Fake News Detection

A Thesis

Presented to

The Academic Faculty

by

ARJUN ROY

(1711MC02)

under the guidance of

Prof. Dr. Pushpak Bhattacharyya,

Dr. Asif Ekbal,

and

Prof. Dr. Stefan Dietze

In Partial Fulfilment

of the Requirements for the M.Tech Degree of

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY PATNA.

July, 2019

In the Loving Memory of my Grand Mother (Thakurmaa) Late. Smt. Amulya Rani Roy.

Your Love and Blessings will always stay with me.

THESIS CERTIFICATE

This is to certify that the thesis titled **Fake News Detection**, submitted by **ARJUN ROY**, to the Indian Institute of Technology, Patna, for the award of the degree of **Master of Technology**, is a bonafide record of the research work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Dr. PUSHPAK BHATTACHARYYA,

Research Guide,
Professor,
Dept. of Computer Science
and Engineering,
IIT-Patna, 801 103,

Prof. Dr. STEFAN DIETZE,

Research Guide,
Professor,
L3S Research Center,
LUH, 30167 Hannover,

Asif Ekbal

Dr. ASIF EKBAL,

Research Guide,
Associate Professor,
Dept. of Computer Science
and Engineering,
IIT-Patna, 801 103,

Place:

Date:

DECLARATION

I certify that

a. The work contained in this thesis is original and has been done by myself under the general supervision of my supervisor.

b. The work has not been submitted to any other Institute for degree of diploma.

c. I have followed the Institute norms and guidelines and abide by the regulation as given in the Ethical Code of Conduct of the Institute.

d. Whenever I have used materials (data, theory and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the reference section.

Arjun Roy

1711MC02

CERTIFICATE OF APPROVAL

This is certified that the thesis entitled **Fake News Detection System**, submitted by **ARJUN ROY** to Indian Institute of Technology Patna, for the award of the degree of M.Tech has been accepted by the examination committee and that the student has successfully defended the thesis in the viva-voce examination held today.

Asif Ekbal

(Supervisor) **(Supervisor)** **(Supervisor)**

(External Examiner) **(Internal Examiner)**

ACKNOWLEDGEMENTS

I express my sincere gratitude to my supervisor and career idol, Prof. Dr. Pushpak Bhattacharyya for giving me the opportunity to be part of his highly qualified research group and providing me his invaluable guidance and suggestions during my M.tech thesis work at IIT Patna. I believe the experience of getting to learn from him and share his ideas and knowledge would reflect throughout my career. I would also like to thank Prof. Dr. Stefan Dietze for giving me the opportunity to be part of his research group in L3S, Germany, giving me his insightful comments during my work there, and letting me get the opportunity to experience the collaborative research work between India and Germany under DAAD scholarship program. I would also take the opportunity to express my warmest regards and thanks to Dr. Pavlos Fafalios for guiding me with his knowledge and giving me his support in every adverse condition during the research work of Chapter 3 in Germany. Next, I would like to appreciate Aman Shah, and Shubhankar Saha for their participation and effort during the work of Chapter 4, as part of their internship work at IIT Patna.

I am grateful to get into the proximity of Dr. Asif Ekbal. Throughout my M.Tech period he has been more than a guide, a parenting figure in my life, showering his love and support, and nurturing my research ability every single day.

Finally, I would take the opportunity to express my love and gratitude to my mom Smt. Gouri Roy, my dad Sri Samarendra Roy, and Swati for being there for me each and every time throughout my highs and lows.

ABSTRACT

Fake news and misinformation are emerging as one of the most important problems of modern times, influencing social, political, economic, communal harmony of human life. Owing to their popularity and reach, misinformation when spread by politicians hidden in between some true information, is studied to have a great effect in manipulating people's choice. But in this age of social and other online media, fake news and claims coming from even unknown and not so popular sources may also penetrate deep into the society, affecting a wide range of audience. Fact verification journalism in this context comes in very useful to debunk such fake claims. Thus, identifying documents which express an opinion towards such claims is also a necessary task to assist fact verification, as they may provide evidence to debunk fake claims. The problem of fake news further gets more complex when multimodal contents and multilingual text are used to spread them, and even becomes difficult for fact verification journalism to timely address them. Considering these three different variants of the problem we propose individual novel approaches to tackle each of them. Particularly, we use an ensemble of deep neural network based approach for the first problem, an hierarchical three-stage pipeline method for solving the second problem, and a multimodal multilingual content based approach for the tackling the third problem of our work. Evaluation on respective benchmark dataset shows the effectiveness of our proposed methods.

[Keywords: Fake news, misinformation, fact verification, multimodal]

TABLE OF CONTENTS

ACKNOWLEDGEMENTS **iii**

ABSTRACT **iv**

LIST OF TABLES **viii**

LIST OF FIGURES **x**

1 Introduction **1**

2 A Deep Ensemble Framework for Fake News Detection and Multi-Class Classification of Short Political Statements **4**

2.1 Introduction 4

2.1.1 Problem Definition and Motivation 5

2.1.2 Contributions 6

2.1.3 Related Work 6

2.2 Methodology 9

2.2.1 Bi-LSTM 10

2.2.2 CNN 12

2.2.3 Combined CNN and Bi-LSTM Model 12

2.3 Data 13

2.4 Experiments and Results 14

2.4.1 Experimental Setup 14

2.4.2 Results and Analysis 14

2.5 Conclusion and Future Works 18

3 Step-by-Step: A three-stage Pipeline for Document Stance Classification towards Claims **20**

3.1 Introduction 20

3.2 Problem Modeling and Approach Overview 22

3.3 Pipeline Implementation 24

3.3.1 Stage 1: Relevance Classification 24

3.3.2 Stage 2: Neutral/Stance Classification 26

3.3.3 Stage 3: Agree/Disagree Classification 28

3.4 Evaluation 29

3.4.1 Evaluation Setup 29

3.4.2 Evaluation results 32

3.5 Related Work 36

3.6 Conclusion 38

4 MulCoB-MulFaV: Multimodal Content Based Multilingual Fact Verification 40

4.1 Introduction 40

4.2 Problem Modeling and Approach Overview 42

4.3 Implementation 44

4.3.1 First Level : Related News Mining 44

4.3.2 Second Level : Veracity Classification 45

4.4 Evaluation 50

4.4.1 Dataset 50

4.4.2 Evaluation Setup 52

4.4.3 Evaluation results 53

4.4.4 Error Analysis 56

4.5 Related Works 58

4.6 Conclusion 60

5 Conclusion and Future Direction 61

LIST OF TABLES

2.1 Example statement of each class. 6

2.2 Meta-data related to each example. P, F, B, H, M is speaker's previous count of Pants-fire, False, Barely-true, Half-true, Mostly-true respectively. 7

2.3 Overall evaluation results 14

2.4 Evaluation of Bi-LSTM model: precision, recall, and F1 score . . . 15

2.5 Evaluation of CNN model: precision, recall, F1 score 15

2.6 Evaluation of Bi-LSTM, CNN combined model: precision, recall, F1 score 15

2.7 Sample text with wrongly predicted label and original label. Spk is speaker, and P, F, B, H, M is speaker's previous count of Pants-fire, False, Barely-true, Half-true, Mostly-true respectively. 16

2.8 Confusion matrix of the Bi-LSTM model 18

2.9 Confusion matrix of the CNN model 18

2.10 Confusion matrix of the Bi-LSTM+CNN combined model 18

3.1 Data distribution of the FNC-I dataset. 29

3.2 Class distribution for *relevance* classification. 30

3.3 Class distribution for *discuss/stance* classification. 30

3.4 Class distribution for *agree/disagree* classification. 30

3.5 Document stance classification performance. 33

3.6 Class-wise performance of the different pipeline stages and of the entire pipeline system. 34

3.7 Confusion matrix of our pipeline system. 35

3.8 Confusion matrix of stage 1. 36

3.9 Confusion matrix of stage 2. 36

3.10 Confusion matrix of stage 3. 36

4.1 Class-wise data distribution of the all dataset. D1 is Dataset1, D2 is Dataset2, and D3 is Dataset3. 52

4.2 Comparative evaluation of our proposed system against all the baseline methods. D1, D2, and D3 are Dataset1, Dataset2, and Dataset3 respectively as described in sec.4.4.1. 53

4.3 Images shared in different fake reports. The column Distortion narrates whether the image is a doctored/ distorted image or not. Column Evidence highlights the facts based on which our model *MulCoB-MulFaV* accurately judged the veracity of the report. 55

4.4 Detailed performance of our proposed model on the three dataset D1, D2, and D3. P, R, and F1 are precision, recall, and F1-score measures respectively. 56

4.5 Examples of wrongly predicted reports by *MulCoB-MulFaV*. *Tar* is target/original label, and *Pred* is predicted label. 57

4.6 Confusion matrix of *MulCoB-MulFaV* on D2 and D3. 58

LIST OF FIGURES

2.1 A relationship network layer. A_x and A_y are two attributes, M_i and M_j are two individual models, $Network_n$ is a representation of a network capturing a relationship 11

2.2 Deep Ensemble architecture 11

3.1 Document stance hierarchy. 23

3.2 Pipeline for document stance classification and difficulty level of each stage. 24

3.3 Diagram outlining the SVM model used in stages 1 and 3. 25

3.4 Diagram outlining the CNN model used in Stage 2. 27

4.1 Given PROBLEM MODEL. 43

4.2 Flowchart diagram for Fake report detection showing sequences of job. r, c, i, and nr_j are given report, textual claim, image, and jth related news report respectively. 43

4.3 Flowchart Diagram of the procedure of mining related news reports. 44

4.4 Diagram showing embedding of text using two different pre-trained word vectors. 46

4.5 Embedded Vectors of word 'Fake' using Glove Wikipedia vecs vs Glove Tweeter vecs in 100d space 47

4.6 Diagram to show generation of priority based image vector p_r. . . . 48

4.7 Diagram to show formation of attended tensor using representation vector Re_{nr} and embedded claim text feature vectors E_c. 49

4.8 Diagram showing formation of evidence vector vec_e and final prediction of veracity of the report based on vec_e. 49

4.9 Bar Graph showing distribution of posts across all the 17 events in MediaEval15 dataset1. Here 1: **Boston Marathon Bombing**, 2: **Bring-back Our girls**, 3: **Columbian Chemicals**, 4: **Rock Elephant**, 5: **Livr mobile app**, 6: **MA flight 370**, 7:**Passport hoax**, 8:**Pig Fish**, 9:**Hurricane Sandy**, 10:**Sochi Olympics**, 11:**Underwater bedroom**, 12:**Solar Eclipse**, 13:**Garissa Attack**, 14:**Nepal Earthquake**, 15:**Girl with Samurai boots**, 16:**Syrian boy**, 17:**Varoufakis and zdf** 50

4.10 Bar Graph showing distribution of posts across all the 36 events in test set of MediaEval16. Here 1: **Gandhi Dancing**, 2: **Half of Everything**, 3: **Hubble Telescope**, 4: **ImmigrantsâĂŹ fear**, 5: **ISIS children**, 6: **John Guevara**, 7: **Mc DonaldsâĂŹ Fee**, 8: **Nazi Submarine**, 9: **North Korea**, 10: **Not Afraid**, 11:**Pakistan Explosion**, 12:**Pope Francis**, 13:**Protest**, 14:**Refugees**, 15:**Rio Moon**, 16:**Snowboard Girl**, 17:**Soldier Stealing**, 18: **Syrian Children**, 19: **Ukrainian Nazi**, 20: **Woman 14 children**, 21: **American Soldier Quran**, 22: **Airstrikes**, 23: **Attacks in Paris**, 24 :**Ankara Explosions**, 25: **Bush book**, 26: **Black Lion**, 27: **Boko Haram**, 28: **Bowie David**, 29: **Brussels Car Metro**, 30: **Brussels Explosions**, 31: **Burst in KFC**, 32: **Convoy Explosion Turkey**, 33: **Donald Trump Attacker**, 34: **Eagle Kid**, 35: **Five Headed Snake**, 36: **Fuji Lenticular Clouds** 51

CHAPTER 1

Introduction

In the field of automated Information Retrieval and Natural Language Processing **Fake News Detection** has emerged as one of the most important problem domain. Fake news can be defined as any natural language short statement, or textual claim, or even a multimodal (having content of different modality) report which is circulated with the intent of spreading misinformation with malicious motives behind it. Spread of fake news as an effect can disrupt and manipulate communal[1], social[2], political[3] harmony , and often has been seen as a tool of influencing political elections [1]..

This thesis work is aimed to solve the problem of fake news in all possible directions. At first, we look into the problem of short political statements made during a speech, interview, etc. about various context, venues, states, etc. by politicians belonging to different ideology (left and right wing) and holding different posts. As it is also possible to keep stats about the number of times a politician has previously lied or said something true, this knowledge is also incorporated in our system. It is also noticed that politician do not always through straight forward lies, but tend to deceive misinformation cleverly under a mask of true information. Thus, binary classification of such statements into plain true and false is not a correct solution, rather it requires to categorized into carefully grained multiple classes based on the content of misinformation present in the statement. Acknowledging all these trails, we try to make use of the meta-information from attributes such as venue, context, etc., along with the statement, try to find whether all these knowledge bears any pattern of relation among the attributes, and based on our findings, we finally categorize each of such political short statements into fine-grained multiple classes.

Next, the attention is shifted to textual claims, that are being circulated in various online platforms, and may come from any source. To control the propagation of fake information through such claims, over the years, fact-checking journalism has emerged

[1] bashirhat-riot
[2] Columbian Chemical hoax
[3] misinformation-on-indo-pak-tension-flood-the-social-media

as a viable solution. In this type of procedure identifying related documents from the web that support or negate a particular claim is an essential task. These documents can provide useful information and evidence which may lead to the debunking of fake claims. But, due to the enormous number of documents present in the web (mostly either uninformative or just discusses the claim without an opinion), it is not always possible for the journalists to go through all such documents to identify their stance towards such claims. In addition, a timely debunking of fake claims is also very important to restrict its effect to a minimal reach. Thus, automation of the process of stance detection of documents toward textual claims is a necessary objective. However, existing (4-class) stance classification approaches are ineffective in detecting documents that negate a claim, even though these documents provide crucial evidence when aiming to detect false claims. We identify this need and attempt to provide an automated solution of identifying stance of documents toward claims, particularly keeping our focus to identify documents that express an opinion.

Recent studies have shown that false claims spread faster, further, and for longer than true claims [2]. But fake claims when made with the support of false or doctored multimedia evidence tends to reach a wider audience with greater convincing ability [3]. Additionally, these false claims are posted in local languages, in general, to increase its reach further. Due to the recent rise of online media platform by people as their main source of information the amount of misinformation is growing exponentially. On the other hand, the journalist needs to manually check in all the related reporting documents that take a stance towards the claim report, to find evidence. Given the enormity of the scale of misinformation, language diversity the existing mechanism of fact-checking fails to address the problem to a reasonable extent. Thus, creating the need for automation of the entire fact-checking journalism procedure. This type of information retrieval falls under the category of content-based information retrieval (CBIR). Even though content based works exist while working with only textual data to verify a fact, but no prior content-based work exists which works towards multimodal fact verification. Acknowledging this necessity, we work towards developing a content-based fact verification system that handles multimodal reports containing multilingual textual claims.

Summarizing in brief, the contributions made in this thesis work can be listed as follows:

- A novel approach to categorize short political statements into fine-grained multi-classes of news (from True to completely fake), by capturing relationships among various meta-information attributes along with the statement.
- A novel approach for the 4-class stance classification problem of news article documents towards textual claims with a dedicated focus to identify documents that express an opinion towards the claim.
- A novel multimodal content-based approach towards fact verification of multi-modal multilingual reports with the aim to automate the manual process of fact verification journalism.

The structure of the rest of this thesis work is arranged as follows:

Chapter 2 discusses the work towards solving the problem of fake detection and classification of short political statements.

Chapter 3 explores the problem of document stance classification towards textual claim. This is a part of the collaborative work between AI-ML-NLP lab (IIT Patna) of India, and L3S research center (Leibniz University Hanover) of Germany.

Chapter 4 identifies the problem of multimodal multilingual fake reports and gives content based solution approach towards solving it.

Finally, Chapter 5 concludes the work and discusses the future direction.

CHAPTER 2

A Deep Ensemble Framework for Fake News Detection and Multi-Class Classification of Short Political Statements

Fake news, rumor, incorrect information, and misinformation detection are nowadays crucial issues as these might have serious consequences for our social fabrics. Such information is increasing rapidly due to the availability of enormous web information sources including social media feeds, news blogs, online newspapers etc. In this paper, we develop various deep learning models for detecting fake news and classifying them into the pre-defined fine-grained categories. At first, we develop individual models based on Convolutional Neural Network (CNN), and Bi-directional Long Short Term Memory (Bi-LSTM) networks. The representations obtained from these two models are fed into a Multi-layer Perceptron Model (MLP) for the final classification. Our experiments on a benchmark dataset show promising results with an overall accuracy of 44.87%, which outperforms the current state of the arts.

2.1 Introduction

> "We live in a time of fake news- things that are made up and manufactured." Neil Portnow.

Fake news, rumors, incorrect information, misinformation have grown tremendously due to the phenomenal growth in web information. During the last few years, there has been a year-on-year growth in information emerging from various social media networks, blogs, twitter, facebook etc. Detecting fake news, rumor in proper time is very important as otherwise, it might cause damage to social fabrics. This has gained a lot of interest worldwide due to its impact on recent politics and its negative effects. In fact, Fake News has been named as 2017's word of the year by Collins dictionary[1].

[1] http://www.thehindu.com/books/fake-news-named-word-of-the-year-2017/article19969519.ece

Many recent studies have claimed that US election 2016 was heavily impacted by the spread of Fake News. False news stories have become a part of everyday life, exacerbating weather crises, political violence, intolerance between people of different ethnics and culture, and even affecting matters of public health. All the governments around the world are trying to track and address these problems. On 1^{st} Jan, 2018, bbc.com published that ”Germany is set to start enforcing a law that demands social media sites move quickly to remove hate speech, fake news, and illegal material.” Thus it is very evident that the development of automated techniques for detection of Fake News is very important and urgent.

2.1.1 Problem Definition and Motivation

Fake News can be defined as completely misleading or made up information that is being intentionally circulated claiming as true information. In this paper, we develop a deep learning based system for detecting fake news.

