Towards detection and explanation of factual inconsistencies

Thesis submitted in partial fulfillment
of the requirements for the degree of

*Master of Science in **Computational Linguistics** by Research*

by

Tathagata Raha
2018114017
tathagata.raha@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
May 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Towards detection and explanation of factual inconsistencies**" by **Tathagata Raha**, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Prof. Vasudeva Varma

To *everyone who were beside me over the last 5 years.*

Acknowledgments

As I reflect on my thesis journey, I am filled with gratitude for the many individuals who have played a crucial role in helping me to achieve this milestone. This thesis, just like everything else that I have done at IIIT has been a collaborative effort with various individuals helping me in their own ways and even a thousand pages would not be sufficient to express my gratitude to all of you.

Firstly, I am deeply indebted to my advisor, Prof. Vasudeva Varma, for giving me the opportunity to be a part of his group and introducing me to the fields of Information Retrieval and Natural Language Processing. I am grateful for his invaluable guidance, encouragement and feedback. He has been a great mentor and a source of inspiration for me. I am very fortunate to have worked with Prof. Manish Gupta, whose valuable insights and constructive feedback have been instrumental in shaping my research journey. I have learnt a lot during my time collaborating with him, whether it be learning how the latest models work, the best way to document research and track progress, or the art of writing papers.

The thesis wouldn't be possible without the support of my lab mates(or my roommates because I would spend more time in my lab than in my hostel!). Without co-authors and co-researchers like Sayar, Sravani, Vijaysaradhi and Harshit this thesis would have been a lot shorter! I also cannot thank enough my seniors and friends at IREL - Sagar, Pawan, Nirmal, Anubhav, Shivprasad, Dhaval, Ankita, Shivansh, who have not only helped in my research but also were there in every ups and down. WIthout them, both my research life and personal life would've been quite boring.

Beyond the lab, I was blessed to come across a bunch of wonderful people who've always been there behind my back, providing all the moral support needed, and have left me with beautiful memories to look at. Thank you Bharathi, Mukund, Rishav, Aditya, Ishan, Monil, Shivaan and Gunjan for showing me what "a family away from a family" really is like. Thank you for bearing with me all this while, and getting a better person out of me with your constant love, care, advises and scoldings. Also, thanks to other amazing people I spent some great moments with on campus - Dipanwita, Prerna, Astitva, Arathy, Abhijit, Nomaan, Mohee, Sridhar, Sumanth.

A big shout-out to Taylor Swift, Ed Sheeran, One Republic, Avicii, Zedd, Halsey, Anne-Marie, Ritviz, Krsna, Seedhe Maut, Tanmay Bhat, Anubhav Singh Bassi, Zakir Khan, Drew Binsky, Mr. Beast, CarryMinati, David Crane, Greg Daniels, Vince Gilligan, Activision and EA Sports for not only being my major sources of entertainment but also inspiring me along the way.

Last but not the least, I would like to thank my father, mother, and brother for their irreplaceable guidance and constant faith in me. I would not be where I am today, having crumbled away, if it weren't for the adamantine pillars of their support, shaping me to become the person I am today.

Thank you all for being there for me and for making this journey one that I will never forget. I could not have done it without all of you, and for that, I am truly grateful.

Abstract

Factual inconsistencies in text, which include a range of errors from minor inaccuracies to substantial distortions, present a significant challenge in the realm of information dissemination. These inconsistencies, whether unintentional or the result of deliberate misinformation, can lead to a skewed understanding and flawed decision-making. In the context of the vast and complex landscape of digital data, the limitation of traditional verification methods becomes evident, highlighting the need for more advanced solutions. In this thesis, we present and explore three critical problems in this domain, each addressing a unique aspect of content credibility and factual consistency.

The first problem we tackled in this thesis is the detection of hostility in online content, specifically focusing on Hindi tweets. Our approach categorizes these tweets into distinct hostile classes: hateful, offensive, defamatory, or fake. We employed pretrained Transformer-based models, particularly IndicBERT, which is adept at processing Hindi text due to its training on a vast corpus of Indian languages. The architecture of our model effectively utilizes information from emojis and hashtags, in addition to the natural language text. A significant enhancement in performance was achieved through Task Adaptive Pretraining (TAPT), leading to increases of 1.35% and 1.40% in binary hostility detection, and improvements of 4.06% and 1.05% in macro and weighted F1 metrics, respectively, for fine-grained classifications. Notably, our system, under the team name 'iREL IIIT', achieved first place in the 'Hostile Post Detection in Hindi' shared task at the CONSTRAINT-2021 workshop.

The second problem addressed in this thesis delves into the realm of automated fact extraction and verification, a pressing challenge in the digital landscape rife with misinformation. Central to our approach is the innovative Fact Extraction and Verification (FEVER) project, which assesses the veracity of claims against a comprehensive body of evidence from Wikipedia. Our contributions include the development of a specialized retrieval model tailored for the FEVER dataset, a strategic approach to sentence selection for optimal evidence gathering, and an exploration of advanced natural language inference (NLI) models, particularly state-of-the-art transformer models. By integrating these components, we not only refine the process of recognizing textual entailment but also significantly enhance the accuracy and efficiency of automated fact-checking.

In tackling the third problem of this thesis, we address the critical issue of detecting and explaining factual inconsistencies in text, a significant challenge in the era of advanced Transformer-based natural language generation models. These models, while adept in tasks like summarization and translation, often struggle with producing hallucinatory and inconsistent content. Our approach introduces the novel Factual

Inconsistency Classification with Explanations (FICLE) method. This technique involves a detailed analysis of sentence pairs to identify inconsistency types and provide comprehensive explanations, including inconsistent fact triples, context spans, and entity types.

Central to our approach is the creation of the FICLE dataset, extensively annotated to cover a range of inconsistency types and their explanations. Utilizing this dataset, we developed a pipeline comprising four neural models, each designed to focus on specific facets of inconsistency detection and explanation. These models utilize various Transformer-based NLU and NLG architectures, with DeBERTa showing notable effectiveness across most sub-tasks. Our results underscore the effectiveness of this approach, demonstrating high performance in inconsistency type classification and entity-type prediction, with weighted F1 scores of around 87% and 86%, respectively, for these tasks. The detection of context spans, while more challenging, achieved an Intersection over Union (IoU) of approximately 65%, indicating the nuanced complexity of this aspect. The contributions of this research are multi-fold: proposing a new problem in factual inconsistency detection, creating a novel dataset, and establishing a baseline pipeline with high performance in inconsistency type classification and entity-type prediction.

Contents

Chapter Page

1 Introduction . . . 1
1.1 Motivation . . . 1
1.2 Hostility Detection . . . 2
1.3 Fact Extraction and Verification (FEVER) . . . 2
1.4 Factual Inconsistency Classification With Explanations (FICLE) . . . 3
1.5 Thesis Contributions . . . 3
1.6 Thesis Outline . . . 4

2 Related Work . . . 6
2.1 Introduction . . . 6
2.2 Natural Language Inference . . . 7
2.2.1 Definition and Scope . . . 7
2.2.2 Key Approaches and Models . . . 7
2.2.3 Datasets and Benchmarks . . . 8
2.3 Fact-Checking Systems . . . 8
2.3.1 Overview of Fact Checking Systems . . . 8
2.3.2 Automated Fact-Checking Techniques . . . 9
2.3.3 Representative Systems and Tools . . . 10
2.4 Explainable NLP . . . 11
2.4.1 Approaches to Explainable NLP . . . 11
2.4.2 Evaluation Metrics for Explainability . . . 12
2.5 Fake News Detection . . . 12
2.5.1 Detection Techniques . . . 12
2.5.2 Datasets and Evaluation . . . 13
2.5.3 Integration with Fact Checking . . . 14

3 Task Adaptive Pretraining of Transformers for Hostility Detection . . . 15
3.1 Introduction . . . 15
3.2 Dataset . . . 16
3.3 Approach . . . 17
3.3.1 Preprocessing and Feature Extraction . . . 17
3.3.2 Architecture . . . 17
3.3.3 Task Adaptive Pretraining . . . 18
3.4 Results . . . 20
3.5 Experimental Details . . . 20

3.6 Conclusion . . . 20

4 Fact Extraction and Verification . . . 22
4.1 Introduction . . . 22
4.2 Related Work . . . 23
4.3 Task Overview . . . 25
4.3.1 Dataset . . . 25
4.3.2 Metrics for Evaluation . . . 26
4.4 Preprocessing the FEVER Dataset . . . 27
4.5 Approach . . . 27
4.5.1 System Overview . . . 27
4.5.2 Retrieval Model . . . 28
4.5.2.1 Preprocessing Wikipedia Dump . . . 28
4.5.2.2 Extracting Proper Nouns . . . 29
4.5.2.3 Article Retrieval . . . 29
4.5.2.4 Evaluation and Error Analysis . . . 30
4.5.3 Sentence Selection . . . 30
4.5.4 Recognizing Textual Entailment . . . 31
4.5.4.1 Baseline Models . . . 31
4.5.4.2 Vanilla Transformer Models . . . 33
4.5.4.3 InferTransformer Models . . . 34
4.5.5 Other Experimental Details . . . 36
4.6 Results and Analysis . . . 37
4.7 Summary and Future Work . . . 38

5 Factual Inconsistencies: Dataset and our Approach . . . 41
5.1 Introduction . . . 41
5.2 Related Work . . . 43
5.3 FICLE Dataset . . . 44
5.3.1 Data Curation . . . 44
5.3.2 Inconsistency Type Classification . . . 45
5.3.3 Annotation Details . . . 47
5.3.3.1 Syntactic Oriented Annotations . . . 47
5.3.3.2 Semantic Oriented Annotations . . . 49
5.3.4 FICLE Dataset Statistics . . . 50
5.3.5 Quality Checks . . . 50
5.3.6 Dataset Fields . . . 51
5.4 Our Approach . . . 52
5.4.1 Model Architecture . . . 52
5.4.2 Predict Inconsistent Spans . . . 54
5.4.3 Predict Inconsistency Type and Claim Component . . . 54
5.4.4 Predict Inconsistent Entity Types . . . 54
5.5 Experiments and Results . . . 55
5.5.1 Source, Relation, Target and Inconsistent Context Span Prediction . . . 56
5.5.2 Inconsistency Type and Inconsistent Claim Component Prediction . . . 57
5.5.3 Inconsistent Entity Type Prediction . . . 58

5.5.4 Qualitative Analysis 58
5.5.5 Experimental Settings 59
5.6 Conclusion and Future Work 60

6 Conclusions and Future Work 61

Bibliography 65

List of Figures

Figure		Page
3.1	Model Architecture	18
4.1	Schematic diagram of the baseline neural network showing the feature vector	32
4.2	Schematic diagram of the NN_SUM Embedding neural network	33
4.3	Sentence pair classification with BERT.	34
4.4	InferSent training scheme for NLI	35
4.5	Loss curve of the model NN_TFIDF trained with 5000 features for vectorizer	38
4.6	Loss curve of the model NN_SUM_EMB trained with GloVe840B300d pretrained embeddings	38
5.1	Factual Inconsistency Classification with Explanation (FICLE) Example: Inputs are claim and context. Outputs include inconsistency type and explanation (inconsistent claim fact triple, inconsistent context span, inconsistent claim component, coarse and fine-grained inconsistent entity types).	42
5.2	Distribution of coarse inconsistent entity types in FICLE.	49
5.3	FICLE: System Architecture	53

List of Tables

Table		Page
3.1	Distribution of Supervised labels in Training set	16
3.2	Distribution of labels in the Test set	17
3.3	Results on the Validation split for every category (% Weighted F1 Scores)	19
3.4	Results on the Validation split for every category (% Macro F1 Scores)	19
3.5	Shared Task Results: Top 3 teams on public leaderboard (% F1 Scores)	19
4.1	Training examples from the FEVER Dataset Showcasing Sentence Claims and Corresponding Evidence	26
4.2	Data distribution across different splits.	27
4.3	Retrieval results	30
4.4	Results of all the models on the validation set. Best performance values are shown in bold	36
4.5	Samples where the best performing model made errors in prediction	40
5.1	Comparison of FICLE with other datasets. #Samples indicates number of contradictory/inconsistent samples (and not the size of full dataset).	43
5.2	Inconsistent Claim Fact Triple, Context Span and Claim Component examples for the context sentence "Prime Minister Narendra Modi enthusiastically hoisted the Indian flag." Subject, relation and target in the claim are shown in bold, italics and underline respectively.	45
5.3	Inconsistency Type and Coarse/Fine-grained Inconsistent Entity Type examples. Inconsistent spans are marked in bold in both claim as well as context.	46
5.4	Minimum, average, and maximum size (words) of various fields averaged across samples in FICLE dataset.	49
5.5	Source, Relation and Target Prediction from Claim Sentence	56
5.6	Inconsistent Context Span Prediction	56
5.7	Joint Prediction of Source, Relation and Target Prediction from Claim Sentence and Inconsistent Context Span using Multi-Task Setting	57
5.8	Inconsistency Type Prediction	57
5.9	Inconsistent Claim Component Prediction (6-class classification)	57
5.10	Coarse Inconsistent Entity Type Prediction. Note that embedding based methods don't work with NLG models.	58
5.11	Accuracy/Weighted F1 for Fine-grained Inconsistent Entity Type Prediction. Note that embedding based methods do not work with NLG models.	59
5.12	Confusion matrix for inconsistency type prediction. We observe a high correlation between actual and predicted values, indicating our model is effective.	59

Chapter 1

Introduction

1.1 Motivation

The advancement of automated data analysis from manual fact-checking to complex algorithmic interpretations signifies a critical response to the escalating volume and intricacy of data. This evolution, essential in the digital age, confronts unique challenges, particularly in detecting and explaining factual inconsistencies. As Reimers and Gurevych[79] note, the challenge extends beyond mere identification. It requires understanding the context and reasons behind these inconsistencies. This task is complicated by the diversity in data formats and the nuances of human language.

The imperative for factual accuracy and the elucidation of inconsistencies is evident across various domains. In journalism, the credibility of news reporting hinges not just on identifying inaccuracies but also on understanding and communicating the reasons behind them[33]. In academic research, the validity of findings relies on accurately interpreting data and explaining discrepancies, if any[14]. The business and finance sectors depend on reliable data for decision-making, where tools that help them in understanding the root causes of inconsistencies are as crucial as detecting them[70]. Similarly, in healthcare, patient care and medical research necessitate not only accurate data but also a clear understanding of any anomalies[92].

The impact of unexplained factual inconsistencies is profound, affecting individual decision-making and public trust. In the era of rapid information dissemination, particularly through social media, the ability to not only identify but also explain misinformation becomes increasingly vital[50]. While advancements in technologies like machine learning have enhanced our capacity for detecting inconsistencies, the challenge remains in providing clear, understandable explanations for them[25]. These open questions in current methodologies is where our thesis attempts to address. By developing methods that not only detect inconsistencies but also elucidate their origins and nature, my work aims to contribute to a more transparent and reliable information landscape.

Addressing these challenges is not solely a technological endeavor; it encompasses ethical considerations, particularly in ensuring that automated processes are unbiased and transparent[65]. The aim is to enhance not just the technological capability but also the ethical soundness of automated systems.

As we progress into an increasingly data-reliant society, the importance of developing sophisticated, accountable methods for both detecting and explaining factual inconsistencies becomes paramount, a challenge that our research is eager to address.

1.2 Hostility Detection

This module is dedicated to hostility detection (identifying hostile content within digital communication like social media) where the focus is on identifying hostile content within Hindi Tweets, a crucial aspect of maintaining the sanctity and user well-being on the World Wide Web. With the surge in internet usage and the proliferation of social media content, a significant portion of online material has become tainted with hostility, including hate speech, offensive language, defamation, and misinformation. This chapter addresses the urgent need to pinpoint such harmful content, aligning with the broader goal of the thesis to develop automated methods for detecting and explaining inconsistencies and harmful elements in digital communication. The chapter explores the use of advanced natural language processing techniques, particularly the adaptation of pretrained Transformer-based models like IndicBERT, tailored for Indian language texts. This innovative approach involves enhancing these models with Task Adaptive Pretraining (TAPT) and integrating additional linguistic features such as emojis and hashtags, following the architecture proposed by Ghosh Roy et al[29].

1.3 Fact Extraction and Verification (FEVER)

In the face of the digital era's challenge of misinformation, this chapter delves into the Fact Extraction and Verification (FEVER) project. This initiative, pivotal in the fight against the spread of "Fake News", leverages advancements in natural language processing to authenticate textual claims against a vast database of factual content from Wikipedia. Unlike traditional Natural Language Inference (NLI) tasks, FEVER requires extracting and verifying evidence from an extensive corpus, thus addressing the urgent need for automated fact-checking in today's information-rich world.

This module discusses the complexities involved in the FEVER project, particularly highlighting its unique dataset comprised of 185,445 claims. These claims, each meticulously verified against Wikipedia's introductory sections, demonstrate the intricacies of automated fact verification. The chapter also explores the challenges faced in this endeavor, such as the trade-off between the speed of annotation and the thoroughness required for evidence gathering. This examination reveals critical insights into the limitations and potential of current automated fact-checking methodologies.

Aligning with the thesis's focus on automated methods for detecting and explaining factual inconsistencies, the chapter makes significant contributions to the field. These include the development of a retrieval model tailored for the FEVER dataset, strategies for efficient sentence selection from retrieved articles, and advanced approaches in Recognizing Textual Entailment (RTE) and NLI using transformer

models. These innovations not only enhance the capabilities of automated fact verification systems but also significantly contribute to the broader context of misinformation management in the digital age.

1.4 Factual Inconsistency Classification With Explanations (FICLE)

In the pivotal module of this thesis, the focus is directed towards the novel approach of factual inconsistency classification with explanations (FICLE). This chapter confronts a significant challenge in Transformer-based natural language generation models: their propensity for producing hallucinatory and inconsistent text. Factual inconsistencies undermine the reliability and credibility of the generated content, raising concerns of misinformation. Thus, detecting and explaining these inconsistencies is crucial. The approach introduced in this chapter advances beyond existing fake content detection studies, which are limited by the state of knowledge bases. The FICLE task involves identifying inconsistencies in sentence pairs (claim and context) and comprehensively explaining them, including the identification of inconsistency types. This approach marks a significant shift from previous studies that either focused on basic contradiction detection or offered limited explanations, thereby enhancing the depth and precision of inconsistency analysis in natural language processing.

Methodologically, the chapter builds upon a specifically curated dataset, annotated for various types of factual inconsistencies and detailed explanations. Originating from the FEVER dataset, it is annotated with inconsistency types and a detailed classification of inconsistent entities, both coarse and fine-grained. This dataset forms the foundation for training a sequence of neural models, each tailored to address specific elements of the FICLE task. The chapter details the architecture of these models, comprising separate components for predicting inconsistent fact triples, inconsistency types, and inconsistent entity types from sentence pairs. This layered approach reflects a nuanced comprehension of linguistic complexity and the intricacies involved in detecting and explaining factual inconsistencies.

Outcome-wise, the chapter critically evaluates various standard Transformer-based models in natural language understanding and generation for their efficacy in the FICLE task. Models like BERT, RoBERTa, DeBERTa, T5, and BART are examined, with DeBERTa showing notable effectiveness for most sub-tasks. The results highlight the particular challenge of accurately detecting contextual inconsistencies, signaling a key area for future research. The chapter's contributions are pivotal, addressing a notable gap in natural language processing and aligning seamlessly with the thesis's broader theme: developing sophisticated, automated methods for detecting and explaining factual inconsistencies, essential for ensuring information reliability in the digital era.

