SADHAN: Hierarchical Attention Networks to Learn Latent Aspect Embeddings for Fake News Detection

Rahul Mishra
University of Stavanger, Norway
rahul.mishra@uis.no

Vinay Setty
University of Stavanger, Norway
vsetty@acm.org

ABSTRACT

Recently false claims and misinformation have become rampant in the web, affecting election outcomes, societies and economies. Consequently, fact checking websites such as snopes.com and politifact.com are becoming popular. However, these websites require expert analysis which is slow and not scalable. Many recent works try to solve these challenges using machine learning models trained on a variety of features and a rich lexicon or more recently, deep neural networks to avoid feature engineering.

In this paper, we propose hierarchical deep attention networks to learn embeddings for various latent aspects of news. Contrary to existing solutions which only apply word-level self-attention, our model jointly learns the latent aspect embeddings for classifying false claims by applying hierarchical attention. Using several manually annotated high quality datasets such as Politifact, Snopes and Fever we show that these learned aspect embeddings are strong predictors of false claims. We show that latent aspect embeddings learned from attention mechanisms improve the accuracy of false claim detection by up to 13.5% in terms of Macro F1 compared to a state-of-the-art attention mechanism guided by claim-text (DeClarE). We also extract and visualize the evidence from the external articles which supports or disproves the claims.

CCS CONCEPTS

• **Information systems → Document representation**; • **Computing methodologies → Neural networks**.

KEYWORDS

fake news; hierarchical attention; latent aspect embeddings

ACM Reference Format:
Rahul Mishra and Vinay Setty. 2019. SADHAN: Hierarchical Attention Networks to Learn Latent Aspect Embeddings for Fake News Detection. In *The 2019 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '19), October 2–5, 2019, Santa Clara, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3341981.3344229

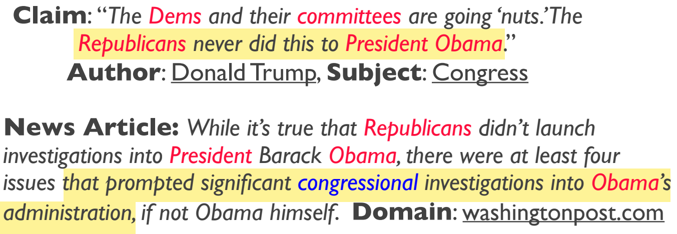
1 INTRODUCTION

The unprecedented growth of the web, online news and social media has led to a paradigm shift in the way people consume information. As a consequence, spread of misinformation or fake news in online media has become faster and wider than ever before. To counter this, several fact checking websites such as snopes.com, politifact.com and fullfact.org are becoming increasingly popular. These websites have dedicated experts manually classifying the credibility of news articles and claims which is slow and tedious.

To address these limitations, several automated machine learning models are proposed in the literature. Early works in this area focused on the tedious task of curating a rich lexicon and other credibility features manually to capture the language of deception [9, 10, 12]. More recent approaches avoid feature engineering by designing deep neural network models which are able to learn non-trivial patterns from the raw text of the claims or facts [14]. However, verifying correctness of claims purely based on the claim text has limited effectiveness due to lack of context information.

To overcome the above problem, recent works incorporate ***external evidence*** retrieved from news media and social media which potentially either supports or refutes the claim [11, 21]. These works propose a word-level attention mechanism guided by the claim text to focus on parts of the external evidence for this purpose. However, it has been shown that word-level attention alone fails to capture the complex structure of the documents [20]. **Hierarchical attention** mechanism which applies attention at sentence level in addition is shown to be more effective for document classification. For example, in Figure 1, we can notice that using word-level attention (in red font) alone makes it hard to determine if the evidence supports or refutes the claim. However, the sentence level attention (highlighted text) is able to capture the context better.

While attention guided by the claim text is effective to some extent for detecting fake news, it has been shown that it is not sufficient [11]. In order to effectively use external evidence for fake news detection, determining its credibility in the context of the given claim and its author (source) is also essential. Popat et. al in [11], propose the use of static representation (one-hot encoding) of source information (domains) together with attention weights for this purpose. However, we postulate that understanding the context and credibility of the evidence requires learning indicative and salient



Claim: "*The Dems and their committees are going 'nuts.'The Republicans never did this to President Obama.*"
Author: Donald Trump, **Subject**: Congress

News Article: *While it's true that Republicans didn't launch investigations into President Barack Obama, there were at least four issues that prompted significant congressional investigations into Obama's administration, if not Obama himself.* **Domain**: washingtonpost.com

Figure 1: Example of a false claim, word and sentence-level attention using latent aspects (Subject, Author and Domain)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICTIR '19, October 2–5, 2019, Santa Clara, CA, USA

ACM ISBN 978-1-4503-6881-0/19/10…$15.00
https://doi.org/10.1145/3341981.3344229

vocabulary, writing style and sentence structures specific to the latent aspects of news. For example, in Figure 1, relying only on the *claim-text attention* does not successfully classify that it is a false claim. Given that the claim is from "Donald Trump" (**Author**), related to the **Subject** "Congress" can guide the attention to a new word such as "congressional" which is missing in the claim text. In addition, the professional writing style of journalists from the **Domain** "washingtonpost.com" further refutes the claim. We hypothesize that the attention due to latent aspects is able to capture the necessary patterns to check if the external evidence supports or refutes the claim. This task is commonly known as **entailment**.

To address these limitations, in this paper, we propose a novel model coined SADHAN[1] to jointly learn embeddings for different latent aspects of news using the hierarchical attention mechanism. Intuitively, the attention mechanism learns a unique global representation (embedding) for each of the latent aspects. These embeddings capture the necessary textual patterns needed to distinguish a claim for being true or false. For example, an embedding learned for the author "Donald Trump" captures the patterns from discussions about false and true claims made by him. Similarly, embeddings representing each of the latent aspects capture the necessary patterns from the representative relevant articles to distinguish the veracity of the claims. We illustrate that these embeddings are indeed able to distinguish false and true news in Section 6 by visualizing these embeddings in two dimensions using t-SNE. Note that the latent aspect embeddings are not limited to the subject, author and domain aspects but they are general purpose and can be used for any latent aspects which are relevant for the task.