Deception detection is a well-studied problem in Natural Language Processing (NLP) and researchers have addressed this problem quite extensively. The problem of detecting fake news in our everyday life, although very much related to deception detection, but in practice is much more challenging and hard, as the news body often contains a very few and short statements. Even for a human reader, it is difficult to accurately distinguish true from false information by just looking at these short pieces of information. Developing suitable hand engineered features (for a classical supervised machine learning model) to identify fakeness of such statements is also a technically challenging task. In contrast to classical feature-based model, deep learning has the advantage in the sense that it does not require any handcrafting of rules and/or features, rather it identifies the best feature set on its own for a specific problem. For a given news statement, our proposed technique classifies the short statement into the following fine-grained classes: *true*, *mostly-true*, *half-true*, *barely-true*, *false* and *pants-fire*. Example of such statements belonging to each class is given in Table 2.1 and the meta-data related to each of the statements is given in Table 2.2.

Table 2.1: Example statement of each class.

Ex	Statement (St)	Label
1	McCain opposed a requirement that the government buy American-made motorcycles. And he said all buy-American provisions were quote ’disgraceful.’	True
2	Almost 100,000 people left Puerto Rico last year.	Mostly-true
3	Rick Perry has never lost an election and remains the only person to have won the Texas governorship three times in landslide elections.	Half-true
4	Mitt Romney wants to get rid of Planned Parenthood.	barely-true
5	I dont know who (Jonathan Gruber) is.	FALSE
6	Transgender individuals in the U.S. have a 1-in-12 chance of being murdered.	pants-fire

2.1.2 Contributions

Most of the existing studies on fake news detection are based on classical supervised model. In recent times there has been an interest towards developing deep learning based fake news detection system, but these are mostly concerned with binary classification. In this paper, we attempt to develop an ensemble based architecture for fake news detection. The individual models are based on Convolutional Neural Network (CNN) and Bi-directional Long Short Term Memory (LSTM). The representations obtained from these two models are fed into a Multi-layer Perceptron (MLP) for multi-class classification.

2.1.3 Related Work

Fake new detection is an emerging topic in Natural Language Processing (NLP). The concept of detecting fake news is often linked with a variety of labels, such as misinformation [4], rumor [5], deception [6], hoax [7], spam [8], unreliable news [9], etc. In literature, it is also observed that social media [10] plays an essential role in the rapid spread of fake contents. This rapid spread is often greatly influenced by social

Table 2.2: Meta-data related to each example. P, F, B, H, M is speaker's previous count of Pants-fire, False, Barely-true, Half-true, Mostly-true respectively.

Ex	St Type	Spk	Spk's Job	State	Party	P	F	B	H	M	Context
1	federal-budget	barack-obama	President	Illinois	democrat	70	71	160	163	9	a radio ad
2	bankruptcy, economy, population	jack-lew	Treasury secretary	Washington, D.C.	democrat	0	1	0	1	0	an interview with Bloomberg News
3	candidates-biography	ted-nugent	musician	Texas	republican	0	0	2	0	2	an oped column.
4	abortion, federal-budget, health-care	planned-parenthood -action-fund	Advocacy group	Washington, D.C.	none	1	0	0	0	0	a radio ad
5	health-care	nancy-pelosi	House Minority Leader	California	democrat	3	7	11	2	3	a news conference
6	corrections-and-updates, crime, criminal -justice, sexuality	garnet-coleman	president, ceo of Apartments for America, Inc.	Texas	democrat	1	0	1	0	1	a committee hearing

bots [11]. It has been some time now since AI,ML, and NLP researchers have been trying to develop a robust automated system to detect Fake/ Deceptive/ Misleading/ Rumour news articles on various online daily access media platforms. There have been efforts to built automated machine learning algorithm based on the linguistic properties of the articles to categorize Fake News. Castillo et al. [12] in their work on social media (twitter) data showed that information from user profiles can be useful feature in determining veracity of news. These features were later also used by Gupta et al. to build a real-time system [13] to access credibility of tweets using SVM-rank. Researchers have also attempted to use Rule-Based and knowledge driven techniques to track the problem. Zhou et al. in their work [14] showed that deceptive senders have certain linguistic cues in their text. The cues are higher quantity, complexity, non-immediacy, expressiveness, informality, and affect; and less diversity, and specificity of language in their messages. Methods based on Information Retrieval from web were also proposed to verify authenticity of news articles. Etzioni et al. in their work [15] extracted claims from web to match with that of a given document to find inconsistencies. To deal with the problem further, researchers have also tried to seek deep learning strategies in their work. Bajaj [16] in his work applied various deep learning strategies on dataset composed of fake news articles available in Kaggle[2] and authentic news articles extracted from Signal Media News[3] dataset and observed that classifiers based on Gated Recurrent Unit (GRU), Long Short Term Memory (LSTM), Bi-directional Long Short Term Memory (Bi-LSTM) performed better than the classifiers based on CNN. Jing Ma et al. in their work [17], focused on developing a system to detect Rumor at EVENT level rather than at individual post level. The approach was to look at a set of relevant posts to a event at a given time interval to predict veracity of the event. They showed that use of recurrent networks are particularly useful in this task. Dataset from two different social media platform, Twitter, and Weibo were used. Chen et al. further built on the work of [5] for early detection Rumors at Event level, using the same dataset. They showed that the use of attention mechanism in recurrent network improves the performance in terms of precision, and recall measure, outperforming every other existing model for detecting rumor at an early stage. Natali Ruchansky et al. [12] used social media dataset (which is also used in [17] for Rumor Detection) and developed a hybrid deep learning model which showed promising performance on both Twitter data and Weibo data. They showed that both, capturing the temporal behavior of the articles as well as learning source characteristics about the behavior of the users, are essential for fake news detection. Further integrating these two elements improves the performance of the classifier.

Problems related to these topics have mostly been viewed concerning binary classification. Likewise, most of the published works also has viewed fake news detection as a binary classification problem (i.e., fake or true). But by observing very closely it can be seen that fake news articles can be classified into multiple classes depending on the fakeness of the news. For instance, there can be certain exaggerated or misleading information attached to a true statement or news. Thus, the entire news or statement can neither be accepted as completely true nor can be discarded as entirely false. This problem was addressed by William Y Wang in his paper [18] where he introduced **Liar** dataset comprising of a substantial volume of short political statements having six dif-

[2] https://www.kaggle.com/mrisdal/fake-news

[3] http://research.signalmedia.co/newsir16/signal-dataset.html

ferent class annotations determining the amount of fake content of each statement. In his work, he showed comparative studies of several statistical and deep learning based models for the classification task and found that the CNN model performed best. Long et al. [19] in their work used the **Liar** [19] dataset, and proposed a hybrid attention-based LSTM model for this task, which outperformed W.Yang's hybrid CNN model, establishing a new state-of-the-art.

In our current work we propose an ensemble architecture based on CNN [20] and Bi-LSTM [21], and this has been evaluated on **Liar** [18] dataset. Our proposed model tries to capture the pattern of information from the short statements and learn the characteristic behavior of the source speaker from the different attributes provided in the dataset, and finally integrate all the knowledge learned to produce fine-grained multi-class classification.

2.2 Methodology

We propose a deep multi-label classifier for classifying a statement into six fine-grained classes of fake news. Our approach is based on an ensemble model that makes use of Convolutional Neural Network (CNN) [20] and Bi-directional Long Short Term Memory (Bi-LSTM) [21]. The information presented in a statement is essentially sequential in nature. In order to capture such sequential information we use Bi-LSTM architecture. Bi-LSTM is known to capture information in both the directions: forward and backward. Identifying good features manually to separate true from fake even for binary classification, is itself, a technically complex task as human expert even finds it difficult to differentiate true from the fake news. Convolutional Neural Network (CNN) is known to capture the hidden features efficiently. We hypothesize that CNN will be able to detect hidden features of the given statement and the information related to the statements to eventually judge the authenticity of each statement. We make an intuition that both- capturing temporal sequence and identifying hidden features, will be necessary to solve the problem. As described in data section, each short statement is associated with 11 attributes that depict different information regarding the speaker and the statement. After our thorough study we identify the following relationship pairs among the various attributes which contribute towards labeling of the given statements.

1. Relation between **Statement** and **Statement type**
2. Relation between **Statement** and **Context**
3. Relation between **Speaker** and **Party**.
4. Relation between **Party** and **Speaker's job**.
5. Relation between **Statement type** and **Context**.
6. Relation between **Statement** and **State**.
7. Relation between **Statement** and **Party**.
8. Relation between **State** and **Party**.
9. Relation between **Context** and **Party**.
10. Relation between **Context** and **Speaker**.

To ensure that deep networks capture these relations we propose to feed each of the two attributes, say A_x and A_y, of a relationship pair into a separate individual model say M_i and M_j respectively. Then, concatenate the output of M_i and M_j and pass it through a fully connected layer to form an individual relationship network layer say $Network_n$ representing a relation. Fig.2.1 illustrates an individual relationship network layer. Eventually after capturing all the relations we group them together along with the five-column attributes containing information regarding speaker's total credit history count. In addition to that, we also feed in a special feature vector that is proposed by us and is to be formed using the count history information. This vector is a five-digit number signifying the five count history columns, with only one of the digit being set to '1' (depending on which column has the highest count) and the rest of the four digits are set to '0'. The deep ensemble architecture is depicted in fig 2.2.

2.2.1 Bi-LSTM

Bidirectional LSTMs are the networks with LSTM units that process word sequences in both the directions (i.e. from left to right as well as from right to left). In our model we consider the maximum input length of each statement to be 50 (average length of statements is 17 and the maximum length is 66, and only 15 instances of the training data of length greater than 50) with post padding by zeros. For attributes like statement type, speaker's job, context we consider the maximum length of the input sequence

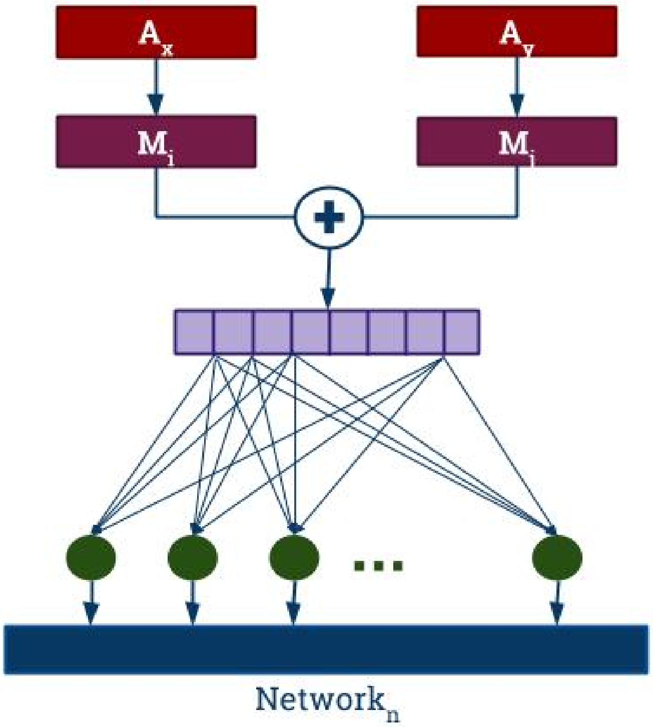


Figure 2.1: A relationship network layer. A_x and A_y are two attributes, M_i and M_j are two individual models, $Network_n$ is a representation of a network capturing a relationship

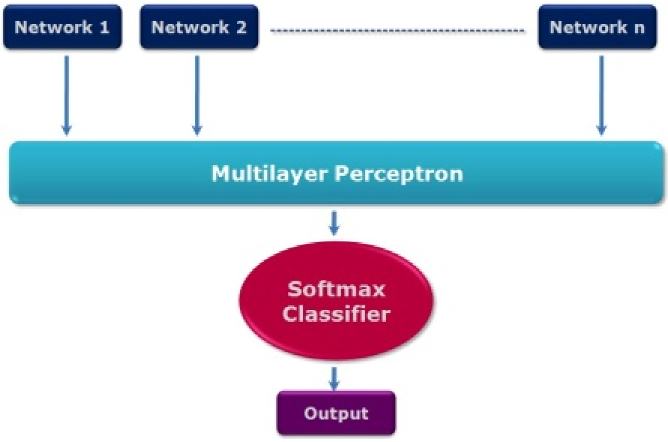


Figure 2.2: Deep Ensemble architecture

to be 5, 20, 25, respectively. Each input sequence is embedded into 300-dimensional vectors using pre-trained Google News vectors [22] (Google News Vectors 300dim is also used in [18] for embedding). Each of the embedded inputs are then fed into separate Bi-LSTM networks, each having 50 neural units at each direction. The output of each of these Bi-LSTM network is then passed into a dense network of 128 neurons with activation function as 'ReLU'.

2.2.2 CNN

Over the last few years many experimenters has shown that the convolution and pooling functions of CNN can be successfully used to find out hidden features of not only images but also texts. A convolution layer of $n \times m$ kernel size will be used (where m-size of word embedding) to look at n-grams of words at a time and then a MaxPooling layer will select the largest from the convoluted inputs.The attributes, namely speaker, party, state are embedded using pre-trained 300-dimensional Google News Vectors [22] and then the embedded inputs are fed into separate Conv layers.The different credit history counts the fake statements of a speaker and a feature proposed by us formed using the credit history counts are directly passed into separate Conv layers.

2.2.3 Combined CNN and Bi-LSTM Model

The representations obtained from CNN and Bi-LSTM are combined together to obtain better performance.

The individual dense networks following the Bi-LSTM networks carrying information about the statement, the speaker's job, context are reshaped and then passed into different Conv layers. Each convolution layer is followed by a Maxpooling layer, which is then flattened and passed into separate dense layers. Each of the dense layers of different networks carrying different attribute information are merged, two at a time-to capture the relations among the various attributes as mentioned at the beginning of 2.2. Finally, all the individual networks are merged together and are passed through a dense layer of six neurons with softmax as activation function as depicted in. The classifier is optimized using Adadelta as optimization technique with categorical cross-entropy as the loss function.

2.3 Data

We use the dataset, named **LIAR** (Wang 2017), for our experiments. The dataset is annotated with six fine-grained classes and comprises of about 12.8K annotated short statements along with various information about the speaker. The statements which were mostly reported during the time interval [2007 to 2016], are considered for labeling by the editors of **Politifact.com**. Each row of the data contains a short statement, a label of the statement and 11 other columns correspond to various information about the speaker of the statement. Descriptions of these attributes are given below:

1. **Label**: Each row of data is classified into six different types, namely
   (a) **Pants-fire:** Means the speaker has delivered a blatant lie .
   (b) **False:** Means the speaker has given totally false information.
   (c) **Barely-true:** Chances of the statement depending on the context is hardly true. Most of the contents in the statements are false.
   (d) **Half-true:** Chances of the content in the statement is approximately half.
   (e) **Mostly-true:** Most of the contents in the statement are true.
   (f) **True:** Content is true.
2. **Statement by the politician:** This statement is a short statement.
3. **Subjects:** This corresponds to the content of the text. For examples, foreign policy, education, elections etc.
4. **Speaker:** This contains the name of the speaker of the statement.
5. **Speaker's job title:** This specifies the position of the speaker in the party.
6. **State information:** This specifies in which state the statement was delivered.
7. **Party affiliation:** This denotes the name of the party of the speaker belongs to.
8. The next five columns are the counts of the speaker's statement history. They are:
   (a) **Pants fire count;**
   (b) **False count;**
   (c) **Barely true count;**
   (d) **Half false count;**
   (e) **Mostly true count.**
9. **Context:** This corresponds to the venue or location of the speech or statement.

The dataset consists of three sets, namely a training set of 10,269 statements, a validation set of 1,284 statements and a test set of 1,266 statements.

2.4 Experiments and Results

In this section, we report on the experimental setup, evaluation results, and the necessary analysis.

2.4.1 Experimental Setup

All the experiments are conducted in a python environment. The libraries of python are required for carrying out the experiments are **Keras**, **NLTK**, **Numpy**, **Pandas**, **Sklearn**. We evaluate the performance of the system in terms of accuracy, precision, recall, and F-score metrics.

2.4.2 Results and Analysis

We report the evaluation results in Table 2.3 that also show the comparison with the system as proposed in [18] and [19].

Table 2.3: Overall evaluation results

Model	Network	Attributes taken	Accuracy
William Yang Wang [18]	Hybrid CNN	All	0.274
Y. Long et al. [19]	Hybrid LSTM	All	0.415
Bi-LSTM Model	Bi-LSTM	All	0.4265
CNN Model	CNN	All	0.4289
Our Proposed Model	RNN-CNN combined	All	**0.4487**

We depict the overall evaluation results in Table 2.3 along with the other existing models. This shows that our model performs better than the existing state-of-the-art model as proposed in [19]. This state-of-the-art model was a hybrid LSTM, with an accuracy of 0.415. On the other hand, our proposed model shows 0.4265, 0.4289 and 0.4487 accuracies for Bi-LSTM, CNN and the combined CNN+Bi-LSTM model, respectively. This clearly supports our assumption that capturing temporal patterns using Bi-LSTM and hidden features using CNN are useful, channelizing each profile attribute through a different neural layer is important, and the meaningful combination of these separate attribute layers to capture relations between attributes, is effective.

Table 2.4: Evaluation of Bi-LSTM model: precision, recall, and F1 score

	precision	**recall**	**F1-score**	**No. of instances**
PANTS-FIRE	0.73	0.35	0.47	92
FALSE	0.47	0.53	0.50	249
BARELY-TRUE	0.58	0.32	0.41	212
HALF-TRUE	0.39	0.46	0.42	265
MOSTLY-TRUE	0.33	0.66	0.44	241
TRUE	0.88	0.14	0.23	207
Avg/Total	0.53	0.43	0.41	1266

Table 2.5: Evaluation of CNN model: precision, recall, F1 score

	precision	**recall**	**F1-score**	**No. of instances**
PANTS-FIRE	0.67	0.39	0.49	92
FALSE	0.36	0.63	0.46	249
BARELY-TRUE	0.50	0.36	0.42	212
HALF-TRUE	0.42	0.46	0.44	265
MOSTLY-TRUE	0.41	0.49	0.45	241
TRUE	0.70	0.16	0.26	207
Avg/Total	0.48	0.43	0.42	1266

Table 2.6: Evaluation of Bi-LSTM, CNN combined model: precision, recall, F1 score

	precision	**recall**	**F1-score**	**No. of instances**
PANTS-FIRE	0.70	0.43	0.54	92
FALSE	0.45	0.61	0.52	249
BARELY-TRUE	0.61	0.32	0.42	212
HALF-TRUE	0.35	0.73	0.47	265
MOSTLY-TRUE	0.50	0.36	0.42	241
TRUE	0.85	0.14	0.24	207
Avg/Total	**0.55**	**0.45**	**0.43**	1266

Table 2.7: Sample text with wrongly predicted label and original label. Spk is speaker, and P, F, B, H, M is speaker's previous count of Pants-fire, False, Barely-true, Half-true, Mostly-true respectively.