1.5 Thesis Contributions

This section summarizes the key contributions of our thesis:

- **Hostility Detection in Hindi Tweets:** The thesis pioneers the use of pretrained Transformer-based models, like IndicBERT, adapted for Indian language texts, specifically for hostility detection in Hindi tweets. This adaptation involves Task Adaptive Pretraining (TAPT) and the integration of unique linguistic features such as emojis and hashtags.

- **Innovative Approaches in Automated Fact-Checking:** The thesis contributes to the FEVER project by developing a specialized retrieval model for the FEVER dataset. This includes strategies for efficient sentence selection and advanced approaches in Recognizing Textual Entailment (RTE) and Natural Language Inference (NLI) using transformer models.

- **FICLE Dataset:** A pivotal contribution of this thesis is the creation of the FICLE (Factual Inconsistency Classification with Explanations) dataset. This unique dataset, originating from the FEVER dataset, is meticulously annotated for diverse types of factual inconsistencies and detailed explanations, making it a valuable resource for detecting and explaining factual inconsistencies.

- **Models for FICLE Task:** Another significant contribution of the thesis lies in the development of neural network models for the FICLE task. These models are innovatively designed to address the multifaceted challenges of factual inconsistency detection and explanation. They include sophisticated architectures for predicting inconsistent fact triples, identifying inconsistency types, and classifying inconsistent entities from sentence pairs. The thesis also conducts a thorough evaluation of various Transformer-based models like BERT, RoBERTa, DeBERTa, T5, and BART, with DeBERTa demonstrating notable effectiveness.

1.6 Thesis Outline

This thesis presents a comprehensive study in the realm of textual analysis, specifically focusing on detecting and explaining factual inconsistencies in text. Through a series of methodical and innovative chapters, this work not only addresses a crucial problem in natural language processing but also makes significant contributions to the field. Below is a summary of the key contributions of this thesis:

- The first chapter sets the stage for our research by highlighting the motivation behind solving the complex problem of detecting and explaining factual inconsistencies in text. It provides a concise summary of our contributions and outlines the structure of the thesis.

- In the second chapter, we delve into a comprehensive review of existing literature pertinent to our research. This chapter helps contextualize our work within the broader field and identifies gaps that our thesis aims to fill.

- The third chapter addresses the challenge of detecting hostility in Hindi tweets. It demonstrates the efficacy of task-adaptive pretraining on IndicBERT, a specialized transformer model for Indian languages, and showcases its superior performance in this context.

- Our fourth chapter introduces a novel pipeline for the Fact Extraction and VERification (FEVER) task. This involves innovative methods for efficient sentence selection from retrieved articles and advanced techniques in Recognizing Textual Entailment (RTE) and Natural Language Inference (NLI) using transformer models.
- The fifth chapter, the pivotal part of this thesis, presents a groundbreaking approach to detecting and explaining factual inconsistencies. It includes the introduction of FICLE, a novel dataset comprising 8055 samples annotated with inconsistency types and various forms of explanations, and details the development of specialized models and pipelines for the FICLE task.
- We conclude our thesis in the sixth chapter by summarizing our diverse contributions. This chapter reflects on the research conducted and suggests promising areas for future investigation and improvement in the field of natural language processing and beyond.

Each chapter of this thesis represents a step forward in understanding and solving the challenges associated with textual inconsistencies, with a special focus on leveraging advanced computational models and techniques. The collective insights and methodologies presented herein not only contribute significantly to the academic community but also hold practical implications for real-world applications in natural language processing and beyond.

Chapter 2

Related Work

2.1 Introduction

The proliferation of digital information has made distinguishing between accurate and inaccurate content a paramount concern. The first chapter laid the foundation by introducing the concept of factual inconsistencies. Here, we shift our focus to survey the landscape of research and development efforts that are at the forefront of addressing this issue. This chapter navigates through the various domains that intersect with the detection and classification of factual inaccuracies, illuminating the progress and the tools that have become instrumental in this field.

The rapid development of Natural Language Processing (NLP) has led to the emergence of diverse approaches toward understanding and interpreting human language, pivotal in identifying factual inconsistencies. These approaches range from Natural Language Inference (NLI) techniques, which discern the relationships between segments of text, to comprehensive fact-checking systems that cross-reference claims with established facts. Moreover, as AI-driven models gain complexity, the call for transparent and explainable methods has given rise to an entire sub-domain of Explainable NLP, underscoring the significance of not only detecting inaccuracies but also understanding the rationale behind these determinations.

Additionally, in a socio-political climate rife with 'fake news,' the ability to automatically identify and flag false information has become an indispensable tool in safeguarding public discourse. The detection of fake news involves an interplay of various NLP techniques, often complementing the broader task of fact-checking.

In this chapter, we examine the key areas, methods, and how well they address the challenges of factual inconsistencies. The goal is to provide an overview of the current state of research and its contributions to maintaining information integrity.

2.2 Natural Language Inference

2.2.1 Definition and Scope

Natural Language Inference (NLI), also known as Recognizing Textual Entailment (RTE), involves determining the relationship between a pair of text elements, typically a premise and a hypothesis. The core objective of NLI is to deduce whether the hypothesis can logically be inferred from the premise, which is crucial in the context of factual inconsistencies. This deduction process is fundamental for a variety of applications including information extraction, text summarization, question answering, and especially in verifying factual accuracy in texts [15][22].

NLI plays a pivotal role in identifying factual inconsistencies as it allows models to evaluate and contrast statements to understand if they support, contradict, or are unrelated to each other. This becomes particularly significant in the era of information where distinguishing between factual and inaccurate information is essential for maintaining the integrity of data-driven systems and decisions [91].

The NLI tasks can generally be categorized into a three-way classification system:

1. **Entailment:** The hypothesis is definitely true given the premise.

2. **Contradiction:** The hypothesis is definitely false given the premise.

3. **Neutral:** The hypothesis may be either true or false given the premise; the premise does not provide enough information to make a definitive decision.

This tripartite classification forms the basis of most NLI systems, enabling them to address the complexities and nuances in human language, which is often ambiguous and context-dependent [30].

2.2.2 Key Approaches and Models

Before the advent of deep learning, NLI was approached with traditional machine learning methods that included feature engineering with lexical, syntactic, and semantic features of the text. Algorithms such as Support Vector Machines (SVMs) and Decision Trees were employed using these handcrafted features to model the relationship between text pairs [57].

With the rise of deep learning, there has been a significant shift in the approaches to NLI. Notably, models based on the transformer architecture such as BERT (Bidirectional Encoder Representations from Transformers) and RoBERTa (A Robustly Optimized BERT Pretraining Approach) have achieved state-of-the-art performance on NLI tasks [24][52]. These models leverage a large amount of data and compute power to learn contextual representations of text, significantly reducing the need for feature engineering and allowing for a more nuanced understanding of language.

2.2.3 Datasets and Benchmarks

To train and evaluate NLI models, several datasets have been developed. Some of the most widely used are:

- **The Stanford Natural Language Inference (SNLI) Corpus:** A collection of 570,000 human-annotated sentence pairs, where the relationships are labeled as entailment, contradiction, or neutral [15].
- **The Multi-Genre Natural Language Inference (MultiNLI) Corpus**: An expansion of SNLI, MultiNLI includes 433,000 sentence pairs with the same three-way classification, but drawn from a variety of genres of spoken and written text [99].
- **Cross-lingual NLI (XNLI)**: This dataset extends the MultiNLI dataset to 15 languages, including low-resource languages. It consists of 7,500 human-annotated examples for each language, which were translated by professional translators from a subset of the English MultiNLI dataset. The XNLI dataset was developed to benchmark the performance of models on cross-lingual sentence understanding tasks [20].
- **SciTail**: SciTail is an entailment dataset created from multiple-choice science exams and web sentences, where the task is to determine whether a given hypothesis is entailed by a given premise. It consists of 27000 examples annotated by 5 annotators where the final entailment label is decided by majority voting. The dataset focuses on a scientific domain, requiring a certain degree of domain-specific knowledge to perform well [45].

Benchmarking performances on these datasets are critical for understanding the progress and current state of NLI systems. However, there are limitations to these benchmarks, including dataset biases and the models' failures to generalize beyond the type of data they were trained on. Furthermore, despite high performance on benchmarks, models often struggle with examples that require world knowledge or common sense reasoning [62].

2.3 Fact-Checking Systems

2.3.1 Overview of Fact Checking Systems

Fact-checking is an essential practice within the ecosystem of information dissemination, aimed at validating the veracity of statements, claims, and assertions made in various media. It has become especially vital in the age of digital media, where the rapid spread of information—and misinformation—can have immediate and profound impacts on public opinion, political processes, and societal trust.

The importance of fact-checking was underscored in a report by the American Press Institute (API), which emphasized its role in enhancing the credibility of the news ecosystem and fostering a more

informed society. The API suggests that accurate information is a cornerstone of democracy, enabling citizens to make choices based on evidence rather than conjecture or deception [4].

Fact-checking has been a manual process, performed by journalists and fact-checkers who delve into primary sources, consult experts, and examine data to assess the accuracy of claims. This traditional approach is thorough but time-consuming and inherently limited by human resources, making it challenging to address the vast quantity of claims made daily across numerous platforms.

The advent of automated fact-checking represents an evolution of the field, harnessing the power of algorithms, artificial intelligence, and computational tools to perform fact-checking tasks with greater speed and at a larger scale. Automated fact-checking systems can quickly sift through massive amounts of data, identify claims in text, cross-reference these against reputable databases and fact-checking archives, and often determine the likelihood of their accuracy. However, these systems also face challenges, including understanding context, nuance, and the reliability of sources. They are not a replacement for human judgment but rather a supplement to enhance the capacity of human fact-checkers.

Researchers from Duke University's Reporters' Lab highlight the integration of automation in fact-checking processes as an innovative development that can assist journalists in real-time, which is especially beneficial during events such as political debates and elections [56].

2.3.2 Automated Fact-Checking Techniques

The landscape of automated fact-checking is complex and rapidly evolving, employing a range of computational techniques to handle the growing volume of information that needs verification. The primary methodology involves a pipeline approach that mirrors the systematic process of human fact-checking but at a significantly larger scale and speed.

The automated fact-checking pipeline typically consists of the following stages:

- **Claim Detection:** The first step in the automated fact-checking pipeline is the identification of statements that are factual and, therefore, verifiable—known as check-worthy claims. This is achieved through natural language processing (NLP), which involves parsing the text, identifying entities, and recognizing statements that make a factual assertion. Machine learning models are trained on labeled datasets to distinguish between factual claims, opinions, and non-factual assertions [37].

- **Evidence Retrieval**: Once a potentially checkable claim is detected, the system proceeds to gather evidence. This involves querying large-scale databases, trusted news repositories, and authoritative sources. Information retrieval algorithms, enhanced by the latest advancements in semantic search and document retrieval, scan through terabytes of structured and unstructured data to find relevant evidence that either supports or contradicts the claim [11].

- **Evidence Verification**: With relevant evidence at hand, the system must now assess the claim's validity. This involves complex algorithms that can understand nuances in the data, compare the

claim to the evidence, and come to a conclusion about its accuracy. Techniques such as stance detection (which determines if the evidence supports or refutes the claim) and check-worthiness analysis are employed. Verification can also involve cross-referencing claims with known facts stored in knowledge bases or inferred through logical reasoning [5].

- **Explanation Generation**: The last step involves generating an explanation for the verdict reached on the claim's veracity. This is crucial for the credibility and accountability of the fact-checking system. Explanations may include the presentation of evidence that led to the verification decision or a summary of the reasoning process the system followed. Recent models in NLP strive to create more interpretable AI, offering transparent insights into their decision-making process [66].

2.3.3 Representative Systems and Tools

Within the domain of fact-checking, several organizations have established themselves as authoritative sources by consistently providing high-quality, reliable fact-checking services. Below is an in-depth look at three such systems:

Full Fact is a pioneering fact-checking charity based in the United Kingdom that stands out for its comprehensive approach to verification. It analyzes the accuracy of claims made by politicians, public institutions, and journalists, and has developed its own automated fact-checking tools, including live fact-checking services and AI tools that monitor broadcasts and online content for checkable claims. Full Fact collaborates with international partners to develop scalable automated fact-checking technologies and works to improve the standards of fact-checking worldwide. Their work underlines the importance of impartiality and methodology in the practice of fact-checking [3].

Snopes, one of the internet's oldest and most influential fact-checking websites, has been debunking myths, rumors, and misinformation since 1994. Snopes is known for its in-depth research into urban legends, viral phenomena, and online hoaxes. The site's approach relies heavily on investigative research and the accumulation of a comprehensive database of fact-checks that are frequently referenced by other media outlets and researchers. Snopes' work highlights the intersection of fact-checking with critical thinking and digital literacy, making it an invaluable resource in the fight against misinformation [1].

FactCheck.org is operated by The Annenberg Public Policy Center of the University of Pennsylvania and is dedicated to monitoring the factual accuracy of U.S. political discourse. FactCheck.org has distinguished itself by providing detailed analyses of political statements, advertising, and debates, emphasizing nonpartisanship and transparency in its fact-checking efforts. The organization's work often includes contextual information that sheds light on how political rhetoric can be used to distort truth, showcasing the need for careful and comprehensive evaluation of claims within political communication [2].

2.4 Explainable NLP

Explainability in Natural Language Processing (NLP) refers to the ability to provide understandable reasons for the decisions made by NLP models. The significance of explainability can be subdivided into two key areas: the need for transparency and its relevance to addressing factual inconsistencies.

In the context of factual inconsistencies, explainability is vital for understanding why a model has arrived at a particular decision, especially when processing information that may contain conflicting facts or misinformation [102]. It enables developers and end-users to identify whether a model is relying on unsound logic or biased data, which is crucial in applications such as news verification, medical diagnosis, and legal decision-making where the factual accuracy is paramount [32].

2.4.1 Approaches to Explainable NLP

A suite of techniques has been developed to enhance the explainability of NLP models. These methodologies provide insights into the internal workings of complex models and shed light on the basis for their outputs.

SHapley Additive exPlanations (SHAP): SHAP is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions. SHAP values offer a cohesive approach to explanation by representing a feature's contribution to a difference from the average prediction [55]. In NLP, SHAP can be applied to quantify the impact of each token or word embedding on the model's prediction, allowing for deeper insight into the model's rationale.

Attention Mechanisms: Originally designed to improve the performance of neural networks, attention mechanisms can also provide a form of explanation by highlighting the parts of the input data (such as words in a sentence) that the model pays more attention to when making a decision [10]. However, the explanatory power of attention is still under debate, as it may not always correlate with feature importance [40].

Datasets specifically curated for inconsistency detection and classification, or for training models to generate explanations, include:

- **e-SNLI:** This dataset extends the Stanford Natural Language Inference (SNLI) dataset by providing human-annotated explanations for the entailment relations [17].
- **CoS-E:** The Commonsense Explanations (CoS-E) dataset offers commonsense reasoning tasks along with explanations, which can be used to train models to generate textual explanations [78].
- **MultiRC:** The Multi-Sentence Reading Comprehension dataset (MultiRC) contains questions where each question requires reasoning over multiple sentences, and it includes annotations for the rationale behind the answer to each question [44].

Generated textual explanations add another layer of interpretability by creating human-readable content that reflects the model's decision-making process. These generated explanations are especially user-friendly, as they provide insights in a format that is naturally consumed by humans—language. Models that generate explanations can be assessed both for the correctness of their predictions and the validity of their explanations. This dual requirement can lead to better-aligned model behavior with human reasoning, as the model not only has to produce the right answer but also justify it in a human-like manner.

For example, in the work of [17], the authors trained a neural network to provide natural language explanations along with its predictions. The explanations not only increased the transparency of the model but also allowed for an additional training signal; models could be trained to predict both labels and justifications, providing a richer learning experience.

2.4.2 Evaluation Metrics for Explainability

Evaluating explainability in NLP models is an area of active research. Metrics and frameworks have been proposed to assess the quality of explanations.

Qualitative measures such as user studies where the interpretability of the model is subjectively evaluated by human judges [25]. Quantitative measures like consistency (how often the explanation remains the same when the input is slightly altered), fidelity (how accurately the explanations reflect the true reasoning of the model), and comprehensibility (how well a human can understand the explanation) are also used [81].

The development of robust evaluation metrics for explainability is still a challenge, and there is no one-size-fits-all metric. As such, the evaluation often involves a combination of different methods to capture the multifaceted nature of explainability (Doshi-Velez & Kim, 2017).

2.5 Fake News Detection

Fake news is a term that has gained immense popularity and attention in recent years. Within the context of factual inconsistencies, fake news can be defined as a deliberate presentation of (typically sensational) misinformation or hoaxes spread via traditional news media (print and broadcast) or online social media. The intent behind fake news is often to mislead, manipulate public opinion, or provoke confusion, and it does not stem from accidental misreporting but from intentional creation of falsehoods [6].

2.5.1 Detection Techniques

Detecting fake news is a multifaceted problem that encompasses various domains such as machine learning, data mining, natural language processing, and social network analysis. Two primary approaches

are employed: linguistic-based approaches, which analyze the text, and network-based approaches, which examine how information disseminates through networks.

Linguistic approaches focus on the content, analyzing the text for tell-tale signs of falsehood such as sensational language, inconsistencies, and the emotional tone. Tools like stylometry, discourse analysis, and natural language processing (NLP) are utilized to scrutinize the text [21].

Network-based approaches, on the other hand, look at the patterns of dissemination and the characteristics of the spread, such as the social network characteristics of spreaders and the speed of the spread. The assumption is that fake news spreads differently compared to genuine articles, often being propagated through bots or coordinated networks [87].

With the advancements in AI, particularly deep learning, the detection techniques have become increasingly sophisticated. Deep learning models such as CNNs, RNNs, and LSTM networks have shown promising results in capturing complex patterns and dependencies within the text. Moreover, transformer-based models like BERT and GPT-3 can understand the context and semantics at a much deeper level than previous models[94].

However, these models require large labeled datasets for training and can be computationally intensive. They also pose challenges in interpretability, which is crucial for understanding why a particular piece of news was flagged as fake.

2.5.2 Datasets and Evaluation

Developing and evaluating fake news detection systems hinge on the availability and quality of datasets. These datasets are typically collections of news articles or social media posts that have been labeled as 'fake' or 'real' by experts or through verifiable sources. Let's delve into some notable datasets:

- **LIAR Dataset:** Short for "Liar, Liar Pants on Fire," this dataset contains around 12.8K manually labeled short statements in various contexts from PolitiFact.com, which includes a range of political topics [98].
- **FakeNewsNet**: A dataset comprising news content from sites like PolitiFact and BuzzFeed. It includes textual and visual content, social context, and dynamic information like user engagement over time [86].
- **CREDBANK:** A large-scale crowdsourced dataset containing over 60 million tweets associated with event credibility annotations. It helps in understanding the spread of information in social networks [64].

The performance of fake news detection models is often evaluated using metrics such as Accuracy, Precision, Recall, and F_1 score. While accuracy measures the proportion of true results among the total number of cases examined, precision, recall, and the F1 score provide a more nuanced view of the model's performance, especially in datasets with imbalanced classes. The choice of metric can significantly affect how the performance of a model is perceived[91].

2.5.3 Integration with Fact Checking

The integration of fake news detection into the fact-checking process is seen as a synergy that amplifies the efficiency and effectiveness of verifying information in the digital age. Automated detection systems can support fact-checkers by conducting preliminary analyses of large volumes of news, flagging suspicious content, and identifying patterns indicative of misinformation. These systems harness the power of machine learning algorithms and natural language processing to scan through text, detect anomalies, inconsistencies, or exaggerated emotional cues that often characterize fake news. Once flagged, these items can be subjected to deeper human-led investigation. The blend of artificial intelligence and human expertise enables a more rapid response to emerging disinformation, especially during critical times such as elections or public health crises.

