A Comprehensive Study on Claim Detection Utilizing Multi-task Learning

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

Master of Engineering
in
Computer Science and Engineering

Submitted by

Shankha Shubhra Das
Regn. No. - 163554 of 2022 - 2023
Exam Roll No. - M4CSE24001
Class Roll No. - 002210502002
Session - 2022 - 2024

Under the supervision of

Dr. Dipankar Das

Department of Computer Science and Engineering
Jadavpur University
188, Raja S.C. Mallick Rd,
Kolkata - 700032, West Bengal, India

Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jadavpur University, Kolkata - 700032

CERTIFICATE OF RECOMMENDATION

This is to certify that the work embodied in this thesis entitled **"A Comprehensive Study on Claim Detection Utilizing Multi-task Learning"** has been satisfactorily completed by **Shankha Shubhra Das** (Registration Number 163554 of 2022 - 2023; Class Roll No. 002210502002; Examination Roll No. M4CSE24001). It is a bonafide piece of work carried out under my supervision and guidance at Jadavpur University, Kolkata, for partial fulfillment of the requirements for the awarding of the **Master of Engineering** in **Computer Science and Engineering** degree of the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jadavpur University, during the academic year 2022 - 2024.

Dr. Dipankar Das,
Department of Computer Science and Engineering,
Jadavpur University.
(Supervisor)

Forwarded By:

Prof. Nandini Mukhopadhyay,
HOD, Department of Computer Science and Engineering,
Jadavpur University.

Prof. Dipak Laha,
Dean, Faculty of Engineering and Technology,
Jadavpur University.

Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jadavpur University, Kolkata - 700032

CERTIFICATE OF APPROVAL

This is to certify that the thesis entitled **"A Comprehensive Study on Claim Detection Utilizing Multi-task Learning"** is a bonafide record of work carried out by **Shankha Shubhra Das** (Registration Number 163554 of 2022 - 2023; Class Roll No. 002210502002; Examination Roll No. M4CSE24001) in partial fulfillment of the requirements for the award of the degree of **Master of Engineering** in **Computer Science and Engineering** in the Department of Computer Science and Engineering, during the period of September 2022 to May 2024. It is understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the thesis only for the purpose of which it has been submitted.

_____
Signature of Examiner 1
Date:

_____
Signature of Examiner 2
Date:

Department of Computer Science and Engineering
Faculty of Engineering and Technology
Jadavpur University, Kolkata - 700032

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that the thesis entitled **"A Comprehensive Study on Claim Detection Utilizing Multi-task Learning"** contains literature survey and original research work by the undersigned candidate, as a part of his degree of **Master of Engineering** in **Computer Science and Engineering, Jadavpur University**. All the information have been obtained and presented in accordance with academic rules and ethical conduct.
I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Shankha Shubhra Das

Examination Roll No.: M4CSE24001

Registration No.: 163554 of 2022 - 2023

Thesis Title: A Comprehensive Study on Claim Detection Utilizing Multi-task Learning

Signature of the Candidate:

. . . . . .

Dedicated to my unwavering pillars of support,
my parents,
whose love and sacrifice have fueled every chapter of my journey.

. . . . . .

Acknowledgements

I am delighted to express my gratitude and regards towards my project guide **Dr. Dipankar Das**, Asst. Professor, Department of Computer Science and Engineering, Jadavpur University, without whose valuable guidance, inspiration, and attention pursuing my project would have been impossible. His expertise and guidance were crucial in shaping both the direction and the quality of this thesis, and I am profoundly thankful for his steadfast dedication.

I express my sincere gratitude to the Head of the Department, **Dr. Nandini Mukhopadhyay**, Professor, Department of Computer Science and Engineering, Jadavpur University, Kolkata, India, for being kind enough to permit me to carry out my research in this prestigious institution.

I am grateful to all the members of **NLP Lab and Magnum Lab**, Department of Computer Science and Engineering, Jadavpur University, for cooperating with me in all possible ways and providing me with a good working environment throughout the work.

I would like to thank my friends Mr. Pritam Pal, and Ms. Shreyashi Karjee, my seniors Mr. Avik Mondal, and Mr. Atanu Mondal, and my juniors Mr. Sohan Choudhury, Mr. Priyam Saha, and Mr. Subharthi Ray for constantly supporting me and helping me in any way possible.

I thank my classmates, Mr. Akash Halder, Ms. Debarati Paul, Mr. Abhijit Sarkar, Ms. Tirtha Das, Mr. Saikat Mazumdar, and Mr. Rajarshi Mitra for their unwavering support, validation, and understanding throughout this journey. Their companionship and camaraderie have been invaluable to me.

I express my regards to my parents for bearing with me, helping me, and being a source of constant motivation during the entire term of the work.

I am so grateful to my family Late Rabindra Nath Das, Late Sobharani Das, Late Aurobinda Mitra, Late Lina Mitra, Mrinal Chandra Das, and Sudarshana Das.

Shankha Shubhra Das

Examination Roll No.: M4CSE24001

Registration No.: 163554 of 2022 - 2023

Department of Computer Science and Engineering,

Jadavpur University.

Abstract

This work delves into the dynamic domain of Natural Language Processing (NLP), which has experienced remarkable strides owing to the advent of machine learning and deep learning methodologies. Within this expansive field, the identification and assessment of claims within textual data have emerged as pivotal tasks, necessitating automated solutions to gauge their validity and relevance. Our exploration extends beyond the confines of English to encompass a spectrum of languages, with a keen focus on Indian linguistic diversity, thereby broadening the applicability of NLP techniques across diverse cultural and linguistic contexts. Central to our investigation is the paradigm of Multi-task Learning (MTL), which offers the promise of concurrent task execution to enhance overall model performance. While MTL has traditionally been applied to similar tasks, our research pioneers its application to dissimilar tasks, thereby pushing the boundaries of its utility and efficacy. To this end, we employ innovative fusion methodologies and a long-head approach in task-specific layer strategies, helping our MTL models with enhanced adaptability and accuracy. Furthermore, our study delves into the intricate domain of Span Identification, particularly honing in on Sentiment and Claim Span Identification tasks. Leveraging transfer learning techniques, we develop robust models capable of effectively extracting specific spans of text containing pertinent information across multiple languages. By imbuing our models with multilingual support, we strive to ensure their versatility and applicability in diverse linguistic contexts, particularly within the rich variety of the Indian linguistic landscape. In synthesizing these methodologies and approaches, our research contributes to the advancement of NLP applications, fostering greater understanding and automation of textual data analysis, especially within the intricate and diverse linguistic fabric of India.

Contents

Abstract i

Table of Contents ii

List of Figures vii

List of Tables ix

List of Abbreviations xi

1 Introduction 1
- 1.1 Natural Language Processing . . . 1
- 1.2 Claim Detection . . . 2
- 1.3 Multi-task Learning . . . 3
- 1.4 Span Identification . . . 5
- 1.5 Research Challenges . . . 6
- 1.6 Motivation . . . 8
- 1.7 Objective . . . 8
- 1.8 Contribution . . . 9
- 1.9 Thesis Outline . . . 10

2 Literature Survey 11
- 2.1 Claim Detection . . . 11
  - 2.1.1 Claim Extraction . . . 12
  - 2.1.2 Misinformation Detection . . . 12
  - 2.1.3 Claim-Paper Clustering . . . 12
  - 2.1.4 Evidence Mining . . . 13
  - 2.1.5 Context-Dependent Claim Detection . . . 13
  - 2.1.6 Context Independent Claim Detection . . . 14
  - 2.1.7 Claim Detection in Scientific Domains . . . 15
- 2.2 Multi-task Learning . . . 15
  - 2.2.1 Multi-Task Representation Learning . . . 16
  - 2.2.2 Single-Sentence Classification . . . 16
  - 2.2.3 Saliency Detection . . . 17
  - 2.2.4 Pairwise Text Classification . . . 17
  - 2.2.5 Relevance Ranking . . . 18
- 2.3 Span Identification . . . 18
  - 2.3.1 Modeling Peer Relation for Span Identification Tasks . . . 19

2.3.2 Span-based Named Entity Recognition . . . 19
2.3.3 Aspect-based Sentiment Span Detection . . . 20
2.3.4 Automatic Identification of Claim Spans on Twitter . . . 21

3 Dataset Preparation **23**
3.1 Claim Detection . . . 23
3.1.1 Binary Claim Detection . . . 23
3.1.2 Multi-class Claim Detection . . . 24
3.1.3 Multi-label Claim Detection . . . 24
3.2 Multi-task Learning . . . 25
3.2.1 Claim Detection . . . 25
3.2.2 Sentiment Analysis . . . 26
3.2.3 Language Identification . . . 26
3.3 Span Identification . . . 27
3.3.1 Claim Span Identification . . . 27
3.3.2 Sentiment Span Identification . . . 27

4 Claim Detection **29**
4.1 Binary Classification . . . 29
4.1.1 Problem Statement . . . 29
4.1.2 Proposed Approach . . . 30
4.1.3 Implementation . . . 30
4.1.3.1 Architecture . . . 30
4.1.3.2 Compilation . . . 32
4.1.3.3 Training . . . 32
4.1.3.4 Prediction and Probability Calculation . . . 33
4.1.4 Experimental Results . . . 33
4.1.4.1 Experimental Setup . . . 33
4.1.4.2 Result . . . 33
4.1.5 Error Analysis . . . 34
4.2 Multi-class Classification . . . 35
4.2.1 Problem Statement . . . 35
4.2.2 Proposed Approach . . . 35
4.2.3 Implementation . . . 36
4.2.3.1 Architecture . . . 36
4.2.3.2 Compilation . . . 38
4.2.3.3 Training . . . 38
4.2.3.4 Prediction and Probability Calculation . . . 38
4.2.4 Experimental Results . . . 39
4.2.4.1 Experimental Setup . . . 39
4.2.4.2 Result . . . 39
4.2.5 Error Analysis . . . 39
4.3 Multi-label Classification . . . 40
4.3.1 Problem Statement . . . 40
4.3.2 Proposed Approach . . . 42
4.3.3 Implementation . . . 42
4.3.3.1 Pre-processing . . . 42
4.3.3.2 Methodology . . . 43

4.3.4 Experimental Results . . . 44
4.3.5 Observations . . . 45
4.4 Discussion . . . 45

5 Multi-task Learning **47**
5.1 Task Descriptions . . . 48
5.1.1 Claim Detection . . . 48
5.1.2 Sentiment Analysis . . . 48
5.1.3 Language Identification . . . 49
5.2 Multi-task Environment . . . 50
5.2.1 Two Tasks Combinations . . . 50
5.2.2 Three Tasks Combination . . . 50
5.3 Proposed Approach . . . 50
5.3.1 Text Preprocessing . . . 51
5.3.2 Tokenization . . . 51
5.3.3 Classification Task . . . 51
5.3.4 Modeling . . . 52
5.4 Implementation . . . 52
5.4.1 Word Embedding . . . 52
5.4.2 BiLSTM Layer . . . 53
5.4.3 GlobalMaxPooling Layer . . . 53
5.4.4 MTL with Task-Specific Long Heads . . . 53
5.4.5 Fusion with Task-Specific Dense Layer . . . 54
5.4.6 Classification . . . 54
5.4.7 Training . . . 55
5.5 Experimental Results . . . 56
5.5.1 Experimental Setup . . . 56
5.5.2 Task Comparison . . . 56
5.6 Error Analysis . . . 58
5.7 Discussion . . . 59

6 Span Identification **61**
6.1 Claim Span Identification . . . 61
6.1.1 Problem Statement . . . 62
6.1.2 Proposed Approach . . . 62
6.1.3 Implementation . . . 62
6.1.3.1 Tokenization . . . 62
6.1.3.2 B-I-O Tagging . . . 63
6.1.3.3 Model Architecture . . . 63
6.1.4 Experimental Results . . . 63
6.1.5 Error Analysis . . . 64
6.1.6 Discussion . . . 64
6.2 Sentiment Span Identification . . . 65
6.2.1 Problem Statement . . . 65
6.2.2 Proposed Approach . . . 66
6.2.3 Implementation . . . 66
6.2.3.1 Data Preprocessing . . . 66
6.2.3.2 Tokenization . . . 67

6.2.3.3 Input Preparation 67
6.2.3.4 Creating and Training the Model 67
6.2.3.5 Fitting the Model 68
6.2.3.6 Prediction using the Model 69
6.2.4 Experimental Results and Error Analysis 69
6.2.5 Discussion 70

7 Conclusion **71**
7.1 Future Work 72

List of Publications **75**

Bibliography **77**

List of Figures

2.1 An example demonstrating the claim detection task and its subtasks. 12
2.2 The standard three components of the framework for automated fact-checking 14
2.3 High-level design of the CDCD approach 14
2.4 Constituency trees for two sentences containing claims. Boxed nodes are common elements between the two trees. 15
2.5 Comparison of different MTL methods 16
2.6 Illustrative examples of the relation between saliency and task similarity. Left: Two tasks are to detect whether the man is smiling and his mouth is open. Right: Two tasks are to detect the horse and the person. The salient regions are close to each other, indicating the potential similarity between the tasks. 17
2.7 Architecture of the MT-DNN model for representation learning. The lower layers are shared across all tasks while the top layers are task-specific. 18
2.8 (a) Illustrations of Subordinate (SUB) and Peer (PR) relations in SpanID tasks. (b) The constructions of augmented data with PR relations in the MRC paradigm. We use NER here for demonstration purposes. 20
2.9 Span-based NER is jointly trained with one supervised VIB component: information compression, and two unsupervised VAE components: span reconstruction and synonym generation. The encoder is shared across the components. Synonyms are collected from an external knowledge base (KB). 21
2.10 Examples illustrating spans for aspect-based sentiment analysis in Vietnamese. 21
2.11 Examples of claim tweets and their ground truth claim spans highlighted in boldface text (blue). 22

3.1 Data Distribution of Binary Claim Detection Train Data 24
3.2 Data Distribution of Binary Claim Detection Test Data 24
3.3 Data Distribution of Multi-class Claim Classification English Data . 24
3.4 Data Distribution of Multi-class Claim Classification Hindi Data . . 24
3.5 Data Distribution of Multi-label Claim Classification Data 26
3.6 Data Distribution of Sentiment Analysis Train Data 26
3.7 Data Distribution of Sentiment Analysis Test Data 26
3.8 Data Distribution of Language Identification Train Data 27

3.9 Data Distribution of Language Identification Test Data 27
3.10 Data Distribution of Claim Span Identification Data 28
3.11 Data Distribution of Sentiment Span Identification Train Data . . . 28
3.12 Data Distribution of Sentiment Span Identification Test Data . . . 28

4.1 The Graphical Representation of the Claim Detection Model 31
4.2 The Confusion Matrix of Binary Claim Detection Task 34
4.3 The Graphical Representation of the Claim Classification Model . . 37
4.4 The Confusion Matrix of Multi-class Claim Classification Task . . . 39
4.5 The Graphical Representation of the Model 44

5.1 Proposed Model Architecture for Dissimilar Tasks: (a) MTL with shared backbone and long task-specific heads, (b) MTL with shared backbone, long task-specific head, and fusion. 52
5.2 Confusion matrices for claim detection for 1-TL, 3-TL and 3-TL$_{fusion}$ 57
5.3 Confusion matrices for sentiment classification for 1-TL, 3-TL and 3-TL$_{fusion}$ 58
5.4 Confusion matrices for language identification for 1-TL, 3-TL and 3-TL$_{fusion}$ 58

6.1 Architecture of Claim Span Identification Model 64
6.2 Architecture of Sentiment Span Identification Model 68

List of Tables

4.1 Parameters used to train the binary claim detection model 32
4.2 Performance Metrics of Binary Claim Detection Task 33
4.3 Error Analysis of Binary Claim Detection Task 34
4.4 Parameters used to train the multi-class claim classification model . 38
4.5 Performance Metrics of Multi-class Claim Classification Task 39
4.6 Error Analysis of Claim Classification Task 40
4.7 Results of three trials 45

5.1 Parameters used to train the model. 56
5.2 Performance comparison of 1-TL vs 3-TL vs 3-TL$_{fusion}$. 57
5.3 Performances of all combinations of MTLs. 58
5.4 Error analysis between 1-TL, 3-TL and 3-TL$_{fusion}$ 59

6.1 Results on Test Split of the Dataset 64
6.2 Error Analysis of Claim Span Identification Task 65
6.3 Parameters used to train the model. 69
6.4 Exact match score of Sentiment Span Identification Model 69
6.5 Error Analysis of Sentiment Span Identification Task 70

List of Abbreviations

A

ABSA	**A**spect-**B**ased **S**entiment **A**nalysis
AI	**A**rtificial **I**ntelligence
AM	**A**rgument **M**ining

B

BERT	**B**idirectional **E**ncoder **R**epresentations from **T**ransformers
BiLSTM	**Bi**directional **L**ong **S**hort-**T**erm **M**emory
BSF	**B**order **S**ecurity **F**orce

C

CCE	**C**ontract **C**lause **E**xtraction
CDCD	**C**ontext-**D**ependent **C**laim **D**etection
CDCs	**C**ontext-**D**ependent **C**laim**s**
COVID	**CO**rona**VI**rus **D**isease
CURT	**C**laim **U**nit **R**ecognition in **T**weets

D

DL	**D**eep **L**earning
DRDO	**D**efence **R**esearch and **D**evelopment **O**rganisation

F

FN	**F**alse **N**egative
FP	**F**alse **P**ositive

G

GPS	**G**lobal **P**ositioning **S**ystem

I

IB	**I**nformation **B**ottleneck

J

JUCCD	**J**adavpur **U**niversity **C**laim **C**lassification **D**RDO
JUCSI	**J**adavpur **U**niversity **C**laim **S**pan **I**dentification

L

LSTM	**L**ong **S**hort-**T**erm **M**emory

M

ML	**M**achine **L**earning
MNLI	**M**ulti **N**atural **L**anguage **I**nference
MRC	**M**achine **R**eading **C**omprehension
MRPC	**M**icrosoft **R**esearch **P**araphrase **C**orpus
MT-DNN	**M**ulti-**T**ask **D**eep **N**eural **N**etworks
MTL	**M**ulti **T**ask **L**earning
MuRIL	**Mu**ltilingual **R**epresentations for **I**ndian **L**anguages

N

NER	**N**amed **E**ntity **R**ecognition
NLG	**N**atural **L**anguage **G**eneration
NLI	**N**atural **L**anguage **I**nference
NLP	**N**atural **L**anguage **P**rocessing

O

OSM	**O**nline **S**ocial **M**edia

Q

QNLI	**Q**uestion-answering **N**atural **L**anguage **I**nferemce
QQP	**Q**uora **Q**uestion **P**airs

R

RNN	**R**ecurrent **N**eural **N**etwork
RoBERTa	**R**obustly **O**ptimized **BERT a**pproach

RTE	**R**ecognizing **T**extual **E**ntailment
S	
SBPD	**S**pan **B**ased **P**ropaganda **D**etection
SVM	**S**upport **V**ector **M**achine
T	
T5	**T**ext-**t**o-**T**ext **T**ransfer **T**ransformer
TE	**T**extual **E**ntailment
TN	**T**rue **N**egative
TP	**T**rue **P**ositive
V	
VAE	**V**ariational **A**uto-**E**ncoder
VIB	**V**ariational **I**nformation **B**ottleneck
X	
XLM	**Cross**-Lingual **L**anguage **M**odel

CHAPTER 1

Introduction

Natural Language Processing (NLP) has witnessed remarkable advancements in recent years, driven by breakthroughs in machine learning and deep learning techniques. These advancements have led to significant improvements in various NLP tasks such as text generation, sentiment analysis, and machine translation. One of the notable pros of these advancements is the ability to handle more complex linguistic structures and nuances, enabling more accurate and contextually relevant results. Additionally, NLP models have become more scalable and efficient, allowing for faster processing of large volumes of text data.