Label	Statement	St Type	Spk	Spk's Job	State	Party	Context	P	F	B	H	M	Predicted Label
barely-true	We know there are more Democrats in Georgia than Republicans. We know that for a fact.	elections	mike-berlon	none	Georgia	democrat	an article	1	0	0	0	0	False

We also report the precision, recall and F-score measures for all the models. Table2.4, Table 2.5 and Table 2.6 depict the evaluation results of CNN, Bi-LSTM and the combined model of CNN and Bi-LSTM, respectively. The evaluation shows that on the precision measure the combined model performs best with an average precision of **0.55** while that of Bi-LSTM model is 0.53 and CNN model is 0.48. The combined model of CNN and Bi-LSTM even performs better with respect to recall and F1-Score measures. The combined model yields the average recall of **0.45** and average F1-score of **0.43** while that of Bi-LSTM model is 0.43 and 0.41, respectively and of the CNN model is 0.43 and 0.42, respectively. On further analysis, we observe that although the performance (based on precision, recall, and F1-score) of each of the models for every individual class is close to the average performance, but in case of the prediction of the class label **TRUE** the performance of each model varies a lot from the respective average value. The precisions of TRUE is promising (Bi-LSTM model:0.88, CNN model: 0.7, Combined model:**0.85**), but the recall (Bi-LSTM model:0.14, CNN model: 0.16, Combined model:0.14) and the F1-score (Bi-LSTM model:0.23, CNN model: 0.26, Combined model:0.24) are very poor. This entails the fact that our proposed model predicts comparatively less number of instances as TRUE, but when it does the prediction is very accurate. Thus it can be claimed that if a statement is predicted as **True** by our proposed model then one can rely on that with high confidence. Although our model performs superior compared to the existing state-of-the-art, still the results were not error free. We closely analyze the models' outputs to understand their behavior and perform both quantitative as well as qualitative error analysis. For quantitative analysis, we create the confusion matrix for each of our models. Confusion matrix corresponding to the experiment 1 i.e with Bi-LSTM model is given in Table2.8, corresponding to experiment 2 i.e with CNN model is given in Table2.9 and corresponding to our final experiment i.e with RNN-CNN combined model is given in Table 2.10.

From these quantitative analysis it is seen that in majority of the cases the test data statements originally labeled with **Pants-Fire** class gets confused with the **False** class, statements originally labeled as **False** gets confused with **Barely true** and **half true** classes, statements originally labeled as **Half true** gets confused with **Mostly True** and **False** class, statements originally labeled as **Mostly true** gets confused with **Half True**, statements originally labeled with **True** gets confused with **Mostly True** class.

It is quite clear that errors were mostly concerned with the classes, overlapping in nature. Confusion is caused as the contents of the statements belonging to these classes are quite similar. For example, the difference between 'Pants-Fire' and 'False' class is that only the former class corresponds to the false information with more intensity. Likewise 'Half True' has high similarity to 'False', and 'True' with 'Mostly True'. The difference between 'True' and 'Mostly True' is that the later class has some marginal amount of false information, while the former does not.

For qualitative analysis, we closely look at the actual statements and try to understand the causes of misclassifications. We come up with some interesting facts. There are some speakers whose statements are not present in the training set, but are present in the test set. For few of these statements, our model tends to produce wrong answers. Let us consider the example given in Table2.7. For this speaker, there is no training data available and also the count history of the speaker is very less. So our models assign an incorrect class. But it is to be noted that even if there is no information about the speaker in the training data and the count history of the speaker is almost empty, still we are able to generate a prediction of a class that is close to the original class in terms of meaning.

It is also true that classifiers often make mistakes in making the fine distinction between the classes due to the insufficient number of training instances. Thus, classifiers tend to misclassify the instances into one of the nearby (and overlapped) classes.

Table 2.8: Confusion matrix of the Bi-LSTM model

Actual\Predicted	Pants-Fire	False	Barely-True	Half-True	Mostly-True	True
Pants-Fire	32	35	3	8	14	0
False	4	131	16	36	59	3
Barely-True	5	31	68	48	60	0
Half-True	0	38	8	123	95	1
Mostly-True	1	20	8	54	158	0
True	2	25	15	47	90	28

Table 2.9: Confusion matrix of the CNN model

Actual\Predicted	Pants-Fire	False	Barely-True	Half-True	Mostly-True	True
Pants-Fire	36	35	6	11	2	2
False	7	156	21	30	28	7
Barely-True	5	66	76	34	29	2
Half-True	2	75	14	123	48	3
Mostly-True	1	53	17	51	119	0
True	3	44	18	44	65	33

Table 2.10: Confusion matrix of the Bi-LSTM+CNN combined model

Actual\Predicted	Pants-Fire	False	Barely-True	Half-True	Mostly-True	True
Pants-Fire	40	34	4	10	4	0
False	7	152	10	67	11	2
Barely-True	4	48	68	83	9	0
Half-True	0	43	7	193	20	2
Mostly-True	2	31	9	112	86	1
True	4	31	13	89	41	29

2.5 Conclusion and Future Works

In this paper, we have tried to address the problem of fake News detection by looking into short political statements made by the speakers in different types of daily access media. The task was to classify any statement into one of the fine-grained classes of fakeness. We have built several deep learning models, based on CNN, Bi-LSTM and the combined CNN and Bi-LSTM model. Our proposed approaches mainly differ from previously mentioned models in system architecture, and each model performs better than the state of the art as proposed in [19], where the statements were passed through one LSTM and all the other details about speaker's profile through another LSTM. On the other hand, we have passed every different attribute of speaker's profile through a different layer, captured the relations between the different pairs of attributes by concatenating them. Thus, producing a meaningful vector representation of relations between speaker's attributes, with the help of which we obtain the overall accuracy of 44.87%. By further exploring the confusion matrices we found out that classes which are closely related in terms of meaning are getting overlapped during prediction. We have made a thorough analysis of the actual statements, and derive some interesting facts. There are some speakers whose statements are not present in the training set but present in the test set. For some of those statements, our model tends to produce the wrong answers. This shows the importance of speakers' profile information for the task. Also as the classes and the meaning of the classes are very near, they tend to overlap due to less number of examples in training data.

We would like like to highlight some of the possible solutions to solve the problems that we encountered while attempted to solve fake news detection problem in a more fine-grained way.

- More labeled data sets are needed to train the model more accurately. Some semi-supervised or active learning models might be useful for this task.
- Along with the information of a speaker's count history of lies, the actual statements are also needed in order to get a better understanding of the patterns of the speaker's behavior while making a statement.

Fake news detection into finely grained classes that too from short statements is a challenging but interesting and practical problem. Hypothetically the problem can be related to **Sarcasm detection** [23] problem. Thus it will also be interesting to see the effect of implementing the existing methods that are effective in sarcasm detection domain in Fake News detection domain.

CHAPTER 3

Step-by-Step: A three-stage Pipeline for Document Stance Classification towards Claims

Fact-checking has emerged as an important means to counter the wide spread of fake news. Identifying documents that support or negate a particular claim is an essential task in this process. In this context, stance classification aims at identifying the position (stance) of a document towards a claim. However, existing (4-class) stance classification approaches are ineffective in detecting documents that negate a claim, even though these documents provide crucial evidence when aiming to detect false claims. In this paper, we propose a three-stage pipeline that treats the initial 4-class problem as three different but connected binary classification tasks, allowing the use of different classifiers and features in each step, thus enabling us to optimise performance on a per step and class basis. Evaluation results of our proposed approach demonstrate the state-of-the-art performance and its ability to significantly improve the classification performance of the important disagree class.

3.1 Introduction

Spread of fake news and false claims have become ubiquitous due to the widespread use and network effects facilitated by social online platforms [24]. A recent study has shown that false claims are re-tweeted faster, further, and for longer than true claims on twitter [2]. Another study found that the top 20 fake news stories about the 2016 U.S. presidential election received more engagement on Facebook than the top 20 election stories from the 19 major media outlets [25]. These findings demonstrate the significance and scale of the fake news problem and the potential effects it can have on contemporary society [26].

In this context, *fact-checking* has emerged as a new type of process where an assertion or statement (a *claim*) is examined to determine its veracity and correctness

[27–29]. Research on the impact of fact-checking has shown that it has a corrective effect on misperceptions among the citizens and discourages politicians from spreading misinformation [30].

Identifying claim-relevant documents that support or negate the claim is essential to address fact-checking. To this end, *stance detection* aims at identifying the perspective (stance) of a document towards a claim, namely whether the document *agrees* with the claim, *disagrees* with the claim, *discusses* about the claim without taking a stance, or is entirely *unrelated* to the claim. In this context, the Fake News Challenge [31] introduced the stance detection task (FNC-I) as an essential building block in an AI-assisted fact-checking pipeline.

However, existing approaches that try to cope with this problem show the same limitation: they are ineffective in detecting instances of the *disagree* class. Specifically, the top three participating systems of FNC-I achieved an F1 score of 3%, 15%, and 11%, respectively, on the *disagree* class (more in Sect. 4.4). However, this class is of key importance in fact-checking since it enables detecting documents that provide evidence for invalidating false claims.

The problem arises from the imbalanced data distribution: the *disagree* class corresponds to less than 3% of the instances in the FNC-I dataset. The FNC-I dataset was derived from the *Emergent* dataset [32] which seems to represent a real world data distribution with *disagree* (*against* in *Emergent*) being the minority class.

In this paper, we focus on this problem and introduce an approach that highly improves the classification performance of the important but underrepresented *disagree* class, without significantly affecting the performance of the *agree* class (the other important class in this *fake news* context). We propose a three-stage pipeline architecture that treats the 4-class classification problem as three different binary classification subtasks with increasing difficulty. The first stage aims at detecting the documents related to the claim (*relevance classification*), the second stage classifies only related documents and focuses on detecting documents that take a stance towards the claim (*neutral/stance classification*), and the third stage classifies the documents taking a stance as agreeing or disagreeing to the claim (*agree/disagree classification*).

Such a step-wise analysis offers the flexibility to use different classifiers and features in each different stage of the pipeline, thus enabling to optimise performance on a per stage and class basis. Evaluation results show that our approach achieves the state-of-the-art performance on the general stance classification problem, slightly outperforming all the existing methods by one percentage point of macro-averaged F1 score. Most importantly, we significantly improve the F1 score of the *disagree* class by 28% compared to the state-of-the-art. A careful analysis of the outputs produced by the system reveals a number of limitations of our approach and demonstrates that there is room for further improvement.

In a nutshell, we make the following contributions:

- We introduce a three-stage pipeline for the 4-class stance classification problem, where each stage focuses on a different binary classification task, and propose dedicated classifiers and features for each stage in the pipeline.
- We provide a comparative evaluation of the overall and class-wise performance of our approach and other methods, demonstrating state-of-the-art performance on the general stance classification problem and significantly improving the performance of the important (and underrepresented) *disagree* class.

The rest of the paper is organized as follows: Section 4.2 formulates the problem and provides an overview of our pipeline approach. Section 3.3 describes supervised models for each stage of the pipeline. Section 4.4 reports evaluation results. Section 4.5 presents related works. Finally, Section 4.6 concludes the paper and discusses interesting directions for future research.

3.2 Problem Modeling and Approach Overview

Given a textual claim c (e.g., *"KFC restaurants in Colorado will start selling marijuana"*) and a document d (e.g., an article), stance classification aims at classifying the stance of d towards c to one of the following four categories (classes):

- **Unrelated**: the document is not related to the claim.
- **Neutral**: the document discusses about the claim but it does not take a stance towards its validity.
- **Agree**: the document agrees with the claim.
- **Disagree**: the document disagrees with the claim.

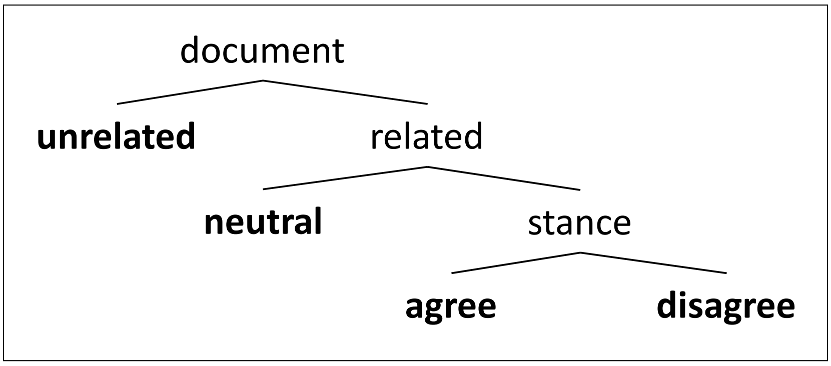


Figure 3.1: Document stance hierarchy.

These four classes can be structured in a tree-like hierarchy as shown in Fig. 3.1. At first, a document can be either *unrelated* or *related* to the claim. Then, a document that is related to the claim can either be *neutral* to the claim or take a *stance*. Finally, a document that takes a stance can either *agree* or *disagree* with the claim. The leaves of the tree are the four classes. Considering this structure, we can now model the stance classification problem as a three-stage pipeline consisting of three connected sub-tasks (or *stages*):

- **Stage 1 (relevance classification)**: identify if a document is related to the claim or not.
- **Stage 2 (neutral/stance classification)**: identify if a document classified as *related* from stage 1 is neutral to the claim or takes a stance.
- **Stage 3 (agree/disagree classification)**: identify if a document classified as *stance* from stage 2 agrees or disagrees with the claim.

Figure 3.2 depicts the proposed pipeline architecture. Each stage of the pipeline can now be modeled as a separate binary classification problem. We hypothesize that relevance classification (stage 1) can filter out documents unrelated to the claim which can facilitate the neutral/stance classification task (stage 2). Likewise, knowing that a document takes a stance towards the claim can facilitate the agree/disagree classification task (stage 3). A weakness of such a pipeline approach is that errors can propagate from one stage to the other, thus errors in earlier stages negatively affect the later stages. It is to be noted though that, in general, relevance classification (stage 1) is a much easier task than neutral/stance classification (stage 2), which in turn is considered easier than agree/disagree classification (stage 3). Our experimental evaluation validates this hypothesis (more in Sect. 4.4).

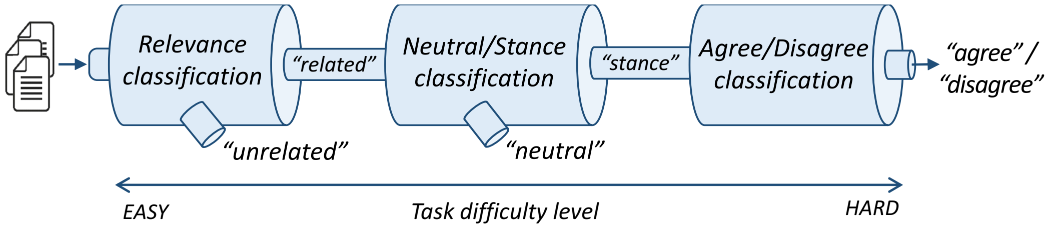


Figure 3.2: Pipeline for document stance classification and difficulty level of each stage.

3.3 Pipeline Implementation

In this section we describe our classification models for each stage of the proposed pipeline, each being implemented through a supervised model tailored towards the specific classification problem at hand.

3.3.1 Stage 1: Relevance Classification

Existing approaches on separating the *related* from the *unrelated* instances have already achieved a high accuracy (>95%) [33]. Such models usually make use of hand-crafted features that aim at reflecting the text similarity between the claim and the document. Since our focus is on the important *disagree* class (which is part of *related* in this stage), we seek a model that penalises misclassifications of this class. To this end, we tried a variety of different classifiers and feature combinations inspired by previous works that perform well on the same problem. Specifically, we tried the following classifiers: support vector machine (SVM) with and without class-wise penalty, gradient boosting trees, AdaBoost, decision trees, random forest with and without class-wise penalty, and convolutional neural networks (CNN). To train the classifiers, we experiment with a variety of text similarity-based features.

We found that a simple SVM classifier with class-wise penalty [34] trained on a set of eight hand-crafted features (described below) outperforms all the other models. In more detail, we solve the following optimization problem:

$$Min_{\varpi,\beta}(\frac{\varpi^T\varpi}{2} + \alpha_1 \sum_{i|\gamma_i \in Relat.}^{m} \epsilon_i + \alpha_2 \sum_{i|\gamma_i \in Unrel.}^{m} \epsilon_i) \tag{3.1}$$

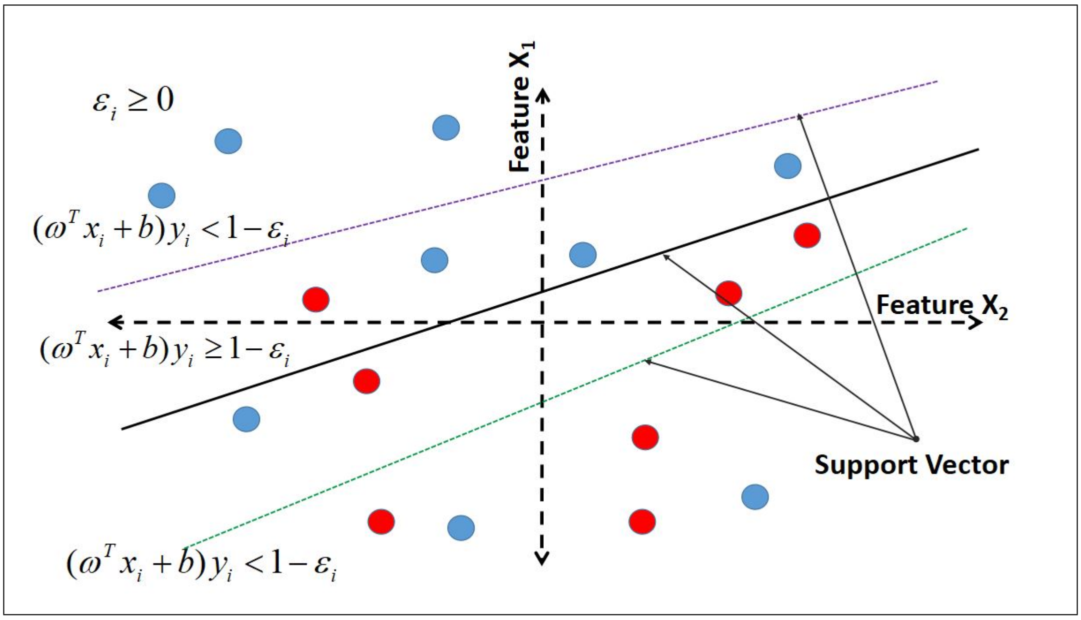

$\varepsilon_i \geq 0$
$(\omega^T x_i + b)y_i < 1-\varepsilon_i$
$(\omega^T x_i + b)y_i \geq 1-\varepsilon_i$
$(\omega^T x_i + b)y_i < 1-\varepsilon_i$
Feature X_1
Feature X_2
Support Vector

Figure 3.3: Diagram outlining the SVM model used in stages 1 and 3.