Leveraging the strengths of both AI and human judgment, the marriage of detection and fact-checking can also serve an educational purpose, highlighting the features of news articles that might indicate untruthfulness and thus informing the public on how to be more critical of the information they consume. By providing explanations for the AI's findings, which can be further validated by professional fact-checkers, the integrated system fosters a greater understanding of misinformation, ultimately encouraging a more discerning readership. The collaboration between automated systems and fact-checkers not only improves the scalability of verification efforts but also acts as a continuous feedback loop, where the insights from fact-checking can refine and enhance the accuracy of fake news detection algorithms.

Chapter 3

Task Adaptive Pretraining of Transformers for Hostility Detection

3.1 Introduction

With the increase in the number of active users on the internet, the amount of content available on the World Wide Web, and more specifically, that on social media, has seen a sharp rise in recent years. A sizable portion of the available content contains hostility, thereby posing potential adverse effects upon its readers. Content that is hostile in the form of, say, a hateful comment, unwarranted usage of offensive language, attempt at defaming an individual, or a post spreading some misinformation circulates faster as compared to typical textual information [60, 96]. Identifying and pinpointing such instances of hostility is of utmost importance for ensuring the sanctity of the World Wide Web and the well-being of its users. Multiple endeavors have been made to design systems that can automatically identify harmful content on the web [8, 9, 48, 58, 75].

In this chapter, we focus on the problem of identifying specific Hindi Tweets which are hostile. We further analyze whether the Tweet can fit into one or more of the following buckets: hateful, offensive, defamation, and fake. The popularity of pretrained Transformer-based [93] models for tasks involving Natural Language Understanding is slowly making them the new baseline for text classification tasks. In such a situation, we experiment with Task Adaptive Pretraining [35]. IndicBERT [43], which is similar to BERT [24] but trained on large corpora of Indian Language text is our primary pretrained Transformer of choice for dealing with Hindi text.

We adopt a model architecture similar to Ghosh Roy et al., 2021 [29], which leverages information from emojis and hashtags within the Tweet in addition to the cleaned natural language text which achieves Macro F1 scores of 90.29, 81.87 and 75.40 for hate speech detection in English, German and Hindi respectively. We are able to portray 1.35% and 1.40% increases for binary hostility detection and, on average, 4.06% and 1.05% increases for fine-grained classifications into the four hostile classes on macro and weighted F1 metrics respectively using Task Adaptive Pretraining (TAPT) before fine-tuning our architectures for classification.

Table 3.1: Distribution of Supervised labels in Training set

Label	**Frequency**
Non-Hostile	3050
Defamation	564
Fake	1144
Hate	792
Offensive	742

3.2 Dataset

The organizers of the Constraint shared task[1] provided the dataset for training and model development [12, 72]. The data was in the form of Tweets primarily composed in the Hindi language and contained annotations for five separate fields. Firstly, a coarse-grained label for whether the post is hostile or not was available. If a Tweet were indeed hostile, it would not carry the 'not-hostile' tag. Hostile Tweets carried one or more tags indicating its class of hostility among the following four non-disjoint sets (the Shared Task organizers provided the definitions for each category):

1. **Fake News**: A claim or information that is verified to be untrue. Example:

2. **Hate Speech**: A post targeting a specific group of people based on their ethnicity, religious beliefs, geographical belonging, race, etc., maliciously intends to spread hate or encourage violence.

3. **Offensive**: A post containing profanity, impolite, rude, or vulgar language to insult a targeted individual or group.

4. **Defamation**: A misinformation regarding an individual or group.

A collection of 5728 supervised training examples were provided, which we split into training and validation sets in an 80-20 ratio by Pareto principle, while a set of 1653 Tweets served as the blind test corpora. The mapping from a particular class to its number of training examples has been outlined in Table 3.1. The distribution of labels within the test set is shown in Table 3.2. Note that the test labels were released after the conclusion of the shared task. Throughout, a post marked as 'not-hostile' cannot have any other label while the remaining posts can theoretically have n labelings, $n \in \{1, 2, 3, 4\}$.

[1] `constraint-shared-task-2021.github.io`

Table 3.2: Distribution of labels in the Test set

Label	**Frequency**
Non-Hostile	873
Defamation	169
Fake	334
Hate	234
Offensive	219

3.3 Approach

In this section, we describe our model in detail and present the foundations for our experiments. We acknowledge that the language style for online social media text differs from that of formal and day-to-day spoken language. Thus, a model whose input is in the form of Tweets should be aware of and leverage information encoded in the form of emojis and hashtags. We base our primary architecture on that of Ghosh Roy et al., 2021 [29] with a few modifications.

3.3.1 Preprocessing and Feature Extraction

Similar to Ghosh Roy et al., 2021 [29], the raw input text is tokenized on whitespaces plus symbols such as commas, colons, and semicolons. All emojis and hashtags are extracted into two separate stores. The cleaned Tweet text, our model's primary information source, is free from non-textual tokens, including smileys, URLs, mentions, numbers, reserved words, hashtags, and emojis. The tweet-preprocessor[2] python library was used for categorizing tokens into the classes mentioned above.

To generate centralized representations of all emojis, we utilize emoji2vec [27] to generate 300-dimension vectors for each emoji and consider the arithmetic mean of all such vectors. We use the ekphrasis[3] Python library for hashtag segmentation. The segmented hashtags are arranged in a sequential manner separated by whitespaces, and this serves as the composite hashtag or 'hashtag flow' feature. Thus, we leverage a set of three features, namely, (a) the cleaned textual information, (b) the collective hashtag flow information, and (c) the centralized emoji embedding.

3.3.2 Architecture

This subsection outlines the flow of information pieces from the set of input features to label generation. We leverage two Transformer models to generate embeddings of size 768 for the cleaned text and hashtag flow features. The two Transformer-based embeddings are passed through two linear layers to yield

[2] github.com/s/preprocessor

[3] github.com/cbaziotis/ekphrasis

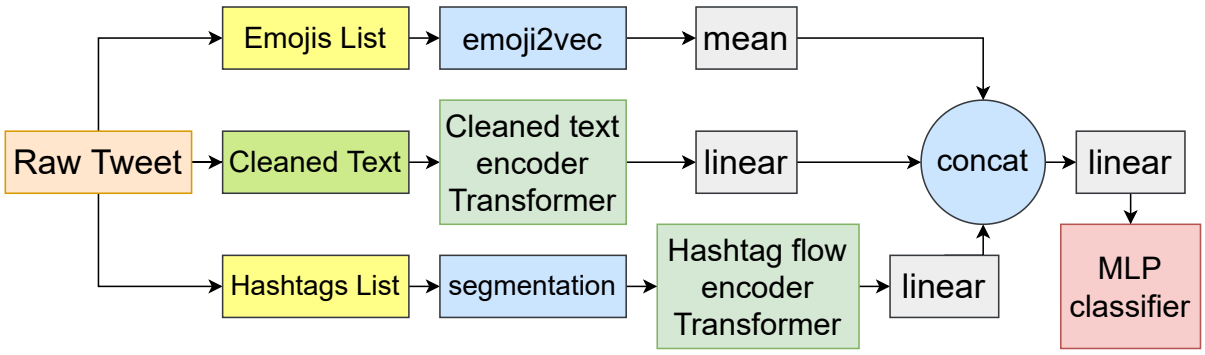

Figure 3.1: Model Architecture

the final vector representations for cleaned text and hashtag collection. The three vectors: cleaned text, composite hashtag, and centralized emoji representation are then concatenated and passed through a linear layer to form the final 1836-dimension vector used for classification. A dense multi-layer perceptron serves as the final binary classifier head. The overall information flow is presented in Figure 3.1. For the multi-label classification task, we trained our architecture individually to yield four separate binary classification models. In all cases, we performed end-to-end training on the available training data based on cross-entropy loss.

3.3.3 Task Adaptive Pretraining

We turn to Gururangan et al., 2020 [35], which showcases the boons of continued pretraining of Transformer models on natural language data specific to particular domains (Domain Adaptive Pretraining) and on the consolidated unlabelled task-specific data (Task Adaptive Pretraining). Their findings highlighted the benefits of Task Adaptive Pretraining (TAPT) of already pretrained Transformer models such as BERT on downstream tasks like text classification. We experimented with the same approach for our task of hostility detection in Hindi, having IndicBERT as our base Transformer model. Our results (in section 3.4) showcase the gains attributed to this continued pretraining with masked language modeling (MLM) objective. Note that only the cleaned text encoder Transformer is undergoing TAPT. The hashtag sequence encoder Transformer is initialized to pretrained IndicBERT weights. We create a body of text using all of the available training samples, and in that, we add each sample twice: firstly, we consider it as is, i.e., the raw Tweet is utilized, and secondly, we add the cleaned Tweet text. A pretrained IndicBERT Transformer is further pretrained upon this body of text with the MLM objective. We use these Transformer model weights for our cleaned text encoder before fine-tuning our complete architecture on the labeled training samples.

Table 3.3: Results on the Validation split for every category (% Weighted F1 Scores)

Metric	**Without TAPT**	**With TAPT**	**Gains**
Hostility (Coarse)	96.87	98.27	1.40
Defamation	86.47	86.31	-0.16
Fake	89.53	90.99	1.46
Hate	85.69	87.06	1.37
Offensive	87.12	88.66	1.54

Table 3.4: Results on the Validation split for every category (% Macro F1 Scores)

Metric	**Without TAPT**	**With TAPT**	**Gains**
Hostility (Coarse)	96.84	98.19	1.35
Defamation	59.43	63.38	3.95
Fake	83.69	86.52	2.83
Hate	70.77	74.20	3.43
Offensive	68.72	74.73	6.01

Table 3.5: Shared Task Results: Top 3 teams on public leaderboard (% F1 Scores)

Metric	**iREL IIIT** (Us)	**Albatross**	**Quark**
Hostility (Coarse)	**97.16**	97.10	96.91
Defamation	**44.65**	42.80	30.61
Fake	77.18	**81.40**	79.15
Hate	**59.78**	49.69	42.82
Offensive	**58.80**	56.49	56.99
Weighted (Fine)	**62.96**	61.11	56.60

3.4 Results

In Tables 3.3 and 3.4, we present metrics computed on our validation set. We observe 1.35% and 1.40% increases in the macro and weighted F1 scores for binary hostility detection and, on average, 4.06% and 1.05% increases in macro and weighted F1 values for fine-grained classifications into the four hostile classes. In all classes (except for 'Defamation' where a 0.16% performance drop is seen for the Weighted F1 metric), the classifier performance is enhanced upon introducing the Task Adaptive Pretraining. In Table 3.5, we present our official results with team name 'iREL IIIT' on the blind test corpora and compare it to the first and second runner-ups of the shared task.

3.5 Experimental Details

We used AI4Bharat's official release of IndicBERT[4] as part of Hugging Face's[5] Transformers library. All of our experimentation code was written using PyTorch[6] [71]. We considered a maximum input sequence length of 128 for both of our Transformer models, namely, the cleaned text encoder and the hashtag flow encoder. Transformer weights of both of these encoders were jointly tuned during the fine-tuning phase. We used AllenAI's implementation[7] of Task Adaptive Pretraining based on the Masked Language Modeling objective. The continued pretraining of IndicBERT upon the curated task-specific text was performed for 100 epochs with other hyperparameters set to their default values. The cleaned text encoder was initialized with these Transformer weights before the fine-tuning phase.

For fine-tuning our end-to-end architecture, we used Adam [46] optimizer with a learning rate of 1e-5 and a dropout [88] probability value of 0.1. All other hyperparameters were set to their default values, and the fine-tuning was continued for 10 epochs. We saved model weights at the ends of each epoch and utilized the set of weights yielding the best macro F1 score on the validation set. The same training and model weight-saving schema was adopted for the coarse binary hostility detector and the four binary classification models for hate, defamation, offensive, and fake posts.

3.6 Conclusion

In this chapter, we have presented a state-of-the-art hostility detection system for Hindi Tweets. Our model architecture utilizing IndicBERT as the primary Transformer encoder, which is aware of features relevant to online social media style of text in addition to clean textual information, is capable of both identifying hostility within Tweets and performing a fine-grained multi-label classification to place them into the buckets of hateful, defamation, offensive, and fake. Our studies proved the efficacy of performing

[4] github.com/AI4Bharat/indic-bert

[5] huggingface.co/

[6] pytorch.org/

[7] github.com/allenai/dont-stop-pretraining

Task Adaptive Pretraining (TAPT) of Transformers before using such encoders as components of a to-be fine-tuned architecture. We experimentally showed 1.35% and 1.40% gains for coarse hostility detection and average gains of 4.06% and 1.05% for the four types of binary classifications, on macro and weighted F1 score metrics, respectively, in both cases. Our system ranked first in the 'Hostile Post Detection in Hindi' shared task with an F1 score of 97.16% for coarse-grained detection and a weighted F1 score of 62.96% for fine-grained classification on the provided blind test corpora.

Chapter 4

Fact Extraction and Verification

4.1 Introduction

In the contemporary digital era, the Internet has become a vast repository of information, instantly accessible and ever-expanding. Yet, this boon of easy access is marred by the concurrent rise of misleading or false content. This issue, often referred to as "Fake News", presents significant challenges. It not only inundates us with questionable claims and assertions but also risks swaying public opinion, thereby undermining democratic principles.

Given these circumstances, relying solely on manual fact-checking is increasingly impractical. Such methods are labor-intensive and subject to human error and bias. Consequently, there is a pressing need for automated fact-checking processes to effectively counter the spread of misinformation. Responding to this need, advancements in natural language processing (NLP) and information retrieval have led to innovative approaches, one of which is the Fact Extraction and Verification (FEVER) shared task, initiated in 2018. This project represents a pivotal effort in tackling misinformation challenges.

FEVER, an acronym for Fact Extraction and Verification, entails assessing the accuracy of textual claims against a comprehensive body of factual evidence, primarily sourced from Wikipedia. Each claim is evaluated to determine if it is supported, refuted, or if there is insufficient information for a conclusive judgment. This process resembles Natural Language Inference (NLI), where a claim's validity is judged against a given premise. However, FEVER differs notably in its requirement to extract evidence from an extensive corpus, unlike the typical NLI tasks that work with paired claim-premise sets.

Central to FEVER is its dataset, comprising 185,445 carefully constructed claims, each verified against the introductory sections of Wikipedia articles. These claims are not merely extracted; they are often modified to enhance the complexity of the verification process. Annotators, tasked with justifying the classification of each claim, selected relevant Wikipedia sentences without prior knowledge of the claims' origins.

Despite its innovative approach, FEVER faces significant challenges. One prominent issue is balancing the annotation speed with the thoroughness of evidence gathering. Often, human-annotated evidence was found to be incomplete, presenting additional hurdles for systems designed to perform comprehensive

verification. This chapter will explore these challenges and the implications of FEVER in the broader context of automated fact-checking and misinformation management in the digital age.

In this work, we navigate these challenges, exploring various strategies for effective evidence retrieval, sentence selection, and determining textual entailment. We not only implement the foundational models for each of these sub-tasks but also delve into state-of-the-art transformer models for NLI, pushing the boundaries of what's possible in automated fact verification.

This chapter presents a series of significant contributions in the domain of fact extraction and verification:

- **Retrieval Model Implementation:** We detail our development of a retrieval model, tailored to address the specific challenges of the FEVER dataset.
- **Sentence Selection Strategy:** We describe our approach to identifying the most relevant sentences from retrieved articles for use as evidence.
- **Recognizing Textual Entailment (RTE):** Central to FEVER, we discuss not only the implementation of a baseline RTE model but also our exploration of an alternative approach to textual entailment, diverging from the traditional document retrieval methods.
- **Exploration of Advanced NLI Models:** We delve into the implementation of advanced transformer models for NLI, leveraging their power to enhance our verification system's accuracy.

In the subsequent sections, we will delve deeper into the nuances of each of these contributions, elucidating the methodologies adopted, challenges encountered, and the innovative solutions implemented to address them.

4.2 Related Work

Fact-Checking and Claim Verification: Fact-checking, the process of assessing the veracity of public statements, has gained substantial importance due to the spread of misinformation in the digital age. Early attempts at automating this task leaned heavily on structured knowledge bases, such as the method proposed by Ciampaglia et al.[19], which used the structure of knowledge graphs like Freebase to validate simple factual claims. However, the limitations of such methods, dictated by the coverage and freshness of the knowledge bases, became apparent. A different approach would be employing the frequency of a claim's appearance in trusted sources as a surrogate for its truthfulness, a strategy that was innovative but also sensitive to claim popularity over its factual correctness[36]. As the field evolved, researchers like Ferreira and Vlachos[28] began exploring the stance detection in source documents, determining whether a given source was in agreement, disagreement, or neutrality with a claim, thereby paving the way for leveraging textual entailment techniques in the realm of fact-checking. The rise of deep learning saw its integration into the fact-checking domain utilizing recurrent neural networks to understand the semantic

relationships between claims and their source documents[76]. While datasets like the LIAR dataset[98] provided a foundation, they often lacked the evidence or source linkage crucial for comprehensive fact-checking. Recognizing the importance of multi-source evidence, Thorne et al.[91] in their precursor works to FEVER emphasized the challenges and necessity of reasoning across multiple documents for effective verification.

Textual Entailment and Natural Language Inference: Textual entailment, the task of determining if one piece of text can be inferred from another, has long been a cornerstone problem in the domain of natural language understanding, exhibiting deep ties with the challenges posed in fact-checking and claim verification. Dagan et al.[22] were among the pioneers to formally define and introduce the concept, catalyzing a surge of research endeavors to address this intricate task. As the field matured, the emphasis shifted from traditional rule-based and alignment-based methods to leveraging large-scale datasets that could train more sophisticated models. A landmark in this trajectory was the introduction of the Stanford Natural Language Inference (SNLI) dataset, which provided a vast collection of sentence pairs annotated for entailment, contradiction, and neutrality[15]. This resource spurred the development of a plethora of neural architectures tailored for the problem. The Transformer-based model, BERT, showcased the potential of deep bidirectional representations in capturing intricate textual relationships[24]. While SNLI laid the foundational groundwork, the subsequent release of MultiNLI expanded the horizons by encompassing a diverse set of genres, thereby pushing models to generalize across varied linguistic structures and styles[99]. In tandem with these developments, the emergence of adversarial examples and stress tests, underscored the need for robustness in entailment models, advocating for a shift beyond mere accuracy metrics and towards a comprehensive evaluation paradigm[31].

Document Retrieval: The quest for effective document retrieval, a foundational pillar in information retrieval, has been a long-standing challenge, ensuring that relevant pieces of text or documents are efficiently retrieved from vast corpora. As the digital landscape expanded, the need for capturing deeper semantic relationships became evident. This realization paved the way for embedding-based models, where dense vector representations in the Word2Vec model, aimed to encapsulate the semantic essence of words and, by extension, documents[69]. These embeddings, when combined with efficient approximate nearest neighbor search algorithms, facilitated the retrieval of semantically similar documents. Advancing further into the neural age, attention-based architectures, epitomized by the Transformer model, offered a fresh paradigm, focusing on capturing intricate contextual relationships within and across documents[94]. Concurrently, models like DRMM highlighted the importance of modeling term matching distributions, marrying traditional term matching with the prowess of deep learning[34]. As the field hurtled forward, the integration of external knowledge bases and the exploration of cross-lingual retrieval, ensuring that the nuances of information extraction are not lost in translation, emerged as promising frontiers in the document retrieval landscape.

4.3 Task Overview

The FEVER task is not just about ascertaining the truth value of a claim; it is a multi-faceted problem that can be broken down as follows:

1. **Claim Identification**: Each participating system is presented with a claim, essentially a sentence whose veracity is unknown.