In the realm of NLP tasks, Claim Detection has emerged as a crucial area, aiming to automatically identify and assess claims made in the text for their validity or relevance. Multi-task Learning has gained traction as a promising approach, where models are trained to perform multiple NLP tasks simultaneously, leveraging shared knowledge and improving overall performance. Span Identification, another important task, focuses on identifying and extracting specific spans of text that contain relevant information, contributing to tasks such as named entity recognition and information extraction. These advancements signify the continuous evolution of NLP towards more sophisticated and versatile applications in various domains.

1.1 Natural Language Processing

NLP[1], a subfield of computer science within the larger field of artificial intelligence (AI), aims to enable computers to interpret and process human language, both spoken and written, in the same way that humans do. It blends computational linguistics, which uses rules to represent human language, with statistical, machine learning, and deep learning approaches. These technologies enable com-

puters to comprehend the meaning, intentions, and emotions conveyed by text or voice, making activities like translation, speech recognition, and summarization possible even in real-time. NLP enables a variety of consumer applications such as voice-activated GPS, digital assistants, and chatbots, while its use in corporate solutions is growing to optimize company operations and increase productivity.

However, it is difficult to create algorithms that correctly comprehend text or audio data since human language is inherently ambiguous. Applications that use natural language must be able to identify and comprehend subtleties like idioms, sarcasm, homophones, metaphors, grammatical exceptions, and sentence structure changesthings that people pick up on intuitively over time but that computers need to be expressly taught.

Many natural language processing (NLP) jobs entail decomposing human textual and auditory input to enable machine understanding, such as:

- **Part of Speech Tagging[2]:** Part of speech tagging, or grammatical tagging, involves identifying the role of a word within a sentence by analyzing its context and usage. For instance, the word "make" can function as both a verb and a noun, as seen in the sentences "I can make a paper airplane" and "What make of car do you own?"

- **Named-Entity Recognition[3]:** Named Entity Recognition (NER) detects words or phrases in the text that refer to specific entities, such as locations, people's names, organizations, and so on. For example, NER can recognize "Kentucky" as a location or "Fred" as a name.

- **Text Classification[4]:** Text Classification is analyzing textual material to detect subtle components such as attitudes, emotions, sarcasm, doubt, and mistrust.

- **Natural Language Generation:** NLG, often known as speech-to-text or reverse voice recognition, is the process of translating structured text into human languages.

1.2 Claim Detection

Claim Detection is the act of recognizing and evaluating claims or assertions in a text that can be true or false and are commonly used to persuade or convince the audience of a specific perspective or argument. This problem is important in a variety of sectors, including information retrieval and natural language processing, as well as politics, science, and health.

The significance of textual claim identification is demonstrated by its function in:

- **Fact-Checking:** Organisations that do fact-checking must guarantee the veracity of assertions made by persons or businesses.
- **Information Retrieval:** Enables the retrieval of important information from vast text databases, making it easier for consumers to look for certain claims or information.
- **Argument Analysis:** Necessary for properly understanding the form and strength of arguments.

However, claim identification presents obstacles such as ambiguity, context dependence, and heterogeneity in format, structure, and language. Identifying claims is becoming increasingly difficult because many online statements are in the CodeMix language, meaning they include multiple languages within a single sentence. In essence, Claim Detection is an important procedure for extracting and analyzing claims from textual material, with applications ranging from fact-checking to information retrieval and argument analysis. It entails using a variety of techniques, including rule-based and machine-learning approaches, but also carefully considering the inherent problems and ethical implications.

1.3 Multi-task Learning

Multi-task learning, also called MTL, has emerged as a potent and promising focus within the domain of machine learning. This powerful paradigm empowers machine learning models to simultaneously tackle multiple tasks, leveraging shared representations to optimize efficiency and learning speed. Deep learning, while transformative in many aspects, has traditionally been hampered by large-scale data requirements and extensive computational demands. MTL offers a breakthrough in addressing these limitations, unlocking new possibilities for efficient and effective learning across diverse tasks.

- **Background of Multi-task Learning:** At its core, MTL mirrors the essence of human learning, where knowledge integration across domains plays a pivotal role in human intelligence. Just as humans acquire general motor skills, like balance and intuitive physics, that serve as the foundation for tackling complex tasks later in life, MTL enables models to rapidly learn from minimal examples and leverage learned concepts for efficient learning across diverse settings. By promoting positive transfer between tasks and

facilitating concept-based learning, MTL aims to develop systems that emulate the cognitive capabilities of human intelligence.

- **Types of Multi-task Learning:** Traditionally, MTL methods have been categorized into hard parameter sharing and soft parameter sharing. The former entails sharing model weights between tasks, whereas the latter allows for task-specific models with parameters designed to encourage similarity between tasks. While this dichotomy captures important aspects of MTL, the breadth and diversity of MTL techniques extend far beyond this classification.

- **MTLs Capability in Multilingualism:** MTL can be used to handle Multilingualism which refers to a high degree of proficiency in two or more languages. It enables individuals to communicate in written and oral modes effectively. It often leads to language mixing, also known as code-mixing, where multilingual speakers switch between multiple languages within a single utterance of text or speech. This is a unique issue that can be taken care of with the use of MTL.

- **Scope of Multi-task Learning:** As researchers delve deeper into the realm of multi-task learning, they interpret its scope as a learning setting with a fixed set of tasks treated equally. This interpretation excludes scenarios with a single "main task" and auxiliary tasks, as well as those with dynamically changing task sets over time. By clarifying the scope of MTL, researchers aim to create a more comprehensive understanding of its application and impact in various domains.

In Multi-task learning, we can concurrently experiment with various tasks like Language Identification, Named-Entity Recognition, Sentiment Analysis, etc. In NLP, Language Identification or language guessing is the problem of determining which natural language given content is in. Named-entity recognition is a subtask of Information Extraction that seeks to locate and classify named entities mentioned in unstructured text into pre-defined categories such as person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc. Sentiment Analysis is the use of NLP, Text Analysis, Computational Linguistics, and Biometrics to systematically identify, extract, quantify, and study affective states and subjective information.

Multi-task learning holds immense promise in verifying scientific claims and enriching knowledge exploration in the data-rich digital age. By advancing MTL methods and seamlessly integrating NLP techniques, researchers can effectively

combat the spread of misinformation and enhance the standard of informed decision-making across various fields. As MTL continues to evolve and diversify, researchers must embrace the challenges and complexities to unlock its full potential, ushering in a new era of transformative scientific discoveries and advancements.

1.4 Span Identification

Span Identification tasks are essential components of practical NLP, spanning a variety of studies inside a single text. These tasks include recognizing and categorizing sequences of tokens. They range in complexity from lemmatization and sentiment analysis to connection extraction. Deep learning advancements have resulted in a wide range of model architectures for span detection, including token classification models, recurrent neural networks, and transformers like BERT.

However, there is no systematic understanding of how various span identification tasks compare. Questions emerge about the efficacy of alternative model designs across tasks and whether certain task features might influence the selection of appropriate models. Answering these concerns might make architectural selection easier and allow for comparisons of existing methodologies and new improvements.

Span Types, such as quote detection and named entity identification, have a variety of features. While there is little research on assessing their similarities and variances, four criteria can capture essential characteristics:

- **Frequency:** The number of spans for a span type in the training corpus; larger frequencies often improve model performance. Architectural options such as transfer learning may reduce the requirement for large training data.

- **Span Length:** It is the geometric mean of span lengths in tokens. Traditional models based on rigid assumptions may struggle with larger spans, but models such as LSTMs or transformers may outperform owing to their ability to describe long-distance relationships.

- **Span Distinctiveness:** Determines how diverse the text inside spans is from the total text of the corpus, with higher distinctiveness suggesting variable word use within spans.

- **Boundary Distinctiveness:** Determines how different the start and end points of spans are from the remainder of the text. High boundary uniqueness indicates recognized span borders, whilst low distinctiveness denotes smoother transitions.

In this work, we concentrated on two major types of Span Identification tasks that are frequently used in many industries: Claim Span Identification and Sentiment Span Identification.

Claim Span Identification includes extracting sentences that contain one or more claims. The key goal is to identify the exact location of the assertion inside the phrase. For example, in the context of combatting fake news, this activity is critical for quickly and reliably recognizing portions of sentences containing erroneous assertions. On the contrary, **Sentiment Span Identification** is likewise concerned with sentence extraction. The system processes a document conveying favorable or negative thoughts about a certain topic. The idea is to detect spans or segments inside phrases that express any type of sentiment.

Both Claim Span Identification and Sentiment Span Identification play critical roles in a variety of applications. For example, in the field of news verification and disinformation detection, Claim Span Identification helps to indicate possibly erroneous or misleading assertions, allowing fact-checking organizations to assess material quickly. Similarly, Sentiment Span Identification is extremely useful in sentiment analysis applications, allowing companies to measure public opinion, customer happiness, and brand sentiment by extracting feelings stated in customer reviews, social media postings, or product feedback.

Furthermore, advances in natural language processing (NLP) and machine learning approaches have resulted in the creation of sophisticated algorithms capable of correctly distinguishing claim spans and sentiment expressions in text data. These developments have considerably increased the efficiency and accuracy of tasks such as false news identification, sentiment analysis, and information extraction, allowing for better-informed decision-making across several domains.

1.5 Research Challenges

In this section, we will briefly outline the challenges encountered during our research. These are presented in a point-wise format below:

- Numerous scholarly studies have investigated Claim Detection and its practical consequences in everyday situations. But, a significant void remains in the evaluation of its relevance in the Indian context. **Example:** Research has explored how claim detection can be used to fact-check news articles, highlighting its importance in combating misinformation. Despite global research efforts, there is a lack of studies examining how claim detection applies to Indian media and discourse.

- Existing research mostly focuses on binary determinations of claim presence

within textual constructions. Little emphasis has been paid to explicit categorization by claim type. **Example:** Many studies only identify whether a statement contains a claim or not, without further analysis. Research seldom differentiates between types of claims, such as factual claims, opinions, or predictions.

- Our thesis seeks to discover assertions and categorize them, overcoming the binary dichotomy. We used a dataset relevant to different opinions regarding vaccination in India, reflecting the unique features of the Indian narrative. **Example:** Instead of merely detecting claims, our study classifies them into categories like medical claims, political claims, etc. Our dataset includes social media posts, news articles, and public statements about vaccines in multiple Indian languages.

- There is a considerable corpus of research on multi-task learning, mostly focused on homogenous or similar task sets. There is very little in-depth research on the dynamics of Multi-task Learning in the context of diverse or dissimilar task designs. **Example:** Studies often look at multi-task learning within related tasks like anger detection and sentiment analysis, which share similarities. Few studies examine how multi-task learning performs when combining tasks like claim detection, sentiment analysis, and language identification.

- Nothing is known about the influence of certain model architectures on multi-task frameworks' overall performance metrics. Our research aims to address this imbalance by investigating various task paradigms and their effects on learning dynamics. We want to understand the usefulness of model designs in tolerating such task variation. **Example:** We study how different combinations of tasks, such as claim detection and sentiment analysis, affect learning outcomes. Our research tests various neural network architectures to determine which ones handle diverse tasks most effectively.

- Existing research on span identification has mostly focused on basic approaches and pre-processing techniques, frequently limited to monolingual English. Our attention moves to multilingual or codemix textual corpora, attempting to find and extract spans from documents among different language modalities. **Example:** Techniques for identifying specific text segments, like names or dates, are usually developed for English-only datasets. We develop methods to identify spans in documents that switch between languages, such as Hindi-English code-mixed texts.

1.6 Motivation

The need for researching Claim Detection in the Indian context arises from an understanding of its ability to address urgent socioeconomic concerns unique to the region. While Claim Detection has received a lot of attention across the world, its applications in India are still untapped. Understanding and categorizing statements in the Indian narrative can give insights into current societal discourses, notably on healthcare, politics, and socioeconomic challenges. By diving into this subject, we want to contribute to a more nuanced knowledge of India's linguistic environment, allowing for more informed decision-making and discourse analysis.

The objective behind Multi-task Learning is to improve the flexibility and efficiency of machine learning models across a wide range of task domains. While previous research has been focused on homogenous task sets, real-world applications frequently demand models to address different tasks at the same time. By investigating the dynamics of Multi-task Learning with diverse tasks, we hope to gain new insights into model architecture design and optimization methodologies. Furthermore, knowing the relationship between task variety and model performance might help to design more resilient and adaptable AI systems capable of solving a wider range of real-world situations. Moreover, multi-task models are appropriate for sustainability and resource efficiency, even if they are only 80% accurate compared to single-task models.

The motivation for Span Identification research is the need for more effective and inclusive natural language processing approaches. While significant progress has been achieved in detecting individual textual components, such as names or phrases, these gains have primarily been limited to monolingual English environments. However, in today's globalized world, multilingual and codemix textual data are more common, necessitating approaches that can accommodate language variety. We want to overcome this gap by concentrating on multilingual span identification and developing approaches that can efficiently parse and extract information from many linguistic modalities, resulting in more inclusive and complete natural language processing capabilities.

1.7 Objective

In this thesis, we intend to look into the many types of opinionated statements that exist in the Indian environment, identifying them based on their distinguishing characteristics. While most research in India focuses on binary categories in Claim Detection, we strive to build a more complex categorization.

In our study on Multi-task Learning, we hope to delve into the complexities

of model adaptation within a broad framework. While conventional wisdom holds that Multi-task Learning thrives on the alignment of similar activities, we want to pioneer approaches that enable the concurrent training of dissimilar tasks, resulting in impressive results.

Within the domain of Span Identification, our goal is to experiment with several pre-trained models and fine-tune them for best performance in the Indian environment. Furthermore, we want to incorporate a multilingual dimension into our research, resulting in a robust model capable of handling the linguistic variety seen across India.

1.8 Contribution

In this segment, we will discuss our contributions in brief.

- **Beyond Binary Classification of Claims:**
  - We moved past simple binary classification to identify complex traits within each claim.
  - Claims were organized into various categories reflecting their unique qualities.
- **Innovative Approaches in Multi-task Learning:**
  - Aimed to improve performance metrics for handling dissimilar tasks.
  - Employed methods such as the fusing of internal layers.
  - Utilized long-head and short-head tactics to enhance learning effectiveness.
  - Detailed examination of these methods will be provided in a later section.
- **Expanded Scope in Span Identification:**
  - Traditionally focused on the English language domain.
  - Expanded to include multilingualism in our experiments.
  - Trained models using India's rich linguistic diversity.
  - Ensured models are adaptable across multiple languages, enhancing the span identification methodology's applicability.

1.9 Thesis Outline

This thesis work has been divided into seven chapters.