Subjected to the constraints,

$$\gamma_i(\varpi^T \chi_i + \beta) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0, \quad i = 1, ..., m \tag{3.2}$$

where, ϖ is weight vector, β is bias, γ_i is the output constraint function, χ_i is the training input vector, and α_1 and α_2 are regularization hyperparameters of penalty terms ϵ_i for *related* and *unrelated* class, respectively. A pictorial representation of the model is depicted in Fig. 3.3. Hyperparameters are tuned through 10-fold cross-validation on the training dataset.

We use the following set of hand-crafted features, selected through extensive feature analysis. The first four features are used in the baseline model provided by the FNC-I organizers, the next two are inspired from [35], and the last two (*keyword* and *proper noun overlap*) are new.

- *N-grams match:* A *n-gram* is the sequence of n continuous words in a given text. The feature value is defined as the number of common n-grams in the claim and the document. It is basically the length of the set formed by intersecting the set of claim's n-grams with the set of the document's n-grams. For our system, we choose bigrams, trigrams and fourgrams.
- *Chargrams match:* Similar to n-gram, chargram is a sequence of n continuous characters. Similar to *ngrams match*, the feature is defined as the number of common chargrams in the claim and the document. We use bi-, tri- and four-chargrams in our system.
- *Binary co-occurrence:* This feature consists of two values. The first one is the

number of words of the claim that appear in the first 255 words of the document, and the second one is the number of words in the claim that appear in the entire body of the document.

- *Lemma overlap:* This feature is similar to the unigram match with the difference that the words are first converted into their lemmatized form.
- *Text similarity:* We calculate the cosine similarity between the text of the claim and each sentence of the document. The maximum similarity value is considered as the feature value.
- *Word2vec similarity:* The cosine similarity between the pre-trained word2vec embeddings [22] of the claim and the document.
- *Keyword overlap:* Keywords are important words that appear in the text. We extract the keywords from the claim and the document using the *cortical.io* tool[1]. The feature is defined as the number of common keywords in the claim and the document.
- *Proper noun overlap:* This feature is same as keyword overlap but instead of keywords we extract proper nouns using the NLTK Part-of-Speech tagger[2] [36].

3.3.2 Stage 2: Neutral/Stance Classification

In this stage, we require a model that misclassifies the smallest number of *stance* instances. Previous works on similar problems have shown that deep learning models [37–39], supervised classifiers with sentiment-related features [40], as well as sentiment features employed in neural models [41], help solving the problem effectively. We tried a variety of approaches including both neural models (Bi-LSTM, CNN) and classical machine learning models (like SVM, gradient boosting and decision trees). We found that a simple CNN model with embedded word vectors and sentiment features outperforms all other methods. CNN is also used by the top performing system of FNC-I, however for the overall 4-class stance classification problem.

To generate sentiment scores for the claims and documents, we use the NLTK sentiment intensity analyzer [42]. NLTK applies a rule-based model for sentiment analysis, and is preferred in comparison to the other systems because it is very effective in aggregating the sentiment polarity of multiple negative words. Our intuition is that a document supporting or refuting a claim will have a strong overall sentiment polarity,

[1] https://www.cortical.io/
[2] https://www.nltk.org/

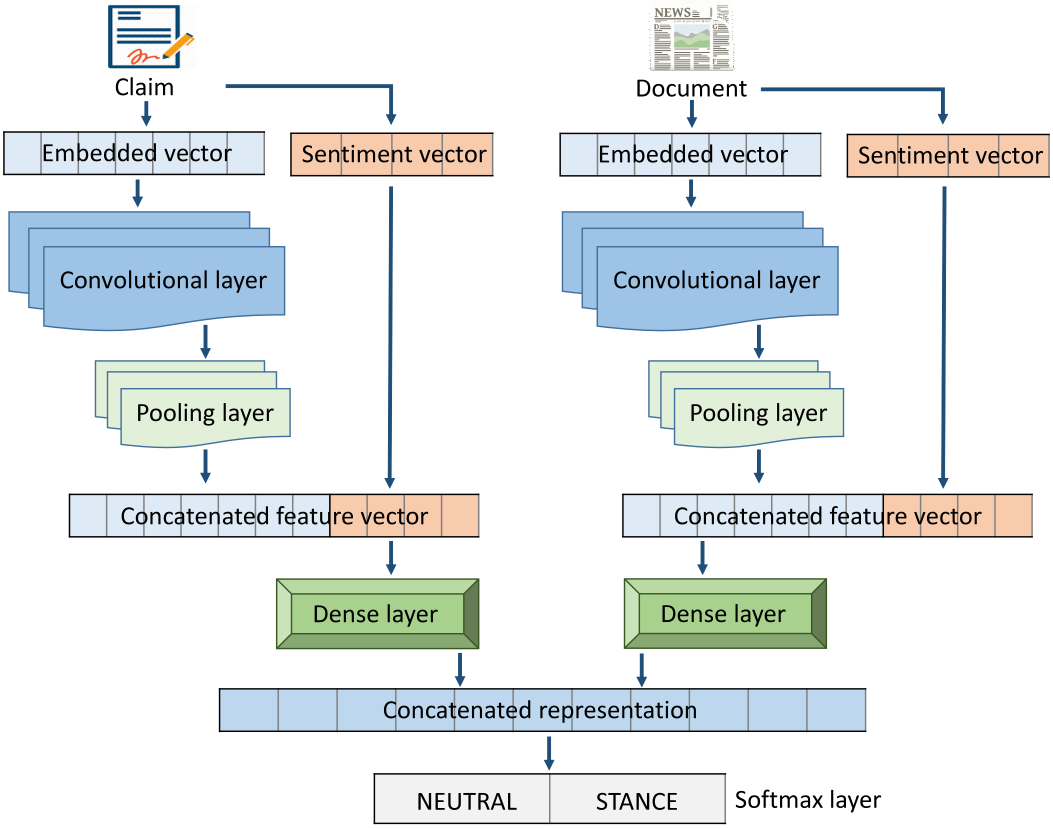


Figure 3.4: Diagram outlining the CNN model used in Stage 2.

while a document not taking a stance towards the claim will have a more neutral overall polarity. In addition, by inspecting several documents that take a stance we noticed that the stance is usually expressed in the first few lines (this is also supported in [43]). Thus, we consider only the document's first 10 sentences in our analysis.

For implementing CNN, we do not follow the approach used by the top performing system of FNC-I because the training data in our case is limited (stage 1 has filtered out *unrelated* instances), and their CNN model underfits and learns only the *neutral* class. Instead, we opt for a much simpler network architecture (depicted in Fig. 3.4): the text of the claim and the first part of the document are converted into embedded vectors (V_c and V_d, respectively). Next, an array of four sentiment scores (positive, negative, neutral, compound) is generated for both the claim and the document using NLTK (arrays S_c and S_d, respectively). The two vectors V_c and V_d are passed through two separate convolutional networks with max-pooling, and obtain the representations R_c and R_d, respectively. R_c is then merged with S_c and R_d with S_d to get the final representations which are then passed through two separate multi-perceptron dense layers with regularization. This gives us the two networks DN_c and DN_d which are combined for the final softmax binary classification. As output class label we consider the one with the highest probability score.

3.3.3 Stage 3: Agree/Disagree Classification

Since the number of documents taking a stance is usually small, especially the number of *disagree* documents, training a deep neural model efficiently is difficult. We hypothesize that a statistical machine learning algorithm trained with a well-defined set of features can be more effective. We again experiment with several models including SVM with and without class-wise penalty, gradient boosting, decision tree, and random forest. We found that an SVM classifier (as depicted in Fig. 3.3) similar to the one of stage 1 obtains the best performance for the *disagree* class. Specifically, we effectively solve the same optimization function, while the model differs from the model of stage 1 in terms of the set of considered features.

As mentioned before, sentiment-related features are useful in the problems related to opinion/stance classification. Thus, similar to stage 2, we again use the sentiment features S_c and S_d (for the claim and document, respectively) generated using NLTK. In addition, we exploit linguistic features generated using the LIWC tool (Linguistic Inquiry and Word Count)[3] [44]. LIWC returns more than 90 features related to various linguistic properties of the input text. After extensive feature analysis, we selected the following 16 features (that also seem to be useful for understanding the agreement/disagreement stance): *analytical thinking*, *clout* (expressing confidence in perspective), *authentic*, *emotional tone*, *conjugation*, *negation*, *comparison words*, *affective processes*, *positive emotions*, *negative emotions*, *anxiety*, *anger*, *sadness*, *differentiation* (distinguishing between entities), *affiliation* (references to others), and *achieve* (reference to success, failure). Moreover, we consider the *refuting words* feature set used in the baseline model provided by the FNC-I organizers. This feature set is generated by matching words from a predefined set of *refuting words* with the document's words. The result is a feature vector of the length same as the number of refuting words. The vector contains “1” in position i if the corresponding refuting word exists in the document, otherwise it contains “0”.

[3] http://liwc.wpengine.com/

Table 3.1: Data distribution of the FNC-I dataset.

	All	**Unrelated**	**Neutral**	**Agree**	**Disagree**
Train	49,972	36,545	8,909	3,678	840
Test	25,413	18,349	4,464	1,903	697

3.4 Evaluation

3.4.1 Evaluation Setup

Dataset

We use the benchmark dataset provided by the Fake News Challenge - Stage 1 (FNC-I)[4] [31], which focuses on the same *4-class* stance classification task. While there are datasets for the related tasks available, like stance detection of tweets or ideological debates, these address binary or three-class classification problems (thereby our three-stage pipeline is not applicable) and focus on detecting the stance of *user opinions* regarding topics, as opposed to the stance of Web documents towards a given true/false claim (more in Sect. 4.5).

The FNC-I dataset was derived from the Emergent dataset [32] and consists of 2,587 documents related to 300 claims. Each document has a summarised *headline* which reflects the stance of its text (this means that each claim can be represented through different headlines of different stances which make the problem harder). The FNC-I dataset contains 49,972 training and 25,413 test instances, related to 200 and 100 different claims, respectively. Each instance has three attributes: a *headline* (which in our case has the role of a *claim*), ii) a *body text* (document), and iii) a *stance label*, having one of the following values: *unrelated*, *discuss* (*neutral*), *agree*, and *disagree*. Table 4.1 shows the class distribution. We see that the dataset is highly imbalanced: there is a very large number of *unrelated* documents (more than 70% in both training and testing datasets), a large number of *neutral* documents (about 18%), and a very small number of agree ($< 8\%$) and disagree ($< 3\%$) documents.

For the first stage of our pipeline (relevance classification), we merge the classes *discuss*, *agree* and *disagree*, to one *related* class. This gives us two classes: *unre-*

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

Table 3.2: Class distribution for *relevance* classification.

	Instances	**Unrelated**	**Related**
Train	49,972	36,545	13,427
Test	25,413	18,349	7,064

Table 3.3: Class distribution for *discuss/stance* classification.

	Instances	**Neutral**	**Stance**
Train	13,427	8,909	4,518
Test	7,064	4,464	2,600

Table 3.4: Class distribution for *agree/disagree* classification.

	Instances	**Agree**	**Disagree**
Train	4,518	3,678	840
Test	2,600	1,903	697

lated and *related*. Table 3.2 shows the corresponding class distribution. For the second stage (neutral/stance classification), we consider only *related* documents and merge the classes *agree* and *disagree* to one *stance* class. This gives us two classes: *neutral* and *stance*. Table 3.3 shows the corresponding class distribution. We notice that the *neutral* documents are about twice the *stance* documents. For the final stage (agree/disagree classification), we only consider the instances of the *agree* and *disagree* classes. Table 3.4 shows the corresponding class distribution. We notice that the classes are imbalanced with the disagree class amounting to only 18.6% (26.8%) of the instances in the training (test) dataset.

Evaluation Metrics

The FNC-I task was evaluated based on a weighted (two-level) scoring system which awards 0.25 points if a document is correctly classified as *related* or *unrelated*, and an additional 0.75 points if it is correctly classified as *neutral*, *agree* or *disagree*. However, as also argued in [33], this metric fails to take into account the highly imbalanced class distribution of the classes *neutral*, *agree*, *disagree*. For example, a classifier that always predicts *neutral* after a correct *related* prediction achieves a score of 0.833, which is higher than the top-ranked system in FNC. Moreover, an effective stance classification model should perform well for the important (for fact-checking and fake news detection) classes, *agree* and *disagree*, since such documents provide crucial evidence when aiming to detect false claims or validate true claims. On the other hand, the *unrelated* and *neutral* classes are not important in this context. For instance, a document that discusses about a claim without taking a stance is not actually useful in fact-checking since it does not provide actual evidence about the veracity of the claim.

Based on the above observations, apart from the FNC evaluation measure, we also consider the following metrics for the overall assessment and comparison: i) the class-wise F1 score (the harmonic mean of precision and recall for each class), ii) the macro-averaged F1 score across all the four classes ($\mathrm{F1}^m$), and iii) the macro-averaged F1 score across the important classes *agree* and *disagree* ($\mathrm{F1}^m_{\mathrm{Agr/Dis}}$).

Baselines

We consider the below eight baseline methods:

- *Majority vote*: The class with the maximum number of instances is always selected (*unrelated* in our case).
- *FNC baseline*[5]: A gradient boosting classifier using a set of hand-crafted features relevant for the task. The features include word/n-gram overlap features, and indicator features for polarity and refutation.
- *SOLAT in the SWEN*[6] [45]: The top-ranked system of FNC-I. This model is based on a weighted average between gradient-boosted decision trees and a deep convolutional neural network. The considered features include: word2vec embeddings, number of overlapping words, similarities between the word count, 2-grams and 3-grams, and similarities after transforming the counts with TF-IDF weighting and SVD.
- *Athene (UKP Lab)*[7] [46]: The second-ranked system of FNC-I. The model is a multilayer perceptron classifier (MLP) with six hidden and a softmax layer. It incorporates the following hand-crafted features: unigrams, cosine similarity of word embeddings of nouns and verbs between claim and document, topic models based on non-negative matrix factorization, latent Dirichlet allocation, and latent semantic indexing, in addition to the features provided in the FNC-I baseline.
- *UCL Machine Reading (UCLMR)*[8] [47]: The third-ranked system of FNC-I. The model is a simple MLP network with a single hidden layer. As features it uses the TF vectors of unigrams of the 5,000 most frequent words, and the cosine similarity of the TF-IDF vectors of the claim and document.
- *ComboNSE* [48]: A deep MLP model which combines neural, statistical and external features. Specifically, the model uses neural embeddings from a deep

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

[6] https://github.com/Cisco-Talos/fnc-1/

[7] https://github.com/hanselowski/athene_system

[8] https://github.com/uclmr/fakenewschallenge

recurrent model, statistical features from a weighted n-gram bag-of-words model, and hand-crafted external features (including TF, ngrams, TF-IDF, and sentiment features).

- *StackLSTM* [33]: A model which combines hand-crafted features (selected through extensive feature analysis) with a stacked LSTM network, using 50-dimensional GloVe word embeddings [49], in order to generate sequences of word vectors of a claim-document pair.

- *LearnedMMD* [50]: Two-layer hierarchical neural network that controls the error propagation between the two layers using a Maximum Mean Discrepancy regularizer. The first layer distinguishes between the *related* and *unrelated* classes, and the second detects the actual stance (*agree*, *disagree*, *neutral*). We report the results we obtained after reimplementing their approach.

We compare the performance of the above baselines with our pipeline method, which we call *StepByStep*. Whereas [35] may be considered as an additional baseline, we were not able to compare performance, since the data used in this method is not provided by the authors (such as the used vocabulary for contradiction indicators), and this method disregards the *neutral (discuss)* class used in the FNC-I dataset and our work.

3.4.2 Evaluation results

Overall classification performance

Table 4.2 shows the performance of all the approaches. First, we notice that considering the FNC evaluation measure, *LearnedMMD* achieves the highest performance (0.85). All other baselines (apart from *Majority vote* and *FNC baseline* whose performance is poor) have a similar score close to 0.82 (from 0.81 to 0.83).

Considering the macro-averaged F1 measure (F1^m), we notice that the ranking of the top performing systems is now very different. Our pipeline approach (*StepByStep*) achieves the highest score (0.62), slightly outperforming the best baseline system (*stackLSTM*) by one percentage point. On the contrary, *LearnedMMD* is now in the fifth position (F1^m = 0.58). This overall performance gain of *StepByStep* is, in particular, due to the robustness of the classifier in predicting the *disagree* class, which is particularly difficult to classify due to the low number of training instances. Specifically, our method improves the F1 score of this class by 28% (from 0.18 to 0.23)

Table 3.5: Document stance classification performance.

System	FNC	$F1^m$	$F1_{\text{Unrel.}}$	$F1_{\text{Neutral}}$	$F1_{\text{Agree}}$	$F1_{\text{Disagr.}}$	$F1^m_{\text{Agr/Dis}}$
Majority vote	0.39	0.21	0.84	0.00	0.00	0.00	0.00
FNC baseline	0.75	0.45	0.96	0.69	0.15	0.02	0.09
SOLAT [45]	0.82	0.58	0.99	0.76	**0.54**	0.03	0.29
Athene [46]	0.82	0.60	0.99	**0.78**	0.49	0.15	0.32
UCLMR [47]	0.82	0.58	0.99	0.75	0.48	0.11	0.30
CombNSE [48]	0.83	0.59	0.98	0.77	0.49	0.11	0.30
StackLSTM [33]	0.82	0.61	0.99	0.76	0.50	0.18	0.34
LearnedMMD [50]	**0.85**	0.58	**1.00**	**0.78**	0.52	0.03	0.28
StepByStep	0.81	**0.62**	0.97	0.75	0.53	**0.23**	**0.38**

compared to the best performing baseline for the same class (*stackLSTM*). All the other comparing baselines achieve less than 0.15 F1 score for this class. With respect to the *agree* class (the other important class in our fake news context), *SOLAT* is the top performing system, slightly outperforming our pipeline method by one percentage point (F1). However, *SOLAT* performs poorly on the *disagree* class achieving only 3% F1 score. Considering now both *agree* and *disagree*, our pipeline method achieves the highest macro-averaged F1 score ($F1^m_{\text{Agr/Dis}}$), improving the state-of-the-art performance by around 12%.