2. **Evidence Retrieval**: The core part of the task is not just verifying the claim but finding supporting or refuting evidence from Wikipedia. This has to be done at the granularity of individual sentences.

3. **Multi-evidence Synthesis**: It's noteworthy that in roughly 16.82% of cases, a single sentence was insufficient. Claims sometimes needed multiple sentences to be validated or refuted, making the task more intricate.

4. **Label Assignment**: Based on the procured evidence, the system then has to label the claim under one of three categories:

    - **SUPPORTED**: If the evidence from Wikipedia substantiates the claim.
    - **REFUTED**: If the claim is contradicted by the evidence found.
    - **NOTENOUGHINFO**: When the available Wikipedia evidence is insufficient to make a conclusive judgment on the claim.

Let us delve a bit deeper into the structural and operational specifics. Suppose we have a vast set of Wikipedia documents represented as $P = \{P_0, P_1, ...\}$. Each of these documents, say P_i, can be visualized as an array of sentences, i.e., $P_i = \{s_0^i, s_1^i, s_2^i, ...s_m^i\}$. Here, s_0^i is notably the title of the document. Given any claim c_i, the goal is to extract a subset of these sentences, S_{P_i}, as potential evidence.

The expected outcome from a system for a given claim c_i is a tuple, (E_i^*, y_i^*). Here:

- $E_i^* = \{s_{e0}, s_{e1}, ...\}$ denotes the subset of sentences from S_{P_i} that form the evidence.
- y_i^* is the label that the system predicts for the claim, belonging to one of the three categories: SUPPORTED (S), REFUTED (R), or NOTENOUGHINFO (NEI).

For a successful verification, both the predicted evidence set E_i^* should encompass the actual evidence set E_i, and the predicted label y_i^* should match the true label y_i.

4.3.1 Dataset

The dataset contains 145K+ annotated claims and relevant evidence(s) pointing to the relevant Wikipedia page. In addition to the training samples, around 20K samples are reserved for development/validation set. Another 20K samples are reserved as the test set on which official evaluation happens. The dataset was constructed in two stages:

Claim Generation The objective of this task was to generate claims from information extracted from Wikipedia.

Claim Labeling - Classifying whether a claim is supported or refuted by Wikipedia and selecting the evidence for it, or deciding there's not enough information to make a decision.

Table 4.1 show the training examples for the Support, Refutes and NotEnoughInfo class labels.

Sentence	**Label**	**Evidence**
Oliver Reed is an actor	SUPPORTS	**Title:** Oliver Reed Robert Oliver Reed (13 February 1938 – 2 May 1999) was an English actor known for his "hellraiser" lifestyle. ...
Lorelei Gilmore's father name is Robert.	REFUTES	**Title:** Lorelei Gilmore Lorelai Victoria Gilmore is a fictional character... Lorelai has a strained relationship with her wealthy parents, Richard and Emily, ...
Henri Christophe is famous for building a palace in Milot.	NOTENOUGHINFO	N/A

Table 4.1: Training examples from the FEVER Dataset Showcasing Sentence Claims and Corresponding Evidence

Out of the 145K claims, we were able to generate a total of 368K samples, as for a given sample there can be more than set of evidences for SUPPORT and REFUTES. Since we don't have any evidences for NOT ENOUGH INFO class, we randomly pick some evidences(by randomly sampling from the Wikipedia sentences) and consider them as the evidences for these labels. For each claim in the NOT ENOUGH INFO, we generate 3 negative samples. Table 4.2 shows the data distribution in the training, development and test sets.

4.3.2 Metrics for Evaluation

The scoring of this task considers classification accuracy and scoring recall. The score is awarded only if the correct evidence is found. This is known as **FEVER score**. It is considered that the correct evidence to be found if at least one complete set of annotated sentences is returned (the annotated data may contain multiple sets of evidence, each of which is sufficient to support or refute a claim).

Split	SUPPORTED	REFUTED	NEI
Training	80,035	29,775	35,639
Dev	6,666	6,666	6,666
Test	6,666	6,666	6,666

Table 4.2: Data distribution across different splits.

In the straightforward setting where the problem is formulated by the stance detection, we consider accuracy as the metric for evaluating the performance of the RTE/Stance classification model.

4.4 Preprocessing the FEVER Dataset

To ensure the consistency and clarity of the FEVER dataset, we adopted two vital preprocessing steps:

1. **Marker Replacement:** The dataset, originating from Penn Treebank formats, utilizes distinct markers for specific symbols. In particular, -LRB- (Left Round Bracket), -RRB- (Right Round Bracket), -LSB- (Left Square Bracket), and -RSB- (Right Square Bracket) are employed. However, these markers do not contribute semantically to the sentences. Therefore, they are replaced with spaces to improve the readability and comprehension of the dataset.

2. **Pronoun Substitution:** Wikipedia articles usually reference the primary entity of the page directly only in their opening lines. Subsequent mentions typically use pronouns like 'He', 'Him', 'She', and 'Her'. To enhance the clarity and specificity of the evidence statements, we replace these pronouns with the actual name of the entity.

    - **Example:** In the Wikipedia page of Mahatma Gandhi `https://en.wikipedia.org/wiki/Mahatma_Gandhi`, the sentence *"He went on to stay for 21 years"* gets transformed into *"Mahatma Gandhi went on to stay for 21 years"* after our preprocessing. We observed that such modifications significantly benefit the performance of the stance detection model.

4.5 Approach

4.5.1 System Overview

Our approach to fact checking and verification within the context of the FEVER dataset is organized into three distinct stages:

Document Retrieval: At this stage, our focus is on efficient and effective document retrieval from the Wikipedia dump. We employ various techniques including measuring document similarity with tf-idf and utilizing cosine similarity metrics.

Sentence Selection: Once we have the relevant documents, the next task is to select the top k sentences that correspond to our claim. To achieve this, we extract embedding vectors of the sequences using advanced models such as BERT, ELMo, and GloVe. These embeddings are then compared using cosine similarity to rank and select the most relevant sentences.

Recognizing Textual Entailment (RTE): In the RTE phase, our baseline implementation draws inspiration from Riedel, Benjamin, et al. (2017) [83]. We adopt a multi-layer perceptron (MLP) architecture with a single hidden layer. This MLP is designed to use term frequencies and TF-IDF cosine similarity between the claim and evidence as its primary features. Further we experimented with the transformer models and the novel infersent architecture based models.

4.5.2 Retrieval Model

In the FEVER task, evidence to substantiate or contest a claim often resides within the vast compendium of Wikipedia articles. The retrieval module of our pipeline serves as a crucial preliminary step, dedicated to extracting pertinent articles from this expansive Wikipedia dump.

Our approach is methodical:

1. **Processing the Wikipedia Dump:** Prior to any extraction, the Wikipedia dump undergoes a systematic preprocessing to ensure its readiness for subsequent stages.

2. **Extracting proper nouns:** We employ linguistic techniques to distill each claim, specifically extracting proper nouns which often hint at the most relevant articles.

3. **Article Retrieval:** Leveraging the extracted nouns, we search the preprocessed Wikipedia dump. Articles that display a high degree of relevance to the claim based on these key terms are retrieved for further analysis.

4.5.2.1 Preprocessing Wikipedia Dump

In order to have a make our searching in the Wikipedia dump efficient using keywords and decrease the time of searching, we need to create an index of the whole Wikipedia dump. We generated two indexes from the Wikipedia dump: one for the titles and another for the sentences in the body. Each of the indexes contain a postlist for every word in the wikipedia vocabulary. A postlist is a space-separated collection of hyphenated code containing the following information:

- The first part of the hyphen signifies the correspoding json file in which the article is stored
- The second part of the hyphen signifies the index or serial number of the article within that file
- The third part contains the number of instances of the word within the article.
- The index of the titles also contains a fourth part which stores the length of the corresponding title.

We have created temporary indexes which consisted of the postlists for each json file in Wikipedia dump and then we merged it using a k-way merging where k = 20. While forming and merging the postlists, we have ensured that the postlists are sorted according to the words in vocabulary. We have also formed an offset file of the index file, which keeps a track where the postlist of a word starts within the index.

4.5.2.2 Extracting Proper Nouns

A pivotal aspect of our methodology involves extracting named entities from the claims. These entities encompass a diverse range: from standard categories such as Locations, Organizations, and Persons, to more specific ones like movie or song titles. To adeptly identify and categorize these varied entities, we integrate the constituency parser from AllenNLP.

The complexity arises when certain entities do not conform to traditional syntactic norms. For instance, the claim "Down with Love is a great movie." presents a challenge. While AllenNLP might not immediately recognize "Down with Love" as a proper noun, it is, in fact, the title of a movie. To navigate such intricacies and ensure comprehensive entity extraction, we've implemented a heuristic approach. This involves considering all words preceding the main verb in the claim as potential entity mentions. Such a strategy augments our extraction accuracy, making it more probable to capture all pertinent named entities within a claim.

4.5.2.3 Article Retrieval

After extracting the proper nouns from the claim and index files, we use them to search in the Wikipedia dump. For every proper noun, we do a binary search on the offset file (as it is alphabetically sorted) to find the start position of the particular word in the index file. While creating the index of the body, we have stored the words present in the first 2 lines only. We search the proper nouns in both the titles and body. We have then assigned a score to articles on basis of the occurance of each of the proper noun in the title and body. We have scored it as follows:

- For an occurance of a proper noun in the title of an article, we have added 10 to the score of the article. For an occurrence of a proper noun in the body, we have awarded 0.1 multiplied by the number of times the proper noun have occurred in the first two sentences of the body. This is to ensure that we give higher priority to the presence of proper nouns within the titles. It is on the basis of the assumption that, proper nouns will have a Wikipedia page of its own and the observation that the evidences are generally present in one of articles of the proper nouns present in the claim.

- On the basis of the above rule, for a proper noun like "Kolkata", our model is also returning different titles like "Kolkata Metropolitan Area", "Port of Kolkata", "Climate of Kolkata", "List of tourist attractions in Kolkata", etc. which although contains the proper noun in the title, these

articles have nothing to do with the city of Kolkata. That's why we have also stored the length of the title and we have penalised the titles having longer length other than the proper nouns.

After scoring, we sort the articles on the basis of the scores and return the top n articles where n is a hyperparameter than can be passed to the retrieval module.

4.5.2.4 Evaluation and Error Analysis

In order to evaluate our Wikipedia Search Engine, we have considered all the claims that are verifiable within in the paper_dev.jsonl file. It accumulated to 6,666 claims. For each claim, we retrieve n number of articles which for each claim. We consider that we have retrieved the correct article if we have an overlap between the retrieved articles and the articles present within the evidence set of a particular claim. We then compare the results of our model with baseline model in the task paper which uses the DrQA system that returns the n-nearest documents for a query using cosine similarity between unigram and bigram Term Frequency-Inverse Document Frequency (TF-IDF) vectors and the articles retrieved by the MediaWiki API which was included from this paper.

Value of n	Baseline Model	Media Wiki	Our Model
3	-	92.60	81.22
5	70.20	93.30	86.52
7	-	93.55	88.10
10	77.24	-	88.46

Table 4.3: Retrieval results

We can see that our model have improved a lot over the baseline model and also gave comparable results with the state-of-the-art MediaWiki API.

4.5.3 Sentence Selection

After we got the probable articles from the retrieval model, we need to extract probable sentences to pass to our natural inference model. Here we have used two methods:

- **Tf-idf features with cosine similarity**: Here we have fit a TfidfVectorizer using the claim and all the sentences from the retrieved articles. We then got the tf-idf representation of the claim and each sentences and found cosine similarity of the claim with each of the sentences. We then select the top 5 sentences with highest cosine similarity as our evidence set.
- **Embeddings from pre-trained BERT**: Similar to the last point, here we retrieve pre-trained BERT embeddings for the claim and the sentences in the articles. Unfortunately, we couldn't use it

because it was taking around 40 secs to get embeddings for all the sentences in an articles for a single claim.

For the claim, "Telemundo is a English-language television network.", the top 5 retrieved sentences are:

- Telemundo News is the flagship daily evening television news program of Noticias Telemundo , the news division of the American Spanish language broadcast television network Telemundo .
- Telemundo is an American Spanish-language terrestrial television network owned by Comcast through the NBCUniversal division
- Telemundo Deportes is the programming division of the NBC Sports Group , owned by the NBCUniversal Television Group division of NBCUniversal , that is responsible for the production of sports events and magazine programs that air on NBCUniversal 's Spanish language television networks Telemundo and Universo .
- It is the second-most watched Spanish language network newscast in the United States ,trailing slightly behind Noticiero Univision in the ratings .
- Telemundo is an American broadcast television television network owned by the Telemundo Television Group division of NBCUniversal , which was launched in 1984 as NetSpan.

We can see that second sentence and the fourth sentence supports our claim.

4.5.4 Recognizing Textual Entailment

In this section we discuss about the methods and experiments we conduct for recognizing the textual entailment. We formulate the problem of textual entailment as a sentence pair classification problem. The textual entailment model takes a claim-evidence sentence pair and outputs one of the 3 classes - SUPPORT if the evidence supports the claim, REFUTES if the evidence refutes the claim and NOT ENOUGH INFO if sufficient information is not available to draw valid conclusions.

In this module, we describe our approaches in 3 sub sections. The baselines, the transformer models and the novel infersent architecture based models.

4.5.4.1 Baseline Models

For our baseline model in textual entailment for the FEVER task, we adopted the architecture presented in the paper by Riedel et al. [83].

This model employs a succinct neural design, leveraging a single-layer fully connected neural network for classification. Key features that form the input to this network are the term frequencies of the claim (TF_C) and evidence (TF_E), alongside the dot product of their TFIDF representations. This architectural representation can be referred to in Fig. 4.1.

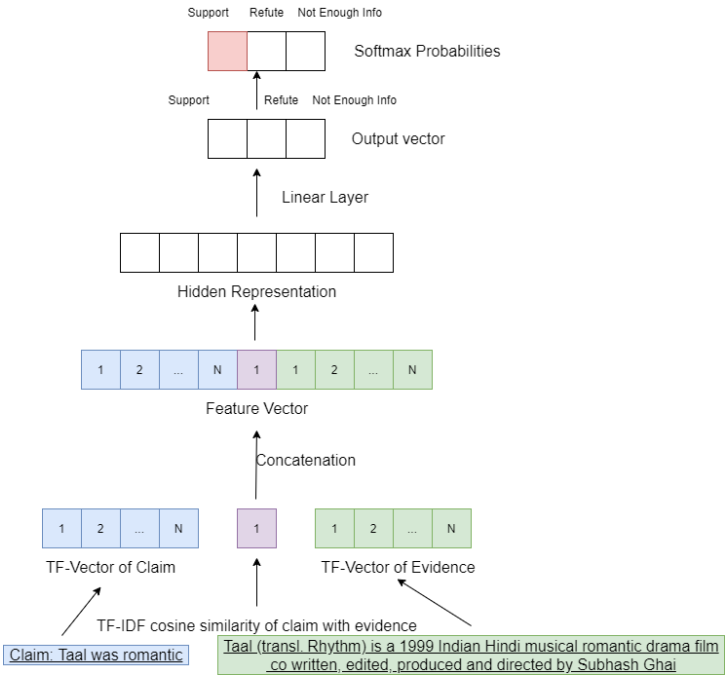

Figure 4.1: Schematic diagram of the baseline neural network showing the feature vector

In processing a claim and its associated evidence, both text pieces are converted into TFIDF vectors. Aligning with the methodology in [83], our vocabulary is restricted to the 5000 most frequent words. Using scikit-learn [73], we generate the TFIDF vectors and evaluate their dot-product similarity. These components are concatenated, forming the full feature vector that feeds into the neural network.

The neural network is designed with a single hidden layer housing 300 units, activated by the ReLU function. Its output layer has three units, employing softmax activation for classification purposes. Our training implementation utilized Keras, backed by Tensorflow.

Our training regimen spanned 10 epochs, deploying rmsprop as the optimizer and operating in batches of 1024 samples. Throughout training, we observed a steady reduction in loss.

We also ventured into an alternative entailment model, termed NN_SUM. This model comprises a neural network with three successive 600-dimensional fully connected layers, all employing ReLU activation. It culminates in an output layer activated by softmax, designed for triclass classification, as visualized in Fig. 4.2.

In this enhanced approach, the claim and evidence undergo transformations through diverse embeddings such as GloVe and FastText. Each sentence's dense representation is derived from summing its constituent word vectors. By merging the dense representations of both claim and evidence, we direct the combined vector through the triple-layered ReLU-activated network. The final layer, activated by softmax, provides the probabilistic distribution across classes.

The training process for this model lasted 20 epochs, also using rmsprop as the optimizer and processing in 1024-sample batches. Consistent with our previous observations, there was a continual decline in loss across epochs.

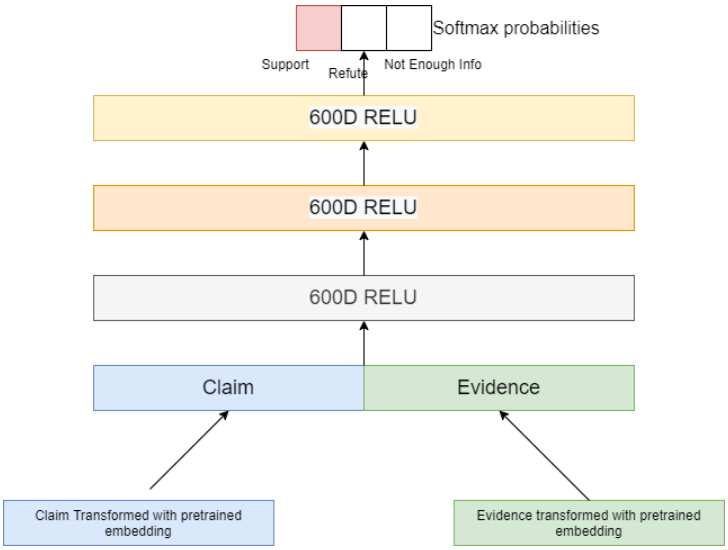

Figure 4.2: Schematic diagram of the NN_SUM Embedding neural network

4.5.4.2 Vanilla Transformer Models

Transfer learning has become a cornerstone in Natural Language Processing (NLP), consistently achieving state-of-the-art results across a myriad of tasks. At the heart of this shift lies the Transformer architecture, introduced by Vaswani et al. in 2017 [93]. Unlike recurrent neural networks (RNNs), Transformers excel in handling ordered data sequences, like natural language, for a variety of tasks such as machine translation, text classification, and summarization. They quickly surpassed traditional models like LSTM networks due to their design which promotes parallelism during training, making large-scale data training feasible. This set the stage for models such as BERT and GPT-2, which are pretrained on extensive language datasets and later fine-tuned for specific applications.

BERT, a noteworthy pre-trained language model, has set new standards in several NLP challenges. Leveraging such pretrained models for domain-specific tasks has yielded remarkable outcomes. Transformers, given their design, are inherently suited for sentence pair classification tasks. This eliminates the need for significant architectural modifications. As shown in Fig. 4.3, the BERT architecture is used for classifying sentence pairs. By encoding claim-evidence pairs as sentences, the segment embeddings within BERT help understand the relationships between sentences. This model is then fine-tuned using labeled claim-evidence pairs across three epochs, a technique we refer to as the "vanilla transformer". We explored two transformer models in particular: BERT and RoBERTa.