- The chapter 1 has the introductions of the topics we are going to discuss.
- In chapter 2, a comprehensive literature survey is presented. This survey delves deep into the recent progress in Claim Detection, Multi-task Learning, and Span Identification.
- The chapter 3 discusses how the raw dataset has been collected and prepared to be fed to different models. We have also provided the necessary statistics for a better understanding of the datasets.
- Next, chapter 4 provides various kinds of classification experiments related to claim detection.
- In chapter 5, we have discussed how multi-task learning models work and how we can improve their performances by using different unique mechanisms.
- Subsequently, chapter 6 delves deep into claim and sentiment span identification techniques and various pre-processing mechanisms accompanying it.
- Finally, chapter 7 discusses the conclusion and future work that can be done additionally in this domain.

Through this research, the aim is to contribute to the advancement of natural language processing, ultimately facilitating more accurate and efficient techniques used in claim detection, multi-task learning, and span identification.

CHAPTER 2

Literature Survey

Recent advancements in natural language processing (NLP) have witnessed significant progress in Claim Detection, Multi-task Learning (MTL), and Span Identification. In Claim Detection, transformer-based models like BERT have excelled, leveraging large-scale datasets and contextual understanding to accurately identify claims in textual content. MTL has emerged as a powerful paradigm, with frameworks such as T5 enabling simultaneous optimization of multiple tasks, leading to improved efficiency and generalization. Span Identification has seen notable strides with the adoption of neural architectures and structured prediction techniques, enhancing the extraction of specific spans in text. These advancements underscore the growing capabilities of NLP systems in understanding and extracting valuable information from textual data, with the potential for further breakthroughs as research in these areas continues to evolve. A few recent publications related to claim detection are highlighted in section 2.1, multi-task learning in section 2.2, and span identification in section 2.3.

2.1 Claim Detection

Claim detection encompasses various methodologies, and it can be performed with or without context, depending on the specific research objectives. Additionally, apart from claim detection, several related tasks, such as Misinformation Detection, Evidence Mining, Scientific NLP tasks, Claim Extraction, and Claim Clustering, can be tackled using similar techniques. To address the diverse range of tasks effectively, we will explore the potential of multi-task learning, aiming to incorporate and leverage the shared knowledge across these related tasks to improve overall performance and efficiency in information verification and natural language processing applications. A basic functioning of the claim detection [5] task is shown in Figure 2.1.

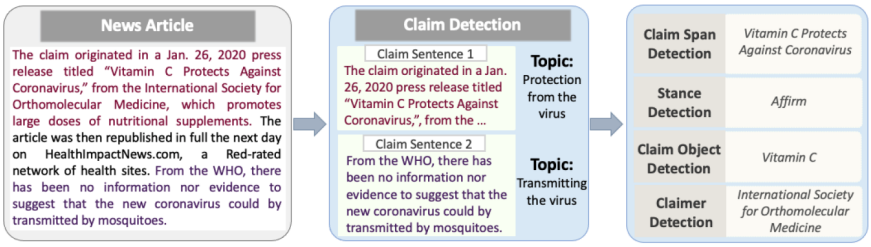


Figure 2.1: An example demonstrating the claim detection task and its subtasks.

2.1.1 Claim Extraction

Pavllo et al. [6] and Smeros et al. [7] develop rule-based heuristics for extracting quotes from general and scientific news articles using weakly supervised models. Levy et al. [8] and Stab et al. [9] propose ML models for claim detection and argument mining, providing publicly available datasets for training extraction models. Hassan et al. [10] and Popat et al. [11] employ claim classification models with fact-checking portals for verifying political claims. Zlatkova et al. [12] focus on claim extraction for images, while Karagiannis et al. [13] present a framework for statistical claims verification. This approach [14], unlike others, is specifically tailored for claims, utilizing advanced language models with and without contextualized embeddings fine-tuned with domain-specific knowledge and capable of processing various input sources like social media postings, blog posts, or news articles.

2.1.2 Misinformation Detection

Fact-checking plays a pivotal role in combatting the dissemination of misinformation, making it an integral component of various Natural Language Processing (NLP) tasks focused on misinformation detection [15]. Within this domain, related tasks encompass fake news detection [16], propaganda detection [17], rumor detection [18], and stance detection [19]. While most of these tasks primarily address misinformation related to politics and society, there has been a notable increase in scientific and health-related misinformation detection, particularly concerning content related to the COVID-19 pandemic [20] [21] [22]).

2.1.3 Claim-Paper Clustering

Few approaches address the challenge of clustering multimodal information, which includes both textual representations of claims and papers and the interconnectedness between them via a bipartite graph. To achieve this, we can draw inspiration from several multimodal clustering approaches proposed by Yao et

al.[23], who suggest a unified convolutional network for terms and documents, and Zhou et al.[24], who utilize weighted graphs encoding attribute similarity. Hamilton et al.[25] introduce a methodology for jointly training embeddings based on text and graph information, while Reimers et al.[26] apply numerical clustering to these embeddings. Additionally, Wang et al.[27] propose a technique for training network embeddings that preserve graph communities, and Duong et al.[28] present interpretable embeddings. Few researchers approaches are the first to tackle the clustering of heterogeneous passages in terms of length and vocabulary type while accounting for their interconnectedness through a bipartite graph.

2.1.4 Evidence Mining

Automated fact-checking, as shown in Figure 2.2, relies on detecting claims in the open domain, which has been approached by Yuan and Yu [29] through a rule-based system for identifying health claims in news headlines, and by Wührl and Klinger [30] with a BERT-based model to detect biomedical claims in social media posts. After claim detection, determining check-worthiness is a critical next step, addressed in the shared task CLEF-CheckThat! [31] [32] for scientific claims related to health from news articles. Another research area involves automatic evidence gathering for scientific claims. Some focus has been on humanities and social sciences [33], while the majority of work pertains to life sciences. Various tools, such as PubTator [34], Textpresso [35], LitSense [36], and EvidenceMiner [37], have been developed for searching PubMed, the largest biomedical publications database. These methods typically employ query analysis to detect named entities, keywords, or metadata patterns to retrieve relevant results from the database, assisting scientists in gathering evidence for their research. In fact-checking, evidence retrieval is only one part of the entire process[15].

2.1.5 Context-Dependent Claim Detection

Assume a Topic and a small set of relevant free-text articles [38] [39] and aim to automatically pinpoint context-dependent claims (CDCs) within these documents. The detected CDCs should be well-phrased and relevant to the given Topic, making them useful in discussions. This task is a part of the argumentation mining field but has distinctive features, requiring context-specific identification of claims and exact claim boundaries [40] [41]. Unlike other tasks, CDCD deals with free-text articles from various domains, adding complexity to claim identification [42]. While related to Argumentative Zoning [43] [42]), CDCD extends beyond sentence classification, aiming to identify relevant claims for a given Topic, not just any claim mentioned in an article. It also differs from Textual Entailment

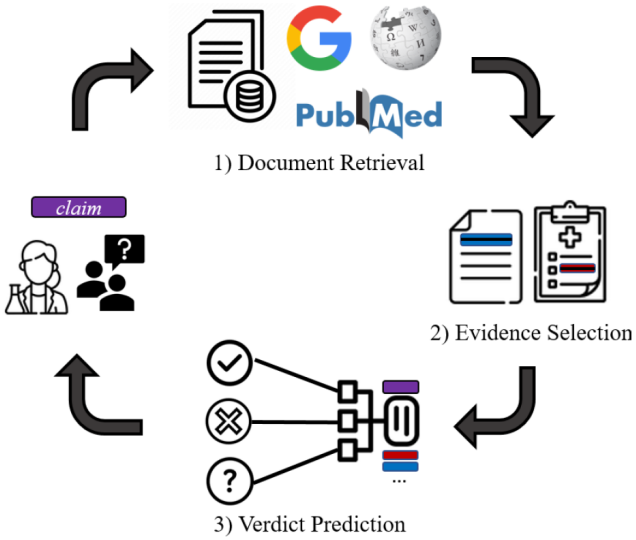


Figure 2.2: The standard three components of the framework for automated fact-checking

(TE) [44]), which focuses on factual assertions, while CDCD deals with persuasive opinions that may not have strict truth values[45]. The CDCs and the Topic may correspond to different sentence types, including imperative and exclamatory sentences. Figure 2.3 shows an abstract workflow of CDCD.

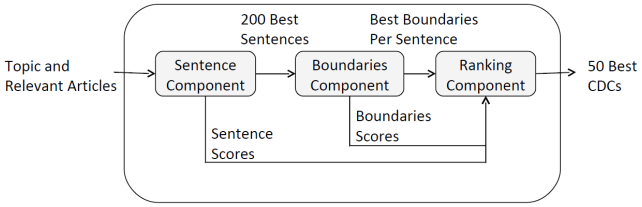


Figure 2.3: High-level design of the CDCD approach

2.1.6 Context Independent Claim Detection

Argumentation mining has garnered increasing attention as it aims to automatically identify structured argument data from unstructured natural language text[46]. Most approaches primarily focus on specific domains, despite argumentative sentences often exhibiting common rhetorical structures across domains. The proposed methodology, however, utilizes structured parsing information for claim detection, achieving comparable performance to state-of-the-art methods without relying heavily on context. Figure 2.4 shows the claim detection pipeline involves splitting the given text document into sentences using a tokenizer, parsing each sentence to obtain a constituency tree, discarding sentences without a verb (VP tag), substituting words at leaves with their stemmed versions for improved gen-

eralization, and finally, employing an SVM to classify each sentence as potentially containing a claim or not.

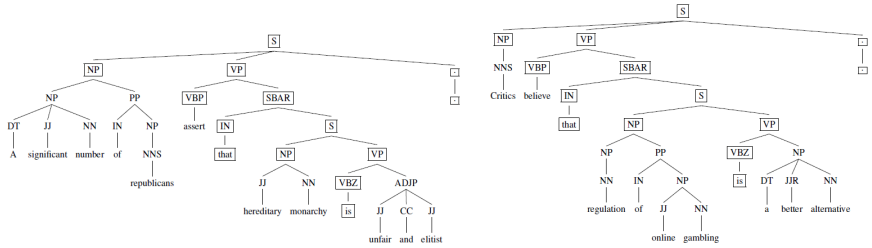


Figure 2.4: Constituency trees for two sentences containing claims. Boxed nodes are common elements between the two trees.

2.1.7 Claim Detection in Scientific Domains

Scientific fact-checking is a part of NLP tasks that involve understanding scientific text, which presents challenges due to complex language and specialized terminology. Large language models pre-trained on general data struggle with NLP tasks in the scientific domain, emphasizing the importance of domain adaptation. Natural Language Inference (NLI) or Recognizing Textual Entailment (RTE) is crucial in automated fact-checking, as it involves inferring if a premise supports or contradicts a hypothesis, serving as a basis for verifying claims. For the scientific domain, datasets like MedNLI [47], SciNLI [48], and NLI4CT [49] provide relevant medical and computational linguistics claims and evidence. Additionally, question-answering tasks, such as open-domain question answering, share similarities with fact-checking, aiming to find answers in unstructured text [50]. Biomedical QA datasets like BioASQ [51] and PubMedQA [52], along with BLURB [53], are important benchmarks. Furthermore, automated evidence synthesis automates systematic reviews for clinical trials [54].

2.2 Multi-task Learning

Multi-task Learning (MTL) and its practical applications have gained increasing attention in recent years. Despite being a relatively new concept, the domain offers a variety of methodologies. MTL enables us to address multiple related tasks effectively, such as Saliency Detection, Single Sentence Classification, Detailed Image Classification, Pairwise Text Classification, and Relevance Learning. Leveraging MTL in claim detection and related tasks opens up promising possibilities for enhancing accuracy and efficiency in these areas. By delving deeper into this field, we can explore the potential of MTL to revolutionize the way

we handle various NLP tasks, including fact-checking and misinformation detection, contributing to more reliable information dissemination in today's data-rich landscape. Figure 2.5 shows a short comparison of different multi-task learning methods[55].

	Transparency in task relation	Feature generation	Conciseness and efficiency	Flexibility
Shallow MTL	✓	✗	✓	✗
Hard-parameter sharing	✗	✓	✓	✗
Soft-parameter sharing	✗	✓	✗	✓
SRDML	✓	✓	✓	✓

Figure 2.5: Comparison of different MTL methods

2.2.1 Multi-Task Representation Learning

Despite the notable achievements of multi-task learning, particularly in representation learning[56], only a limited number of theoretical studies have been conducted [57] [58] [59]. Existing statistical learning bounds mostly apply to linear multitask learning scenarios [60] [61] [62]. Learning-to-learn, also known as inductive bias learning or transfer learning, was proposed by Thrun and Pratt (1998)[63] and investigated theoretically by Baxter (2000)[58]. Baxter's work includes an error analysis, demonstrating that a shared representation performing well on training tasks will also generalize to new tasks from the same "environment". Recent papers presenting dimension-independent bounds can be found in Maurer (2006)[62]; Maurer et al. (2013)[64], and Pentina and Lampert (2014)[65].

2.2.2 Single-Sentence Classification

In the context of natural language processing tasks, a model is presented with a sentence and is tasked with categorizing it into one of the predefined class labels [66]. To illustrate, consider the CoLA task, which involves predicting whether an English sentence is grammatically plausible or not. Similarly, the SST-2 task requires determining the sentiment of a sentence taken from movie reviews and labeling it as either positive or negative. In these scenarios, the model's ability to accurately classify sentences plays a crucial role in various applications. For instance, in the CoLA task, the model's proficiency in identifying grammatical correctness aids in language understanding and comprehension. On the other hand, the SST-2 task's sentiment classification capability is valuable for sentiment analysis in movie reviews, enabling sentiment-based insights and opinion mining.

2.2.3 Saliency Detection

Saliency detection [55] is a process aimed at identifying the most crucial and informative parts of input features. Figure 2.6 shows the relation between saliency and relation. This approach has found applications in diverse domains, including computer vision (CV), and natural language processing (NLP), among others. An example of the salience map approach can be seen where the network is tested by occluding portions of the input to visualize the areas influencing the network's output. In the field of computer vision, Class Activation Mapping (CAM) modifies CNN architectures for image classification, resulting in class-specific feature maps. Grad-CAM, a generalized version of CAM, visualizes the linear combination of the last convolutional layer's feature map activations and label-specific weights, providing heuristic explanations of image saliency. Other saliency-based methods, such as Guided Propagation and Deconvolutional Networks, exist along this research line but are specifically designed for certain architectures, like ReLU Network for Guided Propagation. While direct computation of input gradients has limitations, various approaches have been developed to propagate quantities other than gradients through the network to address these challenges.



Figure 2.6: Illustrative examples of the relation between saliency and task similarity. Left: Two tasks are to detect whether the man is smiling and his mouth is open. Right: Two tasks are to detect the horse and the person. The salient regions are close to each other, indicating the potential similarity between the tasks.

2.2.4 Pairwise Text Classification

In the context of natural language processing tasks, a common scenario involves analyzing pairs of sentences to establish their relationship. To achieve this, a model is designed to determine the connection between the two sentences by considering a predefined set of labels [66]. For example, two widely studied language inference tasks are the Recognizing Textual Entailment (RTE) and the MultiNLI (MNLI) tasks. In these tasks, the goal is to predict whether a given sentence entails, contradicts, or remains neutral concerning another sentence. Additionally, there are other intriguing tasks like the Quora Question Pairs (QQP) and Microsoft Research Paraphrase Corpus (MRPC) datasets. These datasets comprise pairs of sentences, and the objective is to discern whether the sentences in each pair

hold semantic equivalence or not. The model is tasked with determining if the sentences convey the same meaning, making it a paraphrase detection challenge.

2.2.5 Relevance Ranking

Figure 2.7 shows the architecture of a MT-DNN model. The model is presented with a query and a list of potential answers, and its objective is to rank the candidates based on their relevance to the query [66]. The QNLI task, a variant of the Stanford Question Answering Dataset, involves determining whether a given sentence contains the correct answer to the query. While QNLI is typically classified as a binary task in GLUE, this study approaches it as a pairwise ranking task, where the model's goal is to prioritize the candidate that contains the correct answer over the candidate without it. This alternative formulation proves to yield a notable enhancement in accuracy compared to binary classification, as demonstrated in the research.

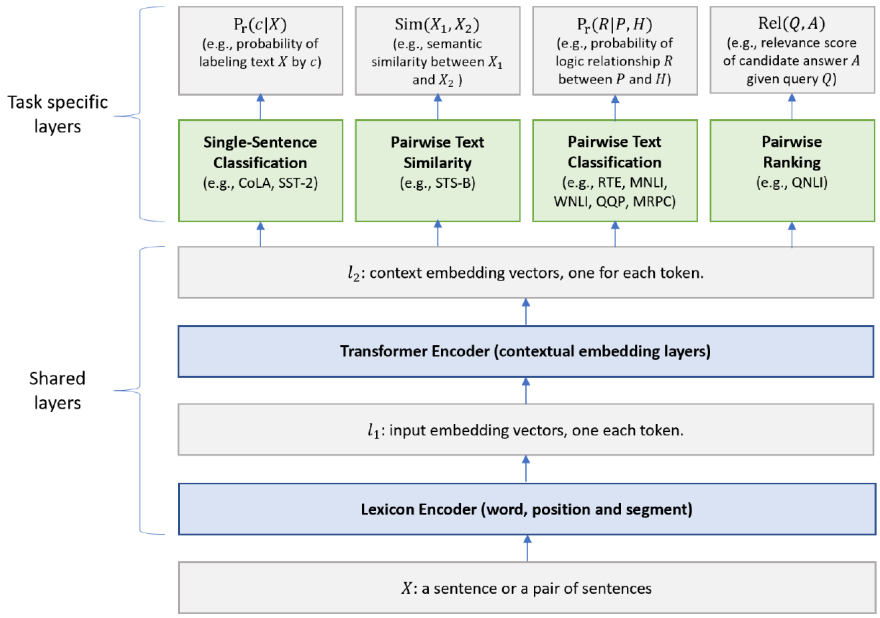


Figure 2.7: Architecture of the MT-DNN model for representation learning. The lower layers are shared across all tasks while the top layers are task-specific.