One of the critical tasks performed by the fact checking websites is to provide evidence for the veracity of the claim. This is a highly cognitive task and usually done manually by experts. Therefore, it is not sufficient to just automate fake news detection but also to automatically extract the supporting evidence. In previous works, word-level attention weights are used to extract the evidence and visualize the words in textual snippets [11]. However, just using word level attention weights to visualize evidence is not very user friendly. In this paper, we propose an algorithm to fuse the word level and sentence level attention weights guided by various latent aspect embeddings to extract evidence snippets which are easier for humans to understand.

In summary, our contributions are:

(1) Hierarchical attention to learn claim and document structure
(2) Jointly learning latent aspects of news using hierarchical attention mechanism
(3) Extensive experiments using three high quality datasets
(4) Visualization and analysis of latent aspect embeddings
(5) Evidence extraction and visualization of attention mechanism for interpretability

Our experiments using data crawled from Snopes and Politifact, show that latent aspect embeddings jointly learned using SADHAN, are very effective in detecting false claims. Specifically, we gain up to 12% improvement in Macro F1 for Politifact and 13.5% for Snopes compared to the state-of-the-art solution based on claim text attention and source embeddings [11]. In addition, we illustrate that the latent aspect embeddings learned by our model are effective for detecting false claims on their own by visualizing them.

2 RELATED WORK

Some of the first methods for detecting fake news have been using linguistic cues [9, 12] and source-based credibility features [10]. However, identifying the specific linguistic cues that are decisive for fake news is not yet fully understood.

Deep learning methods to avoid feature engineering have also been proposed [6, 7, 11, 14]. In [11], the authors concatenate the claim text and content of the article and apply word-level self-attention to detect false claims. We improve on this architecture to include sentence-level attention as well as attention guided by the latent aspect embeddings. In [11], the authors also use word-level attention to extract evidence snippets, we instead propose an algorithm to select top-K sentences based on the attention weights both at word and sentence level.

There are also efforts to address some sub-problems of detecting false claims such as entailment [18, 21]. Another related task is stance detection[2]. While these tasks are not the same as detecting fake news, it could be used for checking the veracity of claims. SADHAN implicitly also depends on entailment among other patterns to detect fake news. To support this, we also use our model to evaluate the Fever dataset published by [18].

Several recent works have shown that using context from social network users and interactions have improved fake news detection [2, 5, 13, 16, 17, 19]. However, these approaches are only suitable when there is sufficient information from social networks associated with the news available. We could integrate SADHAN into these models to achieve further improvements.

The neural architecture of SADHAN is inspired by the hierarchical architecture in [20] originally proposed for document classification. While speaker-based attention has been used before [4], using it for hierarchical attention and multiple latent aspects has never been tried before for fake news detection.

In summary, we are the first to propose a hierarchical attention network which jointly learns latent aspect embeddings to detect fake news.

3 PROBLEM DEFINITION AND PROPOSED MODEL

3.1 Problem Definition

Given a claim $c \in C$ in textual form, along with its latent aspects such as subject, author, domain and a set of candidate relevant documents $D = \{d_1, ..., d_m\}$ as evidence from different domains, the goal is to classify the claim as either "True" or "False".

3.2 SADHAN Model

Now we explain the SADHAN model in detail. The overall architecture is depicted in Figure 2 upper section. Given a training dataset of claims with their ground-truth labels, our goal is to learn a model based on the evidence from the relevant web documents D. To address the two challenges mentioned in Section 1, (1) we use a **hierarchical Bi-LSTM** model to capture the word-level and

[1] Subject, Author, Domain Based Hierarchical Attention Network

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

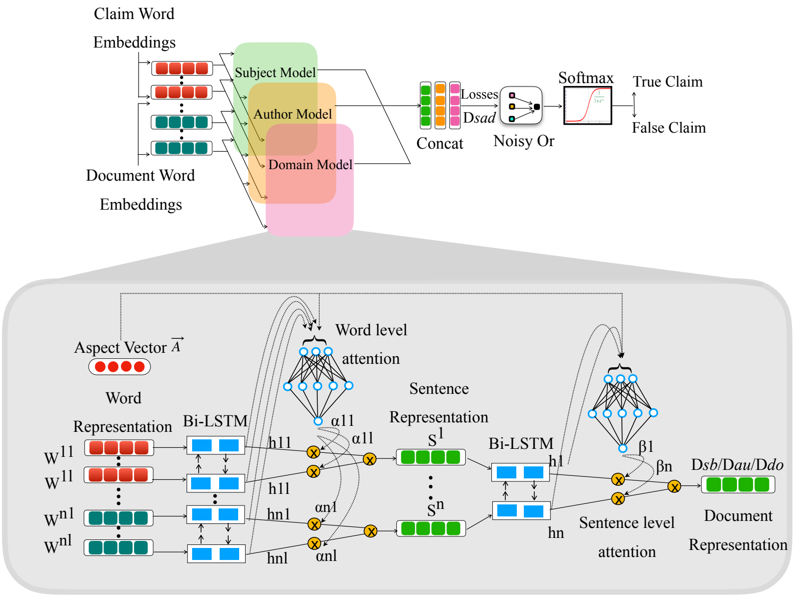


Figure 2: high-level architecture of SADHAN (upper part) and detailed hierarchical attention architecture of latent aspect models (lower part)

sentence-level structure of the documents, (2) An attention mechanism, which uses both claim text and latent aspect attribute vector to compute the attention, is then used to learn the embedding weights of the latent aspects. The intuition behind this design is that each of the latent aspect models jointly guide the attention to vocabulary and the sentences relevant for classifying claims. This architecture as we show in the experiments learns an effective model to identify complex patterns of false claims. For this purpose, SADHAN has different parallel models, one for each of the latent aspects. The detailed architecture of these models is shown in Figure 2 (zoomed in lower section). Specifically, we consider Subject, Author and Domain aspects in this paper but it is generalizable to any additional aspects of the claims and documents. At a high level, each claim-document pair $\{c, d\}$ is passed as the input to each of the three models, along with respective latent aspects. The outputs of these models are concatenated and passed to a fully connected softmax layer for prediction. Losses of all three models are aggregated using a noisy-or function. Finally, since our models operate on claim-document pairs, the classification of the claim c_i is done by the majority voting of outcomes corresponding to each of the $\{c, d\}$ pairs. We now explain the architecture of SADHAN in detail.