We delved into two transformer models: BERT and RoBERTa, further described as follows:

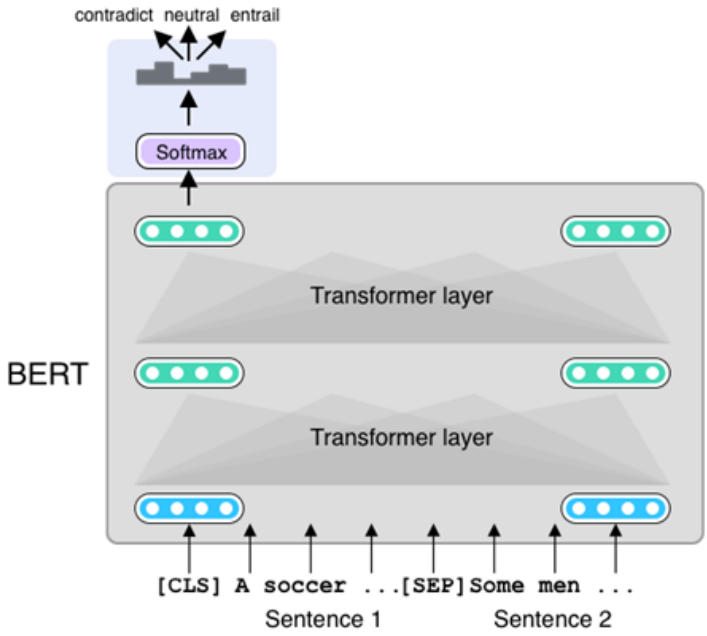

Figure 4.3: Sentence pair classification with BERT.

BERT: Introduced by Devlin et al. [24], BERT represents a transformative moment in NLP. Pretrained on massive amounts of unlabeled text from the web, it has set new benchmarks both as a foundation for research and for end-task applications. BERT's training incorporates two main tasks: Masked Language Modeling and Next Sentence Prediction. The model consists of 12 layers, with dimensions of 768 and 12 attention heads, totaling 110M parameters.

RoBERTa: Liu et al. introduced RoBERTa [52], a variant of BERT with optimization refinements. It diverges from BERT by using larger training data batches and a modified data masking approach. While BERT uses both masked language modeling and next sentence prediction during its pretraining, RoBERTa focuses solely on the former. BERT's data sources encompass the 16GB BookCorpus dataset and English Wikipedia, whereas RoBERTa draws from the extensive Common Crawl News corpus, which spans 63M English articles from September 2016 to February 2019. In benchmarks like GLUE, RoBERTa outperforms BERT. Structurally, it mirrors BERT with 12 layers, 768 dimensions, and 12 attention heads, resulting in 110M parameters.

4.5.4.3 InferTransformer Models

In light of the proven capabilities of NLI-based encoders [20], we integrated them into our RTE framework for the FEVER task. Our method merges the Infersent-inspired architecture with transformer models. The architecture of InferSent by Conneau et al. [20] is depicted in Figure 4.4. Unlike the original study that utilized embeddings from GloVe [74] and FastText [13], our approach uniquely employs transformer-based representations for both claims and evidence.

We were motivated to choose this direction for several reasons:

1. Transformers weren't as widespread during InferSent's inception, positioning our hybrid model as a contemporary iteration built upon LSTM/RNN foundations.

2. Fine-tuning standard transformer models can be computationally intensive. In our approach, we bypass this by only fine-tuning the appended fully connected layers, leaving the transformer layers unchanged. This not only facilitates quicker training but also, as our experiments indicate, competes favorably against the benchmarks set by conventional transformers.

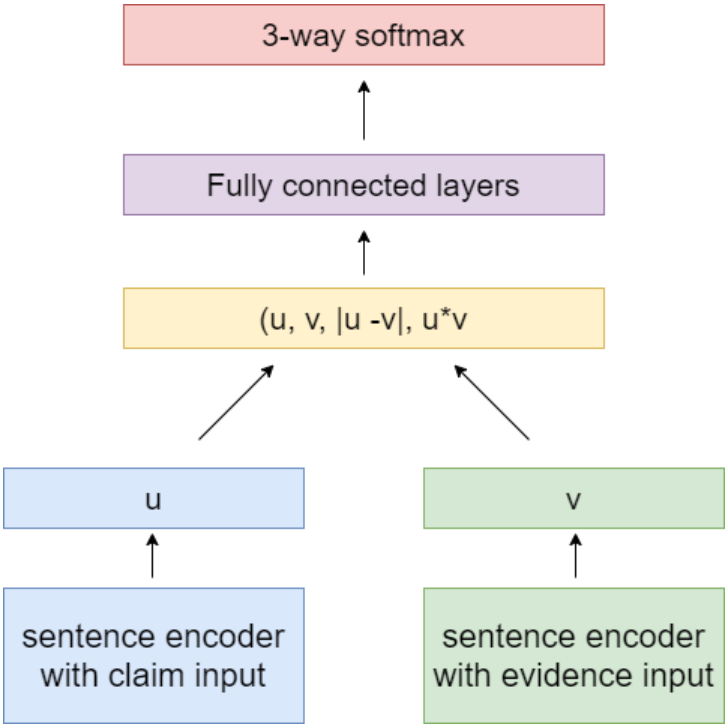

Figure 4.4: InferSent training scheme for NLI

For our experiments, we employed two renowned transformer architectures: BERT and RoBERTa, integrating them into our Infer-Transformer framework.

InferBERT: With BERT at its core, our model derives distinct dense representations for both claim and evidence using the pretrained BERT base uncased variant. Post combination of these representations, they're introduced to a fully connected neural network for a span of 20 epochs. Classification is achieved through a three-way softmax mechanism, with the model being trained on categorical cross-entropy loss.

InferRoBERTa: Following a similar methodology but with RoBERTa as its backbone, our model transforms both claim and evidence to procure dense representations via the pretrained RoBERTa base variant. Post amalgamation, they're processed through a fully connected neural network over 20 epochs. The three-way softmax mechanism ensures classification, trained on the categorical cross-entropy loss.

Model	Description	Precision	Recall	F1	Accuracy	Runtime
NN_TFIDF	5000 features	0.8239	0.8156	0.8192	0.8501	00h:07m:15s
NN_TFIDF	4000 features	0.8324	0.8014	0.8131	0.8503	00h:06m:44s
NN_TFIDF	3000 features	0.8182	0.7958	0.8044	0.8423	00h:04m:59s
NN_TFIDF	2000 features	0.7995	0.7723	0.7814	0.8270	00h:03m:29s
NN_TFIDF	1000 features	0.7692	0.7286	0.7390	0.7975	00h:01m:54s
NN_SUM_EMB	Glove840B(300d)	0.8373	0.8216	0.8254	0.8545	00h:23m:15s
NN_SUM_EMB	FastText	0.8551	0.8054	0.8252	0.8534	00h:24m:44s
Vanilla-BERT	Finetuned BERT model	0.9102	0.9268	0.9179	0.9303	06h:18m:48s
Vanilla-RoBERTa	Finetuned RoBERTa model	**0.9641**	**0.9655**	**0.9648**	**0.9707**	03h:14m:42s
INFER-GloVe	Glove840B(300d)	0.8434	0.8053	0.8209	0.8572	00h:19m:35s
INFER-FastText	FastText	0.8551	0.8054	0.8252	0.8534	00h:22m:14s
INFER-BERT	InferTransformer model (BERT)	0.9161	0.9095	0.9127	0.9267	00h:13m:46s
INFER-RoBERTa	InferTransformer model (RoBERTa)	**0.9504**	**0.9378**	**0.9436**	**0.9534**	00h:12m:27s

Table 4.4: Results of all the models on the validation set. Best performance values are shown in bold

4.5.5 Other Experimental Details

We formulate the problem of textual entailment in two settings, one without retrieval, and another one with retrieval. In the setting without retrieval, it is assumed that for the given claim, the required evidence has been retrieved perfectly and the evidence along with the claim is directly used to predict the textual entailment. Hence the results are better due to the ideal retrieval. As evident from the setting, this is just vanilla textual entailment formulated as a 3-way text classification problem.

In the second setting, we include the retrieval mechanism. Given a claim, first the required evidence has to be retrieved, and for the evidences retrieved, each claim evidence pair is fed into the network for prediction. The predictions are aggregated in the following way to get the evidence.

If any of the prediction is SUPPORTS or REFUTES, then that particular evidence is emitted. If all the predictions return NOT ENOUGH INFO, then, the aggregated result is NOT ENOUGH INFO. The

end to end performance of such a system will be lesser than that of the previous setting as there is some loss of accuracy due to the imperfect retrieval.

4.6 Results and Analysis

Table 4.4 provides a comprehensive view of the performance metrics of various models evaluated on the validation set. Based on the table, a few observations and inferences can be drawn:

- **TFIDF Representations**: The performance of the NN_TFIDF model varies with the number of features. As the number of features reduces from 5000 to 1000, we observe a noticeable decline in performance metrics, especially in F1 and accuracy scores. This indicates the importance of a richer feature set in capturing semantic nuances in the RTE task.

- **Embedding Choices**: The models using summed word embeddings, NN_SUM_EMB, present relatively consistent results. Specifically, the FastText English embeddings outperform the Glove840B embeddings marginally in terms of accuracy and F1 score. This suggests that FastText embeddings, which account for subword information, might be better suited for this task.

- **Vanilla Transformer Models**: As expected, finetuned transformer models like Vanilla-BERT and Vanilla-RoBERTa significantly outperform the other models. Vanilla-RoBERTa, in particular, achieves the highest performance in all metrics, reflecting the model's capacity to understand intricate relationships in the claim and evidence pair.

- **Hybrid InferTransformer Models**: The INFER variants of transformer models provide an interesting insight. They do not perform as well as their vanilla finetuned counterparts but still exhibit impressive results. For instance, INFER-RoBERTa, despite not being the best, produces an F1 score of 0.9436, highlighting the potential of such hybrid models. It also emphasizes the benefits of transformer representations even without extensive finetuning, offering a balance between performance and computational efficiency.

- **Computational Efficiency**: It is noteworthy to consider the training and inference times. The choice of model can be influenced by the trade-off between performance and computational overhead. For instance, INFER-RoBERTa achieves competitive results in considerably less time than Vanilla-RoBERTa.

In conclusion, while finetuned transformer models like Vanilla-RoBERTa offer top-tier performance, the hybrid approaches like InferTransformer models also present a compelling case, especially in scenarios where computational efficiency is paramount. The choice of word embeddings and feature sets can significantly influence the model's efficacy, making it crucial to tailor these choices based on the specific demands of the RTE task.

Fig. 4.5 and Fig. 4.6 show the plots of loss with respect to the number of epochs for each model. We can observe that the loss gradually decreases with the number of epochs for both the models.

Table 4.5 shows some examples where our best model failed to predict the right label.

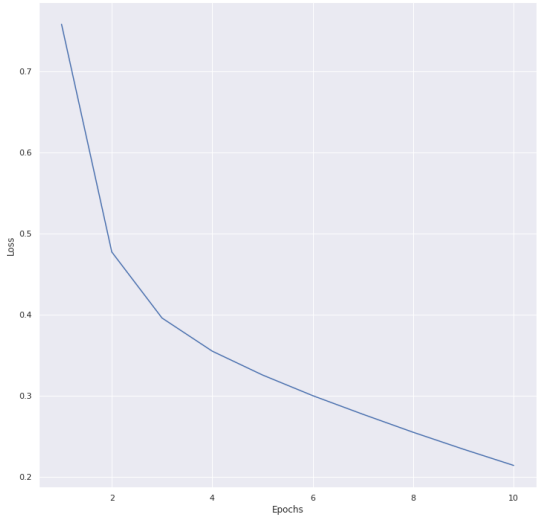

Figure 4.5: Loss curve of the model NN_TFIDF trained with 5000 features for vectorizer

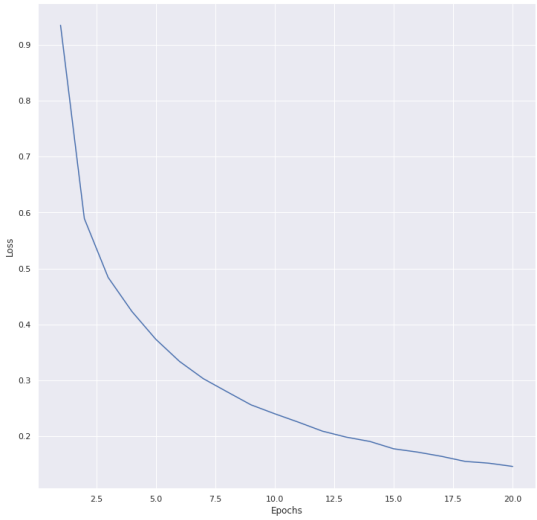

Figure 4.6: Loss curve of the model NN_SUM_EMB trained with GloVe840B300d pretrained embeddings

4.7 Summary and Future Work

In this work, we implemented end to end full pipeline for retrieval of relevant claims and evaluated multiple neural architectures for stance detection. The first model used sparse TFIDF representations of the claim and the evidence to train a network for stance prediction. In the second model, we use dense pretrained word embeddings of the claim and the evidence to train a deep neural network. We observe that the model using dense representations was performing better than the model with sparse representations. The model with more layers also performed better. Further, we explore the transformer architectures in their standard setting where the problem was formulated as a sentence pair classification, very similar to the tasks in the GLUE benchmarks. Further, we proposed and evaluated a hybrid InferTransformer model which uses the Infersent kind of architecture but uses transformer representations as the sentence encoders.

In future work, we plan to explore the use of knowledge graphs for more efficient and comprehensive evidence retrieval. A knowledge graph could provide a better perspective and computationally quicker retrieval of evidence relevant to a given claim compared to the current Wikipedia-based approach.

Additionally, we will investigate pronoun resolution techniques to avoid potential errors when substituting pronouns with named entities.

In the next chapter, we will use the FEVER dataset to form our FICLE dataset to detect factual inconsistencies in text.

Claim	Evidence	Ground Truth	Predicted Label
Paul Simon is a person.	In 2015 , Paul Simon was named as one of the 100 Greatest Songwriters by Rolling Stone	SUPPORTS	REFUTES
Belgium is comprised of three regions.	It is divided into three regions and three communities , that exist next to each other	SUPPORTS	NOT ENOUGH INFO
Vikrant Massey starred in The X-Files.	It was later given a wide release at over 1,700 theaters in the United States and Canada on January 10 , 2014	NOT ENOUGH INFO	SUPPORTS
Octopuses do not release ink into the water as an escape mechanism.	The latter half of their career saw a series of record-breaking tours that earned the group a reputation for excess and debauchery . record-breaking tours Led Zeppelin concerts	NOT ENOUGH INFO	REFUTES
Whoopi Goldberg co-produced an American dance tournament.	Selena began recording English-language songs for her crossover album. crossover crossover (music) Selena Selena (film)	NOT ENOUGH INFO	SUPPORTS
The Taj Mahal attracts significantly less than 7-8 million visitors a year	The Taj Mahal attracts 7 – 8 million visitors a year	REFUTES	NOT ENOUGH INFO

Table 4.5: Samples where the best performing model made errors in prediction

Chapter 5

Factual Inconsistencies: Dataset and our Approach

5.1 Introduction

Factual inconsistencies in text, especially in the context of natural language generation, pose significant challenges in maintaining the reliability and clarity of generated content. Such inconsistencies can lead to confusion, create mistrust among readers, and diminish the overall quality of the text by leading to inaccurate conclusions and interpretations. Tackling this problem has led to various approaches, including the training of robust neural language generation models that aim to reduce hallucinations and improve fidelity, as well as employing human annotators for post-checking. However, the scalability of manual checking remains a concern, highlighting the need for automated detection and explanation of factual inconsistencies.

While Transformer-based natural language generation models like BERT, RoBERTa, and DeBERTa have revolutionized various applications such as summarization, dialogue generation, and machine translation, they are not without limitations. Among these, the generation of hallucinatory and inconsistent text stands out as a critical issue. These models, while state-of-the-art in many respects, often struggle to maintain consistency and accuracy in the text they generate, necessitating further research and development to overcome these challenges. [41].

Accordingly, there have been several studies in the past which focus on detection of false or fake content. Fake content detection studies [**?**, 85, 95] typically verify facts in claims with respect to an existing knowledge base. However, keeping the knowledge base up-to-date (freshness and completeness) is difficult. Accordingly, there have been other studies in the natural language inference (or textual entailment) community [**?**, 68, **?**] where the broad goal is to predict entailment, contradiction or neither. More than a decade back, De Marneffe et al. [23] proposed the problem of fine-grained contradiction detection, but (1) they proposed a tiny dataset with 131 examples, (2) they did not propose any learning method, and (3) they did not attempt explanations like localization of inconsistency spans in claim and context.

In response to these challenges, this chapter introduces the novel problem of factual inconsistency classification with explanations (FICLE). Given a (claim, context) sentence pair, our goal is to predict

Claim	The Invention of Lying is *only a book*.
Context	The Invention of Lying is *a 2009 American fantasy romantic comedy film* written and directed by Ricky Gervais and Matthew Robinson.

Inconsistent Claim Fact Triple	**Source**	The Invention of Lying
	Relation	is
	Target	*only a book*
Inconsistent Context Span		*a 2009 American fantasy romantic comedy film*
Inconsistent Claim Component		Target Head
Inconsistency Type		Taxonomic sisters (book vs film)
Coarse Inconsistent Entity-Type		entertainment
Fine-grained Inconsistent Entity-Type		entertainment_movie

Figure 5.1: Factual Inconsistency Classification with Explanation (FICLE) Example: Inputs are claim and context. Outputs include inconsistency type and explanation (inconsistent claim fact triple, inconsistent context span, inconsistent claim component, coarse and fine-grained inconsistent entity types).

inconsistency type and explanation (inconsistent claim fact triple, inconsistent context span, inconsistent claim component, coarse and fine-grained inconsistent entity types). Fig. 5.1 shows an example of the FICLE task. Two recent studies are close to our work: e-SNLI [17] and TaxiNLI [42]. Unlike detailed structured explanation (including inconsistency localization spans in both claim and context) from our proposed system, e-SNLI [17] contains only an unstructured short sentence as an explanation. Unlike five types of inconsistencies detected along with explanations by our proposed system, TaxiNLI [42] provides a two-level categorization for the NLI task. Thus, TaxiNLI focuses on NLI and not on inconsistencies specifically. Table 5.1 shows a comparison of our dataset with other closely related datasets. In this work, based on linguistic theories, we carefully devise a taxonomic categorization with five inconsistency types: simple, gradable, set-based, negation, taxonomic relations. First, we obtain English (claim, context) sentence pairs from the FEVER dataset [89] which have been labeled as contradiction. We get them manually labeled with inconsistency types and other explanations (as shown in Fig. 5.1 by four annotators. Overall, the dataset contains 8055 samples labeled with five inconsistency types, 20 coarse inconsistent entity types and 60 fine-grained inconsistent entity types, whenever applicable.

We leverage the contributed dataset to train a pipeline of four neural models to predict inconsistency type with explanations: M_1, M_2, M_3 and M_4. Given a (claim, context) sentence pair, M_1 predicts the inconsistent subject-relation-target fact triple $\langle S, R, T \rangle$ in the claim and also the inconsistent span in the context. M_2 uses M_1's outputs to predict the inconsistency type and the inconsistent component (subject, relation or target) from the claim. M_3 uses the inconsistent context-span and inconsistent claim component to predict a coarse inconsistent entity type. M_4 leverages both M_3's inputs and outputs to predict fine-grained inconsistent entity type. Overall, the intuition behind this pipeline design is to first predict inconsistent spans from claim and context; and then use them to predict inconsistency types and inconsistent entity types (when inconsistency is due to entities). Fig. 5.3 shows the overall system architecture for FICLE.

