characteristic of the dataset and while article-claim relevance is crucial in a stance detection task, the usefulness of our IR system depends on the number of unrelated articles. Second, although we combine multiple ranking functions and word representation techniques, we cannot achieve the precision of 100 percent in our IR system which means losing a small number of related pairs (approximately 4 in every 1000).

Future studies should consider metrics beyond accuracy to compare stance detection models. In particular, it would be helpful to compare more models based on their sensitivity, robustness, and data handling capabilities in addition to efficiency and run-time. Also, to investigate the issue of overfitting, it would be beneficial to consider other datasets and evaluate the generalizability and scalability of the models by training them on some topics and testing them on other ones within the same dataset or training and testing models on the same topic across different datasets.

Acknowledgments

The authors would like to acknowledge the use of the services of Chameleon cloud and Jetstream cloud, funded by National Science Foundation (NSF) awards 1419165 and 1445604 respectively.

References

[1] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *J. Econ. Perspectives*, vol. 31, no. 2, pp. 211–36, 2017.
[2] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.
[3] J. Clarke, H. Chen, D. Du, and Y. J. Hu, "Fake news, investor attention, and market reaction," *Inf. Syst. Res.*, to be published, doi: 10.1287/isre.2019.0910.
[4] T. Mihaylov and P. Nakov, "Hunting for troll comments in news community forums," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 399–405.
[5] S. Tavernise, "As fake news spreads lies, more readers shrug at the truth," *New York Times*, Dec. 7, 2016.
[6] C. Silverman, "This analysis shows how fake election news stories outperformed real news on facebook," BuzzFeed News, Nov. 16, 2016. Accessed: Jan. 10, 2021. [Online]. Available: https://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook
[7] J. Thorne and A. Vlachos, "Automated fact checking: Task formulations, methods and future directions," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 3346–3359.
[8] M. Stencel, "Global fact-checking up 50% in past year," *Duke Reporters' Lab*, Feb. 16, 2016. Accessed: Jan. 10, 2021. [Online]. Available: https://reporterslab.org/global-fact-checking-up-50-percent/
[9] M. Stencel and R. Griffin, "Fact-checking triples over four years," 2018. [Online]. Available: https://reporterslab. org/fact-checking-triples-over-four-years
[10] K. Leetaru, "Why can't facebook do better at fact checking photos and videos?," Sep. 2018. [Online]. Available: https://www.forbes.com/sites/kalevleetaru/2018/09/21/why-cant-facebook-do-better-at-fact-checking-photos-and-videos/
[11] C. Shao, G. L. Ciampaglia, A. Flammini, and F. Menczer, "Hoaxy: A platform for tracking online misinformation," in *Proc. 25th Int. Conf. Companion World Wide Web*, 2016, pp. 745–750.
[12] L. Graves, "FACTSHEET: Understanding the promise and limits of automated fact-checking," *Reuters Inst. Study of Journalism, Univ. Oxford, Oxford*, 2018.
[13] B. Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel, "A simple but tough-to-beat baseline for the fake news challenge stance detection task," May 2018. Accessed: Jan. 10, 2021, *arXiv: 1707.03264*[Cs].
[14] M. Reilley, "Journalist's toolbox | a society of professional journalists blog," Mar. 2019. [Online]. Available: https://www.journaliststoolbox.org/2019/03/02/urban_legendsfact-checking/
[15] B. Fortis, "The fake news challenge puts AI to the test," May 2017. [Online]. Available: http://mediashift.org/2017/05/fake-news-challenge-puts-ai-test/
[16] A. Hanselowski *et al.* "A retrospective analysis of the fake news challenge stance-detection task," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 1859–1874.
[17] J. Huang, W. X. Zhao, H. Dou, J.-R. Wen, and E. Y. Chang, "Improving sequential recommendation with knowledge-enhanced memory networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 505–514.
[18] M. Mohtarami, R. Baly, J. Glass, P. Nakov, L. Màrquez, and A. Moschitti, "Automatic stance detection using end-to-end memory networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, 2018, pp. 767–776.
[19] D. Mrowca, E. Wang, and A. Kosson, "Stance detection for fake news identification," Stanf. Univ. Calif. US Rep., 2017.
[20] J. Weston, S. Chopra, and A. Bordes, "Memory networks," in *Proc. Int. Conf. Learn. Representations*, 2015.
[21] S. Sukhbaatar, J. Weston, R. Fergus, and A. Szlam, "End-to-end memory networks," in *Proc. Advances Neural Inf. Process. Syst.*, 2015, pp. 2440–2448.
[22] C. Li, X. Guo, and Q. Mei, "Deep memory networks for attitude identification," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, 2017, pp. 671–680.
[23] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," Sep. 2016. Accessed: Jan. 10, 2021, *arXiv:160106733*.
[24] B. Pan, H. Li, Z. Zhao, B. Cao, D. Cai, and X. He, "Memen: Multi-layer embedding with memory networks for machine comprehension," Jul. 2017. Accessed: Jan. 10, 2021, *arXiv:170709098*.
[25] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2016, pp. 1400–1409.
[26] A. Kumar *et al.* "Ask me anything: Dynamic memory networks for natural language processing," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1378–1387.
[27] B. Dickson, "Deep learning won't detect fake news, but it will give fact-checkers a boost," Feb. 2020. [Online]. Available: https://bdtechtalks.com/2020/02/24/deep-learning-fake-news-stance-detection/
[28] J. Du, R. Xu, Y. He, and L. Gui, "Stance classification with target-specific neural attention networks," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 3988–3994.
[29] M. A. Walker, P. Anand, R. Abbott, and R. Grant, "Stance classification using dialogic properties of persuasion," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, 2012, pp. 592–596.
[30] K. S. Hasan and V. Ng, "Stance classification of ideological debates: Data, models, features, and constraints," in *Proc. 6th Int. Joint Conf. Natural Lang. Process.*, 2013, pp. 1348–1356.
[31] D. Sridhar, L. Getoor, and M. Walker, "Collective stance classification of posts in online debate forums," in *Proc. Joint Workshop Social Dyn. Pers. Attributes Soc. Media*, 2014, pp. 109–117.
[32] A. Rajadesingan and H. Liu, "Identifying users with opposing opinions in twitter debates," in *Proc. Int. Conf. Soc. Comput. Behavioral-Cultural Model. Prediction*, 2014, pp. 153–160.

