INTERNATIONAL JOURNAL OF ADVANCED STUDIES
IN COMPUTER SCIENCE AND ENGINEERING
IJASCSE VOLUME 9 ISSUE 3, 2020

03/31/2020

INTEGRATING STANCE DETECTION AND FACTUALITY CHECKING

Fatima T. Al-Khawaldeh[1], Tommy Yuan[2] & Dimitar Kazakov[3]

[1]Department of Computer Science, University of York, York, United Kingdom
ftma500@york.ac.uk

[2]Department of Computer Science, University of York, York, United Kingdom
tommy.yuan@york.ac.uk

[3]Department of Computer Science, University of York, York, United Kingdom
dimitar.kazakov@york.ac.uk

Abstract *- Fake news is currently seen as a possible risk with a harmful effect on democracy, journalism, and economies, that comes mainly from social media and online websites. To detect fake news, we propose two models trying to check the factuality of a claim against relevant pieces of evidence. In this paper, the stance of each relevant evidence toward a certain claim is detected, then the result of factuality checking will be decided based on the entire aggregation of all available stances in addition to some salient syntactic and semantic features.*

In this paper, we propose two models help distinguish fake news from reliable content. The first model is multi-channel LSTM-CNN with attention, where numeric features are merged with syntactic and semantic features as input. Concerning the second model, word-level and clause-level attention networks are implemented to capture the importance degrees of words in each clause and all clauses for each sentence in evidence. Other crucial features will be used in this model to guide the model in stance detection processes such as tree kernel and semantic similarities metrics. In our work, for stance detection evaluation, the PERSPECRUM data set is used for stance detection, while DLEF corpus is used for factuality checking task evaluation. Our empirical results show that merging stance detection with factuality checking helps maximize the utility of verifying the veracity of an argument. The assessment demonstrates that the accuracy improves when more focus is given on each segment (clause) rather than each sentence, so using the proposed word-level and clause-level attention networks demonstrate more effectiveness against multi-channel LSTM-CNN.

Keywords *- Stance Detection; Factuality Checking; Deep Learning, Tree Kernel, Semantic Similarity.*

I. INTRODUCTION

Nowadays, there is an enormous and exponential growth of incorrect claims (fake news) on the internet and other sources. Fake news is nonfactual information comes from misleading news articles. Most of this fake news is used to direct peoples' opinions by misleading them or sometimes it has a different functional role, e.g., observing peoples' backgrounds and reactions about social or political events in society or the world. Today, it needs just one click to post any news, either fake or real, which could not be discovered manually, so the necessity to develop automatic fact-checking tools increases. People have the right to distinguish the factual information from fake to judge the source reliability and credibility.

Two main tasks should be considered for verification, stance detection and faculty checking to discover the fakeness of a published statement. Stance detection and fact-checking are two main useful tasks that could be used to address the problem of fake news and filter claims according to their degree of truthiness. Since claims are based on a journalist or publisher analysis of one or more events and their causes, there is a considerable amount of uncertainty expressions in the statements, so our research will focus on claim uncertainty detection. Stance detection is the process of finding to which extent a claim is agreed, disagreed, unrelated or discussed against an evidential document. Fact-checking is the method to distinguish whether the veracity of a claim is supported or refuted. In this research, we will concentrate on two main tasks: fact-checking which comprises classifying the truth of the claim and stance detection, which includes defining the viewpoint of an article against a claim.

In the past, factuality checking was manually done which needs excessive efforts to extract evidence that support or attack a claim, or some of them are mainly built-in rule-based approaches by leveraging the exceptional features where a lot of rules should be set [1,2,3,4]. One of the main limitations, it is noticed that most of the previous systems focused on one of the components of fake news detection components, such as stance detection or factuality checking. In spite of the stance, detection is an essential phase of factuality. There is a scarcity of unified datasets that evaluate the combination of these tasks together. In this paper, our contribution is to develop a model that processes both stance and factuality missions together.

In recent years, with the emerging of social media networks, blogs, Facebook, Twitter and others, neural network-based approaches, particularly deep learning models, have become the most popular for factuality checking. Deep learning models have shown its proficiency in solving fake news problem, e.g., Recurrent Neural Networks (RNNs) for representing sequential posts and user engagements for twitter rumours [5, 6, 7, 8], or Convolutional Neural Networks (CNNs) for capturing local features of texts and images [9], or combination of RNN and CNN [10]. Generative Adversarial Networks (GANs) [11] for capturing deceptive writing style features. Stack of CNN and bidirectional LSTM (Bi-LSTM) based models are applied in [12].

Our main contributions are as follows:

- Develop a novel deep neural network model able to infer the degree of stance between different a claim against an evidence article and conduct experiments and study the consequences.
- Develop a novel deep neural network model that can infer the correct factuality label of a claim and conduct experiments and study the consequences.
- We empirically demonstrate that our proposed unified method significantly outperforms the state-of-the-art baselines models.

The remainder of this paper is organized as follows. In section 2, we present false information types. In section 3, and section 4 we continue to summarize the methods and the evaluation in stance detection and fact-checking tasks respectively. In section 5, we discuss the datasets and evaluation metrics used by existing methods. We introduce our proposed models to detect fake news in sections 6 and 7. Finally, we discuss the results and analysis in section 8 and conclude this paper in section 9.

II. FALSE INFORMATION TYPES

Fake news is defined as published false news that misleads users intentionally [13]. Incorrect information is classified into two primary sorts [14] based on intent: misinformation and disinformation or based on knowledge: opinion-based and fact-based. Misinformation is unconscious sharing of false information while disinformation is the intent to deceive readers by sharing incorrect information which is more harmful than misinformation. According to this taxonomy, fake news is fabricated information based on purpose, as shown in figure (1).

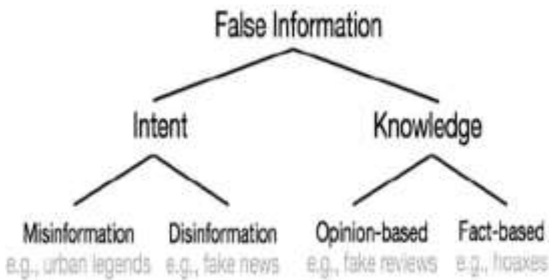

Figure (1): Classification of false information [14]

There are some categories of fake news, according to researchers in [15],[16] and [17]:

- Propaganda: This content is used for political purposes by deceiving people and affecting their opinions intentionally, e.g. "The parliament candidate has announced an amazing financial plan to raise the level of the economy".
- News Fabrication: Publishing news as evidential while they are without factual evidence

e.g. "The ministry of education will raise the salaries for its employees by the end of this year".