Embedding Layer: We use pretrained GloVe embeddings to get representations for each claim and document pair. We also create trainable embeddings for subject, author and domain attributes of 100 dimensions each in size and initialize with uniformly random weights to get the representation of latent attributes in vector space. We learn weights for these embeddings jointly in the model using corresponding hierarchical subject, author and domain attentions from their respective models as shown in Figure 2 (lower part). We concatenate each embedded claim c_i with the corresponding embedded document d_j, which is denoted as $\{c, d\}$. Each $\{c, d\}$ pair consists of n sentences of length l, which is depicted as word sequence w^{11} to w^{nl} in Figure 2 (lower part).

3.3 Latent Aspect Attention

Different authors, while making claims on different subjects, tend to have different styles of speech and selection of words. Similarly, writers and journalists from different domains may have unique style and vocabulary while writing about claims from a specific author and a specific subject. It is an extremely difficult task to curate the salient vocabulary and sentence structures for these complex combinations. Therefore, we automate this task using an attention mechanism which in turn helps in capturing entailment and sentiments necessary to classify the claim. For example, in tweets by Donald Trump words like “great”, “democrats” and “obama” are normally mentioned in specific context and sentiments, which our attention mechanism is able to capture.

Each claim and document pair $\{c, d\}$ is associated with a subject vector $\vec{A}_s$, author vector $\vec{A}_a$ and domain vector $\vec{A}_d$. These aspect vectors are used in addition to claim text to learn attention weights applied to hidden states at both word level and sentence level. The concatenated word embeddings of claim and document pair $\{c, d\}$ are passed on to a Bi-directional LSTM [1] unit which we use as word encoder, output from these Bi-LSTM units are concatenations of forward and backward hidden states for each word. h_{ij} is the hidden state for the i^{th} word of the j^{th} sentence. We compute values of attention weights α^{11} to α^{nl} by using single layer neural net with tanh activation, which uses encoded hidden states of claim and doc pair and aspect attribute vector $\vec{A}$ as input. We then multiply these attention weights α^{11} to α^{nl} with corresponding hidden states to select significant words, which are used to form sentence representations as s^1 to s^n. These sentence representations are then processed by another Bi-LSTM layer, which outputs hidden states h_1 to h_n for each sentence, as shown in the Figure 2(lower part). We compute values of attention weights β^1 to β^n by using another single layer neural net with tanh activation, which uses hidden states of sentences and aspect attribute vector $\vec{A}$ as input. We then multiply these attention weights β^1 to β^n with corresponding hidden states of sentences to select significant sentences, which are used to form document representations as $D_{sb}/D_{au}/D_{do}$ in case of subject, author or domain models correspondingly.

Subject Model: The words which are significant for a specific subject, can be used in various ways by different authors in claims and by different columnists or journalists in articles related to claims, therefore subject attention at the word level tries to learn and attend these words and at the sentence level tries to capture significant sentence formations used for the specific subject.

Author Model: Similar to the subject model, we use author guided aspect attention at word level to select author related words used in articles and sentence representations are learned by aggregating these words. We apply author guided aspect attention at the sentence level to select author specific sentence formations or popular phrases which are frequently used for a specific author and we get document representation D_{au} by aggregating these selected sentences.

Domain Model: Different domains in the web search results may have a unique way of writing articles like selection of words and sentence formations. In similar fashion to subject and author aspect attention, to attend different domains differently and to learn latent patterns, we apply domain guided aspect attention at the word and sentence level and get document representation D_{do}.
More formally, in all three models, sentence representation S^i after word sequence encoding by the Bi-LSTM is the weighted sum of the hidden states of words multiplied by attention weights. Similarly, document representation D is the weighted sum of hidden states of sentences multiplied by attention weights. These are defined as:

$$S^i = \sum_{j=1}^{l_i} \alpha_{ij} h_{ij} \text{ and } D = \sum_{i=1}^{n} \beta_i h_i$$

Where h_{ij} is the hidden state for the j^{th} word and i^{th} sentence. α_{ij} is the attention weight. h_i is the hidden state for i^{th} sentence and β_i is the attention weight. α_{ij} and β_i can be defined as:

$$\alpha_{ij} = \frac{\exp(e(h_{ij}, \vec{A}))}{\sum_{k=1}^{l_i} \exp(e(h^s_{ik}, \vec{A}))} \text{ and } \beta_i = \frac{\exp(e(h_i, \vec{A}))}{\sum_{k=1}^{n} \exp(e(h_k, \vec{A}))}$$

Where e is a *tanh* based scoring function which decides weights for significant words at the word level attention and for significant sentences at sentence level attention. $\vec{A}$ is the latent aspect vector, which is equal to subject vector $\vec{A_s}$ in subject model, author vector $\vec{A_a}$ in author model and domain vector $\vec{A_d}$ in case of domain model. $e(h_{ij}, \vec{A})$ and $e(h_i, \vec{A})$ can be defined as:

$$e(h_{ij}, \vec{A}) = (v_w)^T \tanh(W_{wh} h_{ij} + W_{wA} \vec{A} + b_w)$$

$$e(h_i, \vec{A}) = (v_s)^T \tanh(W_{sh} h_{ij} + W_{sA} \vec{A} + b_s)$$

Where v_w is weight vector at the word level and v_s is weight vector at the sentence level. W_{wh} and W_{wA} are the weight matrices for hidden state and aspect vector and b_w is bias at the word level respectively. W_{sh} and W_{sA} are the weight matrices for hidden state and aspect vector and b_s is bias at the sentence level respectively.

