INTERNATIONAL JOURNAL OF ADVANCED STUDIES IN COMPUTER SCIENCE AND ENGINEERING
IJASCSE VOLUME 9 ISSUE 11, 2020
11/30/2020

# Factual or Non-Factual Claim: Verifying Claims

**Fatima T. Alkhawaldeh**
**Department of Computer Sciences**
**University of York**
**York, UK**
**ftma500@york.ac.uk**

***Abstract* – In debatable topics, people use evidence to reason towards a claim. The claim conveys a stance towards a particular aspect in the evidence. Existing studies mainly focus on identifying claim stance; which is determined by its relevant evidence; however, the task to get a factual claim if the claim is non-factual is not considered. We thus study the question to what extent a false claim can be reconstructed from its premises to be true, either by generating a new factual claim from relevant premises or determining the positions for the misleading information in the false claim and modify it concerning to the evidence. To address such issue, we introduce a factual claim-making task, anew task to predict the factuality of the claim that is associated with evidence that supports or refutes the given claim. If the claim is non-factual, we propose two different models to get a factual claim. In the generator model, we generate a factual claim by applying the generation model. In the modifier model, we depend on the sequence operation model to modify the misleading information. The experimental results on Perspectum dataset show the effectiveness of our models. The performances of the proposed system achieved 76.84% and 78.36% of F1 scores for the generator and modifier mode, respectively.**

***Keywords: stance detection, Hierarchical Reinforcement Learning (HRL) and Deep Learning.***

## I. INTRODUCTION

The fact-checking task is the process to check the veracity of a claim against relevant evidence. In the last time the factuality checking task was performed manually by journalists. Since the internet has become a rapidly growing amount of controversial statements from politicians, biased news reports, rumours and other domains, the necessity to develop an automated model for assessing the truthfulness of claims has increased. As social media has quickly risen to be as a news source, more spread rumor and misinformation is propagated. The progress in natural language processing and information retrieval in addition to the availability of datasets tools help to automate the process of fact-checking.

There is a rich literature on fact-checking aims to measure the truthfulness of a claim, given evidence [1-9]. In [1], the authors released 221 labelled claims in the political domain; they consider intermediate classes as "mostly truth" or "half-truth" when the sentences are not entirely fake or real. In [2], a dataset of approximately 60 million tweets about more than 1,000 news events are labelled according to their credibility. The approach in [3] jointly estimates the credibility of sources and correctness of the claims using the Probabilistic Soft Logic framework. A dataset collected from fact-checking website PolitiFact is released in [4] labelled by multiple classes: pants-fire, false, barely true, half-true, mostly true, and true. Fact check shared task on automatic identification and verification of claims in political debates is in [5], automatically estimate the level of fact-checking of the check-worthy claims. A method with the ability to generalize on unseen data is proposed in [6] to deal with the problems of Fake News Detection. In [7], a multi-class fake news detection framework is applied where a combination of LSTM, CNN, and fully connected network to determine the veracity of fake and real where they integrated multiple pieces of information about a claim. The work in [8] extended the LIAR dataset and labelled a claim, and they use meta-information and "justification," human-written reasoning for factuality checking. They address text entailment of detecting false claims and prove joint learning could enhance both tasks: claim verification and evidence selection [9].

In this paper, we describe a novel task of fact-checking, making a factual claim if not. We suppose that the correction task of false claims can clarify the reason for assuming the claim was real at first, and it was circulated as legitimized. Correcting the false information had a positive effect on people whose beliefs may be affected by responding to misleading information. To achieve that, we propose two models; the generator model is for generating factual claim and the modifier model for moving the false claim to be accurate by modifying the misleading information in the claim. Generator model asses the truth of the claim again relevant evidence then we try to correct the non-factual claims by generating a new claim based on the main aspect of the claim and the substantiating evidence using hierarchical reinforcement learning. Finally, we check whether the new generated claim is true or not. Concerning the modifier model, we propose a sequence operation-based model to detect the wrong information and modify it to make a factual claim. For example, the claim "Animals should have lawful rights", from PERSPECRUM dataset [17] is attacked by the perspective to "animals have no interest and rationality", and the stance label is set to oppose. So according to this perspective and it substantiating evidence the claim is non-factual, so we try to fix the original claim.