- Imposter Content: The published content is originated from a different source, but the publisher posts it as his own, e.g. "The degree of temperature will be higher than yesterday", in this example, the environment agency is the source of this information but not mentioned.
- Satire: Showing a content related to recent news for entertainment purposes but may intentionally deceive readers, e.g. "90 % of products prices will be increased, so we advise you to buy your needs from the moon, it is cheaper."
- Parody: Presenting the belief using a humbling mode, e.g. "Jordanian government will finish the poverty by the end of the year 2021" (this will be by killing the poor people)".
- Manipulated Content: Altering the contextual information by modifying the original news text, e.g. "Two men were suicide yesterday" (original), "Two men killed by throwing them from the 6th floor" (manipulated).
- Advertising and Public Relations: Illegal marketing for commercial purposes, e.g. "The quality of our product is the best in the city".
- False Context of Connection: Irrelevant contextual information of the content with a statement like a title, e.g. title is "Syrian refugees in Jordan", the comment "Trump will visit Syria next week".
- Conspiracy Theories: An interpretation of the event to make an annoyance, especially toward government, e.g. "The Egyptian Government killed Mohammad Mursi".

There is some research work that apply machine and deep learning methods for fake news detection that we briefly review some of them.

III. STANCE DETECTION EVALUATION

The overall architecture for fake news detection has two types of inputs: claims and evidence come from the labelled dataset [18] for training and testing the model. In our project, there are two outputs for stance detection and factuality checking tasks. The feature extraction component has a role in detecting the most important and relevant features from DNN models, and other linguistics features methods. The extracted features are fed to stance label predictions component, the last decision for factuality label will be based on aggregate information from stance label prediction component and other techniques like recognizing textual entailment.

In this section, we will show the result of previously discussed models and talk about some of our experiments on uncertainty using deep learning. The most public factuality corpora will be presented in this section. For each corpus, statistical information, how the annotation of this corpus is done with an example will be shown and the best performance on this corpus.

Stance detection measures the degree of similarity between a claim and evidence. For stance detection task, DNN based models are used for training purposes. One of the best approaches on FNC dataset is Riedel et al. [29], TF and TF-IDF Features are passed to the hidden layer, and the SoftMax layer will output the result. In the second rank on best results on FNC, is Baird et al. [30], in which decision tree model is combined with deep CNN model. Decision tree model extracts sentiment features and other features. Contextual information is obtained by word embedding to the CNN, where final SoftMax layer is responsible for giving the classification detecting results. The final prediction comes from merging the two models. Memory network is applied by Montazami et al. [31].

Stance detection is a component of fact-checking which aggregates the stances to take the final decision of factuality. There are a few models that have been trained on FNC dataset. In this section, we will show the first three ranks of them.

In the **first rank**, the best results on FNC is by Baird et al. [30], 82.02 F-score is achieved, in this system, the decision tree model is combined with deep CNN model. The decision tree model extracts count features, sentiment features and other features. Contextual information is obtained by word embedding to the CNN, where final SoftMax layer is responsible for giving the classification detecting results. The final prediction comes from merging the two models. **The second rank** is UCL Machine Reading by Hanselowski [32] with 81.97 F-score, where latent Dirichlet allocation and latent semantic indexing are the primary training used features. One of the best approaches on FNC dataset and get the **third rank** is by Riedel et al. [29] with 81.72 F-score. TF and TF-IDF Features are passed to the hidden layer, and the SoftMax layer will output the result, they trained their system based on only the most frequent terms using cosine similarity.

Memory network is applied by Mohtarami et al. [31] and obtained good results on twitter dataset, and they reached to 61.67 macro F1 scores. In his models, the target and source are encoded by different DNN models, then the combination of similarity matrix with memory network has the responsibility to assign the correct prediction. These systems used large hand-engineered features like TFIDF, Singular Value Decomposition (SVD), Word2Vec, and sentiment features, the second extract features like unigrams, latent Dirichlet allocation, latent semantic indexing and topic models.

Most of the stance detection models have been trained on FNC dataset, which is imbalanced, so we suggest combining with other similar tasks dataset like entailment and inferences datasets like SNLI dataset [33].

IV. EXISTING WORK OF FACT-CHECKING

There are several research concerns by fact-checking detection like depending on expressive features as TF-IDF features [34] and other features. Karadzhov et al. [35], depending on ground truth as a credible source like google engine and applied LSTMs set on retrieved results to enrich SVMs and multilayer perceptron's to detect the factuality. Evidence are extracted by searching from trusted websites to verify news based on claim queries method. A dataset consists of 992 sets of tweets are used for experiments. Despite satisfactory results by following question answering which needs generating queries and selects the best snippet than the best sentences, it is extremely complicated and requires too much processing to get the final factuality label. The system has the benefit of not depending on highly engineered features. In this automatic system, the information they gain comes from the web, the accuracy may be affected according to how much the retrieved snippet is relevant and to which degree the sources are trusted.

Recurrent/Recursive Neural Networks (RNNs) are used to represent sequential posts and user engagements ([5], [6], [7], [8]), in these research tweet propagation data. Convolutional Neural Networks (CNNs) to capture local features of texts and images [9] are applied to focus on uni-gram word features, or both [10]. This study takes pictures into account to check veracity. Combining images processing with texts needs sperate tasks and mixed datasets. Another risk if the image is right and the text is correct but not related. Generative Adversarial Networks (GANs) have also been used and extended to obtain a “general feature set" for fake news across events to achieve false news early detection [11].

In [5], fake news propagation is represented by RNN based on a bottom-up and top-down tree-structured neural networks focusing on user properties and profiles information. The risky in this system is do which extent the user profile is real and not fake, or there are different reasons behind create this profile and the problem of opposing opinions or comment as “fake deny” or “fake support”. In [6], they developed CSI model specification with three components: Capture module based on LSTM to get textual information of the pattern of temporal engagement to an article, while the Score module extract source characteristic for all users, combination is done between article representation which comes from the first module and user information representation that comes from the second module, they are combined in integrate module to classify the fakes news.

RNN model is used to extract the relationship between creators of news and subject [7]. LSTM is used to extract the representation of temporal textual characteristics (time-series event) of rumour to help to classify the rumour on twitter early. Even this system can learn without training heavy manual features hidden representations incapable of detecting dynamic structures for a long time [8]. Multiple convolutional layers are implemented to merge the input representation from both images and texts. This model obtains competitive results, but it needs a vast data to train, an early detection system to catch the false claims directly after posting is done in [10] by training merged model of CNN and RNN.