With respect to the other two classes (*unrelated* and *neutral*), we first see that *unrelated* achieves a very high F1 score for all the methods. This is expected given the nature of the classification problem and that the majority of instances belong to this class. In the *neutral* class, the top performing systems (*Athene* and *LearnedMMD*) achieve 0.78 F1 score while our approach gives 0.75. Note however that, as we have already argued, similar to the *unrelated* class the *neutral* class is not usually useful in fact checking.

Detailed per-stage performance of StepByStep

We now study the performance of each stage separately as well as the detailed performance of our pipeline system in terms of per-class and macro-averaged precision (P), recall (R) and F1 score. Table 3.6 shows the results.

With respect to stage 1 of our pipeline (relevance classification), precision and recall of the *related* class (the important class is this stage) is 0.91 and 0.93, respectively. We notice that both values are very high, although the data is very imbalanced as shown in Table 3.2 (*unrelated* corresponds to around 28% of all test instances).

Table 3.6: Class-wise performance of the different pipeline stages and of the entire pipeline system.

Stage	Class	P	R	F1
Stage 1	Unrelated	0.97	0.96	0.97
	Related	0.91	0.93	0.92
	Macro-averaged:	0.94	0.95	0.95
Stage 2	Neutral	0.82	0.80	0.81
	Stance	0.67	0.71	0.69
	Macro-averaged:	0.75	0.76	0.75
Stage 3	Agree	0.79	0.75	0.77
	Disagree	0.40	0.44	0.42
	Macro-averaged:	0.60	0.60	0.60
Pipeline	Unrelated	0.97	0.96	0.97
	Neutral	0.74	0.76	0.75
	Agree	0.52	0.53	0.53
	Disagree	0.22	0.23	0.23
	Macro-averaged:	0.61	0.62	0.62

Regarding stage 2 (neutral/stance classification), precision and recall of the important *stance* class is 0.67 and 0.71, respectively, while that of the *neutral* class is 0.82 and 0.80, respectively. In general, this task is harder than relevance classification (stage 1). Here, again the data is imbalanced, with the *stance* instances corresponding to around 37% of the test instances (see Table 3.3). We see that there is room for improvement for the important *stance* class.

The stage 3 of our pipeline deals with the harder problem of agree/disagree classification. We notice that our classifier performs well on the *agree* class (P = 0.79, R = 0.75), but poorly on the *disagree* class (P = 0.40, R = 0.44). We observe that, even a dedicated classifier which only considers *stance* documents struggles detecting many instances of the *disagree* class. Nevertheless, as we saw in the overall evaluation results (Table 4.2), our method outperforms the existing methods by more than 28% of F1 score. There are two main reasons affecting the performance of the *disagree* class: i) the classifiers of stage 2 and 3 may filter out many instances belonging to this class, ii) there is limited amount of training instances for this class (only 840) and also the data distribution is very imbalanced with the *disagree* class corresponding to only 18.6% of the test instances (Table 3.4).

Observing precision and recall of each class for the whole pipeline approach and comparing these values with the values of the same classes in each different stage, illustrates the effects of the filtering process applied in each stage on the performance

Table 3.7: Confusion matrix of our pipeline system.

	Agree	Disagree	Neutral	Unrelated
Agree	1,006	278	495	124
Disagree	237	160	171	129
Neutral	555	252	3,381	276
Unrelated	127	31	523	17,668

of the next stage(s). For instance, we observe that recall of the *disagree* class is 0.44 in stage 3 but only 0.23 overall. The same problem exists for the *agree* class (from 0.75 to 0.53). This suggests that many *stance* documents are misclassified in the previous stages, making the task of stage 3 harder. We try to better understand this problem through an error analysis in the following subsection.

Error analysis

Table 3.7 shows the confusion matrix of our pipeline system. Overall, we note that the *neutral* and *agree* classes seem to be frequently confused, what seems intuitive given the very similar nature of these classes, i.e. a document which discusses a claim without explicitly taking a stance is likely to agree with it. The results for the *disagree* class illustrate that stage 1 misclassifies (as *unrelated*) 18.5% of all *disagree* instances (129 instances, in total), while this percentage is less than 7% for the *agree* and *neutral* classes. This is surprising given the very high performance of stage 1. In stage 2, we see that 171 *disagree* instances (25.5%) are misclassified as *neutral*, while this percentage is similar for the *agree* class (26.2%). Finally, in the last stage, 34% of the *disagree* instances are misclassified as *agree*, which demonstrates the difficulty of this task. As we explained above, the highly unbalanced data distribution and the limited amount of training data are likely to contribute significantly to this picture.

Tables 3.8-3.10 show the confusion matrix of each stage separately, i.e., without the effect of the filtering process. Here again we notice the increasing difficulty of each stage. Stage 1 misclassifies a small number of *related* instances as *unrelated* (less than 8%). Stage 2 misclassifies 29% of the *stance* instances as *neutral*. Finally, stage 3 misclassifies the majority of *disagree* instances (55%) as *agree*, and around 25% of the *agree* instances as *disagree*. It is evident from these results that there is much room for improvement for the last stage of our pipeline.

Table 3.8: Confusion matrix of stage 1.

	Unrelated	Related
Unrelated	17,668	681
Related	529	6,535

Table 3.9: Confusion matrix of stage 2.

	Neutral	Stance
Neutral	3,575	889
Stance	760	1,840

Table 3.10: Confusion matrix of stage 3.

	Agree	Disagree
Agree	1,436	467
Disagree	387	310

To better understand the misclassification problem, we further analyze several misclassified *disagree* instances. We observe that the majority of cases concern a small number of distinct claims. In stage 1, for instance, 56/129 (43.4%) misclassifications are associated with one claim (the claim *"Florida woman underwent surgery to add a third breast"*). The instances of this claim were misclassified as *unrelated* because of vocabulary mismatch: different words are used to express *"breast"* in the claim and the documents (*"boob"*, *"breast"*), and the distance of these words in the embedded vector is, surprisingly, high. Similarly, in stages 2 and 3, there are 40/171 (23.4%) and 49/237 (20.7%) misclassifications, respectively, associated with only one claim. In stage 2, the reason is mainly due to the lack of semantic understanding of the text used to express the disagreement to the claim. For example, a misclassified document uses the phrase *"shot down a report claiming..."* in the 2nd paragraph, while the remaining part of the document does not discuss about the claim itself. In stage 3, the reason is again lack of understanding of similar semantics. As an example, a misclassified document uses the text *"all of it is bullsh*t"*, and *"it is a nice mixture of folklore and truth"*. In these cases, our model fails to understand that the words *bullsh*t* and *folklore* negate the claim.

3.5 Related Work

Stance detection is a classification problem in natural language processing where the stance of a *(piece of) text* towards a particular *target* is explored. Stance detection has been applied in different contexts, including *social media* (stance of a tweet towards an entity or topic) [38, 40, 41, 51, 52], *online debates* (stance of a user post or argument/claim towards a controversial topic or statement) [53–55], and *news media* (stance of an article towards a claim) [31–33]. Our work falls under the context of *news media* where the ultimate objective is the detection of fake news. Below we discuss related works on this area and the difference of our approach.

The 4-class stance classification problem for news media was introduced in the context of the Fake News Challenge (FNC) [31], as *"a helpful first step towards identifying fake news"*.[9] The organizers made available a ground truth dataset as well as a simple baseline method that uses a set of hand-coded features and a gradient boosting classifier. 50 teams participated in FNC using a wide array of techniques. The top performing system (*SOLAT in the SWEN*) used an ensemble of a deep CNN model with embedded word vectors and a gradient boosted tree with lexical features. The second-ranked system (*Athene UKP Lab*) used an MLP classifier with six hidden and a softmax layer. The third-ranked system (*UCL Machine Reading*) used a simple MLP classifier with a single hidden layer. More details about these approaches, together with links to the source codes, are provided in Section 4.4.2.

For the same problem, [48] proposed a deep MLP model with combined neural, statistical, and external features, achieving a state of the art performance. [33] proposed the use of macro-averaged F1 score for evaluating performance in this task because this metric is less affected by highly imbalanced datasets. Moreover, the authors proposed a novel approach which uses a stacked LSTM network and 50-dimensional GloVe word embeddings [49], outperforming all previous methods on macro-averaged F1 score.

All these methods treat the problem as a single 4-class classification task. [35] proposed a different, two-stage approach where *unrelated* documents are first filtered out through relevance classification, and then the *related* documents are classified as *contradict* (*disagree*) or *support* (*agree*), using a gradient boosted decision tree model trained on n-gram features extracted using a specially-designed *contradiction vocabulary*. However, this model ignores the *discuss* class which is very prominent in the FNC dataset. Moreover, the authors do not provide their evaluation datasets and the used contradiction vocabulary. A recent work reported in [50] also applies a two-stage

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

approach where a first stage distinguishes *related* from *unrelated* documents and a second stage detects the actual stance (*agree*, *disagree*, *neutral*). An hierarchical neural network that controls the error propagation between the two stages using a Maximum Mean Discrepancy regularizer [56] has been proposed. We have not been able to replicate the results as reported in [50] after we reimplemented the architecture and having contacted the authors in this regard.

With respect to evaluation datasets, a range of datasets has been made available for related stance classification problems, for instance, for detecting the stance of ideological debates (for/against the debate topic) [57], context-dependent arguments/claims (pro/con a controversial statement) [55], or tweets (favor/against a controversial topic) [51]. However, the ground truth dataset provided by the FNC [31] is, to the best of our knowledge, the only one focusing on the *four-class* stance classification task addressed in this paper, whereas the aforementioned datasets address either binary or three-class classification problems (thereby our three-stage pipeline method is not applicable) and focus on detecting the stance of *user opinions* regarding topics, as opposed to the stance of Web documents (like news articles) towards a given true or false claim.

To our knowledge, no previous work has proposed a *three-stage* pipeline approach for the four-class stance classification problem. We show how, through such a pipeline architecture, we can use different models and features in each stage, thus enabling to better focus on the important class of each stage using simple classification models.

3.6 Conclusion

We have proposed a novel three-stage pipeline for the problem of document stance classification towards claims. Such an approach allows to divide the initial four-class classification problem into three different but connected binary classification tasks. This enables the use of different classifiers and features in each step, crucial to further optimize performance on a per step and class basis. Experimental results on the benchmark dataset demonstrated the state-of-the-art performance of our pipeline model and its ability to improve the performance of the important (for fake news detection) *disagree* class by 28% (F1 score) without significantly affecting the performance of the *agree* class. The results also showed that there is still room for further improvements, mainly due to the lack of semantic understanding of the language used to express agreement or disagreement. As part of future work, we are planning to focus on this problem, aiming to reduce the number of misclassified *agree* and *disagree* documents of each stage.

CHAPTER 4

MulCoB-MulFaV: Multimodal Content Based Multilingual Fact Verification

Verifying fact of multimodal reports is emerging as an important challenge to prevent circulation of fake news reports in various online platform. Moreover, such reports often being posted in local languages penetrate even faster, making the problem more complex. To tackle such problem many fact verifying websites have emerged, which deploys humans to manually find the truthfulness of such reports with the help of multimodal contents that are being used in such report. But in recent times due to infinite growth of the problem owing to various political, and malicious motives, the existing manual system fails to handle the enormity. Existing content based approaches relies only on textual content of such reports and fails to handle the multimodal nature. On the other hand existing multimodal based approaches relies only on report based and user based features, and fails to interpret the fake content with evidence. In this work we propose a novel end to end automated multimodal content based multilingual fact verification system, which automates the task of fact verifying websites, and provides evidence for every judgment. Evaluation results on three benchmark dataset shows the robustness and effectiveness of our approach.

4.1 Introduction

> “ Use a picture. It’s worth a thousand words"   Arthur Brisbane

Since the beginning of *Photojournalism* use of pictures in support of a report has been able to garner more attention and attraction of its readers, and leverage its credibility. In the current age of social media, blogs, online messenger, etc., this mode of news distribution has found exponentially increasing impact in convincing it’s readers. As a benefit, now events and matters which needs immediate and urgent attention, spreads quickly, creating the opportunity for timely action. However, as more number of people started relying heavily on social, and other online media platform as their main source of information, a major drawback is that fake news[1] spreading rapidly and becoming almost omnipresent in every demographics of the current society.

Fake news is a report with the intent of spreading misinformation. The impact of Fake News for political gains has been a hot topic for discussion, especially at places with low technological literacy. It was reported before 2019 general elections in India that the election would be heavily impacted by fake news[2]. Most of the political parties have had their respective IT cells, and are often accused of spending heavily on generating and circulating fake news, leading to defamation of personnel, hate spread resulting in religious tension, mob lynching and misconception creation, diversion of attention from main election issues, etc. Fake claims when made with support of false or doctored multimedia evidence tends to reach a wider audience with greater convincing ability [3]. Some common patterns these fake reports shows are misuse of images taken from a completely unrelated event, use of morphed images, resurfacing claims that were debunked as false previously, etc. Additionally, these false claims are posted in local languages in general to increase its reach further. To control this pool of misinformation a few fact checking websites (e.g PolitiFacts, AltNews), and news media (e.g India Today) emerged as fact verifiers, which deploy teams who manually checks the veracity.

The approach most of these teams follows are a sequence of the following processes:

- search related documents, and related pictures of the reports
- look into these related files
- find evidence that may support or refute the claim
- assess judgment based on the evidence found

But, given the enormity of the scale of misinformation, diversity of language and the lack of tendency to verify information, this current fact checking mechanism fails to address the problem to a reasonable extent. Thus, creating the need of automation of the procedure. In this context content based fake news detection emerged as viable solution which automates the mentioned fact checking mechanism. However, all the

[1] explained-fake-news-asia

[2] india-misinformation-election-fake-news

existing works [27, 58, 59] are targeted towards single modality (textual) data, and fails to handle multimodal fake news surfacing across various online platforms.

In this paper we work towards developing a Multimodal Content Based Multilingual Fact Verification systems across any online platform that automates the steps of fact verification in real time, where we give importance to timely delivery of the judgment. We propose a system of two level sequence to learn and predict the veracity of the report (or post) as **“Real"** or **“Fake"**. The first level aims at mining **k** most relevant images and corresponding articles with respect to the report. The second level uses the mined information from level one and aims at binary classification of the report (Real/Fake).

Evaluation results on benchmark dataset shows effectiveness of our approach. A careful analysis on the outputs produced by the system reveals a number of limitations of our approach and demonstrated that there is room for further improvements.

Briefly stating, we make the following contributions:

- We introduce a multilingual multimodal system to verfiy veracity of a multimodal report.
- We introduce a two level sequential approach to mimic and automate the fact verification procedure of existing fact verification websites in real time.
- We provide the evidence based on which our system makes the judgment.

The rest of the paper is organized as follows: Sect. 4.2 formulates the problem and provides an overview of our approach. Sect. 4.3 describes the implementation details. Sect. 4.4 reports evaluation results. Sect. 4.5 presents related works. Finally, Sect. 4.6 concludes the paper.

4.2 Problem Modeling and Approach Overview

Given a multimodal report (or post) r consisting of a multilingual textual claim c (e.g., *“the earth is flat”*) and an image i (e.g., image of flat earth), the fact verification task is to classify the accumulated knowledge of c and i, to one of the following two categories (classes):

- *Fake:* the report is false and spreads misinformation,
- *Real:* the report is true.

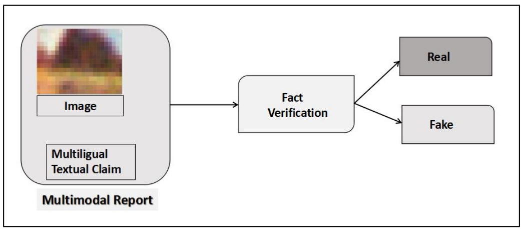


Figure 4.1: Given PROBLEM MODEL.

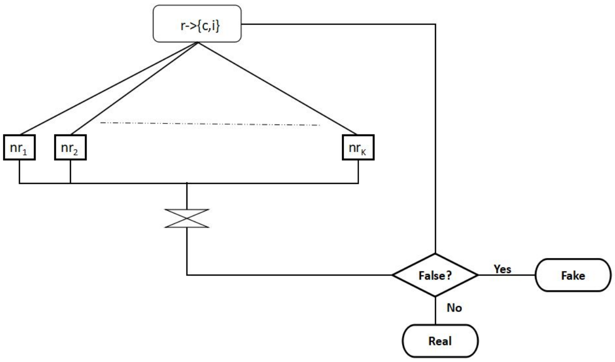


Figure 4.2: Flowchart diagram for Fake report detection showing sequences of job. r, c, i, and nr_j are given report, textual claim, image, and jth related news report respectively.

The pictorial overview of the given problem model is depicted in Figure 4.1. Now each r may have a set of 0 to k related news reports N_r=$\{nr_j\}$, $j \in [0, k]$, where each nr_j consists of an image ir_j and a textual article ar_j, which may provide knowledge required to classify r. A nr_j is said to be related to r *if* relevance between ir_j and i, *relv*(ir_j,i) is above a threshold limit θ i.e

$$nr_j \in N_r \rightarrow relv(ir_j, i) > \theta \tag{4.1}$$

Considering the problem structure in hand now we design our approach. We model the problem into the following two level sequence:

- **First Level (Related news mining):** mine N_r for a given r.
- **Second Level (Veracity classification):** accumulate all the knowledge from N_r, compare it with information from r, and identify if the given r is *Fake* or *Real*.

Figure 4.2 depicts flowchart diagram of the proposed sequential architecture. We hypothesize that the mined related news reports N_r in the first level can provide useful evidence validating or refuting the report r, and accumulating these knowledge from evidence found in first level can facilitate veracity detection of r into classes *Fake* and *Real* in second level.

4.3 Implementation

This section describes in details about the implementation of the algorithms we use in our sequential fact verification system. The first part describes about our approach to find and mine the related news reports for each given input report. The second part narrates the supervised approach we follow to address the classification problem at hand.