## II. OUR PROPOSED MODEL

To detect the factuality of claim, Initially, we segment each sentence into several clauses using sentence-level discourse segmentation then measuring the cosine similarity to decide whether a clause is related to the claim or not. The most related clauses are fed to the model as evidence input. First, we check the evidence clauses with the claim, if the claim has a high correlated then mostly it is correct if not then go to the HRL. Figure (1) shows the general architecture of our factual generator, where each claim is verified with the substantiating evidence clauses if the claim is the truth then no need to generate or modify the claim.

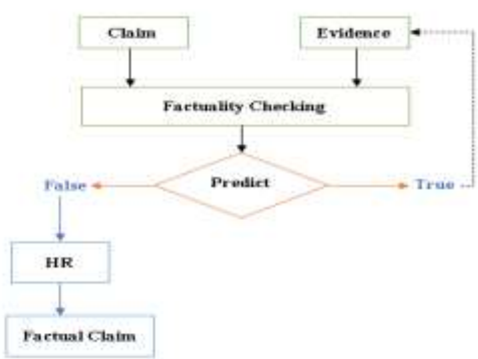


**Fig. 1**: Our Proposed Factual Claim Generator Model

### A. *Factuality Checking*

We propose a novel model: Manhattan Multi-channel LSTM-GRU-CNN, which is a Siamese deep Neural Network. This model uses word embedding vectors to create embedded matrices which are fed to the LSTM-GRU-CNN channels model. Then, we combine all the features come from all channel's LSTM-GRU-CNN for both, claim and evidence into a single numeric value. Manhattan distance [10] is used to get the degrees of the similarity between a claim and evidence. Multi-channel can capture high-level features and long-term dependency and obtained promising performance [11]. For Lexical Embedding, we use pre-trained, WordVec [12], Glove embeddings [13], Elmo [14].

We are using different deep learning models to generate encoded sequences, Bidirectional Long Short-Term Memory (BiLSTM), Bidirectional Long Gated Recurrent Unit (BiGRU) and Convolutional Neural Network (CNN). For BiLSTM, we get the summarized representation from both directions to represent the word sequence obtaining the contextual information of the current word in addition to learn long term dependencies. We also use them to encode the sentences. BiGRU is a variant of BiLSTM that simplifies the gating mechanism and quickly in training, then obtaining the text features fast.

Regarding CNN, we use it to learn featured based on the most salient information and obtain significant and local features. Despite that, CNN is unable to capture the features of the global and long-distance, it is faster to train. All features will be concatenated to form output vectors for both claim and evidence inputs. Claim and evidence output vectors are fed Manhattan distance to check the veracity of claim toward evidence [10]. Figure (2) shows our proposed factuality checker model.

### B. *Generator Model: Factual Claim Generator Based Hierarchical Reinforcement Learning Approach*

For non-factual claim, all related evidence clauses $\{c_1, c_2 .. c_n\}$ will be sent to the HRL; the high-level policy applies word and clause level claim attention to select the more claim relevant clauses. All relevant clauses $\{c_1, c_2 .. c_{n-m}\}$ will be sent to the medium-level policy, where deep communicating agents are implemented for encoding these clauses, helping to decide the next sub-goal (copy or generate). The low-level policy has the role in executing the actions to produce the sequence of the words (choosing words to create the factual claim). Figure (3) show the HRL model to generate a factual claim.