We noticed that DNN models are mainly used with good results. Still, some of these systems have computational limitation like in [35], where retrieving evidence to compare with sometimes taking a long time, especially filtering process. We have noticed that some of these systems need a lot and continuity observing of changing in a sequence of posts in additions to the user and they only trained for only supervised data ([5], [6], [7], [8]). In [11], textual and visual features are extracted to train the models there is a risk that images features have more transferability than texts and the training and experiment have been done on imbalance twitter dataset. Despite good results in [6] but couldn’t be reliable since there is a lack of ground truth information about users where there is a possibility to publish fake data about them and has the problem to predict unobserved users due to depending on user features training. Training on small dataset makes it difficult for CNN to train and detect the meaningful patterns in texts, and CNN doesn’t deal with high dependencies sequences.

V. FAKE NEWS DATASETS

In this section, we will present related works for fake news checking, the most important datasets, models and results. There are few numbers of published Fact-Checking datasets, e.g. Factbank, UDS-IH2, FEVER, symmetric FEVER, FNC, PERSPECTRUM and DLEF. We will show the best models have trained on them.

A. FACTBANK DATASET [19]

FactBank has 3864 sentences and 13506 event factuality values. Example: “Omar Razzaz, the Prime Minister of Jordan, **doubts** that the tax rate will decrease this year”. The predicated are in boldface, while their embedded events are underlined.

EF-AC-GAN is the best to several state-of-the-art models trained and tested on Factbank corpus; they applied the GAN model and obtained Macro-A accuracy of 54.06 [20]. Saur´ı and Pustejovsky [19] focus on the author and embedded sources. A rule-based method to identify factuality of events on FactBank is applied in [21] and obtained 43.97% Macro-Accuracy. Qian et al. combine machine learning and rule-based approaches gained Macro-Accuracy 43.56% [22]. Annotation information in FactBank corpus includes factuality degrees fact, counterfactual, probable, not probable, possible, not certain, certain but unknown output, and unknown or uncommitted. GAN models proved promising results for assessing the integrity of factual claim for news texts, as the generator is trained to capture more syntactic information for uncertain statements. UDS-IH2 Dataset [23]: THE MOST SIGNIFICANT EVENT FACTUALITY DATASET OF UW, FACTBANK, AND MEANTIME COMBINATION.

DATASET	TRAIN	DEV	TEST	TOTAL
FACTBANK	6636	2462	663	9761
MEANTIME	967	210	218	1395
UW	9422	3358	864	13644
UDS-IH2	22108	2642	2539	27289

Table (1): Statistics of the UDS-IH2 Dataset [23]

Example: “Some girl **ate** no dessert”.
Event is bold. The first training and testing on this dataset were by Rudinger et al., and they obtained excellent performance. Stacked bidirectional linear LSTM and Stacked bidirectional tree LSTM are implemented and give a competitive performance [23]. In this system, BiLSTM models have shown its strength for event factuality prediction, which determines whether an event has occurred or not. This system obtained a high Pearson correlation (r) score of 0.857 by taking the capability of a neural network to extract the relevant features without depending on hedges classification.

B. FEVER Dataset

FEVER (Fact Extraction and Verification) dataset of 125,000 claims labelled by supported or refuted or not enough evidence. Two steps applied to get this dataset, claim generation by altering sentenced extracted from Wikipedia pages, then claim to the label. FEVER is used for factuality checking which have three labels distributed as 55% (supported) that means fact, 21% (refuted) that means false truth and 24% (Not Enough Information (NEI) [24]. Table (2) shows some examples of fact-checking labels used in FEVER corpus.
This dataset is large which help to capture more characteristics features when training models but needs more computational effort to retrieve the combination information from Wikipedia which have a massive number of articles also it doesn’t consider textual information or metaknowledge of users. Additionally, the evidence is not available and must be retrieved from Wikipedia also they only come from sentences selected from Wikipedia where there are other evidential sources could be considered. FEVER relies on recruiting the services of several people annotations, which is fewer quality ideals compared to the precise annotations by fact-checking organizations. For factuality checking, we will evaluate it based on FEVEF dataset. We will examine the most distinguished applied methods have been trained on FEVER dataset.

Claim	Fact
Eric Trump is the second son of US President Donald Trump	True (supported)
Eric Trump is unrelated to the current President of the United States	False (refuted)

Table (2): Examples of fact-checking labels used in FEVER corpus

C. Symmetric FEVER dataset

To avoid the idiosyncrasies observed in the claims of FEVER dataset. The authors in [28] make the original claim-evidence pairs of FEVER evaluation dataset symmetric, by augmenting the dataset and making each claim and evidence appear with each label. Therefore, by balancing the artefacts, relying on cues from claim to classify samples is equivalent to a random guess.

D. FNC Datasets

In this research we will use **FNC** datasets [45] which have four labels for stance detection: 1:7%(disagree), 73% (unrelated), 18% (discuss) and 7:3%(agree). Table (3) shows some examples of stance detection labels in FNC corpus. This dataset is imbalanced based on label frequency and a limited label of agree and disagree, which are the most important labels. Most labelled data are for unrelated, which will not be useful to extract the common features and make it easy for DNN models to train.

Claim	**Dylan Thomas Finds Tropical Spider Burrowed Under Skin**
Label	Evidence document number
Disagree	1932, 1057…
Discuss	800, 1602 …
Unrelated	2175, 181…
Agree	1104, 1478…

Table (3): Examples of stance detection labels in FNC corpus

The authors in [25] asked annotators to assign claim faculty label to 221 where evidence is mixed of different type of documents like tables, pdf and excel. Due to the variant format and structure of document which needs more specialized techniques and training methods and the small size of this deserts, it is hard to depend for developing systems [25].

In [26], the authors published a dataset of 74K news articles from trusted websites and untrusted websites like Hoax, Satire and Propaganda. They used linguistic features to detect the faculty of claim if it comes from fake news websites or not. They consider the web site to decide the truth, and they did not take evidence into account [26]. For our stance detection evaluation, we will use FNC.

E. PERSPECTRUM Dataset

Perspective is a neutral point of view as a third-party partner to obtain different representations that emphasis both vital content information and its sentiment fairly and accurately. In [27,55] the authors show that better decision towards a claim could be made by creating different perspectives (viewpoints) and helps to better understand controversial issues. For example, claim A has a supported relationship with perspective A while claim B has a refuted relationship with perspective B, perspectives are generated based on claim text. In other words, rewording claims has supported or undermined relation with perspectives that has supporting evidence.

Claim A: “A government must lessen the economic gap between its rich and poor citizens”.

Perspective A:” The Rich Poor Gap Silences the Political Voice of the Poor”.

Evidence A:” Research has also demonstrated a connection between economic inequality and political voice. The political process is far more responsive to the claims of the privileged, and the privileged are for better organized and engaged in the political process than are less affluent citizens. Recent studies show that government officials are far more likely to support the policy preferences of the wealthy than those of the poor. In short, there is considerable evidence to suggest that there is a growing divide between those who have wealth and political influence and those who do not. Yasmin Dawood, THE NEW INEQUALITY: CONSTITUTIONAL DEMOCRACY AND THE PROBLEM OF WEALTH, Maryland Law Review: 2007”.