Algorithm 1: Evidence Extraction Algorithm

```
Input: Claim $c \in C$; Document $d \in D$; $W_{ws}$, $W_{wa}$, $W_{wd}$ are the
       word level and $W_{ss}$, $W_{sa}$, $W_{sd}$ are the sentence level attention
       weight matrices for subject, author and domain model
       respectively; $K$ is number of sentences in evidence snippet
Output: $E$, an evidence snippet for claim $c$
1  $S = []$ // Initialize an empty list
2  for each sentence $s_i$ in $d$ do
3      $W = []$ // Initialize an empty list
4      for each word $w_{ij}$ in $s_i$ do
5          $W.append((W_{ws}[i,j] + W_{wa}[i,j] + W_{wd}[i,j])/3)$
6      end
7      $W_{avg} \leftarrow sum(W)/len(W)$
         $S[i] \leftarrow W_{avg} + (W_{ss}[i] + W_{sa}[i] + W_{sd}[i])/3$
8  end
9  $indexes \leftarrow argsort(S)[-K:]$ // Get indices of top K elements
       from $S$
10 $E \leftarrow d[indexes]$ // Get sentences corresponding to indices from
       $d$
11 return $E$
```

3.4 Fusion of Models

Representations for each document D are learned from all three models as D_{sb} from subject model, D_{au} from author model and D_{do} from domain model. We concatenate these three representations for the same document and form an overall representation. $D_{sad} = D_{sb} \oplus D_{au} \oplus D_{do}$ We apply a non-linear transformation on overall document representation D_{sad} using tanh dense layer to transform it to binary target space. $D_{bin} = \tanh(W_{bin} D_{sad} + b_{bin})$ where W_{bin} and b_{bin} are the weight matrix and bias for dense layer. We apply a softmax layer to obtain the predictions for each class P_{bin} as $P_{bin} = softmax(D_{bin})$. Finally, we combine the losses of all three models with noisy-or gate as below:

$$Loss = 1 - ((1 - loss_o)) * (1 - loss_s) * (1 - loss_a) * (1 - loss_d))$$

where $loss_o$, $loss_s$,$loss_a$ and $loss_d$ are the losses for overall merged model, subject model, author model and domain model respectively.

3.5 Prediction Per Claim

The prediction outcomes for a claim c paired with each corresponding documents $\{d_1, ..., d_m\}$ are then aggregated by majority voting to assign a class to the claim.

$$\hat{y} = mode\{y_1, y_2, ...y_m\}$$

Where $\hat{y}$ is the final predicted label for claim c and $y_1, y_2, ...y_m$ are the predictions for pairs of claim c and corresponding m documents.

3.6 Evidence Extraction

In this section, we propose a technique to extract evidence snippets supporting or refuting the claim from documents using attention weights at both the word and sentence level from all three models. The pseudocode is shown in Algorithm 1. In line 5, for each word in each sentence of document d, we compute the average of attention weights given by all three models and this gives us overall attention weight for that word. In line 7, we compute the average of overall attention weights for all words in a sentence and add this value to the average of sentence level attention weights for the same sentence from all three models and store this value to list S. We get indices of top K values in S using *argsort* (line 9) and get the corresponding sentence indices from document d (line 10).

4 EXPERIMENTAL SETUP

4.1 Datasets

We use three datasets–Politifact and Snopes released by Popat et al [11] and Fever dataset released by Thorne et al [18].

Politifact Dataset. Politifact has 3568 claims and 29556 documents associated with 3028 domains retrieved from the web search using Bing search API. We discard articles related to fact-checking domains. For each claim, Politifact has one of these six ratings: 'true', 'mostly true', 'half true', 'mostly false', 'false' and 'pants-on-fire'. Similarly to DeClarE we combine 'true', 'mostly true' and 'half true' ratings to 'true' label and rest of them to 'false' label. There are 669 unique authors and 1400 topics in total.

Snopes Dataset. Snopes has around 4341 claims and 29242 documents associated with 3267 domains retrieved from the web using Bing search API. Similar to Politifact we discard all the documents which are from fact checking websites such as Snopes, Politifact, Factcheck and Emergent etc. For each claim, it has a credibility label as 'True' or 'False'.

Fever Dataset. While Fever dataset is not dedicated for fake news detection in itself, it is widely used for the entailment task, which can be viewed as a subtask of fake news detection. We use the fever dataset to illustrate that our model is also effective for the entailment task. This is to validate our hypothesis that our model performs well because it is also able to perform entailment task effectively. Fever dataset has 145449 claim-evidence pairs in train-set, 9999 claim-evidence pairs in development set and 9999 claim-evidence pairs in test set (for more details see [18]). In addition to what is already present in Fever dataset, we use Latent Dirichlet Allocation (LDA) to get the dominant topic for each claim in train, validation and test dataset as Fever dataset doesn't have any aspect attributes. We use the elbow method with topic coherence score to tune the number of topics K, as a result we use $K = 273$.

4.1.1 *Data Imbalance.*

Since Snopes and Politifact datasets have class imbalance, we balance them by setting the `class_weight` parameter to "balanced" in scikit–learn `compute_class_weight` API[3]. On the other hand Fever dataset is already balanced.