2.3 Span Identification

Following significant advances in deep learning, a wide range of model architectures have arisen to support the execution of span identification tasks. These include a wide range of approaches, from traditional isolated token classification models to more advanced probabilistic frameworks such as hidden Markov models,

maximum entropy Markov models, and conditional random fields. The landscape also contains recurrent neural networks, such as LSTMs, which are well-known for their sequential processing capabilities, and transformers, such as BERT, which have revolutionized natural language interpretation through self-attention processes.

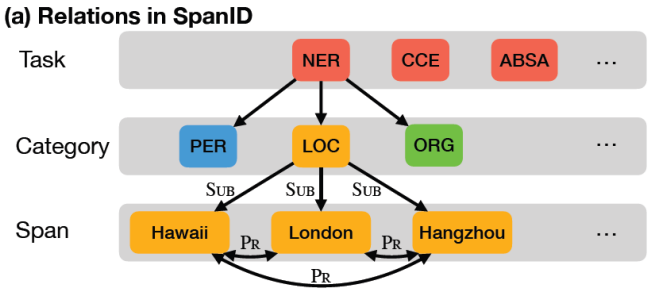
2.3.1 Modeling Peer Relation for Span Identification Tasks

Span Identification (SpanID) tasks entail recognizing distinct spans within input text and categorizing them, which is a critical step in natural language processing (NLP) for scoping significant information [67] [68] [69]. Several domain-specific SpanID tasks have recently come up, including social media Named Entity Recognition (NER) [70], Aspect-Based Sentiment Analysis (ABSA) [71], Contract Clause Extraction (CCE) [72], Span Based Propaganda Detection (SBPD) [73], and Argument Extraction [74], to cater to a wide range of NLP applications. Previously addressed as sequence tagging issues, current techniques have redefined SpanID challenges as Machine Reading Comprehension (MRC) tasks to better use category semantics [75] [75] [76]). However, depending simply on the span-category Subordinate (SUB) connection in training data might result in overfitting and data scarcity concerns. To circumvent these restrictions, researchers recommend using the span-span Peer (PR) connection, which associates two spans as examples of the same category without explicitly labeling them. Models prioritize span semantics above-memorized patterns by recognizing SUB and PR relations together, decreasing overfitting risks, and allowing successful learning even with minimal data for particular categories.

As depicted in Figure 2.8 (a), Span Identification (SpanID) involves extracting spans representing subordinate (SUB) categories from the input text. Early approaches treated SpanID as a sequence tagging task, predicting the category for each token in context. Recent efforts reformulate SpanID as a Machine Reading Comprehension (MRC) problem, generating SUB queries for each category and identifying relevant spans in the input text as responses to these queries (Figure 2.8 (b)).

2.3.2 Span-based Named Entity Recognition

Tishby et al. (1999) [77] developed the information bottleneck (IB) approach, which has been included in cutting-edge (SOTA) deep models such as Variational Autoencoder (VAE) [78] [79] and Deep Variational Information Bottleneck (Deep VIB) [80]. These models, which are classified as supervised generative (e.g., Deep



(b) SpanID in MRC Paradigm

Context:	Gotta dress up for London fashion week and party in style!	
Original data	SUB Query:	Highlight the parts (if any) related to "LOC". Details: the name of politically or geographically defined locations such as cities, provinces, etc.
	Answer:	London
Augmented data	PR Query-1:	Highlight the parts (if any) similar to "Hawaii".
	Answer:	London
	PR Query-2:	Highlight the parts (if any) similar to "Hangzhou".
	Answer:	London

Figure 2.8: (a) Illustrations of Subordinate (SUB) and Peer (PR) relations in SpanID tasks. (b) The constructions of augmented data with PR relations in the MRC paradigm. We use NER here for demonstration purposes.

VIB) and unsupervised (e.g., VAEs) [81], have been used in a variety of domains, including natural language processing.

In the context of Named Entity Recognition (NER), which has traditionally been addressed using sequence models such as BiLSTM and BERT, there has been an increase in span-based models known for their simplicity and effectiveness [82] [83] [84] [85] [86] [87] [88]. This research [89] looks at the benefits of integrating VAE and VIB for the NER challenge by training a span-based NER system with two VAE components and one VIB component. The first VAE component focuses on span reconstruction, similar to Sentence VAEs but works with spans rather than sentences. The second VAE component is synonym generation, which adds semantic information from synonyms to span representations. The workflow is shown in Figure 2.9. The VIB component compresses span representations while conserving essential features for NER.

2.3.3 Aspect-based Sentiment Span Detection

Before making purchase decisions or using services, people frequently seek recommendations from prior customers. With the internet's fast expansion, this guidance is increasingly coming via websites, e-commerce platforms, forums, and product review channels. User evaluations on e-commerce platforms are becoming useful tools for both customers and businesses, providing insights regarding product compatibility and areas for development. Aspect-based sentiment analysis (ABSA) [90] on user input is a difficult process that requires recognizing the men-

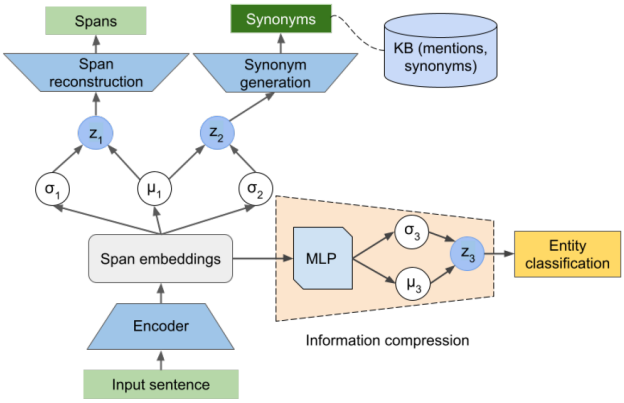


Figure 2.9: Span-based NER is jointly trained with one supervised VIB component: information compression, and two unsupervised VAE components: span reconstruction and synonym generation. The encoder is shared across the components. Synonyms are collected from an external knowledge base (KB).

tioned features and analyzing sentiment. This study [91] focuses on span detection for ABSA, to extract views and sentiments about aspects from customer feedback. Figure 2.10 shows an example of aspect-based sentiment span detection. The user interface of e-commerce platforms has a significant influence on the purchasing experience, with ABSA approaches helping users focus on relevant reviews. While several platforms in Vietnam provide rudimentary feedback systems, there is an increasing need for more advanced ABSA-driven interfaces. To support the study in this area, the publication provides UIT-ViSD4SA, a benchmark Vietnamese smartphone feedback dataset annotated with ABSA spans.

Input	Output
Máy đẹp, sangDESIGN#POSITIVE, **sd thì rất là ok**GENERAL#POSITIVE **máy mượt**PERFORMANCE#POSITIVE. **Pin sd cũng rất lâu mới hết, nhiều khi cả ngày và qua luôn ngày hôm sau mới sạc, sạc rất nhanh khoảng chừng 1 tiếng 5 phút là đầy rồi, ko lâu như iPhones mất gần 3 đến 4 tiếng đầy**BATTERY#POSITIVE. Chỉ sd để lướt web, facebook, youtube. Nghe nhạc rất hay đặc biệt là nghe bằng tai nghe AKG. **Rất xứng đáng với số tiền bỏ ra**GENERAL#POSITIVE., ***Beautifull phone, luxurious***DESIGN#POSITIVE, ***use very ok***GENERAL#POSITIVE ***the machine is smooth***PERFORMANCE#POSITIVE. ***The battery use also takes a long time to run out, sometimes it takes all day and the next day need to charge, very fast charging about 1 hour and 5 minutes is full, not as long as iphone takes nearly 3 to 4 hours to full***BATTERY#POSITIVE. *Only use to surf the web, facebook, youtube. Listening to music is very good, especially listening with AKG heedphones.* ***Well worth the money spent***GENERAL#POSITIVE., 0, 13, "DESIGN#POSITIVE" 15, 31, "GENERAL#POSITIVE" 32, 40, "PERFORMANCE#POSITIVE" 42, 175, "BATTERY#POSITIVE" 315, 346, "GENERAL#POSITIVE"

Figure 2.10: Examples illustrating spans for aspect-based sentiment analysis in Vietnamese.

2.3.4 Automatic Identification of Claim Spans on Twitter

The fast rise of Online Social Media (OSM) platforms has democratized content production while simultaneously facilitating the spread of disinformation, as seen during the COVID-19 epidemic. Identifying and validating claims on OSM has

become a significant research challenge in Argument Mining (AM). 'Claim' [92] is an important term in arguments [93], but present methodologies frequently fail to effectively identify the finer argumentation components inside a claim. This segmentation is critical for extracting further assertions, confirming their legitimacy, and assisting manual fact-checkers. While claim detection has attracted attention, claim span identification, which recognizes the argumentation components of a claim, is still under-explored due to a lack of annotated datasets. To address this gap, CURT (Claim Unit Recognition in Tweets) [94], a massively annotated Twitter corpus, is proposed. Baseline models for claim span detection and claim descriptions, offering specific instructions to models, are introduced and assessed on CURT. Furthermore, DABERTa, a plug-and-play adapter-based variation of RoBERTa that incorporates description information, regularly outperforms traditional baselines and generic models in terms of claim span detection across a variety of criteria. Figure 2.11 shows a few examples of the original tweet and the claim span in it.

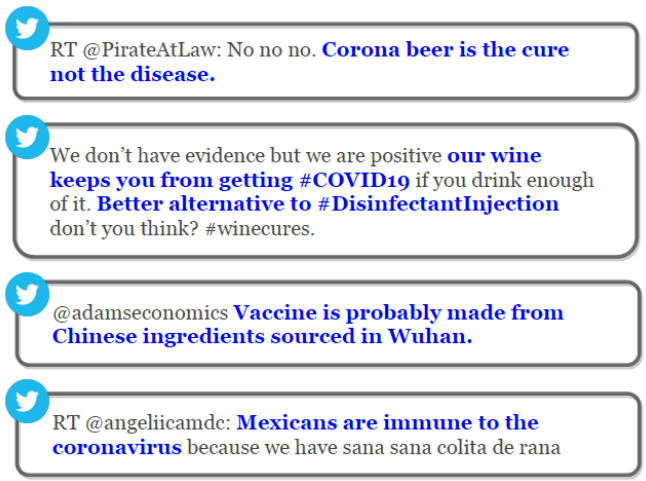


Figure 2.11: Examples of claim tweets and their ground truth claim spans highlighted in boldface text (blue).

CHAPTER 3

Dataset Preparation

In this segment, we will thoroughly examine all the datasets utilized in our thesis. This discussion is meticulously organized into three distinct subsections, each corresponding to one of the three primary tasks of our research: Claim Detection, Multi-task Learning, and Span Identification. In each subsection, we will delve into the specifics of how the data was collected, offering a comprehensive overview of the methods and processes involved. Additionally, we will analyze the trends and statistical insights gleaned from these datasets, providing a detailed understanding of their characteristics and significance. To facilitate better comprehension and to provide a clearer perspective, we will also include various figures and visual representations throughout this section. These visuals will help illustrate the data trends and statistics more effectively, making the information more accessible and engaging.

3.1 Claim Detection

3.1.1 Binary Claim Detection

In this task, the datasets from Sara Rosenthal's [95] paper are used for various tasks. These datasets consist of sentences from LiveJournal blogs and Wikipedia talk pages that have been annotated for opinionated claims. Specifically, there are 2,190 entries from LiveJournal and 2,197 from Wikipedia. Each entry is labeled to indicate whether it contains a claim (Yes or No) and includes sentiment annotations for all the texts. Figure 3.1 shows the train data distribution and Figure 3.2 shows the test data distribution.

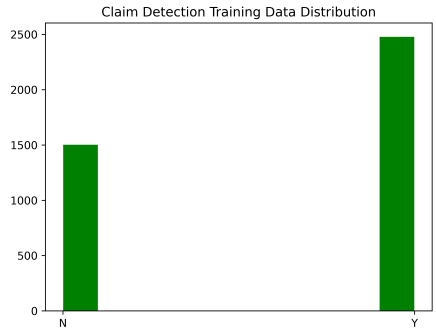


Figure 3.1: Data Distribution of Binary Claim Detection Train Data

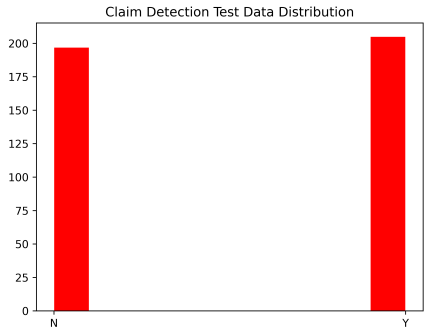


Figure 3.2: Data Distribution of Binary Claim Detection Test Data

3.1.2 Multi-class Claim Detection

For this task, we utilized a modified version of the JUCCD dataset, known as the "Jadavpur University Claim Classification DRDO". This dataset is prepared by DRDO and provided to us. The dataset comprises approximately 700 training samples, split between English and Hindi languages, with around 500 samples in English and roughly 200 in Hindi. The content of these samples primarily revolves around geopolitical and Indian political topics. Each data instance includes the text of a claim and the specific type of claim being made. This bilingual dataset provides a valuable resource for analyzing political discourse and claims within different linguistic contexts. Figure 3.3 shows the English data distribution and Figure 3.4 shows the Hindi data distribution.

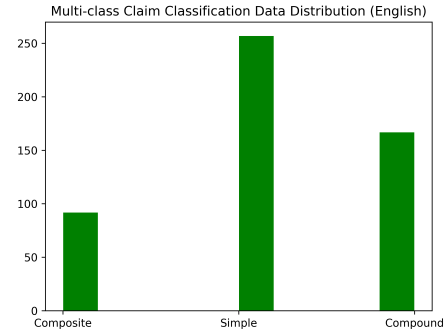


Figure 3.3: Data Distribution of Multi-class Claim Classification English Data

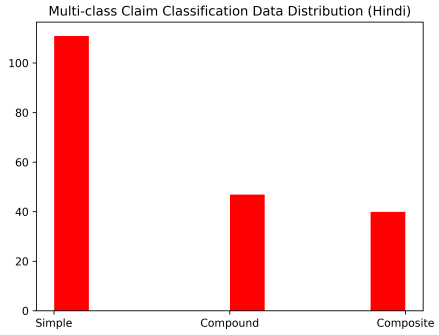


Figure 3.4: Data Distribution of Multi-class Claim Classification Hindi Data

3.1.3 Multi-label Claim Detection

We received a training dataset from a previous paper titled "CAVES: A Dataset to facilitate Explainable Classification and Summarization of Concerns towards

COVID Vaccines." The current version is available on Arxiv [96], and the original article can be found on the ACM site [97]. This dataset includes 9,921 tweets from 2020-21 that expressed anti-vaccination sentiments towards COVID-19 [98]. Human annotators labeled each tweet with one or more labels. Additionally, we were provided with a test set of 500 unlabeled tweets. This test set extends beyond discussions of COVID-19 vaccinations to also cover debates on other vaccines, such as the MMR vaccine and the Flu vaccine. Figure 3.5 shows the category-wise data distribution.

Trends in the dataset:

- The dataset is divided into 12 categories.
- 'Side-effect' is the most common category, appearing 3,805 times.
- 'Religious' is the least common category, with 64 instances.
- 'Pharma,' 'Ineffective,' 'Rushed,' and 'Mandatory' are moderately represented.
- 'None' is associated with 629 items, indicating neutral or unlabeled data.
- The categories 'Conspiracy' and 'Political' suggest discussions on potentially contentious issues.
- 'Pharma' and 'Side-effect' are significant health-related categories.
- The 'Mandatory' and 'Country' categories suggest social impact considerations.
- 'Ineffective,' 'Rushed,' 'Unnecessary,' and 'Conspiracy' reflect negative sentiments.

3.2 Multi-task Learning

3.2.1 Claim Detection

For this task, the datasets from Sara Rosenthal's [95] paper are utilized. These datasets include sentences from LiveJournal blogs and Wikipedia talk pages that have been annotated for opinionated claims. There are 2,190 entries from LiveJournal and 2,197 from Wikipedia. Each entry is labeled to show whether it contains a claim (Yes or No) and includes sentiment annotations for all texts. Figure 3.1 displays the train data distribution and Figure 3.2 displays the test data distribution.