Claim B: “Internet access is a human right”.

Perspective B:” It is a big problem; too many people are file-sharing”.

Evidence B:” The plan to slow down or stop internet connections is the most economical and practical way to deal with file-sharers. Many illegal downloaders are young people, and this plan will prevent the offenders from receiving a criminal record”. Table (4) shows the statistical information of Perspectrum dataset.

Split	*Supporting Pairs*	*Opposing Pairs*	*Total Pairs*
Train	3603	3404	7007
Dev	1051	1045	2096
Test	1471	1302	2773
Total	**6125**	**5751**	**11876**

Table (4): A summary of PERSPECTRUM statistics [27]

The authors in [27], build a dataset that helps for training and testing system for the task of substantiated perspectives discovery given a claim and has a stance regarding it, and supported by evidence texts and annotated as in the following example in figure 2. In this figure, a claim is related to multiple perspectives with taking either support or oppose stance with respect to a claim. Each perspective should have supported evidence to prove it. The authors in [27], confirmed that analyzing diverse perspectives with respect to a claim improve the ability to understand of debatable claims.

A. DLEF DATASET

Both sentence and document-level event factuality are considered to detect the text factuality, considering factuality values CT-, PS+, PS-, Uu, CT+, negated event, speculated event, both negated and speculated event, underspecified event and factual event respectively.

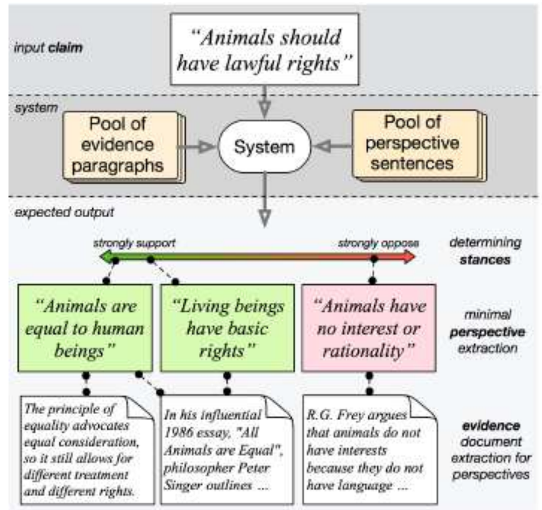

Figure (2): An example of a claim with its perspectives and evidence from PERSPECTRUM Dataset [27]

Table 5 shows the statistics of the DLEF corpus showing CT+ is the majority since most published news is real.

In the following sections, we will present our suggested models, Multi-channel LSTM-CNN with attention, Word Level and Clause Level Attention Network with Syntactic and Semantic Similarity, respectively. Each model achieves stance detection, then the final decision of factuality is detected.

Corpus	Statistics		
English	Documents	CT-	279/16.16%
		PS+	274/15.87%
		PS-	12/0.69%
		Uu	12/0.69%
		CT+	1150/66.59%
		Total	1727
	Sentence-Level Events	CT-	662/11.52%
		PS+	574/9.99%
		PS-	37/6.44%
		Uu	71/1.24%
		CT+	4401/76.61%
		Total	5745
	Avg. Len. of Sentences		14.73
	Avg. Len. of Documents		467.25
Chinese	Documents	CT-	1342/28.87%
		PS+	848/18.24%
		PS-	36/0.77%
		Uu	20/0.43%
		CT+	2403/51.69%
		Total	4649
	Sentence-Level Events	CT-	3923/20.69%
		PS+	2879/15.18%
		PS-	123/0.65%
		Uu	555/2.93%
		CT+	11482/60.55%
		Total	18962
	Avg. Len. of Sentences		29.00
	Avg. Len. of Documents		716.38

Table (5): Statistics of DLEF corpus

VI. THE FIRST PROPOSED MODEL: MULTI-INPUT CHANNEL Bi-LSTM-CNN WITH ATTENTION

In this model, our main contribution is to combine the stance detection with factuality checking to discriminate fake from real news. Firstly, the stance for each perspective concerning a claim is decided by applying our proposed Multi-Input Channel LSTM-CNN with Attention model. All relevant perspectives, either support or oppose the claim, with their supporting evidence, are considered. Then, all stances are aggregated to compute the average of them to decide the final decision for the factuality of claims.

Our proposed model inspired by MVCNN for sentence classification in [56] where the input of this model is multiple word embeddings comes with two main advantages, to get rich information from variant embedding versions, rear words which are not represented by some source embedding could be represented by other to initialize word vectors that can be influenced during training.

Our model architecture consists of four inputs, three texts with three channels each, and one numeric feature: CNN and Bi-LSTM with attention mechanism, concatenation layer and a SoftMax layer. We will describe our model, as illustrated in figure (3).

A. Auxiliary Input: Numerical Feature

In our work, for scoring the stance between a claim and a perspective providing their supportive pieces of evidence, there are many syntactically, and semantically measurements help decide the degree of the stance between a claim and perspectives. the semantic annotation of a text can be extracted by different natural language processing libraries such as pycorenlp[1], spacy[2] and textacy[3].

In our model we use spacy to parse a text then apply textacy to extract the fact about a specific noun phrase(subject) in the claim and the perspective. The extracted facts of both perspective and claim are used for detecting the semantic similarity.

The semantic similarity: similarity measurement will be performed based on Distance Vector space model, distribution by applying Euclidian [39], Cosine similarity [39] and K-L divergence-based [40] respectively. Other metrics, like Manhattan, will be used [41].

The word order similarity: In [46], the researchers show the influence of self-attention to extract word order information, finding differences between recurrent and self-attention models by position decoder. This model can detect both the original and inserted positions by measuring the probability distribution of the sequence is as inserted (labelled as "I") is:

$$P_1 = SoftMax(U_1 \tanh(W_1 H)) \quad \in \mathbb{R}^N \quad (1)$$

[1] https://pypi.org/project/pycorenlp/

[2] https://spacy.io/

[3] https://pypi.org/project/textacy/

The probability distribution of the sequence being the position the word is popped out (labelled as "O"):

$$E = P_I(W_Q H) \quad \in \mathbb{R}^d \quad (2)$$

$$P_O = A_{TT}(E, W_K H) \quad \in \mathbb{R}^N \quad (3)$$

In our model, we will follow the method of the word order similarity between claim and perspective as in [47]. We detect the word order similarity between two sentences S1, S2 by forming the vectors of them V1, V2 and indexing values of words in S1 beginning with 1 then word order similarity is measured by:

$$\mathrm{W_s} = \frac{\|\mathrm{V_1} - \mathrm{V_2}\|}{\|\mathrm{V_1} \times \mathrm{V_2}\|} \quad (4)$$

Finally, the weighted average for all these metrics is computed as auxiliary information to decide the final stance, and then factuality checking is done.

sentiment detection: For sentiment detection there are numerous f research try to enhance the performance classification e.g., Word-level sentiment analysis with reinforcement learning is proposed in [53]. In our model, the sentiment is checked between the claim and the data using TextBlob library which is faster and gives good results. We suggest that if the sentiment value for both claim and perspective are same then the result that will be supplied in input layer is 1 else 0.