4.2 Baselines

We compare our model using several baselines both simple baselines and state-of-the-art techniques:

(1) Simple Convolutional Neural Network (CNN) model which was proposed for sentence classification [3]
(2) Hierarchical LSTM Network (Hi-LSTM) for document classification (without attention) [20]
(3) Self-attention based Hierarchical Attention Network BiLSTM (HAN) [20]
(4) DeClarE which applies claim-text based attention and source based embeddings [11]

To perform ablation testing for our SADHAN model, we incrementally introduce various latent aspect embeddings over the HAN architecture. We represent our models as SHAN, AHAN, and DHAN for each of the latent aspects Subject, Author and Domain respectively. Finally, SADHAN is our full model with all three aspects. Each of these models perform classification at the document level. DeClarE on the other hand performs classification on a per claim basis. Therefore in order to compare the performance of our model to DeClarE we also evaluate an aggregated version of our model represented as SADHAN-agg, which uses mean score from predictions of individual articles to assign a class to the claim.

4.3 SADHAN Implementation

We implement SADHAN using TensorFlow framework. We use 10 fold cross validation for all the models. We compute per-class accuracy, Macro F1 score and AUC as performance metrics for evaluation. We use pre-trained GloVe embeddings of 100 dimensions, trained on 6 billion words. We extract relevant snippets of text from the web documents using cosine similarity to include only highly relevant parts of the web documents. We try different sentence length sizes but we see no noticeable difference in performance. We tune the parameters [4] using a validation set, as a result we use softmax cross entropy with logits as the cost function, learning rate of 0.001 and size of hidden states and cell states of Bi-LSTM units are kept as 200. For drop out regularization we used keep-prob = 0.3. We chose these hyperparameters via grid search.

5 EXPERIMENTAL RESULTS

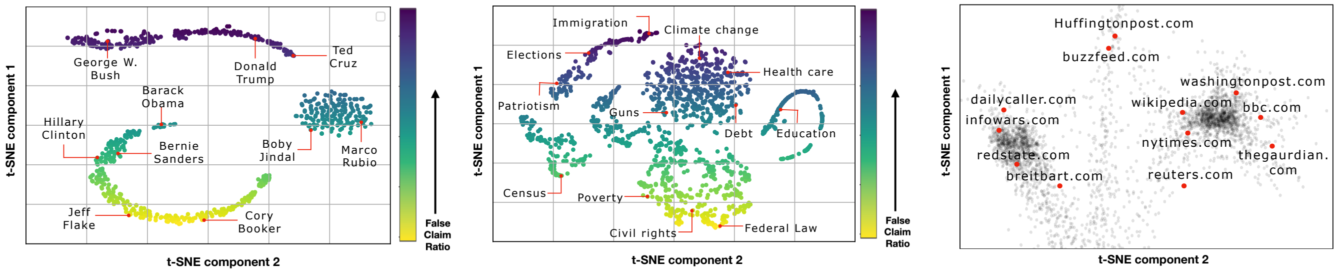
5.1 Results for Politifact Dataset

In Table 1 for Politifact dataset, in case of CNN, we get 59.39% Macro $F1$ accuracy and 58.56% as AUC. Hi-LSTM performs slightly better than CNN with 60.11% Macro $F1$ accuracy and 60.66% as AUC, though we get better false class accuracy with the Hi-LSTM. The reason for this improvement is that Hi-LSTM captures the inherent hierarchical structure of the documents. On the other hand HAN performs significantly better than Hi-LSTM with 64.80% Macro $F1$ accuracy and 64.54% as AUC and provides gain of 6.6% in Macro $F1$ over Hi-LSTM. The reason for this is because the documents retrieved from the web are fairly large even after extracting only relevant snippets using cosine similarity technique. It is hard for LSTM networks to memorize such long sequences. Moreover, LSTM with Attention mechanism only remembers attended words at word level and only attended sentences at sentence level.

As Politifact dataset has all aspect attributes such as subject, author and domain, we apply all individual models. Each of the SHAN, AHAN and DHAN models outperform HAN in Macro $F1$ with Macro $F1$ as 65.36%, 66.83% and 65.05% respectively. AHAN performs slightly better than the other two. This is due to the fact that the subject aspects in Politifact are generic. For each domain in domain attribute, we have high variance because each domain might have articles written by many different writers having different writing styles. The full SADHAN model outperforms all the other models with significant gain of 7.5% in Macro $F1$. This gain can be

[3] https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html

[4] https://github.com/rahulOmishra/SADHAN



(a) Author Embedding Visualization (b) Subject Embedding Visualization (c) Domain Embedding Visualization

Figure 3: Visualization of Latent Embeddings (The darker the color higher the false claim ratio)

Table 1: Comparison of proposed model with various state of the art baseline models for False claim detection on Snopes and PolitiFact datasets

Data	Model	True Acc.	False Acc.	Macro F1	AUC
PolitiFact	CNN	55.92	57.33	59.39	58.56
	Hi-LSTM	55.85	65.86	60.11	60.66
	HAN	60.32	68.20	64.80	64.54
	SHAN	62.29	68.43	65.36	65.23
	AHAN	63.25	70.42	66.83	68.66
	DHAN	60.34	69.76	65.05	65.03
	SADHAN	**69.79**	**75.45**	**71.34**	**72.37**
Snopes	CNN	72.05	74.29	72.63	76.45
	Hi-LSTM	74.21	74.16	74.33	79.20
	HAN	76.76	79.65	77.80	80.33
	DHAN	**77.06**	**81.63**	**78.73**	**82.03**

Table 2: Comparison of proposed model with DeClarE models for False claim detection on Snopes and PolitiFact datasets. SADHAN-agg is statistically significant ($p - value = 1.05e^{-4}, 2.45e^{-2}$ for Politifact and Snopes respectively using pairwise student's t-test)

Data	Model	True Acc.	False Acc.	Macro F1	AUC
PolitiFact	DeClarE (full)	68.18	66.01	67.10	72.93
	SADHAN-agg	**68.37**	**78.23**	**75.69**	**77.43**
Snopes	DeClarE (full)	60.16	80.78	70.47	80.80
	DHAN-agg	**79.47**	**84.26**	**80.09**	**85.65**

attributed to fusion of three models, which considers all aspects of the claim and document pair for classification.