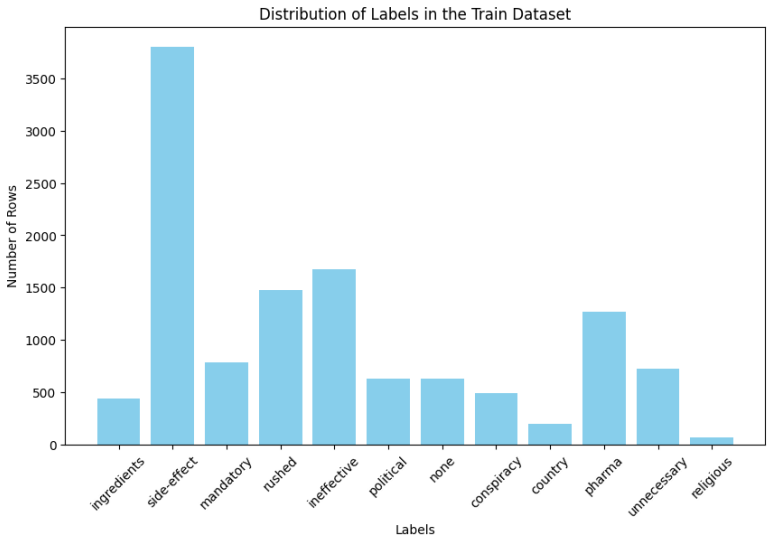


Figure 3.5: Data Distribution of Multi-label Claim Classification Data

3.2.2 Sentiment Analysis

For the Sentiment Analysis task, we utilized a publicly available pre-trained DistilBERT model[1] to generate sentiment labels. We started with the datasets from Sara Rosenthal's [95] paper as our foundation. By applying this model, we were able to assign sentiment labels to all the examples in these datasets. Figure 3.6 displays the train data distribution and Figure 3.7 displays the test data distribution.

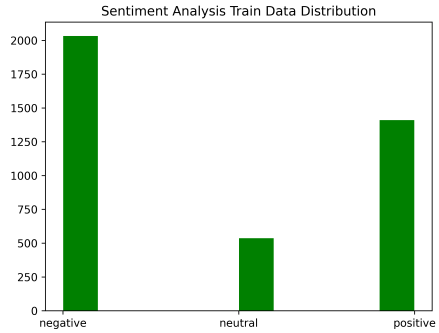


Figure 3.6: Data Distribution of Sentiment Analysis Train Data

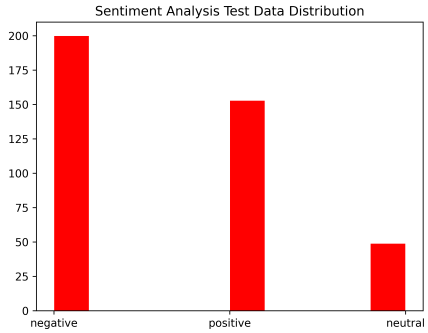


Figure 3.7: Data Distribution of Sentiment Analysis Test Data

3.2.3 Language Identification

A different dataset has been gathered for the Language Identification task, which is a preprocessed version of WiLI-2018[2], the Wikipedia language identi-

[1]https://huggingface.co/lxyuan/distilbert-base-multilingual-cased-sentiments-student
[2]https://bit.ly/language-identification-datasst

fication benchmark dataset. This version includes 22 specific languages: English, Arabic, French, Hindi, Urdu, Portuguese, Persian, Pushto, Spanish, Korean, Tamil, Turkish, Estonian, Russian, Romanian, Chinese, Swedish, Latin, Indonesian, Dutch, Japanese, and Thai. Figure 3.8 displays the train data distribution and Figure 3.9 displays the test data distribution.

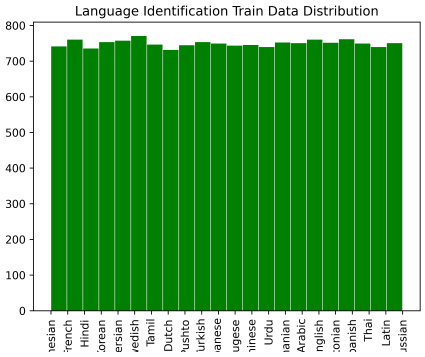


Figure 3.8: Data Distribution of Language Identification Train Data

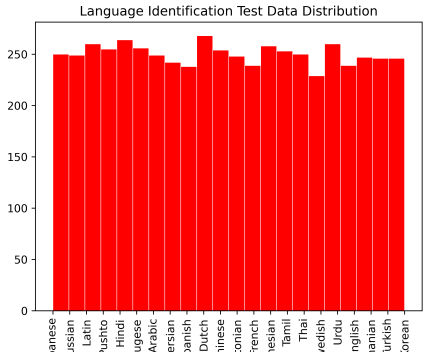


Figure 3.9: Data Distribution of Language Identification Test Data

3.3 Span Identification

3.3.1 Claim Span Identification

For this task, we utilized the JUCSI (Jadavpur University Claim Span Identification) dataset that was specifically provided for our research. This dataset comprises approximately 750 training samples across multiple languages, including English, Hindi, Bengali, and CodeMix. The data predominantly focuses on topics related to COVID-19 vaccines and social distancing measures. Each entry in the dataset includes the original text, an indication of the language used, and the specific span within the text where the claim(s) can be found. This multilingual and topical diversity offers a rich resource for analyzing how different linguistic and cultural contexts handle information related to the pandemic. Figure 3.10 shows the language-wise data distribution.

3.3.2 Sentiment Span Identification

For this task, we utilized a dataset from an Open Source Competition [99] hosted by Kaggle[3], referenced as Sentiment Analysis: Emotion in Text tweets, and used under the Creative Commons Attribution 4.0 International license. This

[3]https://www.kaggle.com/

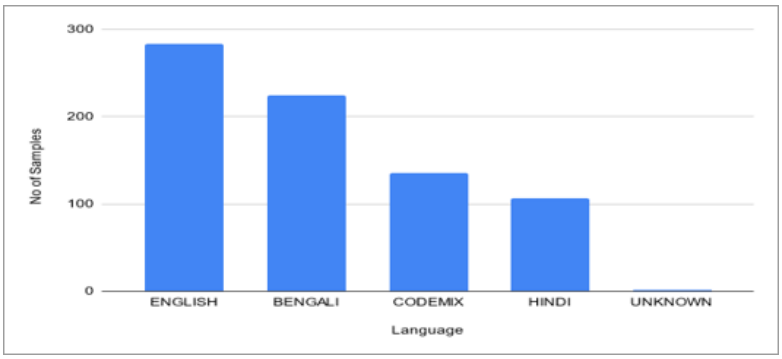


Figure 3.10: Data Distribution of Claim Span Identification Data

dataset includes approximately 27,000 training samples and about 4,000 test samples. Each entry in the dataset comprises the text of a tweet along with a corresponding sentiment label. In the training set, an additional feature is provided: a word or phrase (selected_text) extracted from the tweet that represents the assigned sentiment. Figure 3.11 displays the train data distribution and Figure 3.12 displays the test data distribution.

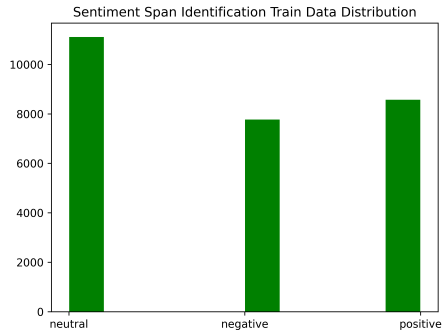


Figure 3.11: Data Distribution of Sentiment Span Identification Train Data

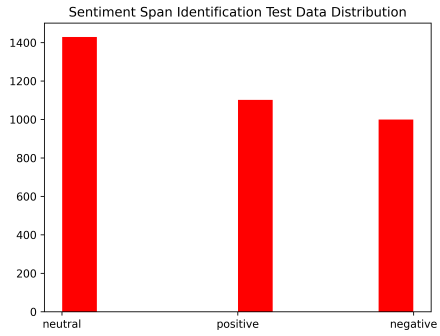


Figure 3.12: Data Distribution of Sentiment Span Identification Test Data

CHAPTER 4

Claim Detection

In this section, we will delve into the diverse classification mechanisms available for Claim Detection, offering a comprehensive examination of each approach. Initially, we will outline the problem statement, provide context for our research, and explain our chosen methodology to address it. Following this, we will provide detailed insights into the practical implementation of our approach, highlighting the strategies employed and the intricacies encountered. Subsequently, we will meticulously analyze the results attained, discussing their significance and implications. Finally, we will conduct a thorough error analysis, identifying areas of improvement and potential avenues for future research and development in this field.

4.1 Binary Classification

4.1.1 Problem Statement

In the realm of automated fact-checking, the process encompasses three primary tasks: claim detection, evidence retrieval, and claim verification. Claim detection involves scouring social media and political sources to pinpoint statements that merit scrutiny. Subsequent phases involve sourcing credible documents to refute detected claims and delivering a verdict.

Attempting to verify every assertion about the real world is unfeasible, necessitating a prioritization process. This objective typically involves assessing the checkworthiness of verifiable claims, a task fraught with subjectivity. Determining the checkworthiness of a claim is complex and can vary depending on the topic and the interests of the user group discussing the claim.

4.1.2 Proposed Approach

In our workflow, once we had acquired our datasets, we meticulously split them into two components: the training set and the validation set. With this division in place, we did a few basic data preprocessing tasks. This initial phase involved cleaning and organizing the data to ensure its suitability for analysis. One of the key steps in this process was text tokenization, a technique where we segmented the textual data into individual tokens or words.

After tokenization, we encountered the challenge of varying sequence lengths within our tokenized texts. To address this, we implemented padding techniques to standardize the length of each sequence to 300 tokens. This standardization ensures consistency and compatibility across the dataset, laying a solid foundation for subsequent analysis.

With our preprocessed data in hand, we tried out the realm of word embeddings, leveraging the power of GloVe300. This method facilitated the transformation of our tokenized text into dense vector representations, capturing intricate semantic nuances within the textual data.

The heart of our project lies in the construction of the model architecture, meticulously crafted to extract meaningful insights from the textual data. Our model architecture is an amalgamation of dual Bidirectional LSTM layers, strategically followed with Dropout layers to mitigate overfitting, and culminating in a GlobalMaxPooling Layer to extract salient features.

As our tokenized texts journey through this intricate neural network architecture, they undergo a transformative process, emerging as probability values at the output layer. These probabilities serve as the foundation for our binary prediction logic, enabling us to derive actionable insights and make informed decisions based on the model's output.

4.1.3 Implementation

4.1.3.1 Architecture

Input Layer: First, we have appointed an input layer for the neural network. The input to the neural network is $X = [k_1, k_2, k_3, ..., k_L]$. Here L represents the length of each input sequence, where $L = 300$.

Embedding Layer: The input sequences are transformed into dense vectors using an embedding layer. As shown in Figure 4.1, to embed words, we utilized pre-trained "GloVe"[100] word embeddings, which are vectors with a dimensionality of $D = 300$. Each token k_i from the sentence X was transformed into a sequence of vectors x_i of length D. Thus, from a tokenized sentence $X = [k_1, k_2, k_3, ..., k_L]$

we obtain $X_{L\times D} = [x_1, x_2, x_3, ..., x_L]$. These vectors were then input into a BiLSTM layer, as illustrated in Figure 4.1. This layer converts words (represented as indices) into dense vectors of fixed size. Here, we have used $trainable = False$ so that these embeddings will remain fixed during training.

Bidirectional LSTM Layers: The BiLSTM[101] is a variant of the recurrent neural network (RNN) architecture known for its bidirectional long-short-term memory. Bidirectional LSTM layers are used to capture the sequential information in the data.

- The First layer consists of two LSTM layers with 512 units each, processing the input sequences in both forward and backward directions and returning sequences.
- Another Bidirectional LSTM layer is added with 64 units.

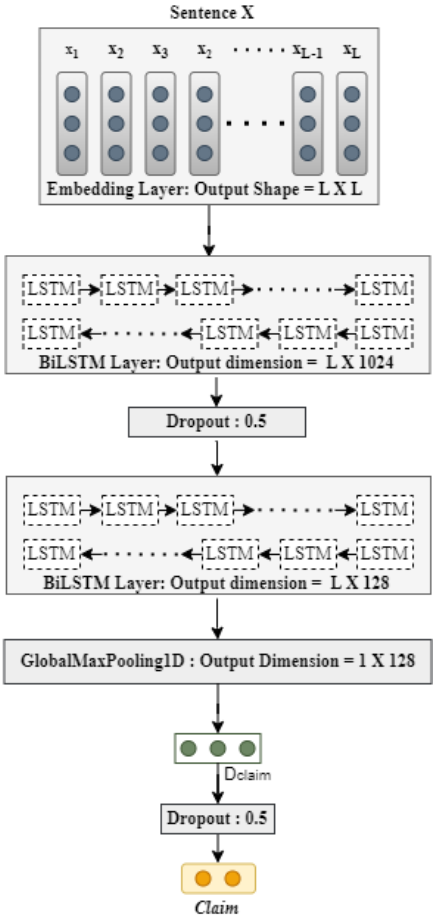


Figure 4.1: The Graphical Representation of the Claim Detection Model

Dropout Layer: Dropout is applied to the output of the LSTM layers to prevent overfitting. It randomly drops 50% of the units during training.

Global Max Pooling Layer: This layer reduces the sequence data to a single vector by taking the maximum value across each feature dimension. It helps in capturing the most relevant information from the sequences.

Dense Layers for Claim Type Prediction: This part is a combination of three layers. These are:

- A dense layer with 64 units and a ReLU activation function is added.
- Dropout of 50% is applied in this layer.
- Finally a dense layer with 1 unit and sigmoid activation function, which outputs the probability of the claim being true.

4.1.3.2 Compilation

The model is compiled with the binary cross-entropy loss function (suitable for binary classification tasks), Adam optimizer with a learning rate of 0.0005, and accuracy as the evaluation metric. Early stopping is employed to monitor the validation loss. If the validation loss does not improve for three consecutive epochs, training is stopped, and the weights corresponding to the best validation loss are restored. The details are shown in Table 4.1

Table 4.1: Parameters used to train the binary claim detection model

Parameters	Value
Embedding	GloVe word embedding 300d
BiLSTM hidden units	512, 64
Dropout	0.5
Loss function	BinaryCrossEntropy
Optimizer	Adam
Metric	Accuracy
Learning rate	0.0005
Epoch	25
Batch size	64

4.1.3.3 Training

As shown in Table 4.1, the model is trained on the training data with labels (0 or 1) for 25 epochs using a batch size of 64. Validation data and corresponding labels are used for validation during training. The early stopping callback is applied during training.

Task	Precision	Recall	F1-Score	Accuracy
Claim Detection	0.711	0.711	0.711	0.711

Table 4.2: Performance Metrics of Binary Claim Detection Task

4.1.3.4 Prediction and Probability Calculation

After training, the model is used to predict claims for the test data. As the model uses a sigmoid activation function in the output layer, it outputs the probability of the claim being true. So, if the probability is more than 0.5 it will be labeled as true, or else false.

4.1.4 Experimental Results

4.1.4.1 Experimental Setup

We use TensorFlow[1] and Keras[2] to implement our proposed model and use the Collaboratory[3] environment to execute the code for the proposed model. We have used the Precision, Recall, F1 Score metrics, and Accuracy for the validation split of our final dataset for evaluation.

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4.3}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.4}$$

4.1.4.2 Result

The findings from our experiment are laid out comprehensively in Table 4.2, offering insights into key metrics such as Precision, Recall, F1-Score, and Accuracy. Furthermore, Figure 4.2 visually represents the confusion matrix, providing a clearer understanding of the data distribution and performance evaluation.

[1]https://www.tensorflow.org/
[2]https://keras.io/
[3]https://colab.research.google.com/

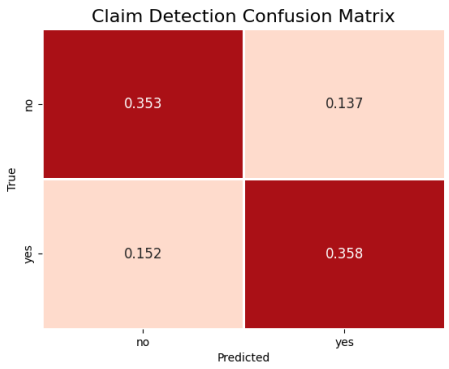


Figure 4.2: The Confusion Matrix of Binary Claim Detection Task

4.1.5 Error Analysis

This segment delves into the error analysis of our suggested framework. Specifically, we've conducted this analysis solely on the validation dataset, showcasing select examples in tabular form. Table 4.3 outlines the error analysis for tasks with varying characteristics.

In Sentence S_1, there's an abundance of numbers and abbreviations, which might have led to our model's failure in this instance. Sentence S_2 presents named entities, adding complexity and resulting in our model erroneously predicting "no" instead of "yes." Sentence S_3 suggests the presence of a claim, but the true label contradicts this, causing confusion for our model. Sentence S_4 poses a challenge in determining the presence of a claim, leading our model to incorrectly predict its absence despite the true label indicating otherwise.

ID	**Text**	**True**	**Predicted**
S_1	– Preceding unsigned comment added by 24.196.105.69 -LRB- talk	no	yes
S_2	There are no reliable sources that assert a Muslim heritage, so Snopes is entirely adequate to refute such nonsense.	yes	no
S_3	Yet he states in his book he was studying the Koran.	no	yes
S_4	In Western Europe, using IE or non-IE as a comparative yardstick is acceptable.	yes	no

Table 4.3: Error Analysis of Binary Claim Detection Task

4.2 Multi-class Classification

4.2.1 Problem Statement

In this study, we're presented with text containing sentences related to claims. Our goal is to categorize these sentences into three types: Simple, Compound, and Composite.