B. *Multi-Input Channel Layer*

The inputs of our model are (claim, perspective and evidence) followed by the multi-channel embedding layer, which has four independent embedding sets:

- Word embeddings that use Glove/Word2Vec can capture the syntactic and semantic information of the words (**W**).
- Position embeddings which aim to detect the location of a pair of entities accurately and to get a sense of the ordering of the input (**P**).
- Part of speech POS tags embeddings: capture more syntactical features by detecting grammatical properties of the word based on linguistic rules or other information (**POS**).
- Sememe embedding it is the smaller unit of the word meaning; this incorporating enhances the performance of semantic representations for words as proved in [36](**S**).

C. *CNN layer:*

This layer receives the embedding vectors to extract the local features by variant filter size producing multiple features then merge them as a single vector.

D. *Bi-LSTM layer is used to get context information.*

Although CNN can capture the high-level features and local dependencies of short sequences, but it doesn't capture the global dependency (long-distance dependency) due to the limitation of CNN filter length. To solve CNN limitation on long sequences, we will use Bi-LSTM to consider long-distance dependency. The LSTM captures the contextual information features and combines the forward and backward representations by using element-wise sum in Bi-LSTM layer which discovers richer semantic information:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (5)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (6)$$

$$g_t = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (7)$$

$$c_t = i_t \otimes g_t + f_t \otimes c_{t-1} \quad (8)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (9)$$

$$h_t = o_t \otimes tanh\,(c_t) \quad (10)$$

$$h_i = \vec{h}_i \oplus \overleftarrow{h}_i \quad (11)$$

$$H = [h_1, h_2 \dots h_t] \quad (12)$$

$$H_m = tanh(H) \quad (13)$$

$$\alpha = softmax(v \cdot {H_m}^T) \quad (14)$$

$$f = tanh(\alpha \cdot H) \quad (15)$$

Where $\otimes$ denotes element-wise multiplication, and $\oplus$ denotes concatenate operation.

E. *Attention layer*

This layer gives more focus on the most relevant words instead of the whole sentence.

Attention-Based Bidirectional Long Short-term Memory Networks (Att-BLSTM) is implemented by applying these equations [58]:

$$M = \tanh(H) \quad (16)$$

$$\alpha = softmax(w^T M) \quad (17)$$

$$r = H\alpha^T \quad (18)$$

F. Concatenation Layer

The output of all attention vectors is fed to the concatenation layer to form a single output vector which then fed to the SoftMax layer.

G. Fully connected layer and SoftMax layer

The output of the concatenation layer is supplied to the SoftMax function to compute the classification probabilities finding the last relationship from the outcome. Following these equations:

s = tanh(r)

$$o_k = W_l \cdot s + b_l \quad (19)$$

$$p_\theta = \frac{exp(o_k)}{\sum_{t=1}^{K} exp(o_t)} \quad (20)$$

Factuality Checking

To decide the veracity of a claim, all stances output is summed then the average of these stances is calculated.

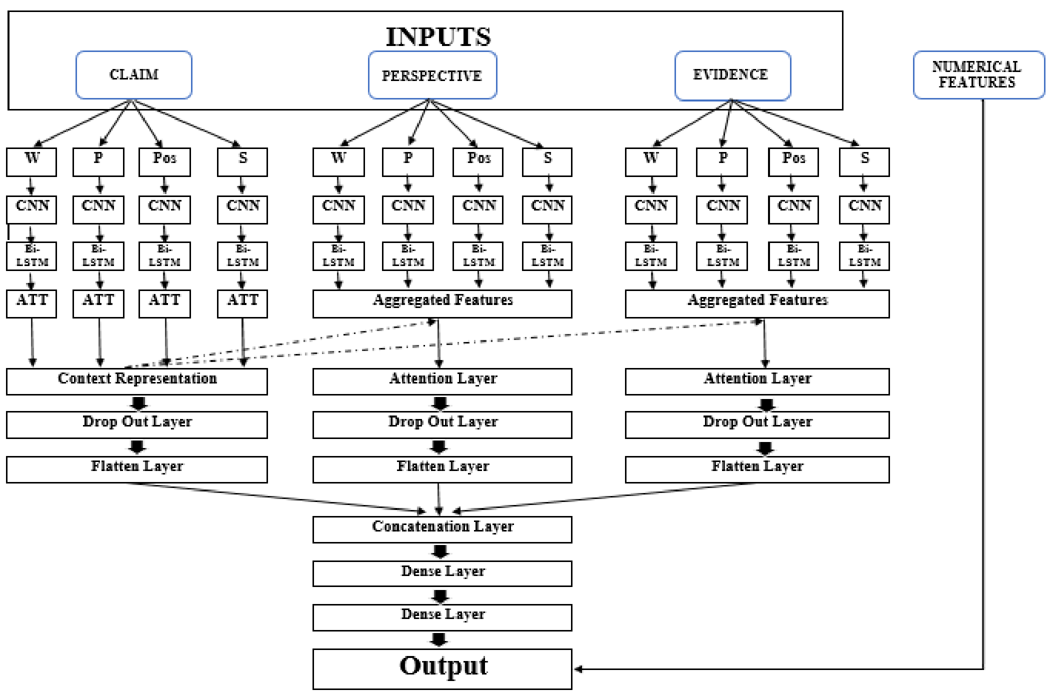

Figure (3): **Multi-channel LSTM-CNN with attention Architecture**

VII. SECOND PROPOSED MODEL: WORD LEVEL AND CLAUSE LEVEL ATTENTION NETWORK WITH SYNTACTIC AND SEMANTIC SIMILARITY

In this model, for stance detection, linguistic analysis is executed initially for each claim against perspective with its supporting evidence, then all stances are averaged to get the final factuality degree of a particular claim. All final results averaged stances for all perspectives are compared; the highest stance value indicates the final factuality decision. For example, suppose the view that has a contradiction concerning the claim has the highest score with the claim, then the factuality of the claim is false.

In order to check the factuality of a claim, we should check all perspectives that either have agree or disagree stance against a claim. Each perspective has its corresponding supporting evidence, both claim-perspectives which support or undermine evidence are inputs for our system. Since the input text may have one or more fact, we propose to segment the text into clauses for better analyzing. Word-level and Clause-level Attention Networks will be implemented to incorporate the knowledge of each clause in sentence.