5.2 Results for Snopes Dataset

For Snopes, we can see in Table 1 that Hi-LSTM with 74.33% Macro $F1$ accuracy and 79.20% as AUC outperforms CNN with 72.63% Macro $F1$ accuracy and 76.45% as AUC by 2.7% in Macro $F1$ and similar to Politifact results, this gain is also attributed to better representation learned in the form of the hierarchical structure of the documents by Hi-LSTM. HAN with 77.80% Macro $F1$ accuracy and 80.33% as AUC gives further gain of 4% on top of Hi-LSTM, due to hierarchical attention at word and sentence level. Since Snopes dataset has only domain attribute, we only use (DHAN) with 78.73% Macro $F1$ accuracy and 82.03% as AUC, which outperforms all the baseline methods and gives gain of 1.2% over HAN.

5.3 Evaluation of claim-level classification

Since DeClarE classifies claims rather than individual documents, we compare aggregated model SADHAN-agg with DeClarE (full) model which applies only claim-text based attention in Table 2.

For Politifact data, SADHAN-agg outperforms DeClarE (full) model by 12% in micro $F1$. We attribute these gains to the latent aspect level attention which is able to capture the context better. While only claim-text based attention learns to attend the words having connotation with claim at word level only.
For Snopes dataset, DHAN-agg with 80.09% Macro $F1$ accuracy and 85.65% as AUC outperforms DeClarE (full) model with 70.47% Macro $F1$ accuracy and 80.80% as AUC by 13.5% in micro $F1$. We attribute these gains to the usage of domain aspect attribute in addition to claim-text for attention computation.

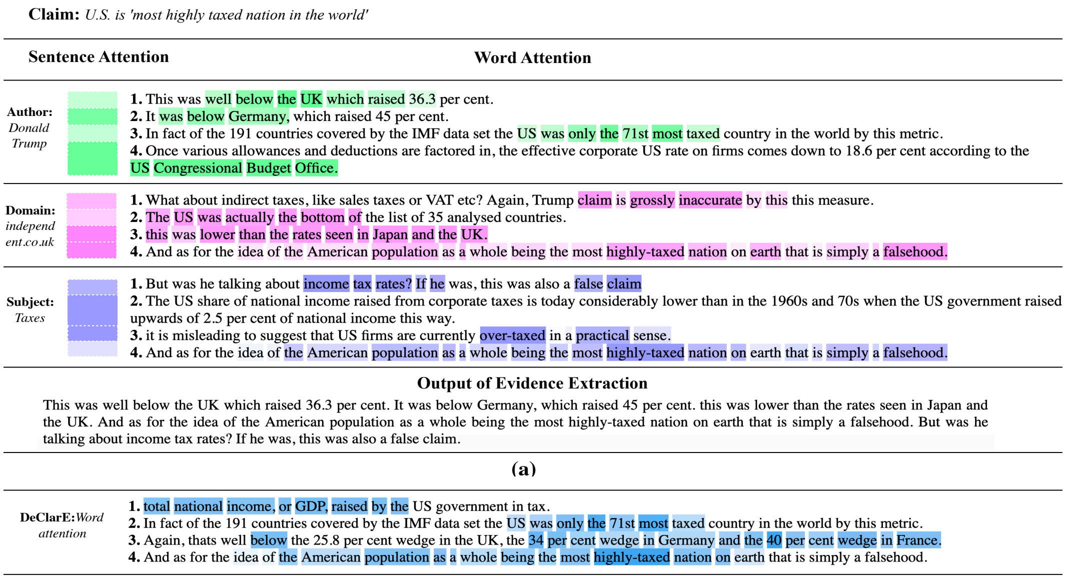
5.4 Results for Fever Dataset

We used Fever dataset to investigate the effectiveness of our model for the textual entailment task. Since Fever data doesn't have any of the three subject, author or domain attributes, we use Latent Dirichlet Allocation (LDA) to get the dominant topic for each claim therefore we apply SHAN model for textual entailment. We get 79.20% accuracy ($p - value = 3.62e^{-4}$ in pairwise student's t-test) with the testset and 83.09% accuracy with devset provided with Fever dataset, which outperforms multi-layer perceptron (MLP) with 73.81% accuracy (Riedel et al. 2017)[15] method used by authors of Fever dataset paper [18], which uses single hidden layer with TF-IDF vector based cosine similarity between the claim and evidence. On the other hand SHAN model could not outperform the decomposable attention model in [8] with 88.0% accuracy. We hypothesize that this is because the derived dominant topics learned for claims using LDA topic model may not be a true representation of original topics of claims. We could improve the performance by using more concrete set of topics such as categories from Wikipedia.

6 DISCUSSION

In this section we analyze the effectiveness of latent aspect embeddings learned by our model and illustrate the interpretability of our model with the help of evidence extraction and attention visualization. We compare snippets extracted by our model to the attention visualization of DeClarE using anecdotal examples.

Author Embeddings: We use t-Distributed Stochastic Neighbor Embedding (t-SNE) to visualize author embeddings in lower dimensional space. We plot only two dimensions from t-SNE with tuned parameters ($perplexity = 10$, $learningrate = 0.1$ and $iterations = 2000$). We show the fraction of false claims associated with each author using a color gradient (cf. Figure 3). As we can see in the plot that the authors having a higher number of false claims are



Claim: *U.S. is 'most highly taxed nation in the world'*

Sentence Attention	Word Attention
Author: *Donald Trump*	**1.** This was well below the UK which raised 36.3 per cent. **2.** It was below Germany, which raised 45 per cent. **3.** In fact of the 191 countries covered by the IMF data set the US was only the 71st most taxed country in the world by this metric. **4.** Once various allowances and deductions are factored in, the effective corporate US rate on firms comes down to 18.6 per cent according to the US Congressional Budget Office.
Domain: *independent.co.uk*	**1.** What about indirect taxes, like sales taxes or VAT etc? Again, Trump claim is grossly inaccurate by this this measure. **2.** The US was actually the bottom of the list of 35 analysed countries. **3.** this was lower than the rates seen in Japan and the UK. **4.** And as for the idea of the American population as a whole being the most highly-taxed nation on earth that is simply a falsehood.
Subject: *Taxes*	**1.** But was he talking about income tax rates? If he was, this was also a false claim **2.** The US share of national income raised from corporate taxes is today considerably lower than in the 1960s and 70s when the US government raised upwards of 2.5 per cent of national income this way. **3.** it is misleading to suggest that US firms are currently over-taxed in a practical sense. **4.** And as for the idea of the American population as a whole being the most highly-taxed nation on earth that is simply a falsehood.