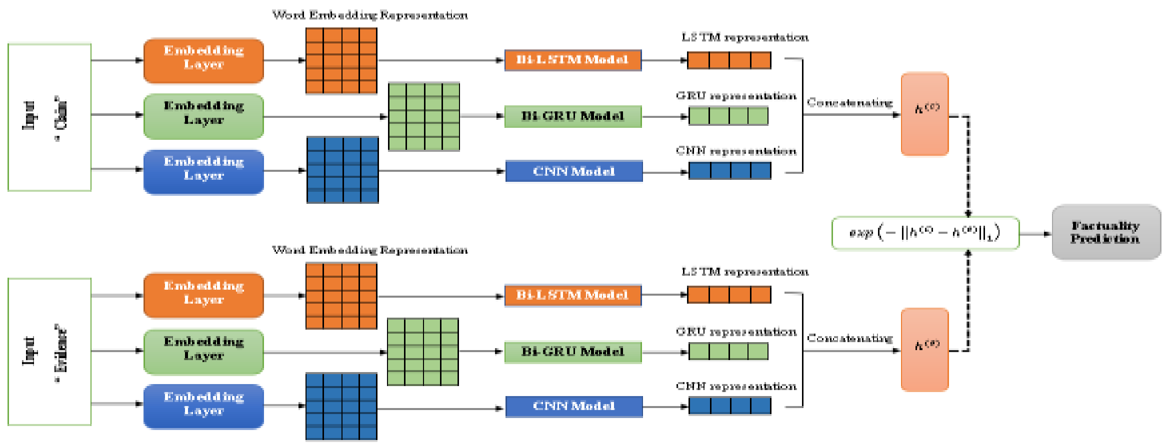


**Fig. 2**: Factuality Checker Model

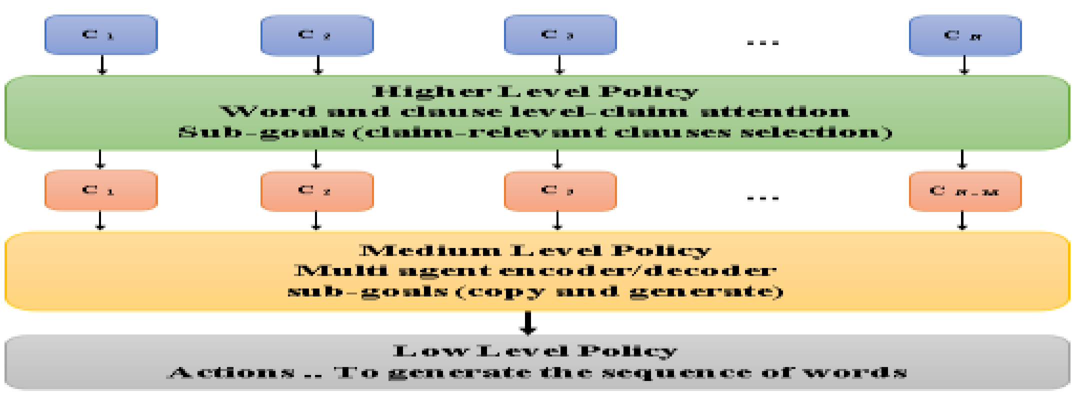


**Fig. 3:** Hierarchical Reinforcement Learning (HRL)

➢ **Higher Level Policy**

For claim-relevant clauses, high-level policy adopts the hierarchical attention mechanism, word-level and clause-level attention networks, to select informative words and clauses relevant to a specific claim.

- **Word-level Claim Attention:** ***Word Encoding***

In word-level claim attention network, word encoding layer concatenates claim representation to each word embedding and then summarizes information by bi-directional GRU. For each evidence, Bi-GRU (Gated Recurrent Units) will be used in order to encode the word information in each clause from forward and backward direction

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

$$\overleftarrow{h}_{ij} = \overleftarrow{GRU}_{(\widehat{w}_{ij})};\ \ i \in [1,\ C],\ j \in [N_i, 1] \quad (2)$$

$$h_{ij} = \vec{h}_{ij} \oplus \overleftarrow{h}_{ij} \quad (3)$$

- **Word-level Claim Attention:** ***Word Attention***

Word attention layer focuses on the terms that are important to the meaning of the clause with respect to the claim, producing clause vectors. Attention mechanism will be implemented to concentrate on those words in the evidence clause with respect to a specific claim CP and combine the representation of all of them to form a clause vector of evidence.

$$u_{ij} = tanh\left(W_w \cdot [h_{ij}; CP] + b_w\right) \quad (4)$$
$$a_{ij} = softmax(u_{ij}) = \frac{exp(u_{ij})}{\sum_{t=1}^{N} exp(u_{it})} \quad (5)$$
$$c_i = \sum_{i=1}^{N_i} a_{ij} \cdot h_{ij} \quad (6)$$

- **Clause-level Claim Attention:** ***Clause Encoding***

Clause encoding layer applies Bi-directional GRU to capture the context clause representations.