4.3.1 First Level : Related News Mining

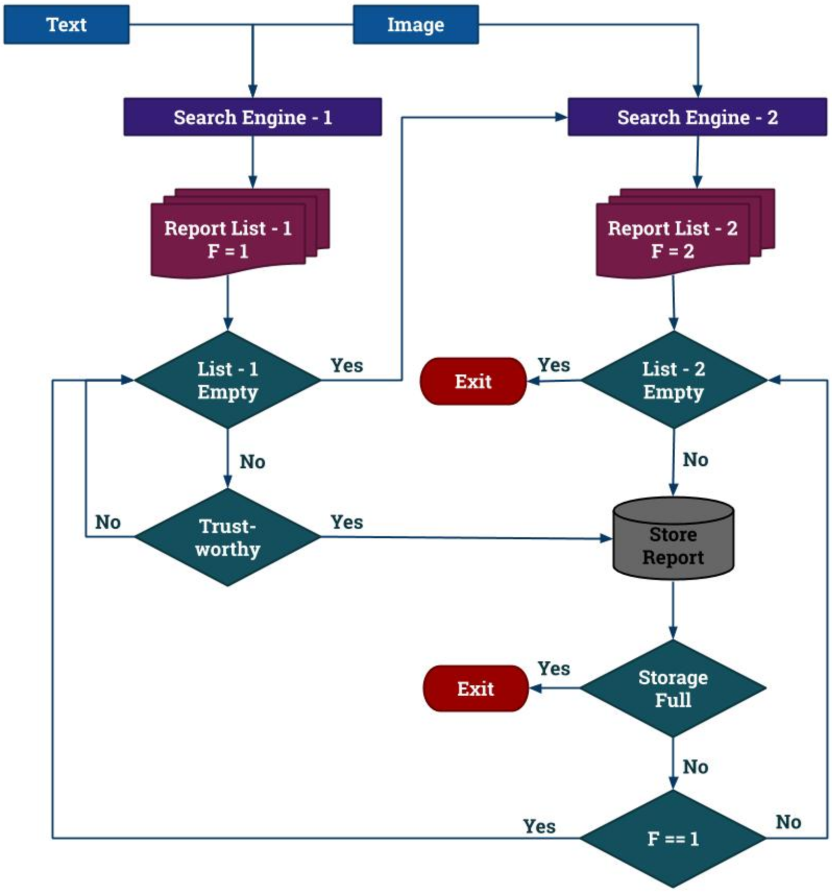


Figure 4.3: Flowchart Diagram of the procedure of mining related news reports.

Content based information retrieval (CBIR) is a long studied problem [60]. In context of fact verification also CBIR has been applied [27, 58, 59] based on single modality (textual data) content. In this work we propose an web search based algorithm to extract relevant multimodal contents from the web using the procedure as depicted as flowchart in Fig.4.3. We use two different web search engines depending on the search criteria, and the efficiency of the search engine on the type of input, to mine the data. This use of two different search engine is finalised based on experimental analysis. At first we collect resultant reporting documents from the first search engine using both image and text of the report r. Then, we check each reporting document individually that whether it is coming from a trustworthy source or not. If ‘no’ then we move to the next reporting document until the list is empty. If ‘yes’ then we append it to the set of related reports N_r and check if the exiting criteria of collecting k number of related reports nr is achieved or not. If ‘yes’ we exit. If ‘no’ then we again go the next reporting document until the list is empty. If empty, then we move to the next search engine and retrieve a new list of related reporting documents using search by image i of r only. Now we keep on appending the reporting documents as nr until the exiting criteria of k is achieved or the list is empty. To note, we do not consider duplicate reports and reporting documents from social media and blogspots in the set N_r.

4.3.2 Second Level : Veracity Classification

In this level at broader sense we need a binary classifier which learns from the mined information from the first level, verify the report given as input against the currently learned knowledge, and predict the veracity of the report into either *Real* or *Fake*. As discussed in sec.4.2, each input report r contains an image i, and a multilingual textual claim c. A related news report nr, contains an image ir and a textual article ar, which is mined from web using i and c used in r as described in sec.4.3.1. Hence, the similarity between the two images i and ir is given importance, and relevance of the knowledge from nr as evidence for verification of r needs to be prioritized in accordance to this similarity. Instead of the entire knowledge from nr there may be certain parts of nr (at times text, at times image, and at times both) which are more relevant to c and needs specific attention. Thus, the sub-goals in hand that needs to be addressed are methods to learn from text data of each ar, image data from i and each ir, handle multilingual input text data of c, learn to arrange priority of knowledge from each nr based on similarity between ir and i, form a single representation vector for each nr, give attention to find the most relevant part of nr w.r.t c, accumulate all the attended knowledge from the set N_r of all the nr to form a single evidence vector vec_e, and finally learn the veracity of r from vec_e. The methods we use to solve each of the sub-goals are as follows:

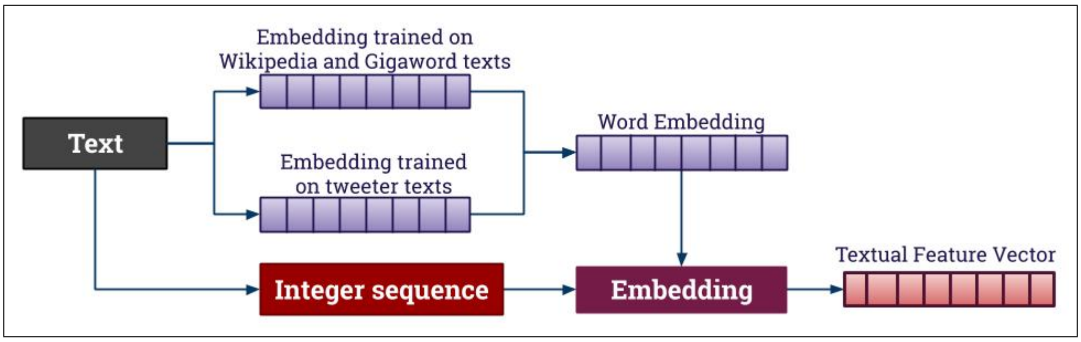


Figure 4.4: Diagram showing embedding of text using two different pre-trained word vectors.

Processing text data (ar and c): In order to process the knowledge from textual data we need to convert the words into vectors. A common practice is embedding the text using pre-trained word vectors. However, a word in context of social media will have different distribution compared to its distribution in context of text of common linguistic domain. To reflect this difference in research works on different domains of formal and informal (social media) linguistic problems, [49] introduced Glove vectors model for formal domain trained on Wikipedia and Gigaword texts (Vec_w), and model for informal domain trained on tweeter texts (Vec_t). Though, both the pre-trained word vec models are trained using the same skip-gram methodology, but due different word distribution in train data, same word gets projected to different direction within the same vector space R^n. This difference has been depicted with an example in a plotted graph (fig.4.5), where embedded vectors of the same word “Fake" using the two different models Vec_w and Vec_t are plotted against each other in $100d$ euclidean space. In the problem we solve, as the claim texts c may come from any source report, expected to contain informal text, whereas the related news reports nr coming from reliable news media, contains published articles ar, expected to contain formal text. Henceforth, using any one of Vec_w and Vec_t to embed both ar and c will not give a meaningful representation for both. On the other hand comparing representation coming from two different embedding distribution, to find and match patterns, will be unfair. Thus, inspired from [61], we decide to use a common representation to embed both c and ar using a combined model Vec_{com} formed by concatenating Vec_w and Vec_t. The embedding method is illustrated by the Fig.4.4

$$Vec_{com} = Vec_w \oplus Vec_t, \tag{4.2}$$

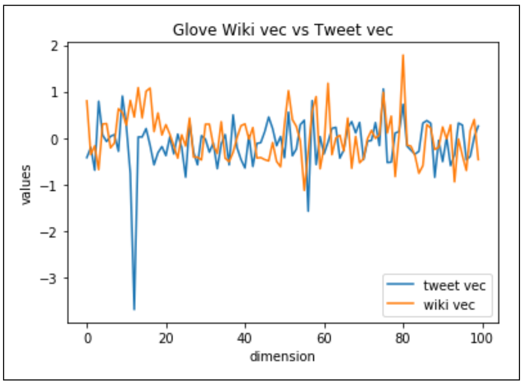


Figure 4.5: Embedded Vectors of word ’Fake’ using Glove Wikipedia vecs vs Glove Tweeter vecs in 100d space

where Vec_w, $Vec_t \in R^n$, $Vec_{com} \in R^{2n}$. Moreover, c comes from multilingual text source, whereas Vec_{com} models embed vectors for only English words. Hence, we translate the text data of c into English text t_c using google translate API. Using Vec_{com} we get the embedded vector E_{ar} from each ar, and embedded vector E_c from t_c.

Representing related articles: To get the representation feature vector B_{ar} of each nr, we pass each Ear obtained in the previous layer through a bidirectional recurrent layer of n LSTM units in each direction.

Processing image data (i and ir): Although images can be directly vectorized using pixel value of each position, but over the last few years the practice of transfer learning by use of existing pre-trained model to extract image features is becoming popular among researchers. To extract image features from both image of input report i and image of related report ir we experiment with different available pre-trained models like VGG19, ResNet, InceptionV2, InceptionResNetV2, etc trained with and without ImageNet [62] data. We find that InceptionResNetV2 [63] a flavour of CNN trained on ImageNet data gives the image features best suited for our classification model. We use this pre-trained model to get feature tensors F_{ir} from each ir and F_i from i.

Learning priority of related report: In our proposed system we try to learn the priority of each related report nr based on the similarity between the tensors representing each ir with the tensor representing given report image i. We find this similarity sim(i,ir) based on the cosine of the angle between the two tensors. This similarity value is then multiplied with the tensor ir to generate the priority based image vector p_r of

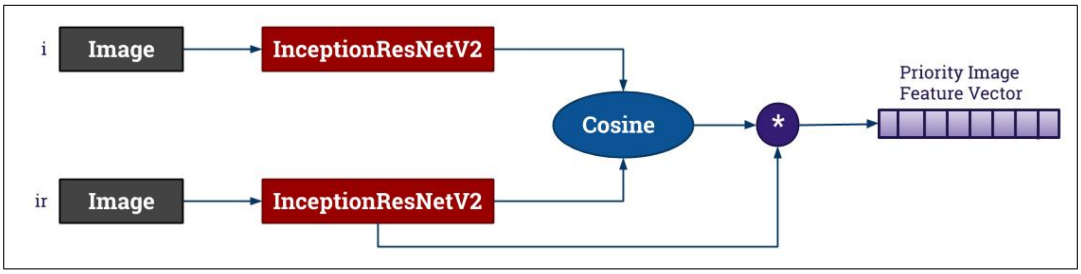


Figure 4.6: Diagram to show generation of priority based image vector p_r.

the respective related report.

$$sim(a, b) = \frac{a \cdot b}{\|a\| \|b\|} \tag{4.3}$$

$$p_r = sim(F_i, F_{ir}) \times F_{ir} \tag{4.4}$$

The procedure of generating the priority based image vector is depicted in Fig.4.6.

Representation of multimodal related report (nr): After generating feature vector B_{ar} and prioritized feature tensor p_r from each related report nr, we form the representation Re_{nr} of each nr by concatenating B_{ar} and p_r.

$$Re_{nr} = B_{ar} \oplus p_r \tag{4.5}$$

Attending relevant part of related report: To get the most dominant part of each related report nr we take each representation Re_{nr} and pass it through a fully connected n perceptron layer with *softmax* activation. We get the tensor $V_{den} \in R^n$. To get the most important part of the claim c where the features map is most significant, we take the embedded claim E_c and pass it through a convolutional layer of n number of $f \times dim$ filters ($E_c \in dim$). Then, finally pass through a global max pooling layer fetching the maximum value from each filter to form the tensor $V_{pool} \in R^n$. Now, we attempt to get the most relevant part of nr with respect to c. We multiply the tensor V_{pool} with V_{den} to get the tensor $At \in R^n$. This procedure has been shown in Fig.4.7.

Forming evidence representation (vec_e): After getting the attended tensors At_j for each related report nr_j, we form the evidence representation vec_e by concatenating all

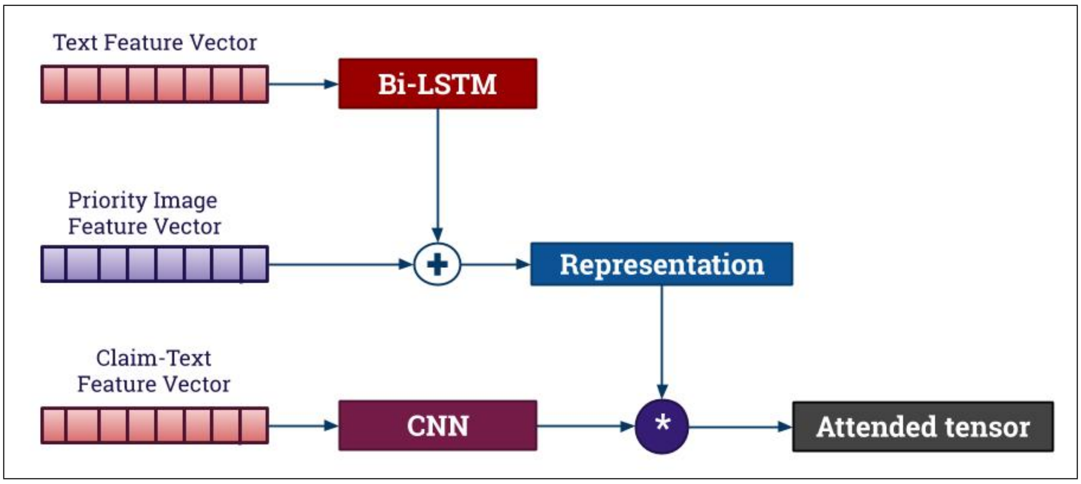


Figure 4.7: Diagram to show formation of attended tensor using representation vector Re_{nr} and embedded claim text feature vectors E_c.

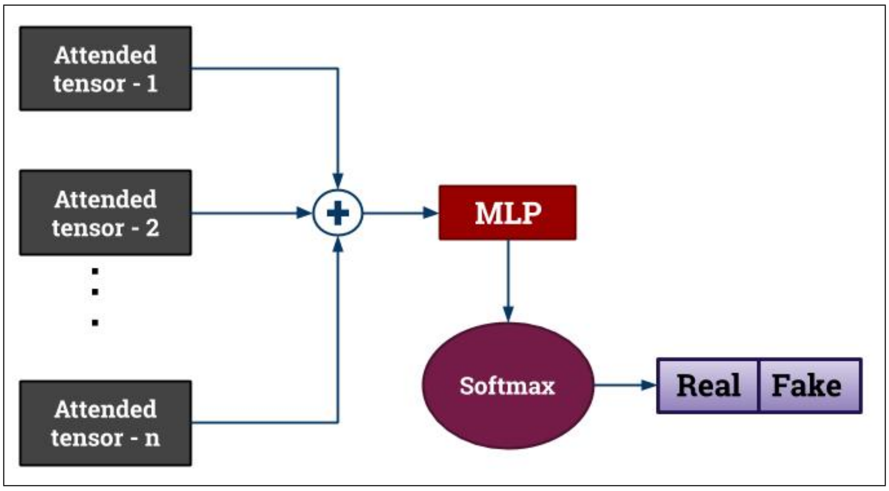


Figure 4.8: Diagram showing formation of evidence vector vec_e and final prediction of veracity of the report based on vec_e.

the k attended tensors.

$$vec_e = At_1 \oplus At_2 \oplus ... \oplus At_j, j \in [0, k] \tag{4.6}$$

Identifying veracity based on evidence: Finally the evidence representation is passed to a fully connected layer with *softmax* activation to get the final prediction of *Real* or *Fake*. Fig.4.8 illustrates the procedure of generation of vec_e and the final prediction based on it.

We use Log likelihood loss function ζ to calculate the prediction loss against the True target. *Adam* a variation of gradient descent method is used to update and find optimal weights across the network in order to optimize the loss function ζ.

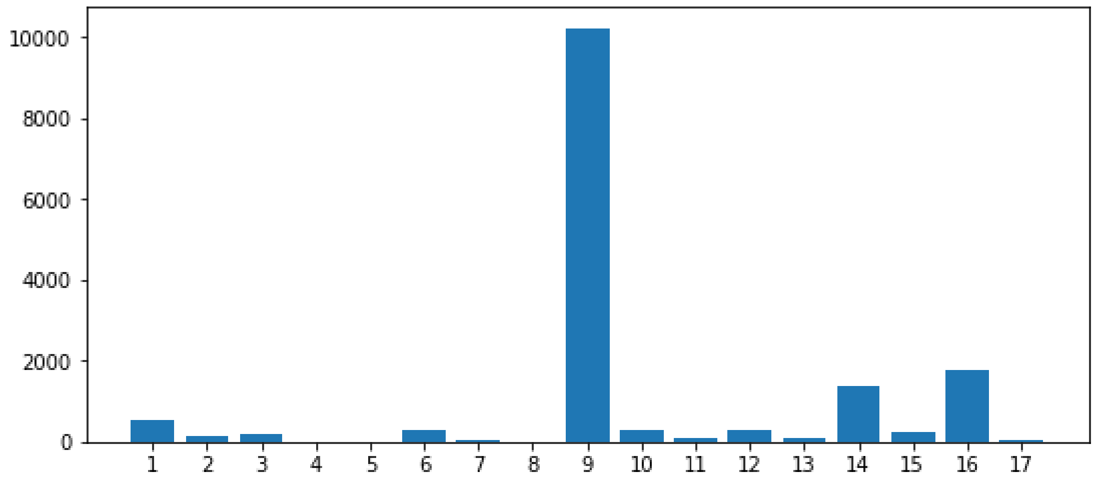


Figure 4.9: Bar Graph showing distribution of posts across all the 17 events in MediaEval15 dataset1. Here 1: **Boston Marathon Bombing**, 2: **Bringback Our girls**, 3: **Columbian Chemicals**, 4: **Rock Elephant**, 5: **Livr mobile app**, 6: **MA flight 370**, 7:**Passport hoax**, 8:**Pig Fish**, 9:**Hurricane Sandy**, 10:**Sochi Olympics**, 11:**Underwater bedroom**, 12:**Solar Eclipse**, 13:**Garissa Attack**, 14:**Nepal Earthquake**, 15:**Girl with Samurai boots**, 16:**Syrian boy**, 17:**Varoufakis and zdf**

$$\zeta = \sum \gamma \ln \hat{\gamma} \tag{4.7}$$

Where γ is the True target label, and $\hat{\gamma}$ is the predicted label.