Output of Evidence Extraction

This was well below the UK which raised 36.3 per cent. It was below Germany, which raised 45 per cent. this was lower than the rates seen in Japan and the UK. And as for the idea of the American population as a whole being the most highly-taxed nation on earth that is simply a falsehood. But was he talking about income tax rates? If he was, this was also a false claim.

(a)

DeClarE: *Word attention*	**1.** total national income, or GDP, raised by the US government in tax. **2.** In fact of the 191 countries covered by the IMF data set the US was only the 71st most taxed country in the world by this metric. **3.** Again, thats well below the 25.8 per cent wedge in the UK, the 34 per cent wedge in Germany and the 40 per cent wedge in France. **4.** And as for the idea of the American population as a whole being the most highly-taxed nation on earth that is simply a falsehood.

(b)

Figure 4: Example 1: Comparison of SADHAN evidence extraction with DeClarE for the claim "U.S. is 'most highly taxed nation in the world'"

clearly separated from authors having a lower number of false claims. Interestingly we also notice the formation of a third cluster, which is related to the authors, who have almost equal number of false claims and true claims. This is also very interesting to see that people of similar ideology like 'Obama', 'Hillary' and 'Sanders' are closer in embedding space. This is evident by the visualization that the author based attention can distinguish very effectively between the authors with less connotation of false claims and the authors with high connotation of false claims, which in-turn helps in deciding the credibility of claims.

Subject Embeddings: Similarly, we plot two dimensions from t-SNE with tuned parameters ($perplexity = 20$, $learningrate = 1.0$ and $iterations = 3000$) to visualize the subject embeddings (cf. Figure 3(b)). We can observe in the plot that the subjects with low and high false claim ratios are separated clearly into clusters. Due to the coarser granularity of the subjects, the separation is not as pronounced as author embeddings. It is however, quiet insightful to see that the topics like 'Climate change' and 'Health care' have very high percentage of false claims and are closer in the two-dimensional space. While 'Federal law' which has very low associated false claims is far away from them.

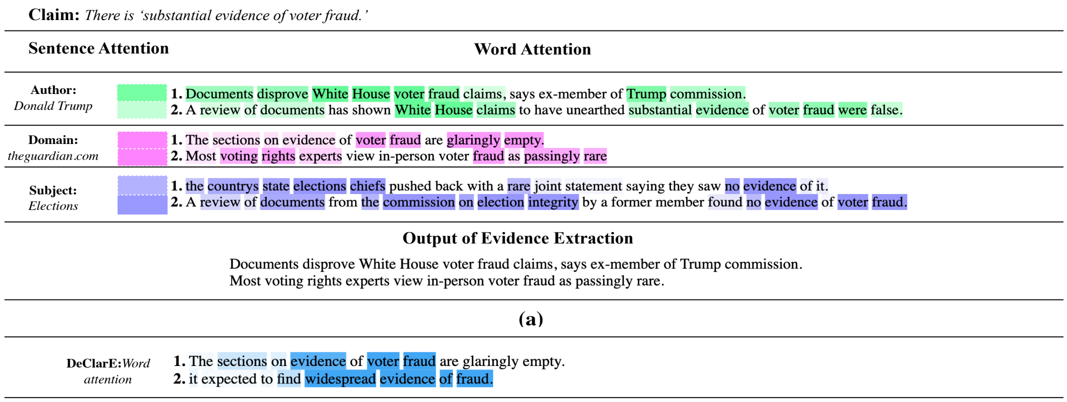
Domain Embeddings: For domain embeddings, we use t-SNE with tuned parameters ($perplexity = 20$, $learningrate = 0.1$ and $iterations = 2000$) to plot two dimensions (Figure 3(c)). Notice that the domain embeddings clearly separate trustworthy domains like 'washingtonpost.com', 'nytimes.com' etc. from non-trustworthy domains like 'inforwars.com' and 'dailycaller.com', making the learned domain embeddings good detectors of fake news.

6.1 Attention Visualization

In this section, we visualize the attention weights for two anecdotal examples (claim and document pairs), both at the word and sentence level for all three models and compare with state-of-the-art DeClarE model in Figure 4 and 5. The depth of the colors in rectangle boxes next to each sentence, represents the distribution of attention weights at the sentence level. Similarly depth of the color of highlights of the words represents the distribution of attention weights at the word level. For all the three models only top 4 sentences in Figure 4 and top 2 sentences in Figure 5 based on both word and sentence level attention weights are shown. As in each of the three models we use both claim and document text on top of aspect attributes to compute attention therefore we get some common trends in both word level and sentence level attention for all three models. Due to usage of different aspect attributes namely subject, author and domain in different models for attention computation, we get very interesting and relevant words and sentences selected in all three, which is not possible otherwise.

As we can see in Figure 4(a), for a claim related to Donald Trump that "U.S. is the most highly taxed nation in the world'", we apply our model to detect if it's true or false. We use a document extracted from the web for which domain is "independent.co.in", author is "Donald Trump" and subject is "Taxes". In author model, we can observe that in Figure 4(a) first row, author based attention is able to capture words like "below Germany", "below the UK" and "Congressional Budget" other than claim oriented words like 'US' and 'Taxed' etc, as these words are highly correlated with the author "Donald Trump" as 'Germany', 'UK' and 'Congressional' are some of the frequent words used by 'Donald Trump' or can be found in the articles related to him.