The contextual information of each clause is obtained by BI-GRU

$$\vec{h}_i = \overrightarrow{GRU}_{(c_i)}; \quad i \in [1, C] \quad (7)$$
$$\overleftarrow{h}_i = \overleftarrow{GRU}_{(c_i)}; \quad i \in [C, 1] \quad (8)$$
$$h_i = \vec{h}_i \oplus \overleftarrow{h}_i \quad (9)$$

- **Clause-level Claim Attention:** ***Clause Attention***

After that, in clause attention Layer, attention mechanism computes the attention weight between each claim-clause representation to produce contextual information conditioned on the claim representation. The attention weight between each clause and the representation of a specific claim will be computed as follows:

$$m_i = tanh(W_c \cdot [h_i; CP] + b_c) \quad (10)$$
$$a_i = softmax(m_i) = \frac{exp(m_i)}{\sum_{t=1}^{C} exp(m_t)} \quad (11)$$

In this policy, to select claim-relevant clauses, conditional probability is used. The selected clauses are sent to the middle-level policy, where the multi-agent encoder is used to generate hidden states for the evidence clauses considering the claimed interest.

➢ **Middle-Level Policy: Multi-Agent Encoder**

The context and states from the environment are used to create all possible sub-goals (to copy or to generate) which should be achieved by the lower agent policy to select a series of actions (words) and produce a new sequence of words. We depend on relevant clauses segments as input, then apply a stack of deep learning models: CNN, Maxpooling layer+ GRU. We use the message sharing mechanism to help other agents' encoders to generate better contextual information conditioned upon the messages received from other agents. For Multi agent encoder, we use the following equations where message passing is applied:

$$\vec{h}_i^{(1)}, \overleftarrow{h}_i^{(1)} = bGRU\left(e_i, \vec{h}_{i-1}^{(1)}, \overleftarrow{h}_{i+1}^{(1)}\right) \quad (12)$$
$$h_i^{(1)} = W_1\left[\vec{h}_i^{(1)}, \overleftarrow{h}_i^{(1)}\right] \quad (13)$$
$$\vec{h}_i^{(l+1)}, \overleftarrow{h}_i^{(l+1)} = bGRU\begin{pmatrix} \mathrm{fun}\left(h_i^{(l)}, \mathrm{mes}^{(l)}\right), \\ \vec{h}_{i-1}^{(l+1)}, \overleftarrow{h}_{i+1}^{(l+1)} \end{pmatrix} \quad (14)$$
$$h_i^{(l+1)} = W_2\left[\vec{h}_i^{(l+1)}, \overleftarrow{h}_i^{(l+1)}\right] \quad (15)$$
$$mes^{(l)} = \frac{1}{N-1}\sum_{n \neq a} h_{n,I}^{(l)} \quad (16)$$
$$fun = v_1^T \tanh\left(W_3 h_i^{(l)} + W_4 mes^{(l)}\right) \quad (17)$$

$e_i$ is word embedding, $h_i^{(1)}$ is the concatenation for both direction for hidden states before consider other agent information, mes is the encoded information from other clauses and $fun$ is score function.

○ **Decoder with Claim and Evidence Attentions**

Inspired by [16], to guide the decoder focus on the claim concentrated aspect, the decoder calculates the attention weights for every word in the claim and evidence which are calculated by Claim attention and Evidence attentions respectively.

- **Claim attention**

$$s_t = GRU_d\left(s_{t-1}, \left[h_{i\ t}^{cl}, E(\mathcal{Y}_{t-1}); c_{t-1}^*\right]\right) \quad (18)$$
$$a_{t,i}^{cl} = v_{cl} \cdot \tanh\left(W_{cl}s_t + U_{cl}h_i^{cl}\right) \quad (19)$$
$$\alpha_{t,i}^{cl} = \frac{exp\left(a_{t,i}^{cl}\right)}{\sum_{i=1}^{|cl|} exp\left(a_{t,i}^{cl}\right)} \quad (20)$$
$$cl_t = \sum_{i=1}^{|cl|} \alpha_{t,i}^{cl} h_i^{cl} \cdot \quad (21)$$

- **Evidence attentions (word attention distribution)**

$$a_{t,j}^{d} = v_d \cdot \tanh\left(W_d s_t + U_d h_j^{l} + Z cl_t\right) \quad (22)$$
$$a_{t,j}^{d} = \frac{exp\left(a_{t,j}^{d}\right)}{\sum_{j=1}^{|w|} exp\left(a_{t,j}^{d}\right)} \quad (23)$$

$d_a^t = \sum_i a_{t,j}^d h_{a,i}^{(l)}$ For each clause (by each agent)

- **Agent Attention**

The last hidden state from each agent sent to the decoder to compute the global agent attention as follows

$$agent^t = (\tanh(W_7 d^t + W_8 s_t + b_2)) \quad (24)$$
$$agent^{att} = softmax(agent^t) \quad (25)$$

$$c_t^{new} = \sum_a agent_a^{att} d_a^t \quad (26)$$

$$P_{subgoal} = sigmoid\left(w_g * (s_t + yt\text{-}1 + c_t^{new}) + b_g\right) \quad (27)$$

st is a state that is computed by decoder by attending to relevant input context provided by the agents, *yt*-1 is the previous target word, $c_t^{new}$ is the agent context vector.