[33] A. K. Chaudhry, D. Baker, and P. Thun-Hohenstein, "Stance detection for the fake news challenge: Identifying textual relationships with deep neural nets," Stanf. Univ. Calif. US Rep., 2017. [Online] Available: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2760230.pdf
[34] B. Galbraith, H. Igbal, V. Veen, D. Rao, J. Throne, and Y. Pan, "Baseline FNC implementation," 2017. [Online]. Available: https://github.com/FakeNewsChallenge/fnc-1-baseline
[35] B. Sean, S. Doug, and P. Yuxi, "Fake news challenge - team solat in the swen," 2017. [Online]. Available: https://github.com/Cisco-Talos/fnc-1/
[36] A. Hanselowski, A. PVS, B. Schiller, and F. Caspelherr, "Description of the system developed by team athene in the FNC-1," 2017. [Online]. Available: https://github.com/ hanselowski/athene_system/blob/master/system_description_athene.pdf
[37] M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," in *Proc. Advances Neural Inf. Process. Syst.*, 2013, pp. 190–198.
[38] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1532–1543.
[39] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva, "Stance detection with bidirectional conditional encoding," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2016, pp. 876–885.
[40] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading wikipedia to answer open-domain questions," 2017, *arXiv:1704.00051*. [Online]. Available: http://arxiv.org/abs/1704.00051
[41] F. Hill, A. Bordes, S. Chopra, and J. Weston, "The goldilocks principle: Reading children's books with explicit memory representations," in *Proc. Int. Conf. Learn. Representations*, 2016.
[42] D. Tang, B. Qin, and T. Liu, "Aspect level sentiment classification with deep memory network," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 214–224.
[43] B. Fortis, *The Fake News Challenge Puts AI to the Test*, Accessed: Jun. 12, 2020, 2017. [Online]. Available: http://mediashift.org/2017/05/fake-news-challenge-puts-ai-test/
[44] N. Ebadi and P. Najafirad, "A self-supervised approach for semantic indexing in the context of COVID-19 pandemic," *arXiv*, pp. arXiv–2010, 2020.
[45] N. Dingwall and C. Potts, "Mittens: An extension of glove for learning domain-specialized representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Human Lang. Technol.*, 2018, vol. 2, pp. 212–217
[46] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *Proc. 5th Int. Conf. Learn. Representations*, 2017.
[47] A. Vaswani *et al.* "Attention is all you need," in *Proc. Advances Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[48] A. Beygelzimer, S. Kale, and H. Luo, "Optimal and adaptive algorithms for online boosting," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2323–2331.
[49] F. Hamborg, N. Meuschke, and B. Gipp, "Bias-aware news analysis using matrix-based news aggregation," *Int. J. Digital Libraries*, vol. 21, pp. 129–147, 2018.
[50] R. Meyer, *How Many Stories Do Newspapers Publish Per Day?*, 2016. [Online]. Available: https://www.theatlantic.com/technology/archive/2016/05/how-many-stories-do-newspapers-publish-per-day/483845/
[51] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.
[52] B. Sean, S. Doug, and P. Yux, "Talos targets disinformation with fake news challenge victory," 2017. [Online]. Available: https://blog.talosintelligence.com/2017/06/talos-fake-news-challenge.html