Dataset	#Samples	Explanations	#Classes	Inconsistency localized?
Contradiction [23]	131	No	10	No
FEVER [89]	43107	No	1	No
e-SNLI [17]	189702	Yes	1	Yes
TaxiNLI [42]	3014	No	15	No
LIAR-PLUS [5]	5669	Yes	3	No
FICLE (Ours)	8055	Yes	5	Yes

Table 5.1: Comparison of FICLE with other datasets. #Samples indicates number of contradictory/inconsistent samples (and not the size of full dataset).

We investigate effectiveness of multiple standard Transformer [93]-based natural language understanding (NLU) as well as natural language generation (NLG) models as architectures for models M_1, M_2, M_3 and M_4. Specifically, we experiment with models like BERT [24], RoBERTa [52] and DeBERTa [38] which are popular for NLU tasks. We also experiment with T5 [77] and BART [51] which are popular in the NLG community. DeBERTa seemed to outperform other models for most of the sub-tasks. Our results show that while inconsistency type classification is relatively easy, accurately detecting context span is still challenging.

Overall, in this work, we make the following main contributions. (1) We propose a novel problem of factual inconsistency detection with explanations given a (claim, context) sentence pair. (2) We contribute a novel dataset, FICLE, manually annotated with inconsistency type and five other forms of explanations. (3) We experiment with standard Transformer-based NLU and NLG models and propose a baseline pipeline for the FICLE task. (4) Our proposed pipeline provides a weighted F1 of $\sim$87% for inconsistency type classification; weighted F1 of $\sim$86% and $\sim$76% for coarse (20-class) and fine-grained (60-class) inconsistent entity-type prediction respectively; and an IoU of $\sim$94% and $\sim$65% for claim and context span detection respectively.

5.2 Related Work

Factual Inconsistency in Natural Language Generations: Popular natural language generation models have been found to generate hallucinatory and inconsistent text [41]. Krysinski et al. [47] and Cao et al. [18] found that around 30% of the summaries generated by state-of-the-art abstractive models were factually inconsistent. There are other summarization studies also which report factual inconsistency of generated summaries [59, 61, 67, 97, 101, 104]. Similarly, several studies have pointed out semantic inaccuracy as a major problem with current natural language generation models for free-form text generation [16], data-to-text [26], question-answering [53], dialogue modeling [39, 63], machine translation [103], and news generation [100]. Several statistical (like PARENT) and model-based metrics have been proposed to quantify the level of hallucination. Multiple data-related methods and modeling

and inference methods have been proposed for mitigating hallucination [41], but their effectiveness is still limited. Hence, automated factual inconsistency detection is critical.

Natural Language Inference: Natural language inference (NLI) is the task of determining whether a hypothesis is true (entailment), false (contradiction), or undetermined (neutral) given a premise. NLI is a fundamental problem in natural language understanding and has many applications such as question answering, information extraction, and text summarization. Approaches used for NLI include earlier symbolic and statistical approaches to more recent deep learning approaches [15]. There are several datasets and benchmarks for evaluating NLI models, such as the Stanford Natural Language Inference (SNLI) Corpus [15], the Multi-Genre Natural Language Inference (MultiNLI) Corpus [99] and Adversarial NLI [68]. FEVER [89] is another dataset on a related problem of fact verification.

Recently there has been work on providing explanations along with the classification label for NLI. e-SNLI [17] provides a one-sentence explanation aiming to answer the question: "Why is a pair of sentences in a relation of entailment, neutrality, or contradiction?" Annotators were also asked to highlight the words that they considered essential for the label. NILE [49] is a two stage model built on e-SNLI which first generates candidate explanations and then processes explanations to infer the task label. Thorne et al. [90] evaluate LIME [80] and Anchor explanations [82] to predict token annotations that explain the entailment relation in e-SNLI. LIAR-PLUS [5] contains political statements labeled as pants-fire, false, mostly-false, half-true, mostly-true, and true. The context and explanation is combined into a "extracted justification" paragraph in this dataset. Atanasova et al. [7] experiment with LIAR-PLUS dataset and find that jointly generating justification and predicting the class label together leads to best results.

There has also been work on detailed categorization beyond just the two classes: contradiction and entailment. Contradiction [23] is a tiny dataset with only 131 examples that provides a taxonomy of 10 contradiction types. Recently, TaxiNLI [42] dataset has been proposed with 15 classes for detailed categorization with the entailment and not the contradiction category. Continuing this line of work, in this chapter, we contribute a new dataset, FICLE, which associates every (claim, context) sentence pair with (1) an inconsistency type (out of five) and (2) detailed explanations (inconsistent span in claim and context, inconsistent claim component, coarse and fine-grained inconsistent entity types).

5.3 FICLE Dataset

5.3.1 Data Curation

The FEVER dataset [89] is a valuable resource designed to support the development and evaluation of models for fact verification. It encourages research in natural language understanding and reasoning, as it requires models to determine the veracity of a claim based on its relationship to the provided evidence. By altering sentences from Wikipedia and removing knowledge of their origins, the FEVER dataset challenges models to rely on their understanding of language and the evidence presented, rather than on

simple pattern matching or other less sophisticated techniques. Comprising 185,445 claims generated by modifying sentences extracted from Wikipedia, the FEVER dataset includes a claim sentence, evidence (or context) sentence from a Wikipedia URL, and a type label ('supports,' 'refutes,' or 'not enough info') for each sample.

With this understanding of the FEVER dataset, we have developed the FICLE dataset through a series of processing steps. We focus on samples with the 'refutes' label to construct our dataset, as our primary goal is to detect inconsistencies and provide explanations for them. This necessitates identifying the location of inconsistencies between a claim and its context and developing a classification system that enhances explainability.

To create the FICLE dataset, we first gather all data points with the 'refutes' label from the FEVER dataset. For each data point, we extract the paragraph containing the evidence and use this information to form the basis of our dataset. The claim and evidence are then further annotated with additional information, which serves to enrich the dataset and facilitate the identification and explanation of inconsistencies. By combining these annotations with the original 'refutes' label data points, we create a comprehensive dataset designed to support the development of models capable of detecting and explaining inconsistencies in natural language generation.

5.3.2 Inconsistency Type Classification

Factual inconsistencies in text can occur because of a number of different sentence constructions, some overt and others that are complex to discover even manually. We design a taxonomy of five inconsistency types following non-synonymous lexical relations classified by Saeed [84, p. 66–70]. The

Inconsistent Claim	Inconsistent Context Span	Inconsistent Claim Component
Prime Minister Swami Vivekananda *enthusiastically hoisted* the Indian flag.	Narendra Modi	Subject-Head
President Narendra Modi *enthusiastically hoisted* the Indian flag.	Prime Minister	Subject-Modifier
Prime Minister Narendra Modi *enthusiastically lowered* the Indian flag.	hoisted	Relation-Head
Prime Minister Narendra Modi *halfheartedly hoisted* the Indian flag.	enthusiastically	Relation-Modifier
Prime Minister Narendra Modi *enthusiastically hoisted* the Indian culture.	flag	Target-Head
Prime Minister Narendra Modi *enthusiastically hoisted* the American flag.	Indian	Target-Modifier

Table 5.2: Inconsistent Claim Fact Triple, Context Span and Claim Component examples for the context sentence "Prime Minister Narendra Modi enthusiastically hoisted the Indian flag." Subject, relation and target in the claim are shown in bold, italics and underline respectively.

Claim	Context	Inconsistency Type	Coarse Inconsistent Entity Type	Fine-grained Inconsistent Entity Type
Kong: Skull Island **is not a** reboot.	The film **is a** reboot of the King Kong franchise and serves as the second film in Legendary's MonsterVerse .	Negation	enter-tainment	brand
The Royal Tenenbaums only stars **Emma Stone**.	The film stars **Danny Glover, Gene Hackman, Anjelica Huston, Bill Murray, Gwyneth Paltrow, Ben Stiller, Luke Wilson, and Owen Wilson**.	Set Based	name	musician
Lindsay Lohan began her career as **an adult fashion model**.	Lohan began her career as **a child fashion model** when she was three, and was later featured on the soap opera Another World for a year when she was 10 .	Simple	time	age
Karl Malone played the **shooting guard position**.	He is considered one of the best **power forwards** in NBA history .	Taxonomic Relation	profession	sport
The Divergent Series: Insurgent is based on the **third** book in the Divergent trilogy.	The Divergent Series : Insurgent is a 2015 American science fiction action film directed by Robert Schwentke, based on Insurgent, the **second** book in the Divergent trilogy by Veronica Roth.	Gradable	quantity	ordinal

Table 5.3: Inconsistency Type and Coarse/Fine-grained Inconsistent Entity Type examples. Inconsistent spans are marked in bold in both claim as well as context.

book mentions the following kinds of antonyms: simple, gradable, reverses, converses and taxonomic sisters. To this taxonomy, we added two extra categories, negation and set-based, to capture the FICLE's complexity. Also, we expanded the definition of taxonomic sisters to more relations, and hence rename it to taxonomic relations. Further, since we did not find many examples of reverses and converses in our dataset, we merged them with the simple inconsistency category. Overall, our FICLE dataset contains these five different inconsistency types.

- Simple: A simple contradiction is a direct contradiction, where the negative of one implies the positive of the other in a pair like *pass vs. fail*. This also includes actions/ processes that can be reversed or have a reverse direction, like *come vs. go* and *fill vs. empty*. Pairs with alternate viewpoints like *employer vs. employee* and *above vs. below* are also included in this category.

- Gradable: Gradable contradictions include adjectival and relative contradictions, where the positive of one, does not imply the negative of other in a pair like *hot vs. cold*, *least vs. most*, or periods of time etc.

- Taxonomic relations: We include three kinds of relations in this type: (a) Pairs at the same taxonomic level in the language like *red vs. blue* which are placed parallel to each other under the English color adjectives hierarchy. (b) When a pair has a more general word (*hypernym*) and another more specific word which includes the meaning of the first word in the pair (*hyponym*) like *giraffe* (hypo) vs. *animal* (hyper). (c) Pairs with a part-whole relation like *nose vs. face* and *button vs shirt*.

- Negation: This includes inconsistencies arising out of presence of explicit negation morphemes (e.g. *not*, *except*) or a finite verb negating an action (e.g. *fail to do X*, *incapable of X-ing*) etc.

- Set-based: This includes inconsistent examples where an object contrasts with a list that it is not a part of (e.g. *cat* vs. *bee, ant, wasp*).

5.3.3 Annotation Details

In order to provide in-depth explanations of inconsistencies, we carried out extensive annotations for every sample in the FICLE dataset. The annotation process was completed in two stages. The initial stage centered on ”syntactically-oriented” annotations, while the subsequent stage emphasized ”semantically-oriented” annotations. These annotations were conducted using the Label Studio annotation tool by a team of four annotators. The annotators possess strong English language skills and are undergraduate computer science students specializing in computational linguistics, aged between 20 and 22 years old. Comprehensive annotation guidelines are discussed in the following subsections.

5.3.3.1 Syntactic Oriented Annotations

During the annotation stage, the annotators provided labels for several syntactic fields for each sample. Examples of these fields can be found in Table 5.2. The following elements were annotated:

- Inconsistent Claim Fact Triple: Claims can consist of multiple facts. Annotators identified the fact that was inconsistent with the context and labeled the spans of source (S), relation (R), and target (T) within the claim fact. In some instances, such as with intransitive verbs, the target may be empty. Annotators also identified and labeled the head and modifier for each of the S, R, and T components. The head refers to the primary noun (for S and T) or verb phrase (for R), while the modifier is the phrase that modifies the meaning of the noun or verb.

- Inconsistent Context Span: Annotators marked a span in the context sentence that was inconsistent with the claim.

- Inconsistent Claim Component: This field can take one of six possible values, depending on which part of the claim fact triple is inconsistent with the context: Subject-Head, Subject-Modifier, Relation-Head, Relation-Modifier, Target-Head, or Target-Modifier.

The annotation process involved the following steps and detailed notes:

1. Carefully read the given pair of (claim, context) sentences, and identify the inconsistency without using external references or knowledge bases.

2. Highlight the ”Source Chunk,” ”Relation Chunk,” ”Target Chunk” in the claim, and the ”Inconsistent Span” in the context, which are involved in the inconsistency as identified above. Definitions for each chunk are provided below:

    - ”Source Chunk” is the linguistic chunk containing the entity lying to the left of the main verb/relating Chunk.
    - ”Relation Chunk” is the linguistic chunk containing the verb/relation at the core of the identified inconsistency.
    - ”Target Chunk” is the linguistic chunk containing the entity lying to the right of the main verb/relating chunk.
    - ”Inconsistent Span” is the chunk in the context sentence that is inconsistent with the source, relation, and target chunks identified in the claim.

3. Next, for each of the source, relation, and target chunks, identify the head and modifier, and label the inconsistent claim component as one of the Subject-Head, Subject-Modifier, Relation-Head, Relation-Modifier, Target-Head, or Target-Modifier. The head is the linguistic head of the chunk, and the modifier is the remaining part of the chunk.

Detailed notes and examples:

- If unsure about the parts of the sentence, use https://corenlp.run/ for the specific sub-phrase that is difficult to break down.

- Source or target chunks can be compound nouns. Sometimes this will be evident through the context. The preference is for the mismatch to be in the modifier as far as possible. Examples are provided in the original text.

- If a claim sentence has the form ”⟨Source⟩ ⟨Relation⟩ ⟨Target1⟩ ⟨Target2⟩”, include only the relevant target in the span (according to where the mismatch is in the evidence). An example is provided in the original text.

- Finite verbs taking the main verb as a complement are part of the verb phrase (hence the relation), such as ”We tried to pass OSN.” These finite verbs are relation modifiers.

- When a sentence has only a subject and a verb in the present perfect tense (”X has Y’ed”), then the auxiliary ”has” is the relation, and the past participle is the target.

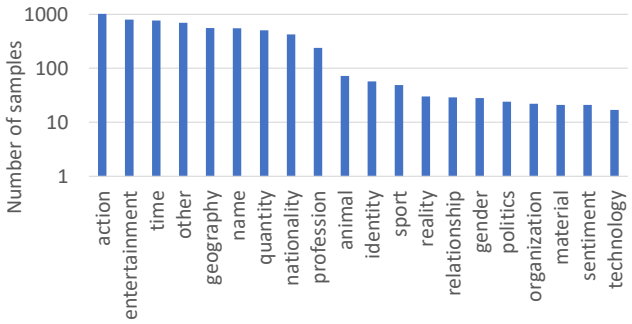

Figure 5.2: Distribution of coarse inconsistent entity types in FICLE.

		Min	Avg	Max
	Claim	3	8.04	31
	Context	5	30.73	138
Incon. Claim	Source	1	2.29	9
	Relation	1	2.17	18
	Target	0	3.39	21
	Incon. Context-Span	1	5.11	94

Table 5.4: Minimum, average, and maximum size (words) of various fields averaged across samples in FICLE dataset.

- For claims of the form "X ⟨verb⟩ ⟨preposition⟩ Y", if the verb is incomplete (semantically) without the preposition, it should be in the relation span; otherwise, it should be in the target span with Y.
- If the evidence does not contain an overt relation, but the inconsistency occurs in the relation of the claim, then in the evidence, mark the entity which is in focus and shared with the claim. An example is provided in the original text.

These detailed notes and examples are included to assist annotators with sentence deconstruction, chunk identification, and understanding various sentence structures.

5.3.3.2 Semantic Oriented Annotations

During the annotation process, annotators focused on labeling various semantic fields for each sample. Examples of these fields can be found in Table 5.3. The following aspects were taken into consideration:

1. Inconsistency Type: Each sample in the dataset was annotated with one of the five predefined inconsistency types, as outlined in Section 5.3.2.
2. Coarse Inconsistent Entity Type: If the inconsistency was due to a specific entity, annotators were required to label the entity with one of the 20 coarse types. The entity types included action, animal, entertainment, gender, geography, identity, material, name, nationality, organization, others, politics, profession, quantity, reality, relationship, sentiment, sport, technology, and time.
3. Fine-grained Inconsistent Entity Type: In addition to the coarse entity type, annotators also labeled a more specific, fine-grained entity type from a list of 60 options, further refining the entity causing the inconsistency.

The annotation process for determining the inconsistency entity type was conducted in two stages. In the first stage, annotators worked on 500 samples without any restrictions on the categories they

could use for labeling, at both coarse and fine-grained levels. Once this initial phase was completed, annotators engaged in discussions to de-duplicate category names and consolidate rare categories with more frequent ones. As a result, a final list of 20 coarse and 60 fine-grained entity types, including "others," was established.

For the second stage, annotators were asked to choose one category from the refined list for each sample. The purpose of this iterative process was to ensure a more consistent and accurate classification of entity types related to inconsistencies in the dataset.

During the entire annotation process, annotators were instructed not to use any external references or knowledge bases. They solely relied on the given claim sentence, context sentence, inconsistent claim triple (subject, relation, object) with head and modifiers, and inconsistent context span to make their judgments.

By following a thorough and structured approach to annotating semantic fields, the dataset allows for a more comprehensive understanding of the inconsistencies present. This, in turn, contributes to the development of more accurate and effective models for detecting and explaining inconsistencies in natural language texts.

5.3.4 FICLE Dataset Statistics

The FICLE dataset is composed of 8,055 English samples, featuring five distinct inconsistency types. The distribution of these types includes: Taxonomic Relations (4,842), Negation (1,630), Set Based (642), Gradable (526), and Simple (415). There are six potential inconsistent claim components, with their respective distributions as follows: Target-Head (3,960), Target-Modifier (1,529), Relation-Head (951), Relation-Modifier (1,534), Source-Head (45), and Source-Modifier (36). The dataset incorporates 20 coarse inconsistent entity types, as illustrated in Fig. 5.2, which are further broken down into 60 fine-grained entity types. Table 5.4 displays the average sizes of various fields across the dataset's samples. The dataset is partitioned into train, validation, and test splits, maintaining an 80:10:10 ratio.

5.3.5 Quality Checks

We assessed the inter-annotator agreement on a subset of 500 samples from the dataset. The evaluation metrics used included Intersection over Union (IoU) and Kappa score. For source, relation, target, and inconsistent context spans, the IoU values were 0.91, 0.83, 0.85, and 0.76, respectively. Additionally, Kappa scores for inconsistency type, coarse inconsistent entity type, and fine-grained inconsistent entity type were 0.78, 0.71, and 0.67, respectively. These scores indicate a good level of agreement among annotators.

Alongside these agreement measures, several basic sanity checks were performed to ensure the quality and consistency of the annotations:

1. Ensuring that the S (Source), R (Relation), T (Target), and M (Mismatch) annotations were present exactly once in each annotated sample.

2. Verifying that the S, R, and T spans were marked on the claim sentence, while M was marked on the context sentence.

3. Investigating cases where a model that accurately predicted S, R, T, etc., struggled to make correct predictions.

4. Examining instances where the mismatch location was marked as Target-Head or Target-Modifier when no target was present in the sentence.

5. Performing random checks on the annotated datasets, and in cases where an annotator had a higher number of bad tagging instances compared to others, conducting more stringent random checks.

By ensuring the quality and accuracy of annotations in the dataset, we aim to provide a solid foundation for the development of natural language processing models that can effectively detect and explain inconsistencies in textual data. This thorough evaluation process contributes to the robustness and reliability of the dataset, ultimately resulting in more accurate and efficient models for inconsistency detection and explanation.

5.3.6 Dataset Fields

Claim (string): A statement or proposition relating to the consistency or inconsistency of certain facts or information.

Context (string): The surrounding information or background against which the claim is being evaluated or compared. It provides additional details or evidence that can support or challenge the claim.

Source (string): It is the linguistic chunk containing the entity lying to the left of the main verb/relating chunk.