### ➢ Low-Level Policy

After receiving sub-goals from the middle-level policy, low-level policy performs basic actions to achieve the specified goal (selecting words), following this equation:

$$P^{voc}(w_t) = softmax\left(MLP([s_t, c_t^{new}])\right)$$

$P^{voc}$ *is* Vocabulary distribution

$$P_{action} = P_{subgoal(generate)} * P^{voc} + P_{subgoal(copy)} * \sum_{i:w_i=w} a_{it} \quad (28)$$

$y_{t-1}$ is the embedding vector of the previously generated word

The final evidence hidden states will *is* used to initialize the first state of the GRU in the decoder. If word is an out-of-vocabulary (OOV) word, then Pvocab (w) is zero; similarly if w does not appear in the source document, then åi:wi=w ati is zero.

### ▪ Multi Rewards

We apply a rewarder function to compute the factuality of the new claim using entailment and semantic similarity metrics to find a policy $\pi*$ that maximizes the reward for each visited state s and action a.

$$\pi_j^{*i}(s) = \arg\max Q_j^{*i}(s,a) \quad (29)$$

For high-level policy, the cumulative reward is calculated between the claim embedding and the selected candidate clause, using cosine similarity as follows:

$$r_i^h = \lambda_1 \sum_{t=i}^{n} \gamma^{t-i} \log cos(v_a, \hat{v}_t) \quad (30)$$

For low-level policy, we calculated the entailment probability score between the evidence (as a premise) and generated a factual claim (as a hypothesis). We apply the Entailment Corrected Reward as in [15].

## *C. Modifier Model: Sequence Operation Based Hierarchical Reinforcement Learning Approach*

In this paper, we propose a hierarchical reinforcement learning approach for claim factuality prediction and adjusting it if its factuality is incorrect. The main idea of the proposed approach is to perform a factuality prediction. In Particular, our approach employs a high-level policy and a low-level policy to perform clause selection and claim to adjust respectively. Our Hierarchical Reinforcement Learning (HRL) approach contains three components: a high-level policy for clause selector, a low-level policy for claim adjustor; and fact predictor for providing reward signals to guide both the above clause selector and claim adjuster. Given an evidence article with a clause sequence and a claim, the high-level policy decides whether a clause mentions this claim. If it is relevant and not noisy, the high-level policy chooses this clause and send them to the low-level policy, which aggregates these clauses as one statement and consider them for claim adjusting by selection one action at each time to change the misleading information in false claim to be true. After low-level action is done, the factuality prediction is employed to provide reward signals to guide the above clause selection and claim to adjust.

- **High-level policy: claim relevant clauses selector**

First, a high-level policy is proposed to select claim-relevant clauses and remove irrelevant clauses. **State**: given the segmented article to the clause and a claim as input, the policy aimed to decide the clam relevant clauses and passed the selected clauses to the low-level policy which take actions to the false claim to alter it to be true. Afterword embeddings $e_i$is performed, we use Bi-GRU to get the vector representation of clause $h_s^{(1)} + h_s^{(1)} + h_s^{(2)} + \cdots + h_s^{(n)}$. After getting the hidden state representations of claim, we perform an average pooling vector $claim^{(l)}$ through the following equations:

$$\vec{h}_i^{(1)}, \overleftarrow{h}_i^{(1)} = bGRU\left(e_i, \vec{h}_{i-1}^{(1)}, \overleftarrow{h}_{i+1}^{(1)}\right) \quad (31)$$

$$h_i^{(1)} = W_1\left[\vec{h}_i^{(1)}, \overleftarrow{h}_i^{(1)}\right] \quad (32)$$

$$claim^{(l)} = \frac{1}{N-1}\sum_j h_j \quad (33)$$

$$st = h_s^{(1)} + h_s^{(1)} + h_s^{(2)} + \cdots + h_s^{(n)} + claim^{(l)} \quad (34)$$