4.4 Evaluation

4.4.1 Dataset

To evaluate our methods we use the ground truth dataset provided in the image verification corpus [64]. The corpus contains three evolving dataset of multimodal fake and real social media posts. The detailed description of each of the dataset are given below.

Dataset1 (D1): Originally published for verification task of MediaEval 2015 [65], this dataset contains a total of 18262 multimodal posts. The posts are distributed across 17 different events, namely, 1: *Boston Marathon Bombing*, 2: *Bringback Our girls*, 3: *Columbian Chemicals*, 4: *Rock Elephant*, 5: *Livr mobile app*, 6: *MA flight 370*, 7:*Passport hoax*, 8:*Pig Fish*, 9:*Hurricane Sandy*, 10:*Sochi Olympics*, 11:*Underwater bed-*

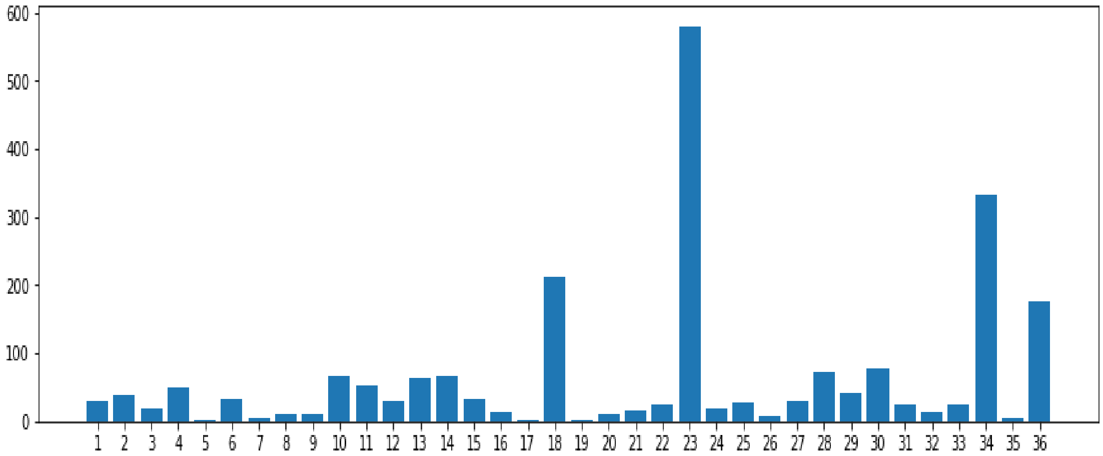


Figure 4.10: Bar Graph showing distribution of posts across all the 36 events in test set of MediaEval16. Here 1: **Gandhi Dancing**, 2: **Half of Everything**, 3: **Hubble Telescope**, 4: **ImmigrantsâĂŹ fear**, 5: **ISIS children**, 6: **John Guevara**, 7: **Mc DonaldsâĂŹ Fee**, 8: **Nazi Submarine**, 9: **North Korea**, 10: **Not Afraid**, 11:**Pakistan Explosion**, 12:**Pope Francis**, 13:**Protest**, 14:**Refugees**, 15:**Rio Moon**, 16:**Snowboard Girl**, 17:**Soldier Stealing**, 18: **Syrian Children**, 19: **Ukrainian Nazi**, 20: **Woman 14 children**, 21: **American Soldier Quran**, 22: **Airstrikes**, 23: **Attacks in Paris**, 24 :**Ankara Explosions**, 25: **Bush book**, 26: **Black Lion**, 27: **Boko Haram**, 28: **Bowie David**, 29: **Brussels Car Metro**, 30: **Brussels Explosions**, 31: **Burst in KFC**, 32: **Convoy Explosion Turkey**, 33: **Donald Trump Attacker**, 34: **Eagle Kid**, 35: **Five Headed Snake**, 36: **Fuji Lenticular Clouds**

room, 12:*Solar Eclipse*, 13:*Garissa Attack*, 14:*Nepal Earthquake*, 15:*Girl with Samurai boots*, 16:*Syrian boy*, 17:*Varoufakis and zdf* and labelled into three different categories: 9404 posts labelled as “**Fake**", 6225 posts labelled as “**Real**", and 2633 posts labelled as “**Humor**". However, detecting “Humor" is altogether a different class of problem, involves sarcasm. Sarcasm [23] although may contain misinformation, but the intent is different than that of spreading fake news. Thus, in this work we disregard all the posts labelled as “Humor". The distribution of all the posts (discarding ‘Humor’) across the events is shown in a plotted bar graph (Figure: 4.9).

Dataset2 (D2): Next we consider the dataset published as a sequel to Dataset1 as verification task in MediaEval 2016 [66]. This is an extension of the previous dataset, where all the data of [65] used as development set, and a set of additional 2228 multi-modal posts spread across 36 new events is used as test set. Out of these 1230 posts are labelled as “**Fake**", and 998 posts are labelled as “**Real**". The plotted graph showing distribution of the test data across the different events is given in Figure 4.10.

Dataset	**Fake**	**Real**	**Total**
D1	9,404	6,225	15,629
D2	1,230	998	2,228
D3	1,255	1,178	2,433

Table 4.1: Class-wise data distribution of the all dataset. D1 is Dataset1, D2 is Dataset2, and D3 is Dataset3.

Dataset3 (D3): This dataset is a subset of the dataset1 [65]. The dataset contains 2433 multimodal posts, formed by removing near-duplicate posts from the original dataset. In the dataset 1255 posts are labelled as **"Fake"**, and 1178 posts are labelled as **"Real"**.

The class-wise and total data distribution of all the dataset is given in table 4.1.

4.4.2 Evaluation Setup

Hyperparameters: For embedding textual data we use $100d$ of Vec_w, and $100d$ of Vec_t. Thus, each E_{ar} and E_c is represented by a Vec_{com} of $200d$. In each of Bi-LSTM layer we use 500 LSTM unit in each direction with 'ReLu' as nonlinear activation function. The fully connected layer taking Re_{nr} as input has 500 perceptron units with 'Softmax' activation function. The convolutional layer taking E_c as input, has 500 filters with each having kernel size 1×200 with 'Relu' activation function. The final layer has 2 perceptron units with 'Softmax' activation.

Evaluation Metrics: The evaluation metrics are used to evaluate the participating teams in tasks [65] and [66] are class-wise precision (P), recall (R), and F1-score (F), and overall macro average F1 (F^m). P gives the measure of how much accurate prediction towards a particular class is given all the predictions made in favor of the class, R gives the measure of accuracy of prediction towards a particular class given all the original target instances of the class. Both P and R have their drawbacks, as high P report for a particular class may occur if a very few prediction is made in favor of the class, on the other hand a very R report for a particular class may occur if all the predictions are made in favor of the class. Thus, prediction towards a particular class can be judged as very accurate and efficient when both P and R is high for the class. This type of judgment is reflected with F which gives a balance measure between P and R. F^m gives the average of F across all classes (two in our case), giving an assessment of the overall performance. Thus, to evaluate the performance of our proposed system against

System	F1^m (D1)	F1^m (D2)	F1^m (D3)	F1^m (Overall)
Majority vote	0.42	0.39	0.35	0.39
MCG-ICT	0.76	0.50	0.75	0.67
CERTH-UNITN	0.69	**0.91**	0.64	0.75
EANN	0.72	0.65	0.71	0.69
MulCoB-MulFaV	**0.79**	0.75	**0.77**	**0.77**

Table 4.2: Comparative evaluation of our proposed system against all the baseline methods. D1, D2, and D3 are Dataset1, Dataset2, and Dataset3 respectively as described in sec.4.4.1.

all the baseline models used, we use F^m, and for in depth analysis of the performance of our proposed system towards predicting each class we use class-wise P, R, and F.

Baselines We consider the below baseline methods to compare our results against:

- *Majority vote*: The class with the maximum number of instances is always selected (*Fake* in our case).
- *MCG-ICT [67]*: A two level classifier, which fuses topic probability features generated from a topic classifier with a message level classifier to get the final veracity prediction.
- *CERTH-UNITN [68]*: A semi-supervised classifier that uses a combination of user level, tweet level and forensic features to predict the veracity of the tweet.
- *EANN [69]*: A multimodal adversarial neural network, that combines a fake detector module with an event discriminator module to learn veracity of a multimodal news without learning the event.

4.4.3 Evaluation results

Overall classification performance. To evaluate comparative performance of our method against all the baseline systems mentioned in sec.4.4.2, for D1 we leave one event out for testing and report the average of cross validation result, for D2 we report test results on the data after training on D1, and for D3 we leave out the intersecting reports between D1 and D3 from training set and then test on D3. Table 4.2 shows the comparative performance of all the approaches, showing that our approach achieves competitive performance against all the baseline system, while being able to interpret every prediction by providing evidence.

With respect to the other baseline systems, we first see that our proposed model *MulCoB-MulFaV* achieves the best performance of **0.78** and **0.77** on D1 and D3 respectively, while achieving the second best performance of 0.75 on D2 in terms of $F1^m$. While averaging over the performances on all the three dataset we achieve the best performance of **0.77** $F1^m$ outperforming the best baseline system (*CERTH-UNITN* [68]) by two percent point. This particular gain in overall performance of *MulCoB-MulFaV* shows that the system is independent of the system on user-based, and event based features, and gives robust and stable performance on new incoming unseen report. The performance of *CERTH-UNITTN* is significantly better on D2, but interestingly decreases by a good margin while evaluating on D1 and D3. *CERTH-UNITTN* makes use of image based forensic report. Even though in D1 (and also trivially D3) there are reports about certain events (e.g Hurricane Sandy) where distorted/doctored images were used extensively and knowledge of forensic report comes very useful, but there are also reports discussing certain events (e.g MH370 Malaysian Ailines, Columbian Chemical plant blast, etc.) where real images (without any distortion) were used, but images were taken from previously reported news about a different event. In these second type of cases forensic report is not expected to provide any crucial feature to identify the veracity of the report, but the list of evidence from related news report in our proposed model based on multimodal content based search provides useful information to the model and helps in detecting the veracity of the report correctly. Table 4.3 describes such cases with the help of examples, where the first two rows shows fake reports using doctored images about the events 'Hurricane Sandy' and 'Greek minister showing middle finger', and the next two rows shows fake reports with real images about events 'MH370 Malaysian Airlines' and 'Columbian Chemical plant blast' respectively.

Detailed per-class and overall performance of *MulCoB-MulFaV*: We now study the detailed performance of our system in terms of per-class precision (P), recall (R) and F1 score. Table 4.4 shows the results. We observe that for each of dataset D1, D2, and D3, measure P for class 'Fake' is very high, which interprets that whenever our system predicts a report as 'Fake', its chances of being a fake report is very high. Although measure R for the same is comparatively lower, but that for class 'Real' is high. It suggests that our proposed system is able to correctly identify real reports more in number. Overall measure F for most of the times (for evaluation on D1 and D2) is

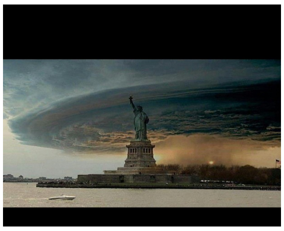
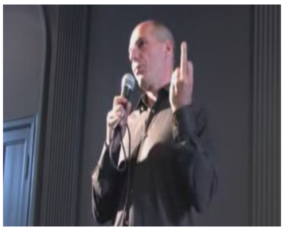
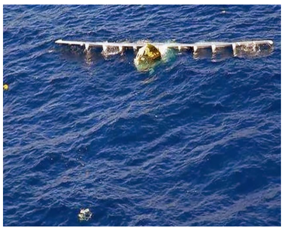
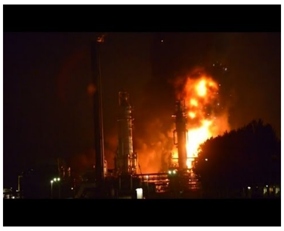
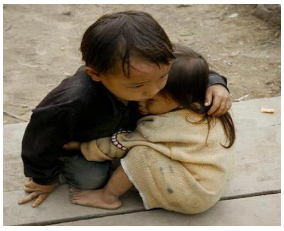
Image	Event	Distortion	Evidence
	Hurricane Sandy	Yes	This photograph claims to show Hurricane Sandy over New York CityâĂŹs Statute of Liberty, as viewed on October 29, 2012. However, this is a poorly **photoshopped** mosaic of a stunning **photograph of a midwest supercell thunderstorm** placed behind a separate photograph of Lady Liberty.
	Greek minister showing middle finger	Yes	German comedian Jan BÃűhmermann claimed he **faked a middle finger** gesture by Greece's former finance minister. He actually didn't, but showed how easy media **can be manipulated**.
	MH370 Malaysian Airlines	No	An aerial view of the fuselage and wings of the **Tunisian plane** that crashed in the sea near the Sicilian city of Palermo, August 6, 2005. REUTERS/Guardia di Finanza/Handout
	Columbian Chemical Pant blast	No	Big bang at the chemical plant in **Zhangzhou City, China**
	Nepal Earthquake	No	Haunting 'Nepal quake victims' **photo from Vietnam**.

Table 4.3: Images shared in different fake reports. The column Distortion narrates whether the image is a doctored/ distorted image or not. Column Evidence highlights the facts based on which our model *MulCoB-MulFaV* accurately judged the veracity of the report.

better for class ‘Fake’, hence proving the usefulness of *MulCoB-MulFaV*.

Dataset	Class	P	R	F1
D1	Fake	0.86	0.78	0.82
	Real	0.71	0.81	0.76
D2	Fake	0.79	0.75	0.77
	Real	0.71	0.74	0.72
D3	Fake	0.80	0.73	0.76
	Real	0.74	0.81	0.77

Table 4.4: Detailed performance of our proposed model on the three dataset D1, D2, and D3. P, R, and F1 are precision, recall, and F1-score measures respectively.

4.4.4 Error Analysis

In sec.4.4.3 we find that though our proposed system is able predict the veracity of a multimodal report with promising results, but still it is not entirely accurate. Moreover, the measure R for class ‘Fake’ is comparatively lower. This suggests that in many reports our model is not able detect the fake content and predicted it as ‘Real’. To analyze this shortcoming we further look into the wrong predictions by *MulCoB-MulFaV* along with the evidence based on which the predictions are being made. We find that for certain events (e.g Samurai Ghosts), although the images used in the report are doctored, but the evidence discovered from trustworthy related news report supports the fake report suggesting it to be ‘Real’. Then, there are certain events (e.g Five Headed snake) for which there is no news from trustworthy source, and the ones from untrustworthy sources gives evidence in support of the wrong prediction. We also find certain reports (e.g Hurricane Sandy) where the claim text of the report describes the fake (doctored/distorted) image posted along the report as ‘Fake’, suggesting the report to be ‘Real’, and our system also predicts the report as ‘Real’, but the given target label is ‘Fake’. All these findings are depicted with examples in Table4.5.

To better understand the misclassification problem on per dataset, we further look into the confusion matrices. Since, for D1 we report average of cross validation result by leaving out one event at a time, we report confusion matrix for evaluation on D2, and D3 in Table 4.6. We notice that on both the occasion i.e for evaluation on D2, and D3,

Image	Claim text	Tar	Pred	Evidence
	'Samurai Ghost': Photo Shows Mysterious Boots Behind Girl	Fake	Real	This photo of a 4-year-old girl on a beach in Zushi, Japan, seems innocent until you look closely behind her legs and back. A closer look appears to reveal a mysterious pair of boots and part of a blue shirt peeking out from behind her.
	Panch Mukhi Sarp(Five Headed Snake) seen in India	Fake	Real	The king cobra (Ophiophagus hannah), also known as the hamadryad, is a venomous snake species in the family Elapidae, endemic to forests from India through Southeast Asia. It is threatened by habitat destruction and has been listed as Vulnerable on the IUCN Red List since 2010.
	Boing Boing: #Fake #Hurricane #Sandy #shark #photo migrates to #Chinese	Fake	Real	The Internet, like the world itself, is particularly prone to create and disseminate false information, such as the one in this photograph that has circulated in the last few hours through social networks and the Internet in general, about the supposed arrival of Hurricane Sandy

Table 4.5: Examples of wrongly predicted reports by *MulCoB-MulFaV*. *Tar* is target/original label, and *Pred* is predicted label.

Dataset		Fake	Real	Support
D2	Fake	923	307	1,230
	Real	245	743	998
D3	Fake	916	339	1,255
	Real	225	953	1,178

Table 4.6: Confusion matrix of *MulCoB-MulFaV* on D2 and D3.

comparatively more number of reports belonging to class 'Fake' gets confused as class 'Real' than vice-versa. This finding further supports the analysis of our system. We find in the previously mentioned examples that this confusion mainly takes place due to lack of proper evidence data to refute the fake report, even though the image being used in the report being a forged (doctored) one. In such cases features such as forensic report of the images is expected to come in handy. As part of our future work we would like to explore the effectiveness of such features on our system in an effort to make our system more robust.

4.5 Related Works

Fake news detection has been applied in different contexts, including *social media* (veracity of tweet, facebook post, etc.) [3, 12, 13, 69, 70], *short statements* (veracity of political statements made by politicians at various context) [18, 71], and *news media* (veracity of claims made in published news articles) [32, 72]. Moreover, depending on the type of input data Fake News detection can be sub categorized into two types, namely, fake news from single modality (mainly from textual data) [70–72], and fake news from multimodality (input containing both textual and visual data) [3, 69, 73].

Fake news detection from single modality: In approaches towards detecting fake news from single modality, use of information from the related documents on the web [27, 58, 59] is emerging as an efficient and meaningful way of solving the problem. These approaches in general are termed as content based (or knowledge based) fact verification. The main usefulness of these approaches are their ability to interpret and explain the fake content and provide evidence for debunking them. The proposed work by [58] and [27] particularly focuses towards detecting fake claims by first retrieving articles from the web referring to the claim text using search engines, then using a distance supervision based classifier they capture different knowledge from these retrieved articles to finally predict the credibility of the claims as true or false. Both the works proposes to capture the interplay of the languages in the retrieved articles, and the reliability of the web sources generating the articles. Each of the retrieved articles assesses credibility of the claim. All the individual assessment are aggregated to give the final prediction. [59] proposed a graph based approach, where they used a set of true and fake articles to construct knowledge graph of triplets, consisting of entities and relation between them. Whenever a new article or claim came, their idea was to extract triplet of entities and relationship from the given text, and then check whether the target triple is true or not based on a given knowledge graph. [74] in their work proposes an end to end deep neural model *DeClarE*, where they does credibility assessment without using any handcrafted features. They search related articles from the web using the claim, then applies claim specific attention on the articles to find the most relevant part of the article with respect to the claim. The attention mechanism helps their model to capture useful words in an article which is being used as evidence for the verdict of their system.