Source Indices (string): Source indices refer to the specific indices or positions within the source string that indicate the location of the relevant information.

Relation (string): It is the linguistic chunk containing the verb/relation at the core of the identified inconsistency.

Relation Indices (string): Relation indices indicate the specific indices or positions within the relation string that highlight the location of the relevant information.

Target (string): It is the linguistic chunk containing the entity lying to the right of the main verb/relating chunk.

Target Indices (string): Target indices represent the specific indices or positions within the target string that indicate the location of the relevant information.

Inconsistent Claim Component (string): The inconsistent claim component refers to a specific linguistic chunk within the claim that is identified as inconsistent with the context. It helps identify which part of the claim triple is problematic in terms of its alignment with the surrounding information.

Inconsistent Context-Span (string): A span or portion marked within the context sentence that is found to be inconsistent with the claim. It highlights a discrepancy or contradiction between the information in the claim and the corresponding context.

Inconsistent Context-Span Indices (string): The specific indices or location within the context sentence that indicate the inconsistent span.

Inconsistency Type (string): The category or type of inconsistency identified in the claim and context.

Fine-grained Inconsistent Entity-Type (string): The specific detailed category or type of entity causing the inconsistency within the claim or context. It provides a more granular classification of the entity associated with the inconsistency.

Coarse Inconsistent Entity-Type (string): The broader or general category or type of entity causing the inconsistency within the claim or context. It provides a higher-level classification of the entity associated with the inconsistency.

All the data fields mentions are of string data type.

5.4 Our Approach

We utilize the FICLE dataset to train models that are capable of classifying factual inconsistencies while also providing explanations. The process is structured in such a way that, given a claim and a contextual sentence, our system carries out predictions in three phases:

- Anticipating the Inconsistent Claim Fact Triple (S,R,T) along with the Inconsistent Context Span
- Identifying the Inconsistency Type as well as the Inconsistent Claim Component
- Determining the Coarse and Fine-grained Inconsistent Entity Type

The entire system is composed of a sequence of four neural models, referred to as M1, M2, M3, and M4, which collectively aim to predict inconsistency type along with explanations. In this section, we delve into the intricacies of these three stages and the overall pipeline.

5.4.1 Model Architecture

We carry out our experiments using five pre-existing models, two of which are designed for natural language generation (NLG). More specifically, we fine-tune Transformer [93] encoder-based models

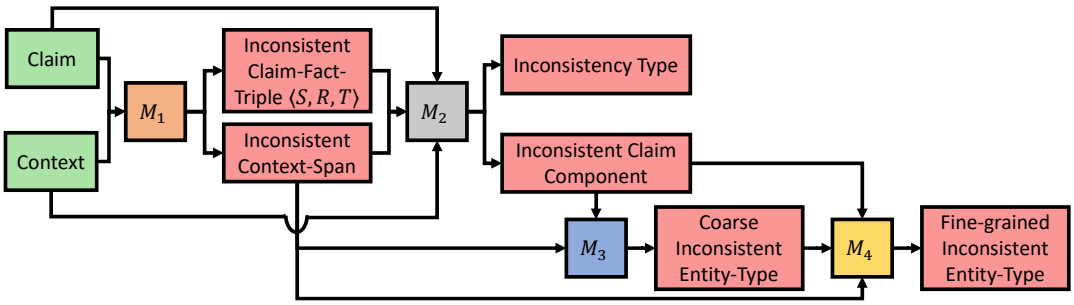

Figure 5.3: FICLE: System Architecture

such as BERT [24], RoBERTa [52] and DeBERTa [38]. We also employ two NLG models, BART [51] and T5 [77], both of which are widely recognized in the NLG field.

BERT (Bidirectional Encoder Representations from Transformers) [24] is essentially a transformer encoder with 12 layers, 12 attention heads, and 768 dimensions. The model we use has been pretrained on the Books Corpus and Wikipedia, utilizing the MLM (masked language model) and the next sentence prediction (NSP) loss functions. RoBERTa [52] is an improved pretraining methodology for natural language processing (NLP) systems, which builds upon BERT. It was trained with 160GB of text over a larger number of iterations, up to 500K, using batch sizes of 8K and a larger byte-pair encoding (BPE) vocabulary of 50K subword units, excluding NSP loss. DeBERTa [38] is trained using a distinct attention mechanism where content and position embeddings are separated. This model also boasts an enhanced mask decoder which effectively leverages absolute word positions. BART [51] is a denoising autoencoder used for pretraining sequence-to-sequence models. It's trained by corrupting text with an arbitrary noising function and learning to reconstruct the original text. T5 [77], on the other hand, is also a Transformer encoder-decoder model pretrained on Colossal Clean Crawled Corpus. It models all NLP tasks in generative form.

To encode inputs or outputs for these models, we prepend different semantic units using special tokens like claim, context, source, relation, target, contextSpan, claimComponent, type, coarseEntityType, and fineEntityType. NLG models (BART and T5) are tasked with generating the inconsistency type and all explanations, and are trained using cross-entropy loss. For NLU models (BERT, RoBERTa, DeBERTa), we prepend input with a [CLS] token and use its semantic representation from the final layer alongside a dense layer to predict inconsistency type, inconsistent claim component, and entity types with categorical cross-entropy loss. For the NLU models, the source, relation, target, and context span are predicted using start and end token classifiers (employing cross-entropy loss), as typically applied in the question-answering literature [24].

5.4.2 Predict Inconsistent Spans

In this stage, our primary focus is on training models to estimate the source, relation, and target using the claim sentence as input. Additionally, we try out four distinct approaches for predicting the inconsistent context span, detailed as follows:

1. Structure-ignorant: In this approach, the input is composed of the claim and context sentence. The goal is to directly estimate the inconsistent context span while overlooking the structure of the claim in terms of "source, relation, target".

2. Two-step: In the initial stage, the claim and context sentences are used as input to predict the source, relation, and target (SRT). In the subsequent stage, the input is enriched with the source, relation, and target, along with the claim and context, with the aim of predicting the inconsistent context span.

3. Multi-task: Here, the claim and context sentence form the input, and the objective is to concurrently predict the source, relation, target, and the inconsistent context span.

4. Oracle-structure: In this case, the input is the claim and context sentence as well as the confirmed truth (source, relation, and target). All of these are jointly used to estimate the inconsistent context span.

5.4.3 Predict Inconsistency Type and Claim Component

This stage operates under the assumption that (1) SRT from the claim and (2) inconsistent context span have already been predicted. Consequently, the input at this stage comprises the claim, context, predicted SRT, and predicted inconsistent context span. Utilizing these inputs, to forecast the inconsistency type and inconsistent claim component, we test three distinct methods, articulated as follows:

1. Individual: In this method, the inconsistency type and the inconsistent claim component are forecasted independently.

2. Dual-stage: The initial step of this process predicts the inconsistent claim component. Following this, the predicted inconsistent claim component is added to the input, with the second step focusing on predicting the inconsistency type.

3. Simultaneous-task: This approach employs a multi-task learning framework to concurrently predict both the inconsistency type and the inconsistent claim component.

5.4.4 Predict Inconsistent Entity Types

To identify inconsistent entity types, we construct a number of models. Each of these takes two primary inputs: the inconsistent context span and the corresponding span from the claim that relates to the inconsistent claim component. We explore the following distinct models:

1. Independent: Forecasts the coarse and fine-grained inconsistent entity types independently.

2. Sequential: The initial stage predicts the coarse inconsistent entity type. Following this, the input is supplemented with the predicted coarse inconsistent entity type, which is then used to forecast the fine-grained type.

Moreover, we strive to harness the semantics from the names of entity classes. Consequently, we employ the NLU models (BERT, RoBERTa, DeBERTa) to extract embeddings for entity class names and train NLU models to predict the class name that is most semantically similar to the representation (of the [CLS] token) of the input. These models are trained using cosine embedding loss. Particularly, with the help of class (or, entity type) embeddings, we train the following models. It's important to note that NLG models cannot be trained using class embeddings; therefore, this experiment is conducted exclusively with NLU models.

1. Single Embedding: We independently predict coarse and fine-grained inconsistent entity types using entity type embeddings.

2. Dual-phase Embedding: In the first phase, the coarse inconsistent entity type is predicted using class embeddings. The second phase enhances the input by adding the predicted coarse inconsistent entity type, and predicts the fine-grained type using class embeddings.

3. Hybrid Dual-phase: In the first phase, the coarse inconsistent entity type is predicted using class embeddings. The second phase, similar to the dual-phase embedding method, adds the predicted coarse inconsistent entity type to the input. However, it predicts the fine-grained type using a traditional multi-class classification approach without class embeddings.

Following various experimental iterations with model choices for the three stages discussed in this section, we discover that the configuration depicted in Fig. 5.3 yields optimal results. We have also experimented with other designs such as (1) predicting all outputs (inconsistency type and all explanations) simultaneously in a 6-task setting using only the claim and context as input, and (2) identifying claim component solely as S, R or T rather than distinguishing between heads and modifiers. Nevertheless, these alternate approaches did not result in improved outcomes.

5.5 Experiments and Results

In our effort to predict elements such as the source, relation, target, and inconsistent context span, we employ metrics such as exact match (EM) and intersection over union (IoU). The EM is a numerical value ranging from 0 to 1, which quantifies the degree of overlap between the predicted span and the true span based on tokens. An EM value of 1 indicates a perfect match between the model's prediction and the true span in terms of characters, while an EM value of 0 signifies no match. In a similar vein, the IoU metric calculates the intersection over the union, also in terms of tokens. For classification undertakings,

Model	Exact Match			IoU		
	Source	Relation	Target	Source	Relation	Target
BERT	0.919	0.840	**0.877**	0.934	0.876	**0.895**
RoBERTa	0.921	**0.865**	0.871	0.936	0.883	0.885
DeBERTa	0.918	0.857	0.864	0.932	0.874	0.893
BART	0.981	0.786	0.741	0.986	0.873	0.842
T5	**0.983**	0.816	0.765	**0.988**	**0.945**	0.894

Table 5.5: Source, Relation and Target Prediction from Claim Sentence

Model	Exact Match			IoU		
	Structure-ignorant	Two-step	Oracle-structure	Structure-ignorant	Two-step	Oracle-structure
BERT	0.483	0.499	0.519	0.561	0.541	0.589
RoBERTa	0.542	0.534	0.545	0.589	0.584	0.632
DeBERTa	0.538	0.540	**0.569**	0.591	0.587	**0.637**
BART	0.427	0.292	0.361	0.533	0.404	0.486
T5	0.396	0.301	0.352	0.517	0.416	0.499

Table 5.6: Inconsistent Context Span Prediction

such as predicting the type of inconsistency as well as the coarse and fine-grained inconsistent entity type, we make use of accuracy and weighted F1 as our performance metrics.

It should be noted that factual inconsistency classification is a pioneering task, and as such, there are no established baseline methods for comparison.

5.5.1 Source, Relation, Target and Inconsistent Context Span Prediction

Table 5.5 presents the outcomes for the prediction of the source, relation, and target derived from claim sentences. Evidently, the T5 model exhibits superior performance, except in relation to the prediction of relation and target utilizing the exact match metric. Additionally, Table 5.6 reveals a somewhat unexpected result, as the structure-ignorant method marginally outperforms the two-step method. As anticipated, the Oracle method paired with DeBERTa emerges as the most effective approach. When predicting the context span, the NLG models (BART and T5) exhibit a noticeable shortfall in performance in comparison to the NLU models. Finally, the results pertaining to the joint prediction of the source, relation, target, and inconsistent context span are disclosed in Table 5.7. While T5 and BART prove more adept at predicting the source, relation, and target, DeBERTa distinctly excels at predicting the inconsistent context span.

Model	Exact Match				IoU			
	Source	Relation	Target	Context Span	Source	Relation	Target	Context Span
BERT	0.769	0.665	0.752	0.524	0.801	0.708	0.804	0.566
RoBERTa	0.759	0.686	0.780	0.572	0.828	0.745	0.836	0.617
DeBERTa	0.788	0.704	0.819	**0.604**	0.843	0.768	0.844	**0.650**
BART	0.973	**0.816**	**0.836**	0.501	0.979	**0.874**	**0.895**	0.549
T5	**0.981**	0.764	0.717	0.570	**0.988**	0.870	0.842	0.602

Table 5.7: Joint Prediction of Source, Relation and Target Prediction from Claim Sentence and Inconsistent Context Span using Multi-Task Setting

Model	Accuracy			Weighted F1		
	Individual	Two-step	Multi-task	Individual	Two-step	Multi-task
BERT	0.84	0.84	0.84	0.86	0.86	0.86
RoBERTa	0.85	0.85	0.86	0.86	0.86	**0.87**
DeBERTa	0.86	0.85	**0.87**	0.86	**0.87**	**0.87**
BART	0.57	0.60	0.73	0.59	0.64	0.74
T5	0.53	0.61	0.74	0.58	0.66	0.74

Table 5.8: Inconsistency Type Prediction

Model	Accuracy		Weighted F1	
	Individual	Multi-task	Individual	Multi-task
BERT	0.83	0.88	0.83	0.88
RoBERTa	0.85	**0.89**	0.85	**0.89**
DeBERTa	0.88	**0.89**	**0.89**	**0.89**
BART	0.80	0.75	0.81	0.76
T5	0.81	0.75	0.81	0.75

Table 5.9: Inconsistent Claim Component Prediction (6-class classification)

5.5.2 Inconsistency Type and Inconsistent Claim Component Prediction

The outcomes for the inconsistency type and inconsistent claim component prediction are depicted in Tables 5.8 and 5.9, respectively. It should be highlighted that these two challenges represent 5-class and 6-class classifications correspondingly. An examination of the results indicates that the joint multi-task model exhibits superior performance when compared to the other two methodologies. Furthermore, DeBERTa consistently emerges as the most effective model across all configurations. Regarding this top-performing model, the F1 scores for the different types of inconsistency are enumerated as follows: Taxonomic Relations (0.92), Negation (0.86), Set Based (0.65), Gradable (0.78), and Simple (0.81).

5.5.3 Inconsistent Entity Type Prediction

The accuracy and weighted F1 scores for the prediction of coarse and fine-grained inconsistent entity types are detailed in Tables 5.10 and 5.11, respectively. Upon examining these tables, several observations can be made:

1. DeBERTa exhibits superior performance compared to the other models in predicting both coarse and fine-grained inconsistent entity types.

2. In terms of predicting coarse inconsistent entity types, an approach that employs embeddings proves to be more effective than the conventional classification approach. The underlying reason for this lies in the richness of semantics present within the entity class names, which the embedding-based approach skillfully utilizes.

3. When predicting fine-grained inconsistent entity types, the two-step method surpasses the individual method in its effectiveness, regardless of whether embeddings are employed.

4. The two-step mix method, which applies an embedding-based method for the prediction of coarse inconsistent entity types followed by the standard 60-class classification for fine-grained types, delivers the most optimal performance.

5.5.4 Qualitative Analysis

To gain further insight into the limitations of our model, we examine the confusion matrix for the inconsistency type prediction for our most successful model as shown in Table 5.12. It's notable that many instances that fall into the 'set-based' category are incorrectly labeled as 'taxonomic relations' by the model, leading to a diminished F1 score for the set-based class. In general, much of the confusion occurs between 'taxonomic relations' and other categories.

Model	Accuracy		Weighted F1	
	Individual	Individual Embedding	Individual	Individual Embedding
BERT	0.82	0.84	0.78	0.84
RoBERTa	0.83	0.86	0.80	0.85
DeBERTa	**0.85**	**0.87**	**0.81**	**0.86**
BART	0.73	-	0.71	-
T5	0.74	-	0.73	-

Table 5.10: Coarse Inconsistent Entity Type Prediction. Note that embedding based methods don't work with NLG models.

Model	Individual	Two-step	Individual Embedding	Two-step Embedding	Two-step Mix
BERT	0.65/0.59	0.74/0.71	0.64/0.62	0.72/0.70	0.75/0.71
RoBERTa	0.69/0.65	0.75/0.73	0.72/0.68	**0.76/0.73**	0.76/0.75
DeBERTa	**0.70/0.67**	**0.77/0.74**	**0.73/0.70**	**0.76/0.73**	**0.78/0.76**
BART	0.50/0.44	0.64/0.59	-	-	-
T5	0.56/0.48	0.67/0.62	-	-	-

Table 5.11: Accuracy/Weighted F1 for Fine-grained Inconsistent Entity Type Prediction. Note that embedding based methods do not work with NLG models.

		Predicted				
		Taxonomic Relations	Negation	Set Based	Gradable	Simple
Actual	Taxonomic Relations	**456**	16	4	17	9
	Negation	11	**123**	3	0	4
	Set Based	17	4	**22**	1	1
	Gradable	16	1	2	**51**	0
	Simple	6	2	2	2	**36**

Table 5.12: Confusion matrix for inconsistency type prediction. We observe a high correlation between actual and predicted values, indicating our model is effective.

In the context of coarse entity types, we observe the highest F1 scores for time, action, quantity, nationality, and geography entity types, while the lowest scores are found for animal, relationship, gender, sentiment, and technology entity types.

Moreover, when examining inconsistency spans within the context, we note that the average length of correct predictions (3.16) is significantly less than that of incorrect predictions (8.54), when compared to the lengths of ground truth spans. In the case of incorrect predictions, we note a trend where the coverage of ground truth tokens by the predicted tokens typically decreases as the length of the inconsistency span grows. Furthermore, we categorized incorrect span predictions into four groups: additive, reordered, changed, and subtractive. 'Additive' refers to predictions having more terms than the ground truth, 'reordered' signifies predictions having the same terms but in a different order, 'changed' indicates the model generated some new terms, and 'subtractive' represents predictions missing some terms compared to the ground truth. We discovered that approximately 91% of the errors were of the subtractive type, suggesting that our inconsistency span predictor model tends to be too concise and its performance could be improved by reducing the sampling probability for the end of sequence token.

5.5.5 Experimental Settings

The experimental procedures were carried out on a computational platform equipped with four GEFORCE RTX 2080 Ti graphical processing units (GPUs). For training all the models, we employed

a batch size of 16 and utilized the AdamW optimizer [54], conducting the training process for a total of 5 epochs. The models implemented in these experiments included bert-base-uncased, roberta-base, microsoft/deberta-base, facebook/bart-base, and t5-small. The learning rate was configured to be 1e-4 specifically for BART and T5 models, while a learning rate of 1e-5 was set for the rest of the models. These configurations were chosen after careful consideration to balance the trade-off between computational efficiency and model performance.

5.6 Conclusion and Future Work

In this study, we delved into the issue of identifying and elucidating various forms of factual inconsistencies in text. Our contributions include the creation of a novel dataset, FICLE, which encompasses approximately 8,000 samples, each marked with meticulous inconsistency labels for associated (claim, context) pairs. We undertook numerous experiments employing diverse natural language understanding and generation models to address the issue at hand. Our findings suggest that the most effective strategy incorporates a sequence of four models. This sequence starts with predicting inconsistency spans in both the claim and the context, followed by determining the type of inconsistency, and concluding with predicting the type of inconsistent entity. Furthermore, we observed that DeBERTa yielded the most favorable results. Looking forward, we intend to broaden the scope of this work by considering multilingual contexts. Additionally, we aim to enhance this work by introducing the capability to detect and localize inconsistencies across several sentences within a given paragraph. This could help us gain a more holistic understanding of the inconsistencies within broader text units.