**Action**: A stochastic policy uses the state information for deciding to select the clause or not. We adopt a logistic function (conditional probability) to decide whether this clause is relevant for a claim or not.

$$\text{action} = sigmoid\left(\text{W st, b}\right) \quad (35)$$

**Reward**: For high-level policy, the high-level cumulative reward is calculated between the claim embedding and the selected candidate clause, using cosine similarity and the signal from fact predictor as follows:

$$claim^{(l)} = \frac{1}{N-1}\sum_j h_j \quad (36)$$

$$clause^{(l)} = \frac{1}{M-1}\sum_i h_j \tag{37}$$

$$r_i^h = \lambda_1 \sum_{t=i}^{n} \gamma^{t-i} \log cos\left(claim^{(l)}, clause^{(l)}\right) + \text{fact predictor} \tag{38}$$

$\lambda$ is weight parameter and $\gamma$ is the discount factor

- **Low-level policy: claim adjuster**

we model it as an attention-based [18] pointer network, which assigns normalized probability to each position where the misleading information may occur. **The clauses representation represents the state** $clause^{(l)}$ and hi is each position representation of claim. **Action**: we adopt an attention-based policy to take an action, sequence operation, where i denotes each position in the input claim.

$$m_i = tanh\left(W_c \cdot \left[h_i; clause^{(l)}\right] + b_c\right) \tag{39}$$

$$a_i = softmax(m_i) = \frac{exp(m_i)}{\sum_{t=1}^{C} exp(m_t)} \tag{40}$$

M (action l $clause^{(l)}$; i) = softmax (w *hi), hi is the position word where an action should be taken. The actions are inserted, delete or replace the word

**Reward**: fact predictor, we apply double-layer attention mechanism Intra-sequence Attention Layer and Inter-sequence Attention Layer, for claim contextual features extraction as shown in figure 4. We propose a new model, incorporate the claim-relevant clauses with a double-layer attention mechanism: Intra-sequence Attention Layer and Inter-sequence Attention Layer to capture latent correlation features among the claim-relevant clauses sequence. Intra-sequence Attention Layer (intra-relation reasoning) and Inter-sequence Attention Layer used to obtain the characteristic representation of the claim-relevant clauses and find the characteristic representation of the claim-relevant clauses

$$V = BIGRU(clause) \tag{41}$$

$$V_c = \tanh(V) \tag{42}$$

$$\alpha = softmax(v \cdot V_c\ ^T) \tag{43}$$

$$ri = \tanh(\alpha \cdot V) \tag{44}$$

$r$ is a representation of the claim-relevance clause

$$V_{cs} = \tanh(R_t) \tag{45}$$

$$\alpha_t = softmax(v_t \cdot V_{ct}\ ^T) \tag{46}$$

$$r_e = \tanh(\alpha_t \cdot R_t) \tag{47}$$

$R_t$ *is* clause sequence features, $r_e$ *is* characteristic representation for all claim-relevant clauses. A SoftMax layer performs the final factuality prediction output as the classifier.