In our model, fast RST for discourse parsing will be adopted to segment evidence sentence into several clauses since it is fast and robust [48]. Incorporating the representation of a claim by extracting the self-attentions to get the importance of different perspective clauses for a specific clause in each perspective, and then predicting the stance of a particular perspective clause toward a target a claim. Word-level Claim Attention and Clause-level claim Attention, illustrated in this part of paper, are captured to distinguish the rich information by applying stack of layers. Inspired of the most similar model to our work which is suggested for sentiment classification in [50], we develop our model which differs from their model since we pay attention to the claim text while they focus on aspect for the text. Another difference is that we don't merge claim representation for word embedding as they do for aspect which is appended to the word embeddings. The first time we use the claim representation information is in attention layer. Another suggestion to get better representation for clause is appending the constituency and dependency parsed information to the embedding of each word. We propose an attention-based bidirectional LSTM model with CNNs where rich feature representations for each sentence could be obtained based on contextual information in addition to the salient information. In addition to the sentence input, the constituency and dependency trees are fed to the input of the proposed model, and then follow these steps:

A. *Word-level Claim Attention and Clause-level claim Attention*

a. *Word Encoding Layer*

For each perspective, text representation using word embedding will be used for capturing the context of a word in addition to its constituency and dependency representation. Both representations are concatenated, then fed to CNNs layer to extract the local features and computing the maximum feature value per filter by MaxPooling layer. The context representation is the input for LSTM layer to obtain context representation. Bi-LSTM will be used to encode the information in each perspective clause from forward and backward direction

$$\overrightarrow{h}_{ij} = \overrightarrow{LSTM}_{(\widehat{w}_{ij})};\quad i \in [1, C], j \in [1, N_i] \quad (21)$$

$$\overleftarrow{h}_{ij} = \overleftarrow{LSTM}_{(\widehat{w}_{ij})};\quad i \in [1, C], j \in [N_i, 1] \quad (22)$$

$$h_{ij} = \overrightarrow{h}_{ij} \oplus \overleftarrow{h}_{ij} \quad (23)$$

b. *Word Attention Layer*

Attention mechanism will be implemented to concentrate on those words in the claim clause regarding a specific perspective CP and combine the representation of all of them to form a clause vector of perspective.

$$u_{ij} = tanh\left(W_w \cdot \left[h_{ij}; CP\right] + b_w\right) \quad (24)$$

$$a_{ij} = softmax\left(u_{ij}\right) = \frac{exp(u_{ij})}{\sum_{t=1}^{N} exp(u_{it})} \quad (25)$$

$$c_i = \sum_{i=1}^{N_i} a_{ij} \cdot h_{ij} \quad (26)$$

c. *Clause Encoding Layer*

BI-LSTM obtains the contextual information of each clause

$$\overrightarrow{h}_i = \overrightarrow{LSTM}_{(c_i)};\quad i \in [1, C] \quad (27)$$

$$\overleftarrow{h}_i = \overleftarrow{LSTM}_{(c_i)};\quad i \in [C, 1] \quad (28)$$

$$h_i = \overrightarrow{h}_i \oplus \overleftarrow{h}_i \quad (29)$$

d. Clause Attention Layer

The attention weight between each clause and the representation of a specific claim clause will be computed as follows:

$$m_i = tanh(W_c \cdot [h_i; CP] + b_c) \quad (30)$$

$$a_i = softmax(m_i) = \frac{exp(m_i)}{\sum_{t=1}^{C} exp(m_t)} \quad (31)$$

The sentence representation based on the attention vectors will be computed:

$$s = \sum_{i=1}^{C} a_i \cdot h_i \quad (32)$$

e. SoftMax Layer

A SoftMax classifier will be used to perform stance classification.,

$$o = W_l \cdot s + b_l \quad (33)$$

$$p_\theta = \frac{exp(o_k)}{\sum_{t=1}^{K} exp(o_t)} \quad (34)$$

B. TREE KERNEL (TK) FOR ARGUMENT STRUCTURE

Tree Kernel detects the syntactically structured representations for each claim clause and perspective clauses. In this model, we suppose the most similar structure is closer to form the same type of argument, so possibly controversial statements (claims) and evidence (opinion, results of studies etc..) have their own structure.

We use the Partial Tree kernel, that proposed in [51] to estimate similarities between two sentence tree structures. The Partial Tree Kernel detects a richer feature space by extracting shared child subsets of the two nodes, considering the order of children, considering two trees T1, T2 with n1, n2 nodes of them respectively, so the shared subtrees cases are:

1. If the nodes in the trees are different then $\Delta(n_1, n_2) = 0$
2. If the nodes are the same and one or both is a leaf, then $\Delta(n_1, n_2) = \lambda\mu^2$
3. Otherwise

$$\Delta(n_1, n_2) = \lambda \left(\mu^2 + \sum_{I_1, I_2, \ell(I_1)=\ell(I_2)} \mu^{d(I_1)+d(I_2)} \prod_{k=1}^{\ell(I_1)} \Delta(c_{i_{1k}}, c_{i_{2k}})\right)$$
(35)

$I_1 = i_{11}, i_{12}, i_{13}, \ldots, \ell(I_1)$ and $I_2 = i_{21}, i_{22}, i_{23}, \ldots, \ell(I_2)$ are sequences of indices of the child nodes of n_1 and n_2 be computed as the dot product between two such representations, of different trees. Tree kernels count the numbers of shared subtrees between trees T1 and T2.

Detect conflicting statement (CS)which have negation words like never which have antonym, lot, little, numeric mismatch [37] and focusing on dissimilar information [38].

$$\textbf{sentence stance output} = \boldsymbol{avg}(\boldsymbol{clauses}\ \textbf{stances outputs}) + \textbf{avg}(\textbf{Tree Kernel outputs})$$

$$\textbf{evidence stance output} = \boldsymbol{avg}(\textbf{sentences stance output})$$

We will consider Factuality Values in [42] where the authors employed modality and polarity to describe the specified degree of events and express polarity such as certain (CT), probable (PR), and possible (PS), while polarities are positive (+) and negative (-) [54]. For factuality checking task, hierarchical deep learning models will be applied where model-A input is stance aggregation, conflict views and sentiment values, the output of model-A combined with the encoded in the formation of semantic roles for sentences pair, which are the input for model-B which produces the factuality result — the dataset published in [43] used in our model.