Fake news detection from multimodality: The problem statement in fake news detection from multimodality is more complex in nature and involves learning veracity of a report from input type of more than one modality (mainly from two types of modality, image and text). In this regard [64] introduced the image verification corpus which is an evolving corpus of fake and real posts with images shared in social media. The corpus contains dataset introduced as verification task in MediaEval 2015 [65] and in MediaEval 2016 [66]. Recent works towards multimodal fact verification [3, 69] explores extraction of image features using pre-trained CNN models, and text data features using pre-trained word embedding models followed deep neural network, then merging the image and text features to feed into another deep neural network to learn the veracity of the multimodal reports. While all the works towards this problem including the works [67, 68] by the participating teams in [65, 66] showed high performance and impressive result, but they all failed to interpret or explain the fake content or provide any evidence for the predicted verdict.

Our work falls under the context of *social media* where the objective is the detection of fake news from multimodality data. In contrast to the existing works on this problem our approach provides evidence for each of the judgment, which interprets the fake content present in the report. Moreover, our work gives an end to end automated real time solution to fake news detection problem which unlike [12, 13, 67, 68, 70] does not depend on domain specific, user specific features, or require any additional knowledge along with the report, and is able to detect veracity of a report irrespective of its source.

To our knowledge, no previous work has proposed such an end to end automated, real time, domain independent, interpretable content based solution to multimodal fake news detection problem, thus justifying novelty and real world applicability of our approach.

4.6 Conclusion

In this work we have proposed a novel multimodal content based approach for the problem of multimodal multilingual fake news detection. Such an approach allows us to automate the work of fact verification which is being followed by established fact verifying websites, to manually detect veracity of multimodal reports. Our approach also able to justify each prediction and interpret the fake contents in such fake reports by providing evidence, collected from the news reports related to such fake multimodal contents. Experimental results on three benchmark dataset demonstrated the robustness of our proposed approach by achieving best overall performance compared to all the baseline methods. Evaluation and analysis of results also exhibits that there is possibility for further improvements, mainly due to the lack of forensic understanding of doctored multimodal contents. As part of our future work, we are planning to focus on including such features to make our system have better understanding of the problem, aiming to reduce the number of misclassification of 'Fake' multimodal reports as 'Real'.

CHAPTER 5

Conclusion and Future Direction

In this thesis work we first explored fake news spread by politicians via statements made by them. As politicians have wide reach among masses, thus fake information coming from them spreads faster and have a deep impact in society. Next, we acknowledged that the modern settings of social media and dependence of large number of masses on them as the main source of information, enables any person to spread false claims very rapidly. In this context, we worked towards detecting documents, which expresses opinion toward textual claims. Finally, we explored the problem of multimodal multilingual fake reports. We provide automated individual solutions to address each of the problems. Evaluation results on publicly available benchmark datasets shows effectiveness of each of our proposed approaches. In future we want to work towards minimizing the errors found for each of our proposed approaches, as discussed in each respective chapter. Additionally, we are also working towards developing and deploying a real time application, to detect fake reports (of any modality) in context of Indian politics.

Publications

Arjun Roy, Kingshuk Basak, Asif Ekbal, Pushpak Bhattacharyya. "A Deep Ensemble Framework for Fake News Detection and Classification". In proceedings of CICLing 2019.

Arjun Roy, Asif Ekbal, Stefan Dietze, Pavlos Fafalios. "Step-by-Step: A three-stage Pipeline for Document Stance Classification towards Claims". Under review process awaiting acceptance in the proceedings of CIKM 2019.

Arjun Roy, Asif Ekbal, Pushpak Bhattacharyya. "MulCoB-MulFaV: Multimodal Content Based Multilingual Fact Verfication". To be communicated as a scientific journal in ACM-TIST.

REFERENCES

[1] A. Bovet and H. A. Makse, “Influence of fake news in twitter during the 2016 us presidential election,” *Nature communications*, vol. 10, no. 1, p. 7, 2019.

[2] S. Vosoughi, D. Roy, and S. Aral, “The spread of true and false news online,” *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[3] D. Khattar, J. S. Goud, M. Gupta, and V. Varma, “Mvae: Multimodal variational autoencoder for fake news detection,” in *The World Wide Web Conference*, ser. WWW ’19. New York, NY, USA: ACM, 2019, pp. 2915–2921. [Online]. Available: http://doi.acm.org/10.1145/3308558.3313552

[4] M. Fernandez and H. Alani, “Online misinformation: Challenges and future directions,” in *Companion Proceedings of the The Web Conference 2018*, ser. WWW ’18. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2018, pp. 595–602. [Online]. Available: https://doi.org/10.1145/3184558.3188730

[5] T. Chen, L. Wu, X. Li, J. Zhang, H. Yin, and Y. Wang, “Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection,” *CoRR*, vol. abs/1704.05973, 2017.

[6] V. Rubin, N. Conroy, and Y. Chen, “Towards news verification: Deception detection methods for news discourse,” 01 2015.

[7] E. Tacchini, G. Ballarin, M. L. D. Vedova, S. Moret, and L. de Alfaro, “Some like it hoax: Automated fake news detection in social networks,” *CoRR*, vol. abs/1704.07506, 2017.

[8] N. Eshraqi, M. Jalali, and M. H. Moattar, “Spam detection in social networks: A review,” in *2015 International Congress on Technology, Communication and Knowledge (ICTCK)*, Nov 2015, pp. 148–152.

[9] V. Duppada, “"attention" for detecting unreliable news in the information age,” in *AAAI Workshops*, 2018.

[10] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake news detection on social media: A data mining perspective,” *SIGKDD Explor. Newsl.*, vol. 19, no. 1, pp. 22–36, Sep. 2017. [Online]. Available: http://doi.acm.org/10.1145/3137597.3137600

[11] A. Bessi and E. Ferrara, “Social bots distort the 2016 u.s. presidential election online discussion,” *First Monday*, vol. 21, no. 11, 2016. [Online]. Available: https://firstmonday.org/ojs/index.php/fm/article/view/7090

[12] C. Castillo, M. Mendoza, and B. Poblete, “Information credibility on twitter,” in *Proceedings of the 20th International Conference on World Wide Web*, ser. WWW ’11. New York, NY, USA: ACM, 2011, pp. 675–684. [Online]. Available: http://doi.acm.org/10.1145/1963405.1963500

[13] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier, “Tweetcred: Real-time credibility assessment of content on twitter,” in *International Conference on Social Informatics*. Springer, 2014, pp. 228–243.

[14] L. Zhou, D. P. Twitchell, T. Qin, J. K. Burgoon, and J. F. Nunamaker, “An exploratory study into deception detection in text-based computer-mediated communication,” in *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, Jan 2003, pp. 10 pp.–.

[15] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, “Open information extraction from the web,” in *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, ser. IJCAI’07. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007, pp. 2670–2676. [Online]. Available: http://dl.acm.org/citation.cfm?id=1625275.1625705

[16] S. Bajaj, “âĂIJ the pope has a new baby ! âĂİ fake news detection using deep learning,” 2017.

[17] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, “Detecting rumors from microblogs with recurrent neural networks,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial*

Intelligence, ser. IJCAI'16. AAAI Press, 2016, pp. 3818–3824. [Online]. Available: http://dl.acm.org/citation.cfm?id=3061053.3061153

[18] W. Y. Wang, "“liar, liar pants on fire”: A new benchmark dataset for fake news detection," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2017, pp. 422–426. [Online]. Available: http://aclweb.org/anthology/P17-2067

[19] Y. Long, Q. Lu, R. Xiang, M. Li, and C.-R. Huang, "Fake news detection through multi-perspective speaker profiles," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Asian Federation of Natural Language Processing, 2017, pp. 252–256. [Online]. Available: http://aclweb.org/anthology/I17-2043

[20] Y. Kim, "Convolutional neural networks for sentence classification," in *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014, pp. 1746–1751. [Online]. Available: http://aclweb.org/anthology/D14-1181

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[22] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13. USA: Curran Associates Inc., 2013, pp. 3111–3119. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999792.2999959

[23] A. Joshi, P. Bhattacharyya, and M. J. Carman, "Automatic sarcasm detection: A survey," *ACM Comput. Surv.*, vol. 50, no. 5, pp. 73:1–73:22, Sep. 2017. [Online]. Available: http://doi.acm.org/10.1145/3124420

[24] D. M. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild *et al.*, "The science of fake news," *Science*, vol. 359, no. 6380, pp. 1094–1096, 2018.

[25] J. Chang, J. Lefferman, C. Pedersen, and G. Martz, "When fake news stories make real news headlines," *Nightline. ABC News. https://abcnews.go.com/Technology/fake-news-stories-make-real-news-headlines/story?id=43845383*, 2016.

[26] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–36, 2017.

[27] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum, "Credibility assessment of textual claims on the web," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 2173–2178.

[28] Y. Wu, P. K. Agarwal, C. Li, J. Yang, and C. Yu, "Toward computational fact-checking," *Proceedings of the VLDB Endowment*, vol. 7, no. 7, pp. 589–600, 2014.

[29] N. Hassan, C. Li, and M. Tremayne, "Detecting check-worthy factual claims in presidential debates," in *Proceedings of the 24th acm international on conference on information and knowledge management*. ACM, 2015, pp. 1835–1838.

[30] B. Nyhan and J. Reifler, "The effect of fact-checking on elites: A field experiment on us state legislators," *American Journal of Political Science*, vol. 59, no. 3, pp. 628–640, 2015.

[31] D. Pomerleau and D. Rao, "Fake news challenge stage 1 (FNC-I): Stance detection," http://www.fakenewschallenge.org/, 2017.

[32] W. Ferreira and A. Vlachos, "Emergent: a novel data-set for stance classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 1163–1168. [Online]. Available: https://www.aclweb.org/anthology/N16-1138

[33] A. Hanselowski, A. PVS, B. Schiller, F. Caspelherr, D. Chaudhuri, C. M. Meyer, and I. Gurevych, "A retrospective analysis of the fake news challenge stance-detection task," in *Proceedings of the 27th International Conference on Computational Linguistics*. ACL, 2018, pp. 1859–1874. [Online]. Available: http://aclweb.org/anthology/C18-1158

[34] K. Veropoulos, C. Campbell, and N. Cristianini, “Controlling the sensitivity of support vector machines,” in *Proceedings of the International Joint Conference on AI*, 1999, pp. 55–60.

[35] X. Wang, C. Yu, S. Baumgartner, and F. Korn, “Relevant document discovery for fact-checking articles,” in *Proceedings of the 2018 World Wide Web Conference*. Lyon, France: International World Wide Web Conferences Steering Committee, April 2018, pp. 525–533.

[36] S. Bird and E. Loper, “Nltk: the natural language toolkit,” in *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics, 2004, p. 31.

[37] S. Ruder, J. Glover, A. Mehrabani, and P. Ghaffari, “360 stance detection,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Association for Computational Linguistics, 2018, pp. 31–35. [Online]. Available: http://aclweb.org/anthology/N18-5007

[38] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva, “Stance detection with bidirectional conditional encoding,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. ACL, 2016, pp. 876–885. [Online]. Available: http://aclweb.org/anthology/D16-1084

[39] N. Rakholia, “âĂIJ is it true ? âĂİ âĂŞ deep learning for stance detection in news,” 2017.

[40] M. Lai, D. I. Hernández Farías, V. Patti, and P. Rosso, “Friends and enemies of clinton and trump: Using context for detecting stance in political tweets,” in *Advances in Computational Intelligence*. Cham: Springer International Publishing, 2017, pp. 155–168.

[41] Q. Sun, Z. Wang, Q. Zhu, and G. Zhou, “Stance detection with hierarchical attention network,” in *Proceedings of the 27th International Conference on Computational Linguistics*. ACL, 2018, pp. 2399–2409. [Online]. Available: http://aclweb.org/anthology/C18-1203

[42] C. J. Hutto and E. Gilbert, “Vader: A parsimonious rule-based model for sentiment analysis of social media text,” in *ICWSM*, 2014.

[43] C. Conforti, M. T. Pilehvar, and N. Collier, “Towards automatic fake news detection: Cross-level stance detection in news articles,” in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 2018, pp. 40–49.

[44] J. Pennebaker, M. E. Francis, and R. J. Booth, “The development and psychometric properties of liwc2015,” 2015.

[45] S. Baird, D. Sibley, and Y. Pan, “Talos targets disinformation with fake news challenge victory,” https://blog.talosintelligence.com/2017/06/talos-fake-news-challenge.html, 2017.

[46] A. Hanselowski, P. Avinesh, B. Schiller, and F. Caspelherr, “Description of the system developed by team athene in the fnc-1,” Technical report, Tech. Rep., 2017.

[47] B. Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel, “A simple but tough-to-beat baseline for the fake news challenge stance detection task,” *arXiv preprint arXiv:1707.03264*, 2017.

[48] G. Bhatt, A. Sharma, S. Sharma, A. Nagpal, B. Raman, and A. Mittal, “Combining neural, statistical and external features for fake news stance identification,” in *Companion Proceedings of the The Web Conference 2018*, ser. WWW ’18. International World Wide Web Conferences Steering Committee, 2018, pp. 1353–1357. [Online]. Available: https://doi.org/10.1145/3184558.3191577

[49] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[50] Q. Zhang, S. Liang, A. Lipani, Z. Ren, and E. Yilmaz, “From stancesâĂŹ imbalance to their hierarchical representation and detection,” in *Companion Proceedings of the The Web Conference*, 2019.

[51] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, “Semeval-2016 task 6: Detecting stance in tweets,” in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 2016, pp. 31–41.

[52] J. Du, R. Xu, Y. He, and L. Gui, "Stance classification with target-specific neural attention," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 3988–3994. [Online]. Available: https://doi.org/10.24963/ijcai.2017/557

[53] M. A. Walker, P. Anand, R. Abbott, and R. Grant, "Stance classification using dialogic properties of persuasion," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ser. NAACL HLT '12. ACL, 2012, pp. 592–596. [Online]. Available: http://dl.acm.org/citation.cfm?id=2382029.2382124

[54] D. Sridhar, J. Foulds, B. Huang, L. Getoor, and M. Walker, "Joint models of disagreement and stance in online debate," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. ACL, 2015, pp. 116–125. [Online]. Available: http://aclweb.org/anthology/P15-1012

[55] R. Bar-Haim, I. Bhattacharya, F. Dinuzzo, A. Saha, and N. Slonim, "Stance classification of context-dependent claims," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017, pp. 251–261.

[56] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

[57] K. S. Hasan and V. Ng, "Stance classification of ideological debates: Data, models, features, and constraints," in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013, pp. 1348–1356.

[58] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum, "Where the truth lies: Explaining the credibility of emerging claims on the web and social media," in *Proceedings of the 26th International Conference on World Wide Web Companion*, ser. WWW '17 Companion. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2017, pp. 1003–1012. [Online]. Available: https://doi.org/10.1145/3041021.3055133

[59] J. Pan, S. Pavlova, C. Li, N. Li, Y. Li, and J. Liu, “Content based fake news detection using knowledge graphs,” in *The Semantic Web âĂŞ ISWC 2018 - 17th International Semantic Web Conference, 2018, Proceedings*, ser. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), D. VrandeÄ■iÄĞ, Ed. Germany: Springer Verlag, 12 2018, pp. 669–683.

[60] M. T. Maybury, *Intelligent multimedia information retrieval.* Aaai Press, 1997.

[61] J. Garten, K. Sagae, V. Ustun, and M. Dehghani, “Combining distributed vector representations for words,” in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 2015, pp. 95–101.

[62] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[63] C. Szegedy, S. Ioffe, and V. Vanhoucke, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *AAAI*, 2016.

[64] C. Boididou, S. Papadopoulos, M. Zampoglou, L. Apostolidis, O. Papadopoulou, and Y. Kompatsiaris, “Detection and visualization of misleading content on twitter,” *International Journal of Multimedia Information Retrieval*, vol. 7, no. 1, pp. 71–86, 2018.

[65] C. Boididou, K. Andreadou, S. Papadopoulos, D.-T. Dang-Nguyen, G. Boato, M. Riegler, Y. Kompatsiaris *et al.*, “Verifying multimedia use at mediaeval 2015.” in *MediaEval*, 2015.

[66] C. Boididou, S. E. Middleton, S. Papadopoulos, D. Nguyen, D. Tien, M. Riegler, G. Boato, A. Petlund, and Y. Kompatsiaris, “The vmu participation@ verifying multimedia use 2016,” 2016.

[67] J. Cao, Z. Jin, Y. Zhang, and Y. Zhang, “Mcg-ict at mediaeval 2016 verifying tweets from both text and visual content.” in *MediaEval*, 2016.

[68] C. Boididou, S. Papadopoulos, D.-T. Dang-Nguyen, G. Boato, and Y. Kompatsiaris, "The certh-unitn participation@ verifying multimedia use 2015." in *MediaEval*, 2015.

[69] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, and J. Gao, "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, 2018, pp. 849–857.

[70] N. Ruchansky, S. Seo, and Y. Liu, "Csi: A hybrid deep model for fake news detection," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM '17. New York, NY, USA: ACM, 2017, pp. 797–806. [Online]. Available: http://doi.acm.org/10.1145/3132847.3132877

[71] A. Roy, K. Basak, A. Ekbal, and P. Bhattacharyya, "A deep ensemble framework for fake news detection and classification," *arXiv preprint arXiv:1811.04670*, 2018.

[72] S. Singhania, N. Fernandez, and S. Rao, "3han: A deep neural network for fake news detection," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 572–581.

[73] C. Boididou, S. E. Middleton, Z. Jin, S. Papadopoulos, D.-T. Dang-Nguyen, G. Boato, and Y. Kompatsiaris, "Verifying information with multimedia content on twitter," *Multimedia Tools and Applications*, vol. 77, no. 12, pp. 15 545–15 571, 2018.

[74] K. Popat, S. Mukherjee, A. Yates, and G. Weikum, "Declare: Debunking fake news and false claims using evidence-aware deep learning," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 22–32.