Simple sentences are those found in tweets that contain direct responses or statements from notable individuals, public figures, government entities, or courts. Example: court-ordered stay order against the application of three Krishi bills. It provides a clear statement that can be easily verified for its accuracy.

Compound sentences, on the other hand, involve two or more sentences that are closely intertwined or dependent on each other. Example: Narendra Modi said In Parliament If China doesnt stop aggression in the border area, there may be a high probability of starting a war; After winning the gold medal in Javelin through an honorary post of Captain was given to Neeraj Chopra (Indian Athletic) from Indian BSF force. It illustrates how the truth of one statement hinges on the truth of another.

Composite sentences are those where the veracity of the statement may be challenging to confirm, either due to a lack of concrete evidence or complexities surrounding the context. For instance, verifying statements attributed to historical figures like Lal Bahadur Shastri or events like India's independence from British rule can be intricate. Similarly, statements made by political figures in certain contexts may lack clear evidence or accountability. If anyone wants to take action based on those statements, he/she can easily say that this is a fake document, so we are unable to prove the actual reality.

It's worth noting that handling claims within tweet texts, especially across different languages, poses challenges due to the informal and noisy nature of tweets. Nonetheless, with the right frameworks and techniques, classifying claims from both Hindi and English tweets is feasible, albeit complex.

4.2.2 Proposed Approach

In our workflow, we began by dividing our datasets into training and validation sets. We then performed fundamental data preprocessing tasks, including tokenization, converting text to lowercase, removing stop-words, and handling special characters. This initial stage was crucial for cleaning and organizing the data, ensuring its readiness for analysis.

Following tokenization, we addressed the challenge of varying sequence lengths

in our tokenized texts by employing padding techniques to standardize each sequence to 300 tokens. This standardization ensured consistency and compatibility across the dataset, providing a strong foundation for further analysis. With our preprocessed data, we utilized GloVe300 for word embeddings, transforming the tokenized text into dense vector representations. This method captured intricate semantic nuances within the textual data.

The core of our project involved constructing a model architecture designed to extract meaningful insights from the data. The architecture included an Input Layer, Embedding Layer, Bidirectional LSTM layers, Dropout layers to combat overfitting, and a GlobalMaxPooling Layer to extract prominent features.

As the tokenized texts passed through this neural network, they underwent a transformative process, culminating in probability values at the output layer, facilitated by the softmax activation function. From these probabilities, we derived our output label using the argmax function, which identified the label with the highest output probability.

4.2.3 Implementation

4.2.3.1 Architecture

Input Layer: $Input(shape = (L,))$ creates an input layer. The input to the neural network is $X = [k_1, k_2, k_3, ..., k_L]$. Here L is the length of the input sequences, where $L = 300$.

Embedding Layer: As shown in Figure 4.3 $Embedding(vocab_length, L,$ $weights = [embedding_matrix], input_length = T, trainable = False)(i)$ creates an embedding layer. It converts input sequences into dense vectors of fixed size. vocab_length is the size of the vocabulary, L is the dimensionality of the output vectors, embedding_matrix is a pre-trained embedding matrix, and $trainable = False$ indicates that the embeddings will not be updated during training.

Bidirectional LSTM Layers: Two Bidirectional LSTM layers are used. LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) that is capable of learning long-term dependencies. The first layer has 512 units and returns sequences, while the second layer has 64 units and also returns sequences. Both layers use the hyperbolic tangent (tanh) activation function.

Dropout Layer: It applies dropout regularization, randomly setting a fraction of input units to 0 at each update during training, which helps prevent overfitting.

Global Max Pooling Layer: $GlobalMaxPooling1D()$ reduces each sequence to a single vector by taking the maximum value of each feature across the sequence.

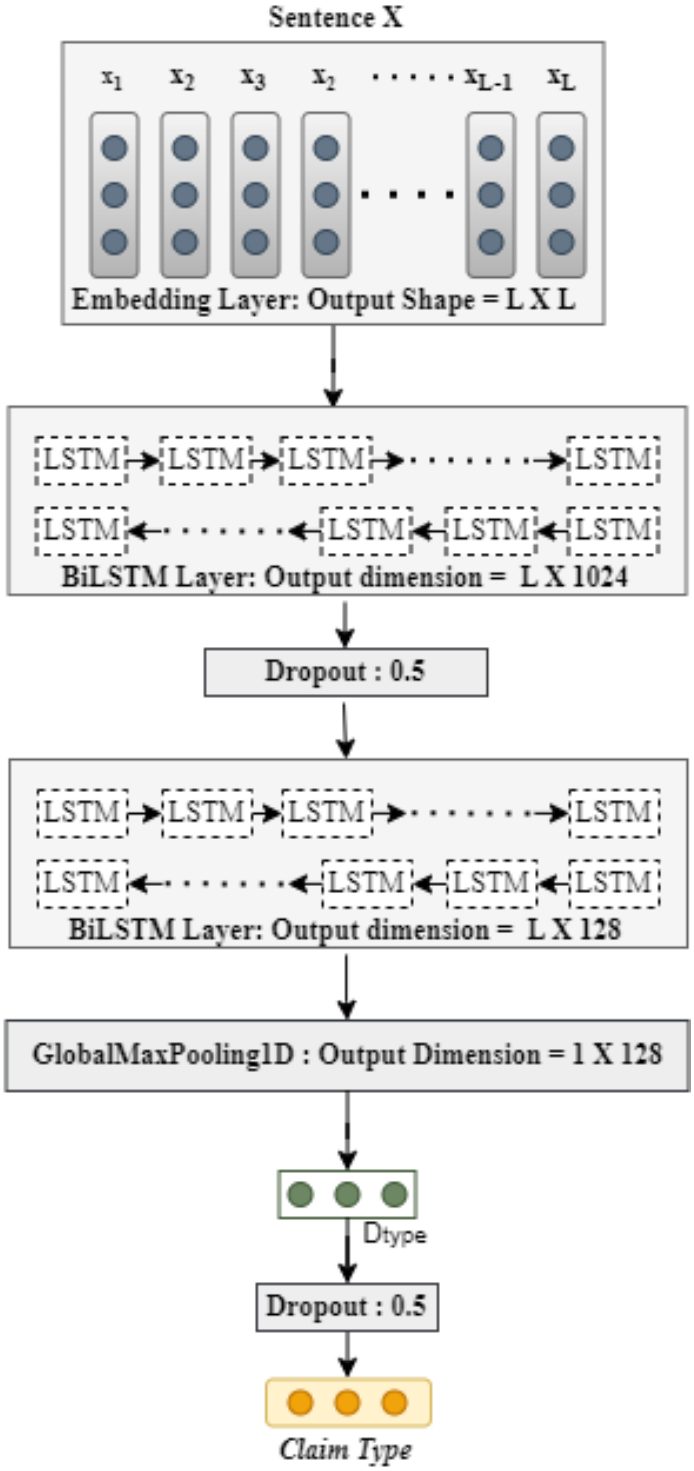


Figure 4.3: The Graphical Representation of the Claim Classification Model

This helps in capturing the most important features of the sequences.

Dense Layers for Claim Type Prediction: The first dense layer has 64 units with ReLU activation function, followed by a dropout layer with a dropout rate of 0.5. Another dense layer with 3 units (as there are 3 types of claims) and a softmax activation function is added to output the probabilities of each claim type.

4.2.3.2 Compilation

The model is compiled using the Sparse Categorical Crossentropy loss function, Adam optimizer with a learning rate of 0.0005, and accuracy as the evaluation metric as shown in Table 4.4. Early stopping is employed to monitor the validation loss. If the validation loss does not improve for three consecutive epochs (patience=3), the training is stopped, and the weights corresponding to the best validation loss are restored.

Table 4.4: Parameters used to train the multi-class claim classification model

Parameters	Value
Embedding	GloVe word embedding 300d
BiLSTM hidden units	512, 64
Dropout	0.5
Loss function	SparseCategoricalCrossEntropy
Optimizer	Adam
Metric	Accuracy
Learning rate	0.0005
Epoch	25
Batch size	64

4.2.3.3 Training

As depicted in Table 4.4, the model is trained using the fit() function on training data and training labels for 25 epochs with a batch size of 64. Validation data and validation labels are used for validation during training. The early stopping callback is applied during training with the patience of 3 epochs.

4.2.3.4 Prediction and Probability Calculation

After training, the model is used to predict claim types for the test data. The predicted probabilities for each claim type are calculated using softmax activation. The claim type with the highest probability is considered the predicted claim type.

Task	Precision	Recall	F1-Score	Accuracy
Claim Detection	0.418	0.495	0.451	0.636

Table 4.5: Performance Metrics of Multi-class Claim Classification Task

4.2.4 Experimental Results

4.2.4.1 Experimental Setup

We use TensorFlow[4] and Keras[5] to implement our proposed model and use the Collaboratory[6] environment to execute the code for the proposed model. We have used the Precision (4.1), Recall (4.2), F1 Score (4.3), and Accuracy (4.4) for the validation split of our final dataset for evaluation.

4.2.4.2 Result

In Table 4.5, we have detailed the results of our experiment, providing a comprehensive overview of important metrics like Precision, Recall, F1-Score, and Accuracy. Additionally, Figure 4.4 serves to visually depict the confusion matrix, enhancing comprehension of both data distribution and performance assessment.

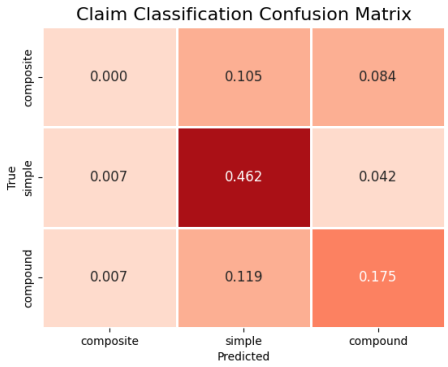


Figure 4.4: The Confusion Matrix of Multi-class Claim Classification Task

4.2.5 Error Analysis

This section explores the error analysis of our proposed framework. We've specifically focused on analyzing errors using the validation dataset, presenting specific examples in a tabular format. Table 4.6 provides a breakdown of the error analysis for tasks that exhibit different characteristics.

In Sentence S_1, our model identified the sentence as a simple claim when it's actually a composite claim. This flaw in our model is evident as composite claims are

[4]https://www.tensorflow.org/
[5]https://keras.io/
[6]https://colab.research.google.com/

frequently misidentified as simple sentences, as indicated by the confusion matrix in Figure 4.4. Sentence S_2, despite being a compound claim, was also wrongly classified as a simple claim, aligning with the observations from the confusion matrix. Sentence S_3 presents a rare instance where a compound claim was misclassified as a composite claim, possibly due to the presence of numerous named entities. In Sentence S_4, distinguishing between a composite and compound claim proves challenging. Regrettably, our model labeled it as a compound claim, contrary to its designation as a composite claim in our dataset.

ID	Text	True	Predicted
S_1	UN: Widespread russian Torture in Ukraine is Deliberate #Bakhmut #Defeatrussia #Ukraine #Standwithukraine #Zelenskyy #Kyiv #Kakhovka #Terrorism #Flood #Torture #Warcrimes	Composite	Simple
S_2	The number of people who have recuperated from the disease surged to 4,44,59,514, while the case fatality rate was recorded at 1.18 per cent. #Covid19	Compound	Simple
S_3	No wonder Rishi Sunak allows #BorisTheLiar to continue to have influence - Not only did #RichRishi also go against #Covid19 rules too, just look how much he backed lying Boris at every opportunity #GeneralElectionN0W #ToriesOut	Compound	Composite
S_4	Our babies are our future. The development of our nation depends on our babies. Let the students learn lessons peacefully. #WhatsHappeningInMyanmar #terrorism #rejectPDF #NUG #MyanmarNews	Composite	Compound

Table 4.6: Error Analysis of Claim Classification Task

4.3 Multi-label Classification

4.3.1 Problem Statement

Vaccination is becoming increasingly crucial in the ongoing battle to safeguard public health, reducing the risks and spread of numerous diseases. Recent events underscore the pivotal role immunization plays in mitigating the widespread impacts of the COVID-19 pandemic. However, the necessity for widespread vaccination extends beyond the current crisis to encompass a broader spectrum of ailments such as childhood illnesses and seasonal flu outbreaks. Nevertheless, skepticism

surrounding vaccine acceptance is a complex landscape shaped by political maneuvers, concerns over potential adverse effects, and other nuanced issues. A comprehensive understanding of the diverse factors surrounding vaccines is essential for effectively addressing this intricate subject. In the digital era, social media serves as a vast repository of diverse viewpoints, facilitating rapid insights into vaccine-related discussions. People worldwide hold varying perspectives on vaccination, often expressed through platforms like Twitter. A single tweet can convey a multitude of viewpoints and subtly influence others' perceptions. However, some tweets may diverge from the positive intentions of vaccination and should not be widely disseminated. Consequently, there is an urgent need to develop machine-learning algorithms capable of categorizing COVID-19 vaccine-related tweets.

The primary objective is to establish a robust multi-label categorization system capable of assigning meaningful labels to tweets, encompassing a wide array of vaccine-related concerns expressed by authors. It is important to note that a single tweet may fall into multiple categories, reflecting the author's diverse concerns. This is our submitted work for the shared task Artificial Intelligence on Social Media (AISoMe) in FIRE 2023[102].

The taxonomy of concerns for classification includes the following:

1. Unnecessary: Suggests immunizations are unnecessary or advocates for alternative treatments.

2. Mandatory: Opposes mandatory vaccination, arguing against compulsory immunization.

3. Pharma: Criticizes pharmaceutical companies, alleging profit-driven motives or opposing such corporations.

4. Conspiracy: Implies deeper conspiracies beyond profit motives, such as tracking individuals or labeling COVID as a hoax.

5. Political: Expresses concerns about the political manipulation of vaccines for governmental or political agendas.

6. Country: Criticizes vaccination based on its country of origin or manufacturing.

7. Rushed: Raises concerns about the expedited or insufficiently tested vaccination process.

8. Ingredients: Worries about vaccine components or technology, such as fetal cells or mRNA vaccines affecting DNA.

9. Side-effect: Expresses concerns about negative effects or fatalities associated with vaccination.

10. Ineffective: Claims vaccines are ineffective or worthless.

11. Religious: Opposes vaccination on religious grounds.

12. None: Indicates no specific reason stated or reasons other than those listed.

4.3.2 Proposed Approach

In tackling this challenge, our initial emphasis was on thoroughly preparing the tweets for analysis. We meticulously preprocessed the data, ensuring its quality and relevance. Subsequently, we broke down each tweet into tokens and fed them into our model, leveraging a fine-tuned BERT Model renowned for its effectiveness in natural language processing tasks. Following this, we implemented a distinctive prediction method, allowing for the assignment of multiple labels to a single tweet. This approach acknowledges the nuanced nature of discussions surrounding vaccination, recognizing that a single tweet may encompass various concerns or viewpoints simultaneously.

4.3.3 Implementation

4.3.3.1 Pre-processing

The dataset's tweets underwent preprocessing to eliminate irrelevant elements, thereby enhancing the BERT model's word embeddings' efficiency. Tweets often include links, emojis, punctuation, and Twitter embeds like images or GIFs, which can impact the BERT model's performance.

Our preprocessing pipeline included the following steps:

1. **Lowercasing:** All words within the tweets were converted to lowercase to enhance the classification process.

2. **Emojis to words:** Emojis and emoticons carry contextual information and are valuable for sentiment analysis. We transformed them into text format (:emojiname:) using the 'emoji' library [103], removing the colons afterward.

3. **Removal of non-alphanumeric characters:** Python's 're' library was employed to discard irrelevant characters such as colons, semicolons, brackets, and hashtags. Additionally, we retrieved the raw words by eliminating colons from the processed emojis.

4. **Removal of URLs:** URLs present in the tweets were eliminated using regular expressions.

Input: #VAERS 16y♀#Pfizer died pulmonary embolism suspected #Cardiacarrest #vaccinedeaths #adolescent Symptoms: Cardiac arrest, Circulatory collapse, Computerised tomogram thorax abnormal, Death, Lung assist device therapy, Pulmonary embolism https://t.co/dTAdjik8gO.

Output: vaers 16y **female sign** pfizer died pulmonary embolism suspected cardiacarrest vaccinedeaths adolescent symptoms cardiac arrest circulatory collapse computerized tomogram thorax abnormal death lung assist device therapy pulmonary embolism.

4.3.3.2 Methodology

Model: BERT-BASE-UNCASED [104] is a model built on transformer architecture, pre-trained on a vast English text corpus in a self-supervised manner. The uncased variant of the model operates on lowercase text, simplifying its training and usage. Its pretraining involved two main methods:

1. **Masked Language Modeling (MLM):** In this approach, a sentence undergoes random masking of 15% of its words. The model then predicts these masked words, enhancing its understanding of contextual word relationships.

2. **Next Sentence Prediction (NSP):** This method aids the model in comprehending relationships between consecutive sentences. By presenting two sentences as input, BERT predicts the correct sequence of these sentences, fostering its grasp of longer-term dependencies in text.

Experimental Setup: First, the tweets undergo tokenization, and then the labels are transformed into a binary matrix, which serves as input for classification. Sklearn's MultiLabelBinarizer facilitates this conversion, generating a 1x12 matrix for our current task, which involves 12 labels for tweet classification.