$$\text{out} = softmax(v \cdot r_e + b) \tag{48}$$

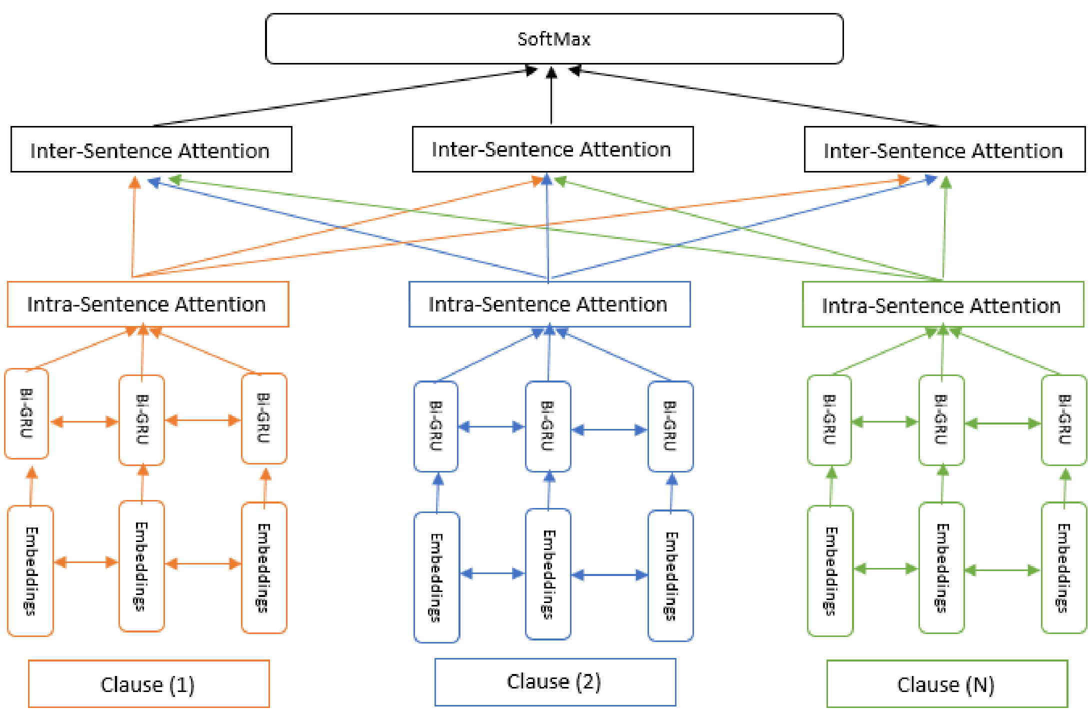


**Fig. 4:** Fact Predictor Model

## III. EXPERIMENTAL RESULTS

**Dataset** We used the publicly available dataset PERSPECTRUM provided by [17]. PERSPECTRUM, a dataset of claims, perspectives and evidence paragraphs. It contains 907 claims, 11,164 perspectives and 8,092 evidence paragraphs. For the factuality checking model, to check the factuality of claim, we suppose that it has support stance with its related perspective, then it should be factual. If the substantiating evidence of the related perspective with is refuted stance, then the claim should be non-factual. For factual claim, the claim should have entailment relation with the connected perspective

## IV. EVALUATION MODEL

The factual claim (generated) should have support stance with the original related perspective. We apply Bi-GRU Siamese network with attention. Bi-GRU output vector will be multiplied by a weight, which is determined by the claim representation. Using a BI-GRU based Siamese architecture (it is two networks with the same structure and the same weight, each process one sentence in a pair) to model both claim m and perspective p. where hi is the hidden state of the GRU at time-step i,(or annotation), briefing all the information of the sentence, c(a,h) is annotation attention mechanism assigns a weight ai to each word annotation, which indicates its importance and z is e the final representation and y is the label of the relation between the generated claim and a support stance-perspective. Figure 5 shows the general architecture for our proposed evaluation model.

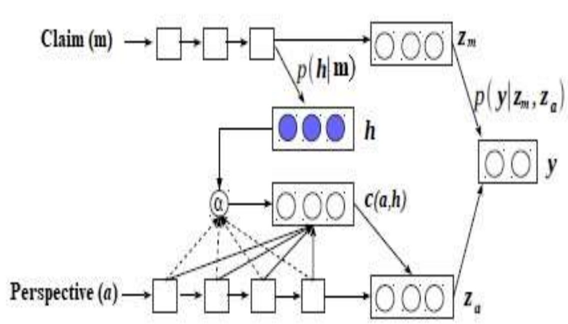


**Fig. 5**: Evaluation Model

In addition to the evaluation model, we apply other Evaluation metrics. Use the ROUGE scores [19] to evaluate the quality of generated factual claim. ROUGE-1, ROUGE-2 and ROUGE-L measure respectively the unigram overlap, bigram-overlap, and the longest common sub-sequence between the predicted and reference. We also, evaluate our approach on the Perspectrum dataset using F-score. **The F-SCORE** balances the generated text's precision and recall by measuring the harmonic mean of the two measures, Recall and Precision. Precision is the fraction of n-grams in the model-generated text that is present in the reference text. The recall is the fraction of the n-grams in the reference text that are present in the candidate text [9]. F- SCORE = 2 × (Precision · Recall) /(Precision + Recall). For the correct label, we try to generate a claim has a supportive stance with the perspective related to. In other words, the claim is entailed by the perspective. Our experiments demonstrate that our system performs well in generating a factual claim.