In similar fashion in domain model in Figure 4(a) second row, domain based attention is able to capture words 'grossly inaccurate' and 'falsehood' and in Figure 5(a) second row, words like 'glaringly empty' and 'passingly rare', which are otherwise not possible to get attended with just claim only attention. As many articles from same domain, might be written by the same columnist or journalist and hence domain attention tries to capture their writing style and usage of specific phrases or words.



Claim: *There is 'substantial evidence of voter fraud.'*

Sentence Attention **Word Attention**

Author: *Donald Trump*

1. Documents disprove White House voter fraud claims, says ex-member of Trump commission.
2. A review of documents has shown White House claims to have unearthed substantial evidence of voter fraud were false.

Domain: *theguardian.com*

1. The sections on evidence of voter fraud are glaringly empty.
2. Most voting rights experts view in-person voter fraud as passingly rare

Subject: *Elections*

1. the countrys state elections chiefs pushed back with a rare joint statement saying they saw no evidence of it.
2. A review of documents from the commission on election integrity by a former member found no evidence of voter fraud.

Output of Evidence Extraction

Documents disprove White House voter fraud claims, says ex-member of Trump commission.
Most voting rights experts view in-person voter fraud as passingly rare.

(a)

DeClarE:*Word attention*

1. The sections on evidence of voter fraud are glaringly empty.
2. it expected to find widespread evidence of fraud.

(b)

Figure 5: Example 2: Comparison of SADHAN evidence extraction with DeClarE for the claim "There is substantial evidence of voter fraud"

In case of subject model in Figure 4(a), subject based attention learns to attend words and sentences which are related to the subject. As we can see 'Taxes' as subject captures words 'over-taxed' and 'income tax ' etc but also at the sentences level, it is able to capture very interesting sentences like sentence 2. In case of DeClarE model however, the model is unable to attend the most important words and sentences except few, like in sentence 4, though it attends words like 'highly taxed nation' etc but fails to attend word 'falsehood' as we can see in Figure 4(b). As DeClarE model doesn't have sentence level attention, it's therefore not able to use the evidence provided by sentence 4 to decide the appropriate label.

Finally, we show a snippet extracted by our evidence extraction algorithm in Figure 4(a) fourth row and 5(a) fourth row. The value of K is 5 in Figure 4(a) and 2 in 5(a), which means snippet contains top 5 sentences and top 2 sentences based on our evidence extraction method. It is evident that such a sentence extraction technique can be really effective in case of extractive text summarization tasks.

7 CONCLUSIONS AND FUTURE WORK

In this paper we presented an hierarchical attention mechanism to jointly learn various latent aspect embeddings for news. For example, these latent aspects can be subject, author and domain related to the claim and news articles. This allows us to capture salient vocabulary and complex structure at the document level. Compared to the only claim-text based attention, the attention weights, which are jointly learned guided by both claim text and different latent aspects are more effective for detecting if the claims are True or False. This is apparent from our experiments conducted on Snopes, Politifact and Fever dataset. We also propose an algorithm to extract evidence snippets supporting or refuting the claim from news articles using attention weights at both the word and sentence level from all three models. We show a t-SNE visualization that the learned embeddings are also good predictors of fake news. We also show examples where the evidence extracted using our latent aspect embeddings are superior to simple word level attention used in DeClarE. In future, we plan to conduct a detailed user study on the informativeness and interpretability of these evidence snippets.

REFERENCES

[1] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional LSTM networks for improved phoneme classification and recognition. In *ANN*. Springer, 799–804.

[2] Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. 2014. TweetCred: Real-Time Credibility Assessment of Content on Twitter. In *SocInfo*. 228–243.

[3] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[4] Yunfei Long, Qin Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang. 2017. Fake news detection through multi-perspective speaker profiles. In *IJCNLP*, Vol. 2. 252–256.

[5] 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.

[6] Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor Detection on Twitter with Tree-structured Recursive Neural Networks. *ACL* (2018), 1980–1989.

[7] Jing Ma, Wei Gao, and Kam-Fai Wong. 2019. Detect Rumors on Twitter by Promoting Information Campaigns with Generative Adversarial Learning. February (2019).

[8] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. (2016). arXiv:1606.01933

[9] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2017. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104* (2017).

[10] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2017. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In *WWW*. 1003–1012.

[11] Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning. In *EMNLP*. 22–32.

[12] Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A Stylometric Inquiry into Hyperpartisan and Fake News. In *ACL*, Vol. 1. 231–240.

[13] Feng Qian, Chengyue Gong, Karishma Sharma, and Yan Liu. 2018. Neural User Response Generator : Fake News Detection with Collective User Intelligence. *Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)* (2018), 3834–3840.

[14] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *EMNLP*. 2931–2937.

[15] Benjamin Riedel, Isabelle Augenstein, Georgios P. Spithourakis, and Sebastian Riedel. 2017. A simple but tough-to-beat baseline for the Fake News Challenge stance detection task. (2017), 1–6. arXiv:1707.03264

[16] 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.

[17] Kai Shu, Suhang Wang, and Huan Liu. 2019. Beyond News Contents: The Role of Social Context for Fake News Detection. In *WSDM*. ACM, 312–320.

[18] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355* (2018).

[19] Liang Wu and Huan Liu. 2018. Tracing Fake-News Footprints: Characterizing Social Media Messages by How They Propagate. In *WSDM*. 637–645.

[20] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL: HLT*. 1480–1489.

[21] Wenpeng Yin and Dan Roth. 2018. TwoWingOS: A Two-Wing Optimization Strategy for Evidential Claim Verification. In *EMNLP*. 105–114.