The tokenized tweets are comprised of three components: input_ids, attention_mask, and segment_ids. Input_ids represent the raw tokenized tweets, while attention_masks distinguish relevant tokens from those to be disregarded. Segment_ids aid in distinguishing different parts of the input, such as separate sentences.

These input components are fed into a Keras layer, which embeds and encodes the tokens into a dense vector. Subsequently, two dense layers follow, with 128 and 64 units respectively, processing the embeddings and generating new features.

The output layer, also a dense layer, comprises 12 units corresponding to the 12 classification labels. It takes the features from the preceding dense layers as input and produces probabilities for each label.

To mitigate overfitting during training, dropout layers are employed after each dense layer while fine-tuning the model.

The model's graphical representation is depicted in Figure 4.5.

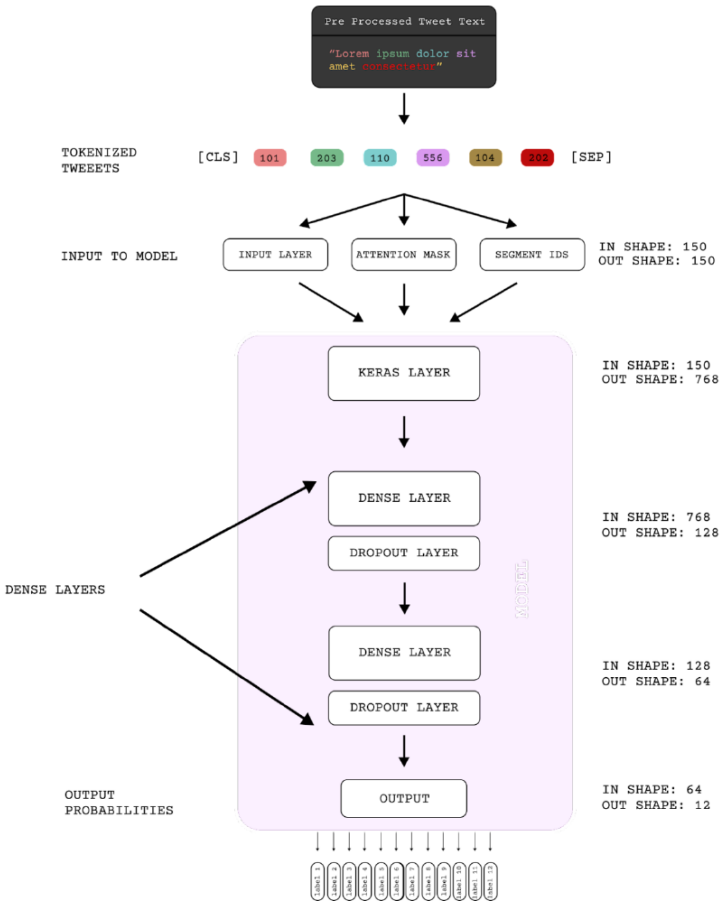


Figure 4.5: The Graphical Representation of the Model

Prediction: We utilized the fine-tuned BERT model to create embeddings for the tweets and determine the probability scores for each tweet across twelve classes. If the probability of a class exceeded the threshold of 0.25, it was considered a predicted class for that tweet.

4.3.4 Experimental Results

The objective is to classify data into multiple classes and labels. Evaluation of performance is based on two metrics: the macro-F1 score and the Jaccard Coefficient, applied across 12 distinct categories. Table 4.7 displays the outcomes of our submission for this task. Among three attempts, our most successful model

Run file	Summary of Methodology	Macro-f1	Jacc
run1.csv	Overall model is the same as discussed above and for this run, We set the learning rate at 1e-4	0.54	0.58
run3.csv	Overall model is the same as discussed above and for this run, We set the learning rate at 1e-6	0.5	0.55
run2.csv	Overall model is the same as discussed above and for this run, We set the learning rate at 1e-8	0.4	0.5

Table 4.7: Results of three trials

achieved a macro-F1 score of 0.54 and a Jaccard Coefficient score of 0.58.

4.3.5 Observations

In our study, we employed a transformer-based model pre-trained on a sizable dataset of COVID-19-related anti-vaccination tweets. Our primary objective was to classify tweets into 12 distinct categories. Surprisingly, our findings indicate that the transformer-based model surpasses traditional natural language processing classifiers such as Naive Bayes, Logistic Regression, and Support Vector Machine. This superiority stems from the transformer model's capacity to generate more expressive word embeddings, leading to enhanced performance on our task.

Considering the transformer-based model's affinity for extensive data, we suggest exploring data augmentation techniques to bolster our model's performance. Additionally, we propose employing adversarial training to enhance the model's resilience and robustness. These advancements have the potential to further elevate our categorization accuracy.

4.4 Discussion

This chapter delves into a multifaceted exploration of various claim-related tasks, encompassing Claim Detection, Multi-class Claim Classification, and Multi-label Claim Classification. Through rigorous experimentation, we investigated the efficacy of several models including LSTM, BiLSTM, and BERT across these tasks, contextualizing our research within datasets rooted in the Indian landscape. While our efforts shed light on the strengths and weaknesses of each model, there remain ample opportunities for refinement and advancement. For instance, incorporating diverse embedding techniques could enhance the robustness of our models, while

harnessing transformer-based architectures across all tasks could potentially yield more efficient and accurate results, given their proven efficacy in claim-related contexts. Additionally, expanding our experimentation to encompass a broader spectrum of data could offer deeper insights and validation of our findings, paving the way for further advancements in claim detection methodologies tailored to the Indian context.

CHAPTER 5

Multi-task Learning

In today's digital age, where the internet is readily available to all, people freely express their thoughts and emotions across various social media platforms like Twitter, Facebook, and Reddit. These platforms host a multitude of posts written in different languages, containing sentiments ranging from positive to negative, along with claims on various topics. To ensure comprehension across diverse audiences, there's a need for language detection or translation tools.

Natural Language Processing (NLP) encompasses fields such as sentiment analysis, sarcasm detection, claim analysis, and language classification, which aim to decipher sentiments, identify sarcasm or irony, detect hate speech, offensive language, political claims, etc. Traditionally, researchers focused on individual tasks like sentiment analysis or claim detection. However, some sentences may encompass multiple aspects, necessitating separate models for sentiment, sarcasm, and claim detection.

Multi-task learning (MTL) as the name suggests, refers to a single shared machine-learning model that is capable of performing multiple different tasks simultaneously [105]. While multi-task learning (MTL) models have been explored extensively for almost a decade, most emphasize similar tasks like sentiment analysis, sarcasm detection, irony detection, hate speech detection, etc. MTL models are scarce for dissimilar tasks. This study proposes an MTL model for dissimilar tasks like claim detection, language identification, and sentiment analysis.

There are a few advantages of Multi-Task Learning (MTL) compared to single-task learning, presented in bullet points[106]:

- **Facilitates Generalization:** MTL enables achieving generalization for multiple tasks simultaneously.
- **Enhanced Task Performance:** Each task improves performance by leveraging associations with other participating tasks.

- **Sustainability and Power Efficiency:** MTL embraces sustainability and saves power by employing a single model to handle multiple tasks concurrently.
- **Reduced Complexity:** MTL reduces the complexity from the user's perspective, offering a streamlined approach to handling various tasks.

The main objectives of this study are to evaluate whether integrating different classification tasks into an MTL model enhances overall performance compared to single-task models and to determine how each task benefits from MTL. Two types of MTL models are proposed: a simple MTL model and an MTL model with fusion techniques.

5.1 Task Descriptions

5.1.1 Claim Detection

Identifying claims within text entails recognizing and evaluating claims that may be true or false, and are often intended to persuade or move an audience towards a certain position or argument. This job has wide-ranging implications, including information retrieval and natural language processing, as well as disciplines like politics, science, and health.

Finding textual claims is important for a variety of reasons. First, in fact-checking, organizations can guarantee that comments made by individuals or groups are correct. Second, it facilitates the search for specific claims or information inside massive text databases. Third, argument analysis requires an awareness of the structure and power of arguments. Overall, the ability to detect textual assertions is important for checking facts, gathering information, and successfully analyzing arguments.

Nonetheless, recognizing claims has obstacles such as ambiguity, dependency on context, and variation in style, structure, and language.

In conclusion, Claim Detection is an important process for extracting and analyzing claims from textual content, with applications ranging from fact-checking to information retrieval and argument analysis. It entails using a variety of methodologies, including rule-based and machine-learning methods, as well as dealing with the inherent difficulties and ethical implications.

5.1.2 Sentiment Analysis

Sentiment analysis evaluates whether a text has a positive, negative, or neutral tone. The goal of sentiment mining is to understand people's opinions in a way

that helps businesses grow. It goes beyond basic polarity to include joy, sadness, and anger. Sentiment analysis, which employs a variety of Natural Language Processing approaches such as rule-based, automatic, and hybrid algorithms, is becoming increasingly important in understanding sentiment across different data types.

With individuals expressing their opinions and feelings more publicly than ever before, sentiment analysis is an essential tool for monitoring and understanding sentiment across varied datasets. Businesses may automatically analyze client feedback via surveys and social media interactions to determine which parts delight or upset their consumers, allowing them to modify their products and services appropriately.

For example, using sentiment analysis to assess over 4,000 open-ended replies in customer satisfaction surveys might reveal the causes of consumer happiness or dissatisfaction at each stage of the process. Whether it's tracking brand sentiment to quickly handle unsatisfied consumers or comparing sentiment between quarters to determine what actions are required, sentiment analysis enables organizations to dive deeper into qualitative data and effectively comprehend sentiment movements.

5.1.3 Language Identification

Language Identification is one of the most popular tasks in Natural Language Processing. Language Identification or Language Guessing is the task of determining the natural language of the content. It is considered as a special case of Text Categorization or Text Classification. Language Identification is used as a preprocessing task in various Natural Language Processing tasks.

Language Identification is the task of automatically detecting the language(s) present in a document based on its content. Language Identification task commonly assumes that the documents are written in one of the closed sets of known languages from the training examples. Hence, it is the task of selecting the most likely language from the set of training languages.

In Language Identification, documents are modeled as a stream of characters (Cavnar and Trenkle, 1994; Kikui, 1996), often approximated by the corresponding stream of bytes (Kruengkrai et al., 2005; Baldwin and Lui, 2010a) for robustness over variable character encoding.

5.2 Multi-task Environment

5.2.1 Two Tasks Combinations

In this part, we will look at how to pair two jobs at the same time. Essentially, our model will be trained to handle two distinct tasks simultaneously.

- **Claim Detection + Sentiment Analysis:** First, we will look at the combination of Claim Detection and Sentiment Analysis. Claim Detection distinguishes between two labels (yes and no), whereas Sentiment Analysis categorizes text into three labels (positive, negative, and neutral).
- **Sentiment Analysis + Language Identification:** Next, we will consider the combination of Sentiment Analysis and Language Identification. Sentiment Analysis sorts text into three Sentiment Analysis categories (positive, negative, and neutral), whereas Language Identification selects a language from a pool of 22 choices.
- **Claim Detection + Language Identification:** Finally, we will discuss the pairing of Claim Detection and Language Identification tasks. Claim Detection involves identifying two labels (Yes and No), while Language Identification requires recognizing one of 22 different languages.

5.2.2 Three Tasks Combination

In this section, we'll explore the integration of three tasks simultaneously. Here, our model will be trained to handle three distinct tasks concurrently.

- **Claim Detection + Sentiment Analysis + Language Identification:** In this case, the tasks we've chosen to combine are Claim Detection, Sentiment Analysis, and Language Identification. Claim Detection involves distinguishing between two labels (Yes and No), Sentiment Analysis involves categorizing text into three labels (Positive, Negative, and Neutral), and Language Identification entails recognizing one of 22 different languages.

5.3 Proposed Approach

In this section, we outline our proposed approach, which aims to create a unified multi-task learning model capable of classifying three distinct types of tasks: claim detection, language identification, and sentiment analysis. Here's how we proceed:

5.3.1 Text Preprocessing

Before diving into classification, we preprocess each sentence S to ensure uniformity and cleanliness. This involves several steps:

- Removal of HTML tags.
- Conversion of S into lowercase.
- Elimination of punctuation and extra spaces.
- Handling of usernames: Any username starting with '@' is replaced with '@user'.
- Handling of website links: Any website link is substituted with 'http'.
- If S contains any emoji convert that emoji to its corresponding text using the "emoji" package[1] of Python.

5.3.2 Tokenization

We convert each preprocessed sentence S into a sequence of tokens $[k_1, k_2, k_3, ..., k_n]$. Since sentence lengths vary, we standardize them by padding with zeros to achieve a fixed size. Consequently, every sentence S transforms into a token sequence $[k_1, k_2, k_3, ..., k_L]$ where $L = 300$.

5.3.3 Classification Task

For our dissimilar classification task, each tokenized sentence $X = [k_1, k_2, k_3, ..., k_L]$ consists of L words (tokens) where k_is are words (tokens) and $L = 300$. The sentence is associated with three labels:

- **Claim Label:** Indicates whether the sentence contains a claim (Yes/No).
- **Language Label:** Specifies the language of the sentence (one out of 22 options).
- **Sentiment Label:** Represents the sentiment conveyed by the sentence (positive/negative/neutral).

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

5.3.4 Modeling

Our primary objective is to predict the appropriate labels for each sentence using a single neural network. This network is trained to simultaneously handle the classification tasks of claim detection, language identification, and sentiment analysis, leveraging the shared knowledge and representations across these tasks for improved performance.

5.4 Implementation

5.4.1 Word Embedding

To embed words, we utilized pre-trained "GloVe"[100] word embeddings, which are vectors with a dimensionality of $D = 300$. Each token k_i from the sentence X was transformed into a sequence of vectors x_i of length D. Thus, from a tokenized sentence $X = [k_1, k_2, k_3, ..., k_L]$ we obtain $X_{L\times D} = [x_1, x_2, x_3, ..., x_L]$. These vectors were then input into a BiLSTM layer, as illustrated in Figure 5.1.

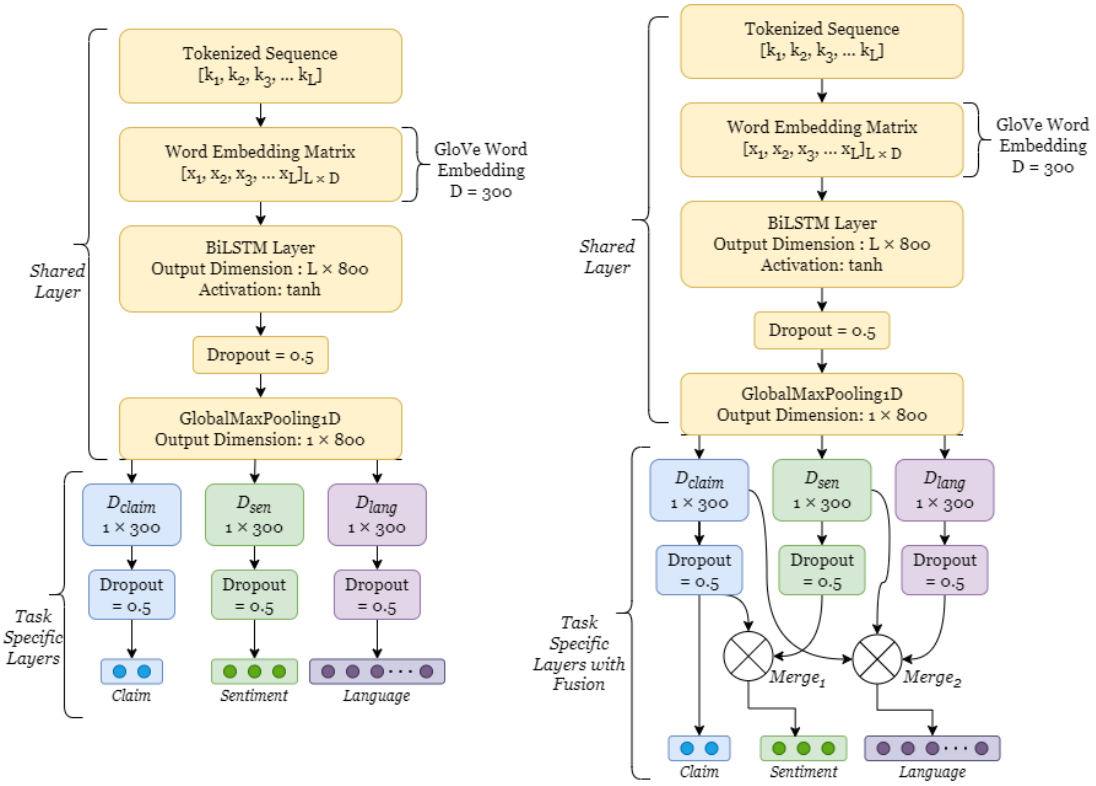


Figure 5.1: Proposed Model Architecture for Dissimilar Tasks: (a) MTL with shared backbone and long task-specific heads, (b) MTL with shared backbone, long task-specific head, and fusion.

5.4.2 BiLSTM Layer

The BiLSTM[101] is a variant of the recurrent neural network (RNN) architecture known for its bidirectional long-short-term memory. Traditional neural networks suffer from short-term memory issues and the vanishing gradient problem. LSTMs address these problems by effectively recognizing patterns, retaining crucial information, and mitigating the vanishing gradient issue. BiLSTMs, being more potent than unidirectional LSTMs, can analyze inputs from both the beginning and the end of a sequence. This bidirectional analysis enables them to utilize features from both past and future contexts, thereby enhancing language comprehension compared to unidirectional LSTMs.