We aggregate all stances to get factuality in the final stage. Suppose that we have a claim and two perspectives with opposite stances supported evidence related to. Each evidence is split to clauses and compare each clause with clause representation of claim. All probabilities are aggregated and averaged to decide the final stance of each sentence

VIII. ANALYSIS AND RESULTS

We use the Perspectrum Dataset [27]to evaluate the effectiveness of our stance detection task. Each claim has its corresponding perspectives from various debate websites. Our approach is compared to the work was published as a baseline in [27] and the work at EMNLP [44]. The authors in [27] proved that stance detection is an essential phase for information fact-checking. They applied a supervised method for stance detection based on a language representation model, BERT (Bidirectional Encoder Representations from Transformers) to examine the perspectives of controversial claims which gives a clear understanding. BERT is used to obtain the representation of a perspective concerning a claim by merging them on one input separated by a token, then passed to SoftMax layer to detect the agreement between perspective and a claim. In [44], the authors have enhanced this model by augmenting it with a novel consistency constraint for stance detection.

A recent work [27] proposes a supervised method for stance detection based on a language representation model called BERT (Bidirectional Encoder Representations from Transformers) [52]. In their paper, they enhance the stance detection model by augmenting it with a novel consistency constraint to where the consistency measures the similarity of the latent representations for claim and perspective.

We evaluate the effectiveness of our stance detection model against the STANCY model. The F1 score of BERT baseline in [27] and consistency aware BERT in [44],77.63, 79.95 F1 respectively, where the last outperforms all the other baselines.

To investigate the reasons why combine text and numerical features in the first model enhance the performance of stance classification, we train our model with and without the numeric features. The model shows a higher performance about 2.5 points in F1-score by incorporating the numerical features which give similarity computation for claim against perspective. We suggest combine numerical similarity score after computing the text stance classification then decide the final score for stance detection. We notice that due to the high dimensionality of text (vector space features), no impact could be obtained when combining the numerical features in the same level of the model. For example, suppose the text has a dimension of 500 features, adding one numeric feature will not affect, so the we find the score based on numeric features separately then add them to the score of the model. the result of the two scores is averaged for the final score. In this model, we show that using multiple parallel (channels) deep neural networks achieves a performance improvement of about 3 points in F1-score. Using different types and versions of word embeddings ensure better representations for words and consider more syntactical and semantical features and capture different aspects of linguistic properties for detecting the semantic relation of text pair. multichannel initialize the model by diverse types and versions of pretrained word embeddings capture more features. for each channel, multiple input feature maps are processed by the same type of layer but with different operation as in CNN layer, multiple filters of variant sizes. Following different channels for data processing help detect different kinds of features

For the second model, two level of semantic computations considered. The syntactically structured representations of text pair are computed by tree kernel and the semantic similarity score is computed by the clause level model. tree kernel is used to compare tree structures of tow texts as parse trees) which is benefit for complex structures of sentences as a measure of syntactic similarity. Another useful enhancement in this model is depending on clause attention where each is more able to select informative words and clauses corresponding to claim. More improvement performance is obtained by incorporating the constituency and dependency information to identify a hierarchical structure with clauses and connects words according to their relationships.

For the factuality checking task, the most appropriate dataset for evaluating our work, DLEF corpus [18] is used for training and testing. The final output of each evidence toward a claim is one of five label CT+, CT-, PS+, PS-, Uu. The authors implemented LSTM neural network with both intra-inter sequence attention and consider adversarial training to get the factuality of an evidence article against a sentence. The Experimental results showed that applying this model on DLEF corpus is useful, producing 76.28 accuracy.

In our experiments, our proposed model, Word Level and Clause Level Attention Network with Syntactic and Semantic Similarity outperforms the Multi-Channel Bi-LSTM-CNN With Attention model. Both proposed model for stance detection, 81.29,80.56, respectively. The obtained f-score for factuality checking are 83.06, 82.97.

Table 6 and table 7 compares our proposed models with other state-of-the-art methods of stance detection and factuality checking, respectively.

Model	F-score
Word Level and Clause Level Attention Network with Syntactic and Semantic Similarity	81.29
Multi-Channel Bi-LSTM-CNN With Attention	80.56
Consistency aware BERT [44]	79.95
BERT baseline [27]	77.63

Table 6: Comparison with previous results for stance detection

Model	F-score
Word Level and Clause Level Attention Network with Syntactic and Semantic Similarity	81.29
Multi-Channel Bi-LSTM-CNN With Attention	80.56
Intra-inter sequence attention [18]	76.28

Table 7: Comparison with previous results for factuality checking

IX. CONCLUSION

We have described a novel model that combines stance detection and fact-checking. We demonstrated experimentally that these integrations are helpful both for stance detection and for fact-checking.

In the future, we plan to train our models on the unified corpus for the Arabic language as it is the only dataset that combines both stance detection and factuality components [49]

REFERENCES

[1] Sandulescu, Vlad, and Martin Ester. "Detecting singleton review spammers using semantic similarity." Proceedings of the 24th international conference on World Wide Web. ACM, 2015.

[2] Qazvinian, Vahed, et al. "Rumor has it: Identifying misinformation in microblogs." Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011.

[3] Kumar, Srijan, Robert West, and Jure Leskovec. "Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes." Proceedings of the 25th international conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2016.

[4] Horne, Benjamin D., and Sibel Adali. "This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news." Eleventh International AAAI Conference on Web and Social Media. 2017.

[5] Jing Ma, Wei Gao, and Kam-Fai Wong. 2018b. Rumor Detection on Twitter with Tree-structured Recursive Neural Networks. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vol. 1. 1980–1989.

[6] Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. Csi: A hybrid deep model for fake news detection. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. ACM, 797–806.

[7] Jiawei Zhang, Limeng Cui, Yanjie Fu, and Fisher B Gouza. 2018. Fake News Detection with Deep Diffusive Network Model. arXiv preprint arXiv:1805.08751 (2018).

[8] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting Rumors from Microblogs with Recurrent Neural Networks.. In IJCAI. 3818–3824.

[9] Yang Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, and Philip S Yu. 2018. TI-CNN: Convolutional Neural Networks for Fake News Detection. arXiv preprint arXiv:1806.00749 (2018).

[10] Yang Liu and Yi-fang Brook Wu. 2018. Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks. In AAAI.

[11] Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 849–857.

[12] Xu, Brian. Combating Fake News with Adversarial Domain Adaptation and Neural Models. Diss. Massachusetts Institute of Technology, 2019.

[13] Allcott, Hunt, and Matthew Gentzkow. "Social media and fake news in the 2016 election." Journal of economic perspectives31.2 (2017): 211-36.

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

[15] Tandoc Jr, Edson C., Zheng Wei Lim, and Richard Ling. "Defining “fake news” A typology of scholarly definitions." Digital Journalism 6.2 (2018): 137-153.

[16] House of Commons (2018). Disinformation and Fake News: An Interim Report. Fifth Report of Session 2017–19. House of Commons: Digital, Culture, Media and Sport Committee.