Nima Ebadi received the BSc degree from the Electrical Engineering Department, Sharif University of Technology, in 2014. Since September 2016, he has been with the Electrical Engineering Department, University of Texas at San Antonio (UTSA) as a doctor of Philosophy (PhD) student. He is currently a research fellow with the Secure AI and Autonomy Lab, UTSA. His research interests include machine learning, deep learning, natural language processing, and conversation as a platform.

Mohsen Jozani received the master's degree in technology entrepreneurship from the University of Tehran, and the PhD degree in information technology from the University of Texas at San Antonio. He is currently an assistant professor of information systems and decision sciences, Louisiana State University in Shreveport. As a multidisciplinary explorer, his research interests include machine learning, user reviews, e-commerce, and m-commerce. His doctoral research investigates the economic impact of recommendation systems on mobile app markets.

Paul Rad received the 1st BS degree in computer engineering from the Sharif University of Technology, in 1994, the 1st master's degree in artificial intelligence from the Tehran Polytechnic, the 2nd master's degree in computer science from the University of Texas at San Antonio (Magna Cum Laude), in 1999, and the PhD degree in electrical and computer engineering from the University of Texas at San Antonio, and the PhD degree in electrical and computer engineering on cyber analytics from the University of Texas at San Antonio, San Antonio, TX. He is a co-founder and associate director of the open cloud institute, and an associate professor with the information systems and cyber security from the University of Texas at San Antonio. He was a recipient of the most outstanding graduate student with the College of Engineering, 2016, Achieving Rackspace Innovation Mentor Program Award" for establishing Rackspace patent community board structure and mentoring employees, 2012, Achieving Dell Corporation Company Excellence (ACE) Award in Austin for exceptional performance and innovative product research and development contributions, 2007, and Dell Inventor Milestone Award, Top three Dell Inventor of the year, 2005. He holds 15 U.S. patents on cyber infrastructure, cloud computing, and big data analytics with more than 300 product citations by top fortune 500 leading technology companies such as Amazon, Microsoft, IBM, Cisco, Amazon Technologies, HP, and VMware. He has advised more than 200 companies on cloud computing and data analytics with more than 50 keynote presentations. He serves on the advisory board for several startups, high performance cloud group chair at the cloud advisory council, OpenStack foundation member, the number 1 open source cloud software, San Antonio Tech Bloc founding member, Children's Hospital of San Antonio Foundation board member.

Kim-Kwang Raymond Choo (Senior Member, IEEE) received the PhD degree in information security from the Queensland University of Technology, Australia, in 2006. He currently holds the Cloud Technology Endowed professorship with the University of Texas at San Antonio (UTSA). He was included in Web of Science's Highly cited researcher with the field of Cross-Field, 2020, and in 2015 he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen-Nuremberg. He is the recipient of the 2019 IEEE Technical Committee on Scalable Computing Award for Excellence in Scalable Computing (Middle Career Researcher), 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, British Computer Society's 2019 Wilkes Award Runner-up, 2019 EURASIP JWCN Best Paper Award, Korea Information Processing Society's JIPS Survey Paper Award (Gold) 2019, IEEE Blockchain 2019 Outstanding Paper Award, Inscrypt 2019 Best Student Paper Award, IEEE TrustCom 2018 Best Paper Award, ESORICS 2015 Best Research Paper Award, 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, Fulbright Scholarship in 2009, 2008 Australia Day Achievement Medallion, and British Computer Society's Wilkes Award in 2008. He is also an IEEE Computer Society's Distinguished Visitor (Jan 2021–Dec 2023).