Chapter 6

Conclusions and Future Work

Despite extensive research in Natural Language Inference (NLI) and misinformation detection, the specific challenge of detecting factual inconsistencies in real text remains under-explored. Traditional approaches often concentrate on identifying contradictions or false information within a given context or with a knowledge base, yet they lack the depth to unravel subtle factual inaccuracies embedded in authentic texts. This gap highlights the need for a nuanced approach that not only discerns these inaccuracies but also provides comprehensive explanations. Our work delves into this relatively uncharted territory, proposing a novel framework that leverages linguistic theories and advanced natural language processing techniques to identify and explain factual inconsistencies. Our work introduces Factual Inconsistency Classification with Explanations (FICLE), a novel approach that combines linguistic theories and advanced natural language processing techniques. FICLE is designed to identify and articulate factual inconsistencies, providing a groundbreaking contribution to the realm of factual accuracy analysis in real-world text narratives.

Chapter 1 introduces the thesis, highlighting the essential shift from manual fact-checking to advanced algorithmic analysis in response to the growing complexity and volume of data in the digital age. It underscores the necessity of not only detecting factual inconsistencies but also comprehensively understanding and explaining their contexts and causes across various sectors, including journalism, healthcare, and finance. The chapter lays out the motivation and framework for the research, touching upon key topics such as hostility detection, Fact Extraction and Verification (FEVER), and Factual Inconsistency Classification with Explanations (FICLE). Concluding with a thesis outline, this introductory chapter establishes the foundation for the ensuing detailed exploration into analysing factual inconsistencies, setting the direction for the entire research work.

Chapter 2 serves as a comprehensive survey of the current landscape in detecting and classifying factual inaccuracies within digital information. It methodically navigates through key areas such as Natural Language Inference, automated fact-checking systems, Explainable NLP, and the detection of 'fake news.' The chapter offers insights into the definitions, scopes, and methodologies of these domains, highlighting the evolution of techniques and tools that have become crucial in this field. By examining a range of approaches—from the intricate mechanisms of NLP to the integration of fact-checking with

fake news detection—this chapter provides a holistic view of the efforts and advancements made in maintaining the integrity of information in the digital age.

Chapter 3 addresses the critical issue of hostility detection in Hindi tweets, an increasingly pertinent topic given the proliferation of harmful content on social media. Focusing on identifying tweets that fall into categories such as hateful, offensive, defamatory, or fake, the chapter introduces an advanced system utilizing IndicBERT, a Transformer-based model trained on Indian language texts. This model is adept at processing online social media-style text alongside clean textual information. The chapter outlines the successful implementation of Task Adaptive Pretraining (TAPT) to enhance the performance of this Transformer encoder before its integration into the classification architecture. The results are noteworthy, showing improvements of 1.35% and 1.40% in binary hostility detection and 4.06% and 1.05% in fine-grained classifications into the four hostile categories, based on macro and weighted F1 metrics, respectively. The system demonstrated its superiority by achieving first place in the 'Hostile Post Detection in Hindi' shared task, with an F1 score of 97.16% for coarse-grained detection and a weighted F1 score of 62.96% for fine-grained classification.

In **Chapter 4**, the focus shifts to addressing the pervasive issue of misinformation, commonly known as "Fake News," in the digital landscape. This chapter introduces the Fact Extraction and Verification (FEVER) initiative, a groundbreaking approach to counteract the spread of false information by verifying textual claims against a vast repository of facts, primarily from Wikipedia. The chapter explores the complexities of implementing FEVER, including the creation of a retrieval model specifically designed for the FEVER dataset, strategies for sentence selection, and advancements in Recognizing Textual Entailment (RTE) using both baseline and state-of-the-art transformer models. A key highlight is the development of a novel InferTransformer model, blending Infersent architecture with transformer representations, showcasing its efficiency in stance detection and fact verification. Through a detailed analysis of various neural architectures and feature sets, the chapter provides valuable insights into optimizing automated fact-checking systems to effectively combat the challenges posed by misinformation in today's digital era.

We pivot towards the complex challenge of identifying factual inconsistencies in text in **Chapter 5**, specifically in the context of Transformer-based natural language generation models. This chapter introduces the novel FICLE dataset, meticulously annotated with various types of inconsistencies in around 8,000 (claim, context) pairs, serving as a foundational resource for addressing this challenge. A sequence of four models is proposed and tested using Transformer-based NLU and NLG architectures, including BERT, RoBERTa, DeBERTa, T5, and BART. The study reveals that accurately predicting inconsistency spans and types, and identifying inconsistent entities, is a multifaceted task, with DeBERTa emerging as the most effective model. The findings underscore the complexity of factual inconsistency detection and the potential of the proposed pipeline, which achieves significant F1 scores and Intersection over Union (IoU) rates in various sub-tasks.

Future Work

Multiple possible future works are possible in the work presented in this thesis like exploring with large language models, multimodal setting etc. Below points discuss these points in details.

- **Expansion to Multilingual and Cross-Lingual Contexts:** Adapting the FICLE model for multilingual and cross-lingual contexts involves handling diverse linguistic structures and idioms. The goal is to develop a robust system capable of detecting factual inconsistencies across different languages, enhancing the model's applicability globally. This expansion can significantly benefit international journalism, legal proceedings, and academic research by providing a more inclusive and versatile tool for factual verification.

- **Exploring Multimodal Settings:** Integrating FICLE with multimodal data, such as images and videos, aims to create a comprehensive system for inconsistency detection in multimedia content. This adaptation will allow the model to correlate textual information with visual or auditory cues, providing a more holistic understanding of content validity. Such a multimodal approach could revolutionize fact-checking in digital media, social networks, and educational content.

- **Evaluation with Large Language Models:** Leveraging large language models like GPT-4 for evaluating and enhancing FICLE's capabilities aims to harness their advanced understanding of complex narratives. This approach will test the model's efficiency in handling nuanced and intricate factual data. The potential benefits include improved accuracy in inconsistency detection, making the model more reliable for critical applications in journalism, academia, and beyond.

- **Incorporating Broader Contextual Understanding:** Enhancing FICLE to analyze broader contexts, such as entire documents, aims to provide a deeper understanding of the narrative or argument structure. This development will allow the model to detect inconsistencies not just in isolated statements but in the context of the larger discourse. Such an enhancement could be particularly beneficial in research and legal analysis, where the context often holds the key to understanding the material.

- **Implementation in Different Domains:** Tailoring FICLE for specific domains like journalism, healthcare, and the legal sector involves adapting the model to recognize domain-specific nuances and terminologies. This specialization aims to make the model a valuable tool for professionals in these fields, ensuring the accuracy and reliability of information in critical areas like public health, legal proceedings, and news reporting.

Related Publications

- **Tathagata Raha**, Mukund Choudhary, Abhinav Menon, Harshit Gupta, KV Aditya Srivatsa, Manish Gupta, Vasudeva Varma. **Neural Models for Factual Inconsistency Classification with Explanations.** In Proceedings of the ECML-PKDD 2023.
- **Tathagata Raha**, Sayar Ghosh Roy, Ujwal Narayan, Zubair Abid, Vasudeva Varma. **Task adaptive pretraining of transformers for hostility detection.** In Proceedings of CONSTRAINT 2021, Collocated with AAAI 2021.

Bibliography

[1] About us, snopes.com.

[2] Our mission, factcheck.org, 06 2009.

[3] What is full fact? - fullfact.org, 2019.

[4] Fact-checking and accountability journalism project archives, 12 2020.

[5] T. Alhindi, S. Petridis, and S. Muresan. Where is Your Evidence: Improving Fact-checking by Justification Modeling. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 85–90, Brussels, Belgium, 2018. Association for Computational Linguistics.

[6] H. Allcott and M. Gentzkow. Social Media and Fake News in the 2016 Election. *Journal of Economic Perspectives*, 31(2):211–236, May 2017.

[7] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein. Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, 2020.

[8] P. Badjatiya, M. Gupta, and V. Varma. Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. In *The World Wide Web Conference*, WWW '19, page 49–59, New York, NY, USA, 2019. Association for Computing Machinery.

[9] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, page 759–760, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[10] D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate, May 2016. arXiv:1409.0473 [cs, stat].

[11] R. Baly, M. Mohtarami, J. Glass, L. Màrquez, A. Moschitti, and P. Nakov. Integrating Stance Detection and Fact Checking in a Unified Corpus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 21–27, New Orleans, Louisiana, 2018. Association for Computational Linguistics.

[12] M. Bhardwaj, M. S. Akhtar, A. Ekbal, A. Das, and T. Chakraborty. Hostility detection dataset in hindi, 2020.

[13] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword info. *arXiv preprint arXiv:1607.04606*, 2016.

[14] C. L. Borgman. *Big data, little data, no data : scholarship in the networked world*. The Mit Press, 2016.

[15] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, 2015. Association for Computational Linguistics.

[16] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[17] O.-M. Camburu, T. Rocktäschel, T. Lukasiewicz, and P. Blunsom. e-SNLI: Natural Language Inference with Natural Language Explanations, Dec. 2018. arXiv:1812.01193 [cs].

[18] Z. Cao, F. Wei, W. Li, and S. Li. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[19] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini. Computational Fact Checking from Knowledge Networks. *PLOS ONE*, 10(6):e0128193, June 2015.

[20] A. Conneau, R. Rinott, G. Lample, A. Williams, S. Bowman, H. Schwenk, and V. Stoyanov. XNLI: Evaluating Cross-lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium, 2018. Association for Computational Linguistics.

[21] N. K. Conroy, V. L. Rubin, and Y. Chen. Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4, Jan. 2015.

[22] I. Dagan, O. Glickman, and B. Magnini. The PASCAL Recognising Textual Entailment Challenge. In J. Quiñonero-Candela, I. Dagan, B. Magnini, and F. d'Alché Buc, editors, *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, volume 3944, pages 177–190. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[23] M.-C. De Marneffe, A. N. Rafferty, and C. D. Manning. Finding contradictions in text. In *Proceedings of ACL-08: HLT*, pages 1039–1047, 2008.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[25] F. Doshi-Velez and B. Kim. Towards A Rigorous Science of Interpretable Machine Learning, Mar. 2017. arXiv:1702.08608 [cs, stat].

[26] O. Dušek and Z. Kasner. Evaluating semantic accuracy of data-to-text generation with natural language inference. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 131–137, 2020.

[27] B. Eisner, T. Rocktäschel, I. Augenstein, M. Bosnjak, and S. Riedel. emoji2vec: Learning emoji representations from their description. *CoRR*, abs/1609.08359, 2016.

[28] 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*, pages 1163–1168, San Diego, California, 2016. Association for Computational Linguistics.

[29] S. Ghosh Roy, U. Narayan, T. Raha, Z. Abid, and V. Varma. Leveraging multilingual transformers for hate speech detection. In *Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation*. CEUR, 2021.

[30] D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing - RTE '07*, page 1, Prague, Czech Republic, 2007. Association for Computational Linguistics.

[31] M. Glockner, V. Shwartz, and Y. Goldberg. Breaking NLI Systems with Sentences that Require Simple Lexical Inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia, 2018. Association for Computational Linguistics.

[32] B. Goodman and S. Flaxman. European Union Regulations on Algorithmic Decision Making and a "Right to Explanation". *AI Magazine*, 38(3):50–57, Sept. 2017.

[33] L. Graves, B. Nyhan, and J. Reifler. Understanding Innovations in Journalistic Practice: A Field Experiment Examining Motivations for Fact-Checking. *Journal of Communication*, 66(1):102–138, Feb. 2016.

[34] J. Guo, Y. Fan, Q. Ai, and W. B. Croft. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 55–64, Indianapolis Indiana USA, Oct. 2016. ACM.

[35] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

[36] N. Hassan, F. Arslan, C. Li, and M. Tremayne. Toward Automated Fact-Checking: Detecting Check-worthy Factual Claims by ClaimBuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1803–1812, Halifax NS Canada, Aug. 2017. ACM.

[37] N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A. K. Nayak, V. Sable, C. Li, and M. Tremayne. ClaimBuster: the first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment*, 10(12):1945–1948, Aug. 2017.

[38] P. He, X. Liu, J. Gao, and W. Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

[39] O. Honovich, L. Choshen, R. Aharoni, E. Neeman, I. Szpektor, and O. Abend. q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. *arXiv preprint arXiv:2104.08202*, 2021.

[40] S. Jain and B. C. Wallace. Attention is not Explanation, May 2019. arXiv:1902.10186 [cs].

[41] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Bang, A. Madotto, and P. Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, page To appear, Nov 2022.

[42] P. Joshi, S. Aditya, A. Sathe, and M. Choudhury. Taxinli: Taking a ride up the nlu hill. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 41–55, 2020.

[43] D. Kakwani, A. Kunchukuttan, S. Golla, G. N.C., A. Bhattacharyya, M. M. Khapra, and P. Kumar. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*, 2020.

[44] D. Khashabi, S. Chaturvedi, M. Roth, S. Upadhyay, and D. Roth. Looking Beyond the Surface: A Challenge Set for Reading Comprehension over Multiple Sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana, 2018. Association for Computational Linguistics.

[45] T. Khot, A. Sabharwal, and P. Clark. SciTaiL: A Textual Entailment Dataset from Science Question Answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[46] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

[47] W. Kryściński, N. S. Keskar, B. McCann, C. Xiong, and R. Socher. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, 2019.

[48] R. Kumar, A. K. Ojha, M. Zampieri, and S. Malmasi, editors. *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Santa Fe, New Mexico, USA, Aug. 2018. Association for Computational Linguistics.

[49] S. Kumar and P. Talukdar. Nile: Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742, 2020.

[50] D. M. J. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, M. Schudson, S. A. Sloman, C. R. Sunstein, E. A. Thorson, D. J. Watts, and J. L. Zittrain. The science of fake news. *Science*, 359(6380):1094–1096, Mar. 2018.

[51] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020.

[52] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[53] S. Longpre, K. Perisetla, A. Chen, N. Ramesh, C. DuBois, and S. Singh. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, 2021.

[54] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[55] S. Lundberg and S.-I. Lee. A Unified Approach to Interpreting Model Predictions, Nov. 2017. arXiv:1705.07874 [cs, stat].

[56] J. Luther, Mark Stencel. Annual census finds nearly 300 fact-checking projects around the world, 06 2020.

[57] B. MacCartney. *Natural language inference*. Stanford University, 2009.

[58] T. Mandl, S. Modha, G. K. Shahi, A. K. Jaiswal, D. Nandini, D. Patel, P. Majumder, and J. Schäfer. Overview of the HASOC track at FIRE 2020: Hate Speech and Offensive Content Identification in Indo-European Languages). In *Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation*. CEUR, December 2020.

[59] Y. Mao, X. Ren, H. Ji, and J. Han. Constrained abstractive summarization: Preserving factual consistency with constrained generation. *arXiv preprint arXiv:2010.12723*, 2020.

[60] B. Mathew, R. Dutt, P. Goyal, and A. Mukherjee. Spread of hate speech in online social media. *Proceedings of the 10th ACM Conference on Web Science*, pages 173–182, 06 2019.

[61] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, 2020.

[62] R. T. McCoy, E. Pavlick, and T. Linzen. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. 2019.

[63] M. Mesgar, E. Simpson, and I. Gurevych. Improving factual consistency between a response and persona facts. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 549–562, 2021.

[64] T. Mitra and E. Gilbert. CREDBANK: A Large-Scale Social Media Corpus With Associated Credibility Annotations. *Proceedings of the International AAAI Conference on Web and Social Media*, 9(1):258–267, Aug. 2021.

[65] B. D. Mittelstadt, P. Allo, M. Taddeo, S. Wachter, and L. Floridi. The ethics of algorithms: Mapping the debate. *Big Data & Society*, 3(2):205395171667967, Dec. 2016.

[66] P. Nakov, D. Corney, M. Hasanain, F. Alam, T. Elsayed, A. Barrón-Cedeño, P. Papotti, S. Shaar, and G. Da San Martino. Automated Fact-Checking for Assisting Human Fact-Checkers. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 4551–4558, Montreal, Canada, Aug. 2021. International Joint Conferences on Artificial Intelligence Organization.

[67] F. Nan, R. Nallapati, Z. Wang, C. dos Santos, H. Zhu, D. Zhang, K. McKeown, and B. Xiang. Entity-level factual consistency of abstractive text summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2727–2733, 2021.

[68] Y. Nie, A. Williams, E. Dinan, M. Bansal, J. Weston, and D. Kiela. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, 2020.

[69] V. Novotný, M. Štefánik, E. F. Ayetiran, P. Sojka, and R. Řehůřek. When FastText Pays Attention: Efficient Estimation of Word Representations using Constrained Positional Weighting. *JUCS - Journal of Universal Computer Science*, 28(2):181–201, Feb. 2022.

[70] C. O'neil. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Penguin Books, 09 2016.

[71] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[72] P. Patwa, M. Bhardwaj, V. Guptha, G. Kumari, S. Sharma, S. PYKL, A. Das, A. Ekbal, M. S. Akhtar, and T. Chakraborty. Overview of constraint 2021 shared tasks: Detecting english covid-19 fake news and hindi hostile posts. In *Proceedings of the First Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation (CONSTRAINT)*. Springer, 2021.

[73] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[74] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[75] N. Pinnaparaju, V. Indurthi, and V. Varma. Identifying fake news spreaders in social media. In *Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (eds.) CLEF 2020 Labs and Workshops, Notebook Papers. CEUR-WS.org*, sep 2020.

[76] 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*, pages 22–32, Brussels, Belgium, 2018. Association for Computational Linguistics.

[77] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[78] N. F. Rajani, B. McCann, C. Xiong, and R. Socher. Explain Yourself! Leveraging Language Models for Commonsense Reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy, 2019. Association for Computational Linguistics.

[79] N. Reimers and I. Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3980–3990, Hong Kong, China, 2019. Association for Computational Linguistics.

[80] M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[81] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier, Aug. 2016. arXiv:1602.04938 [cs, stat].

[82] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[83] 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.

[84] J. Saeed. *Semantics*. Introducing Linguistics. Wiley, 2011.

[85] B. Shi and T. Weninger. Discriminative predicate path mining for fact checking in knowledge graphs. *Knowledge-based systems*, 104:123–133, 2016.

[86] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu. FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media. *Big Data*, 8(3):171–188, June 2020.

[87] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu. Fake News Detection on Social Media: A Data Mining Perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36, Sept. 2017.

[88] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[89] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, 2018.

[90] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. Generating token-level explanations for natural language inference. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 963–969, 2019.

[91] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, and A. Mittal. The Fact Extraction and VERification (FEVER) Shared Task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium, 2018. Association for Computational Linguistics.

[92] E. J. Topol. *Deep Medicine : How Artificial Intelligence Can Make Healthcare Human Again*. Basic Books, March, 03 2019.

[93] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[94] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Red Hook, NY, USA, Dec. 2017. Curran Associates Inc.

[95] N. Vedula and S. Parthasarathy. Face-keg: Fact checking explained using knowledge graphs. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 526–534, 2021.

[96] S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[97] A. Wang, K. Cho, and M. Lewis. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, 2020.

[98] 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)*, pages 422–426, Vancouver, Canada, 2017. Association for Computational Linguistics.

[99] A. Williams, N. Nangia, and S. Bowman. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, 2018. Association for Computational Linguistics.

[100] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi. Defending against neural fake news. *Advances in neural information processing systems*, 32, 2019.

[101] S. Zhang, J. Niu, and C. Wei. Fine-grained factual consistency assessment for abstractive summarization models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 107–116, 2021.

[102] X. Zhang, J. Zhao, and Y. LeCun. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[103] C. Zhou, G. Neubig, J. Gu, M. Diab, F. Guzmán, L. Zettlemoyer, and M. Ghazvininejad. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, 2021.

[104] C. Zhu, W. Hinthorn, R. Xu, Q. Zeng, M. Zeng, X. Huang, and M. Jiang. Enhancing factual consistency of abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 718–733, 2021.