[17] Zannettou, Savvas, et al. "The web of false information: Rumors, fake news, hoaxes, clickbait, and various other shenanigans." arXiv preprint arXiv:1804.03461 (2018).

[18] Qian, Zhong, et al. "Document-Level Event Factuality Identification via Adversarial Neural Network." *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019.

[19] Saurí, R., & Pustejovsky, J. (2009). FactBank: a corpus annotated with event factuality. Language resources and evaluation, 43(3), 227.

[20] Z. Qian, P. Li, Y. Zhang, G. Zhou, and Q. Zhu, “Event factuality identification via generative adversarial networks with auxiliary classification.” in IJCAI, 2018, pp. 4293–4300.

[21] Saurí, Roser, and James Pustejovsky. "Are you sure that this happened? assessing the factuality degree of events in text." Computational Linguistics 38.2 (2012): 261-299.

[22] Qian, Zhong, Peifeng Li, and Qiaoming Zhu. "A two-step approach for event factuality identification." 2015 International Conference on Asian Language Processing (IALP). IEEE, 2015.

[23] R. Rudinger, A. S. White, and B. Van Durme, “Neural models of factuality,” arXiv preprint arXiv:1804.02472, 2018.

[24] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, “Fever: a large-scale dataset for fact extraction and verification,” arXiv preprint arXiv:1803.05355, 2018.

[25] AndreasVlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science, pages 18–22, Baltimore, MD, USA, June. Association for Computational Linguistics.

[26] Rashkin, Hannah, et al. "Truth of varying shades: Analyzing language in fake news and political fact-checking." Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017.

[27] Chen, Sihao, et al. "Seeing things from a different angle: Discovering diverse perspectives about claims." arXiv preprint arXiv: 1906.03538 (2019).

[28] Schuster, Tal, et al. "Towards debiasing fact verification models." *arXiv preprint arXiv: 1908.05267* (2019).

[29] Riedel, B., Augenstein, I., Spithourakis, G. P., and Riedel, S.(2017). A simple but tough- to-beat baseline for the fake news challenge stance detection task. CoRR, abs/1707.03264.

[30] Baird, S., Sibley, D., and Pan, Y. (2017). Talos targets disinformation with fake news challenge victory. In Fake News Challenge.

[31] Mohtarami, Mitra, Ramy Baly, James Glass, Preslav Nakov, Lluís Màrquez, and Alessandro Moschitti. "Automatic stance detection using end-to-end memory networks." arXiv preprint arXiv:1804.07581 (2018).

[32] Hanselowski, Andreas, P. V. S. Avinesh,

Benjamin Schiller, and Felix Caspelherr. Description of the system developed by team athene in the fnc-1. Technical report, 2017.

[33] Bowman, Samuel R., et al. "A large annotated corpus for learning natural language inference." arXiv preprint arXiv:1508.05326 (2015).

[34] Karadzhov, G., Gencheva, P., Nakov, P., and Koychev, I. (2017a). We built a fake news & click-bait filter: What happened next will blow your mind! In Proceedings of the 2017 International Conference on Recent Advances in Natural Language Processing, RANLP '17, Varna, Bulgaria.

[35] Karadzhov, G., Nakov, P., Màrquez, L., Barrón-Cedeño, A., and Koychev, I. (2017c). Fully automated fact checking using external sources. CoRR, abs/1710.00341.

[36] Qin, Yujia, et al. "Enhancing Recurrent Neural Networks with Sememes." arXiv preprint arXiv:1910.08910 (2019).

[37] Lingam, Vijay, et al. Deep learning for conflicting statements detection in text. No. e26589v1. PeerJ Preprints, 2018.

[38] Wang, Zhiguo, Haitao Mi, and Abraham Ittycheriah. "Sentence similarity learning by lexical decomposition and composition." arXiv preprint arXiv:1602.07019 (2016).

[39] Baeza-Yates, Ricardo, and Berthier Ribeiro-Neto. Modern information retrieval. Vol. 463. New York: ACM press, 1999.

[40] Huang, Anna. "Similarity measures for text document clustering." Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand. Vol. 4. 2008.

[41] Choi, Seung-Seok, Sung-Hyuk Cha, and Charles C. Tappert. "A survey of binary similarity and distance measures." Journal of Systemics, Cybernetics and Informatics 8.1 (2010): 43-48.

[42] Roser Saur´ı. 2008. A Factuality Profiler for Eventualities in Text. Ph.D. thesis, Brandeis University, Waltham, MA, USA.

[43] Habernal, Ivan, et al. "The argument reasoning comprehension task: Identification and reconstruction of implicit warrants." arXiv preprint arXiv:1708.01425 (2017).

[44] Popat, Kashyap, et al. "STANCY: Stance Classification Based on Consistency Cues." arXiv preprint arXiv:1910.06048 (2019).

[45] Available at www.fakenewschallenge.org .

[46] Yang, Baosong, et al. "Assessing the ability of self-attention networks to learn word order." arXiv preprint arXiv:1906.00592 (2019).

[47] Pawar, Atish, and Vijay Mago. "Calculating the similarity between words and sentences using a lexical database and corpus statistics." arXiv preprint arXiv:1802.05667 (2018).

[48] Heilman, Michael, and Kenji Sagae. "Fast rhetorical structure theory discourse parsing." arXiv preprint arXiv:1505.02425 (2015).

[49] Baly, Ramy, et al. "Integrating stance detection and fact checking in a unified corpus." arXiv preprint arXiv:1804.08012 (2018).

[50] Wang, Jingjing, et al. "Aspect Sentiment Classification with both Word-level and Clause-level Attention Networks." IJCAI. Vol. 2018. 2018.

[51] Moschitti, Alessandro. "Making tree kernels practical for natural language learning." *11th conference of the European Chapter of the Association for Computational Linguistics*. 2006.

[52] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[53] Chen, Ruiqi, et al. "Word-level sentiment analysis with reinforcement learning." IOP Conference Series: Materials Science and Engineering. Vol. 490. No. 6. IOP Publishing, 2019.

[54] Saurí, R., & Pustejovsky, J. (2009). FactBank: a corpus annotated with event factuality. Language resources and evaluation, 43(3), 227.

[55] Chen, Sihao, et al.(b) "PerspectroScope: A Window to the World of Diverse Perspectives." arXiv preprint arXiv:1906.04761 (2019).

[56] Yin, W., & Schütze, H. (2016). Multichannel variable-size convolution for sentence classification. arXiv preprint *arXiv:1603.04513.*

[57] [Dong and Dong 2003] Dong, Z., and Dong, Q. 2003. Hownet-a hybrid language and knowledge resource. In Proceedings of NLPKE.

[58] Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016, August). Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the* association for computational linguistics (volume 2: Short papers) (pp. 207-212).