The results based on our Evaluation Model, are 77.84 for the generator model and 80.26 for the modifier model. For the generator model, ROUGE-1 ROUGE-2 Rouge-L values are 27.41, 7.93 and 25.83, respectively. For the modifier model, ROUGE-1 ROUGE-2 Rouge-L values are 28.50, 9.73 and 27.36, respectively. The results show that modifying the misleading information in the false claim is more effective for obtaining factual claim rather than generating a new claim from its premises. The performances of the proposed system achieved 76.84% and 78.36% of F1 scores for the generator and modifier mode, respectively. Our detailed analysis shows that our model modifier performs better that generator model according to all evaluation methods.

## V. CONCLUSION AND FUTURE WORK

This paper proposes a novel task for supervised learning-based, making a factual claim approach. We develop neural network-based models that use a claim context information to make a factual claim in the case of non-factual claim. We find that the neural network-based model performs better when modifying the misleading information instead of generating a new claim from its premises. For claim generating model, we explored the problem of encoding long evidence articles to generate a factual claim and demonstrated that the use of hierarchical reinforcement learning could improve the generation by automatic evaluation. Analysis proves that this enhancement is due to the ability of multi-agent to cover the relevant information of the claim and generate a factual claim. For claim modifying model, sequence operation-based method with the hierarchical reinforcement learning (HRL) effectively addresses the non-factual claim problem. For future work, there is a need for further research and more advanced models to reduce the available false information and analyzing its impact on different domains. Furthermore, we will study to what extent people accept the information that has been corrected and the possibility to change their minds

## REFERENCES


1. Vlachos, Andreas, and Sebastian Riedel. "Fact-checking: Task definition and dataset construction." *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. 2014.
2. Mitra, Tanushree, and Eric Gilbert. "Credbank: A large-scale social media corpus with associated credibility annotations." *Ninth International AAAI Conference on Web and Social Media*. 2015.
3. Samadi, Mehdi, et al. "Claimeval: An integrated and flexible framework for claim evaluation using credibility of sources." *Thirtieth AAAI Conference on Artificial Intelligence*. 2016.
4. Wang, William Yang. "" liar, liar pants on fire": A new benchmark dataset for fake news detection." *arXiv preprint arXiv:1705.00648* (2017).
5. Nakov, Preslav, et al. "Overview of the CLEF-2018 CheckThat! Lab on automatic identification and verification of political claims." *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, Cham, 2018.
6. Hanselowski, Andreas, et al. "A retrospective analysis of the fake news challenge stance detection task." *arXiv preprint arXiv:1806.05180* (2018).
7. Karimi, Hamid, et al. "Multi-source multi-class fake news detection." *Proceedings of the 27th International Conference on Computational Linguistics*. 2018.
8. Alhindi, Tariq, Savvas Petridis, and Smaranda Muresan. "Where is your Evidence: Improving Fact-checking by Justification Modeling." *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. 2018.
9. Yin, Wenpeng, and Dan Roth. "TwoWingOS: a two-wing optimization strategy for evidential claim verification." *arXiv preprint arXiv:1808.03465* (2018).
10. Mueller, Jonas, and Aditya Thyagarajan. "Siamese recurrent architectures for learning sentence similarity." *thirtieth AAAI conference on artificial intelligence*. 2016.
11. Yoon, Joosung, and Hyeoncheol Kim. "Multi-channel lexicon integrated CNN-BiLSTM models for sentiment analysis." *Proceedings of the 29th conference on computational linguistics and speech processing (ROCLING 2017)*. 2017.
12. Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.
13. Pennington,Jeffrey,Richard Socher, and Christopher D.Manning."Glove: Global vectors for word representation. " Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.
14. Peters, Matthew E., et al. "Deep contextualized word representations." arXiv preprint arXiv:1802.05365 (2018).
15. Pasunuru, Ramakanth, and Mohit Bansal. "Reinforced video captioning with entailment rewards." *arXiv preprint arXiv:1708.02300* (2017).
16. Celikyilmaz, Asli, et al. "Deep communicating agents for abstractive summarization." *arXiv preprint arXiv:1803.10357* (2018).
17. Chen, Sihao, et al. "Seeing things from a different angle: Discovering diverse perspectives about claims." *arXiv preprint arXiv:1906.03538* (2019).
18. Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473* (2014).
19. Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." *Text summarization branches out*. 2004.