5.4.3 GlobalMaxPooling Layer

We incorporated two BiLSTM layers followed by a dropout layer of 0.5 (as shown in Figure 5.1) and introduced a "GlobalMaxPooling" layer. The "GlobalMaxPooling1D" operation selects the maximum value from the hidden output vectors. If the output of the second dropout layer is represented as $[\hat{y_1}, \hat{y_2}, \hat{y_3}, ... , \hat{y_M}]_{L \times M}$ where $\hat{y_i}$ are vectors of length L and M is the number of hidden units of a BiLSTM layer, then it is processed according to the equation provided. Subsequently, the result Z_GlobalMaxPooling1D is fed into a dense layer consisting of 300 neurons.

$$Z_{GlobalMaxPooling1D} = [Max(\hat{y_1}), Max(\hat{y_2}), Max(\hat{y_3}), ..., Max(\hat{y_M})]_{[1 \times M]} \tag{5.1}$$

However, we conducted experiments without the "GlobalMaxPooling" layer to assess performance under different conditions. In the absence of the GlobalMaxPooling layer, the output of the 2nd BiLSTM layer is directly fed into a dense layer containing 300 neurons.

5.4.4 MTL with Task-Specific Long Heads

Figure 5.1(a) gives the architectural representation of the MTL with long task-specific heads or layers, where the output of the "GlobalMaxPooling" Layer is fed

into individual dense layers corresponding to each task.

$$D_{unit1} = ReLU(Z_{GlobalMaxPooling1D}) \tag{5.2}$$

$$D_{unit2} = Dropout(D_{unit1}) \tag{5.3}$$

$$D_{final} = ReLU(D_{unit2}) \tag{5.4}$$

5.4.5 Fusion with Task-Specific Dense Layer

Figure 5.1(b) gives the architectural representation of the MTL with a task-specific dense layer; where instead of using a single dense layer of 300 neurons, separate dense layers of 300 neurons are used for each task to perform the task-specific learning.

$$D_{*} = ReLU(Z_{GlobalMaxPooling1D}) \tag{5.5}$$

where D_{*} represents task-specific dense layer outputs for each task (D_{claim}, D_{sen}, D_{lang})[2] for our dissimilar tasks.

Now, the outputs of the task-specific dense layers are passed to dropout layers, and then we merge the outputs from the previous layers and feed them into the final task-specific dense layers as follows:

$$Merge_1 = Dropout(D_{claim}) \otimes Dropout(D_{sen}) \tag{5.6}$$

$$Dense_{sen} = ReLU(Merge_1) \tag{5.7}$$

and,

$$Merge_2 = D_{claim} \otimes D_{sen} \otimes Dropout(D_{lang}) \tag{5.8}$$

$$Dense_2 = ReLU(Merge_2) \tag{5.9}$$

where $\otimes$ represents the concatenation of the outputs of the dense or dropout layers.

5.4.6 Classification

For classification, we have used separate dense layers for each task. For MTL with long task-specific heads, the outputs of the individual dropout layers are fed into task-specific dense layers which use *softmax* as their activation function.

$$P_{*} = softmax(D_{*}) \tag{5.10}$$

$$\hat{Y}_{*} = \underset{\mathbf{j}}{argmax}(P_{*}) \tag{5.11}$$

[2] *claim, sen, lang* are the abbreviations of *claim, sentiment* and *language* respectively.

where P_* represents the probability value of each class of each classification task and $\hat{Y_*}$ represents the predicted class value.

For MTL with task-specific dense layers and fusion, the output of $Dropout_{claim}$ is fed as an input to the claim detection layer, fed the output of $Dense_1$ as an input to the sentiment classification layer, fed the output of $Dense_2$ as an input to language identification task layer.

$$P_{claim} = sigmoid(Dropout_{claim}) \tag{5.12}$$

$$P_{sen} = softmax(Dense_1) \tag{5.13}$$

$$P_{lang} = softmax(Dense_2) \tag{5.14}$$

and,

$$\hat{Y_*} = \underset{\mathbf{j}}{argmax} \begin{cases} P_{claim} \\ P_{sen} \\ P_{lang} \end{cases} \tag{5.15}$$

where P_{claim}, P_{sen}, P_{lang} represents the probability value of each class of claim, sentiment, and language classification tasks respectively and $\hat{Y_*}$ represents the predicted class value for each classification task.

5.4.7 Training

To train our proposed model, the final dataset is split out in an 80-20 ratio using the "scikit-learn"[3] 'train-test split' package, where 80% of the data was used to train the model (train split) and 20% of the data was used to test the model (test split).

For the multi-task loss function, the CrossEntropy loss is used for each of the tasks and monitors the loss for the test split of the dataset.

$$L_{total} = \sum_{i=1}^{3} L_i \tag{5.16}$$

where L_i is the loss for different tasks.

For training, the Adam optimizer [107] is used with a learning rate of 0.0005 and the number of epochs taken to train the model is 25, but we have used the EarlyStopping[4] method provided by TensorFlow's 'keras' module to reduce overfitting in our model. The parameters that are used to train the model are

[3]https://scikit-learn.org/

[4]https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping

given in Table 5.1.

Table 5.1: Parameters used to train the model.

Parameters	Value
Embedding	GloVe word embedding 300d
BiLSTM hidden units	$2 \times 400 = 800$
Dropout	0.5
Loss function	CrossEntropy
Optimizer	Adam
Learning rate	0.0005
Epoch	25
Batch size	64

5.5 Experimental Results

5.5.1 Experimental Setup

We use TensorFlow[5] and Keras[6] to implement our proposed model and use the Collaboratory[7] environment to execute the code for the proposed model. We have used the Precision (4.1), Recall (4.2), F1 Score (4.3) metrics, and confusion matrix for the validation split of our final dataset for evaluation.

5.5.2 Task Comparison

We evaluate our proposed model for all the possible combinations of tasks.

One Task Learning (1-TL): In 1-TL, the standalone classifiers for each sentiment, claim, and language classification are developed .

Two Task Learning (2-TL): Here all possible combination of two tasks i.e. sentiment + claim, language + claim, and sentiment + language are taken for classification.

Three Task Learning (3-TL): Here all three tasks are taken i.e., claim + sentiment + language.

For our dissimilar tasks, the performance of 2-TL and 3-TL is compared with the baseline classifier 1-TL. For dissimilar tasks in 1-TL and 2-TL, the simple MTL architecture (Figure 5.1(a)) is followed and for 3-TL both the simple MTL and MTL with fusion (Figure 5.1(b)) is followed. In Table 5.3 the performance

[5]https://www.tensorflow.org/
[6]https://keras.io/
[7]https://colab.research.google.com/

of all the combinations of MTL for dissimilar tasks is given and in Table 5.2, the performance of the standalone classifiers and 2 types of 3-TLs is given.

Table 5.2: Performance comparison of 1-TL vs 3-TL vs 3-TL$_{fusion}$.

	Task	Precision	Recall	F1-Score
1-TL	Claim	0.711	0.711	0.711
	Sentiment	0.66	0.669	0.664
	Language	0.788	0.738	0.723
3-TL	Claim	0.589	0.564	0.534
	Sentiment	0.597	0.528	0.55
	Language	0.762	0.705	0.715
3-TL$_{fusion}$	Claim	0.61	0.599	0.59
	Sentiment	0.63	0.566	0.586
	Language	0.722	0.706	0.707

Claim Detection: For claim detection task, the performance of 3-TL$_{fusion}$ model is better than 3-TL model. However, all multi-task models perform worse than the 1-TL model. The best precision, recall, and f1-score are 0.711, 0.711, and 0.711 respectively which all are achieved by the 1-TL model. Although it is noticeable that the claim detection task benefits when combined with the sentiment classification task as we can see in Table 5.3. The confusion matrices for claim detection of 1-TL, 3-TL, and 3-TL with fusion are given in Figure 5.2.

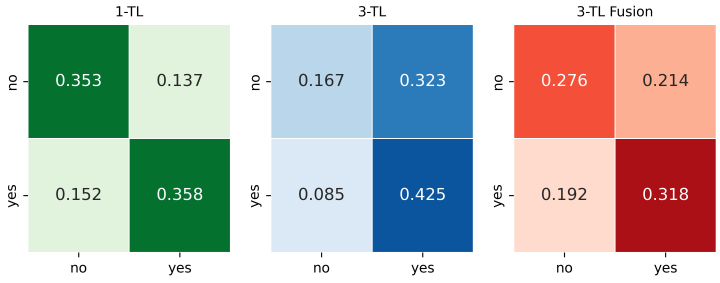


Figure 5.2: Confusion matrices for claim detection for 1-TL, 3-TL and 3-TL$_{fusion}$

Sentiment Classification: In the case of sentiment classification, the 3-TL$_{fusion}$ model performs better than 3-TL model. The best f1-score of 0.664 is achieved by the single task (1-TL) model and sentiment + claim combination of MTL. The best precision of 0.709 and recall of 0.642 is achieved by the sentiment + claim combination of MTL (2-TL). The confusion matrices for sentiment classification of 1-TL, 3-TL, and 3-TL with fusion are given in Figure 5.3.

Language Identification: For the language identification task, the single task model overperforms all other combinations of MTL. In this scenario, the performance in 3-TL is better than the 3-TL$_{fusion}$ model. The best precision,

Table 5.3: Performances of all combinations of MTLs.

	Claim			Sentiment			Language		
Task	P	R	F	P	R	F	P	R	F
claim	**0.71**	**0.71**	**0.71**	-	-	-	-	-	-
sen	-	-	-	0.66	**0.67**	**0.66**	-	-	-
lang	-	-	-	-	-	-	**0.79**	**0.74**	**0.72**
claim+sen	0.71	0.71	**0.71**	**0.71**	0.64	**0.66**	-	-	-
sen+lang	-	-	-	0.57	0.59	0.58	0.75	0.70	0.69
lang+claim	0.59	0.57	0.54	-	-	-	0.69	0.69	0.68
3-TL	0.59	0.56	0.53	0.60	0.52	0.55	0.76	0.71	0.72
3-TL$_{fusion}$	0.61	0.60	0.59	0.63	0.57	0.59	0.72	0.71	0.71

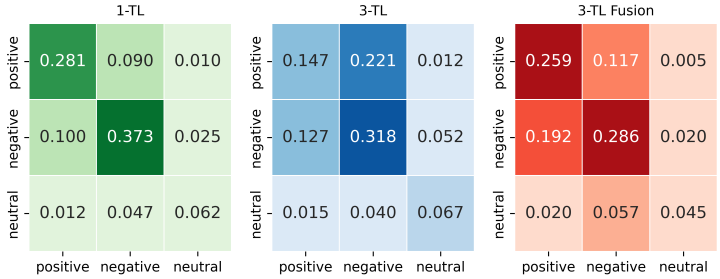


Figure 5.3: Confusion matrices for sentiment classification for 1-TL, 3-TL and 3-TL$_{fusion}$

recall, and f1-score are 0.788, 0.738, and 0.723 which are achieved by the 1-TL model. The confusion matrices for language identification of 1-TL, 3-TL, and 3-TL with fusion are given in Figure 5.4.

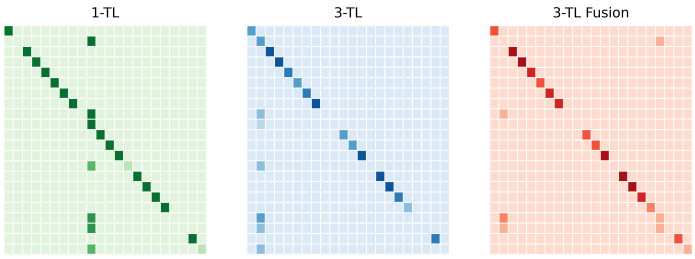


Figure 5.4: Confusion matrices for language identification for 1-TL, 3-TL and 3-TL$_{fusion}$

5.6 Error Analysis

In this section, the error analysis of our proposed MTL framework is discussed. This error analysis has been done only on the validation dataset and presented

a few of the examples in Tabular format. For simplicity purposes, only the error analysis between the baseline classifier, 3-TL, and 3-TL$_{fusion}$ is compared.

The error analysis for dissimilar tasks is given in Table 5.4. From Table 5.4, in the example S_1, it is seen that although claim and language labels are correctly assigned, the 3-TL model failed to correctly predict the positive sentiment of the sentence. This is quite justified according to the confusion matrix in Figure 5.3 that the model tends to predict negative for sentences having positive sentiment. For the example S_2, for claim detection, 1-TL and 3-TL$_{fusion}$ models failed to predict the claim correctly, but the 3-TL model did. This also follows the confusion matrix in Figure 5.2. For sentence S_3, only the 1-TL model correctly predicted the claim but the MTL models couldn't. The 3-TL model also incorrectly predicts it as a sentence in Dutch whereas it is an English sentence. Only the 1-TL model correctly predicted the language of the sentence for sentence S_4 which explains the confusion matrix in Figure 5.4. 1-TL and 3-TL$_{fusion}$ models predict this sentence as a sentence with negative sentiment despite this being a neutral sentence. This is also obvious according to the confusion matrix in Figure 5.3 which shows these models tend to predict neutral sentences as negative sentences.

Table 5.4: Error analysis between 1-TL, 3-TL and 3-TL$_{fusion}$

			Claim		Sentiment		Language	
id	Text	Task	True	Pred	True	Pred	True	Pred
S_1	I will admit it has less than the Sabbath albums before it, but it still very much holds onto the blues.	1-TL	yes	yes	pos	pos	eng	eng
		3-TL	yes	yes	pos	neg	eng	eng
		3-TL$_F$	yes	yes	pos	pos	eng	eng
S_2	Maybe I could do my own statistics.	1-TL	yes	no	neu	neu	eng	eng
		3-TL	yes	yes	neu	neu	eng	indo
		3-TL$_F$	yes	no	neu	neg	eng	eng
S_3	Have not got around to sorting out the history yet.	1-TL	no	no	neg	neg	eng	eng
		3-TL	no	yes	neg	neg	eng	dut
		3-TL$_F$	no	yes	neg	pos	eng	eng
S_4	müller mox figura centralis circulorum doctorum vindobonesium fiebat quibus intererant petrus	1-TL	no	no	neu	neg	lat	lat
		3-TL	no	yes	neu	neu	lat	por
		3-TL$_F$	no	no	neu	neg	lat	spa

5.7 Discussion

In this study, our main motive was to study the performance of our model for different dissimilar tasks and draw some insights from that. After all the experiments, there were a few noticeable points we delved deep into.

Firstly, it is observed that the performances of dissimilar tasks are not that good in our MTL setting. This is because in dissimilar tasks, the learning from one task cannot help other tasks properly and we see a performance drop in MTL for our dissimilar tasks.

Secondly, as already discussed in Section 5.4, for dissimilar tasks, the Long-Head approach has been used. The reason behind this is the simple fact that the dissimilar tasks have very few things in common among them and each task needs extra standalone attention. For this reason, for our dissimilar tasks, we have used more layers in the individual task-specific layers.

Thirdly, it can be seen that when the experiments are carried out in the infused architectures as shown in Figure 5.1(b), the performances are better in most of the cases. After seeing these statistics it can be said that learnings of one task help the other tasks in an MTL architecture.

Finally, our work has some limitations. Firstly, the size of our final dataset is small, so there needs to be a large size dataset to validate this model. Secondly, there is a high level of imbalance in our final dataset, so there may be a performance bias in the model for those tasks. Thirdly, we have not explored the current state-of-the-art transformer-based models such as "BERT". Our future work includes trying to balance the existing imbalance classes, add more data to our existing dataset, add other similar types of classification tasks, and explore transformer-based architectures.

CHAPTER 6

Span Identification

In this chapter, we will explore Span Identification tasks in depth. Our investigation encompassed not only English but also underrepresented languages such as Hindi and CodeMix, particularly within the Indian cultural framework. We concentrated on identifying spans related to claims and sentiments, underscoring their increasing relevance in business contexts. This approach, which integrates multiple languages and cultural settings, underscores the critical need for Natural Language Processing (NLP) models capable of performing efficiently across diverse linguistic and cultural landscapes.

6.1 Claim Span Identification

To elaborate, Span Identification involves detecting specific segments within a text that correspond to particular types of information, such as claims or sentiments. Our research ventured into less-explored languages, recognizing the necessity to bridge the gap in NLP resources and tools for these languages. By incorporating Hindi and CodeMix, which is a blend of English and other languages commonly used in informal communication in India, we aimed to create models that are more inclusive and capable of understanding a wider range of linguistic nuances.

In the context of claim span identification, we focused on pinpointing exact phrases that make assertions or statements, which is crucial for applications like fact-checking, news summarization, and information extraction.

Our multilingual and multi-context methodology not only addresses linguistic diversity but also adapts to the cultural variations that influence language use. This comprehensive approach ensures that NLP models are robust and versatile, capable of delivering accurate results in different languages and contexts. As businesses continue to expand globally, the ability to analyze and interpret text across

various languages and cultures becomes increasingly vital, making our research an essential contribution to the field of NLP.

6.1.1 Problem Statement

The rapid rise of social media has boosted content creation and information sharing, benefiting many. However, these platforms also harbor those spreading rumors, fake news, propaganda, and misinformation. Claims shape our worldview but are often misused by fake news spreaders to deceive. The COVID-19 Infodemic exemplifies this, with false information about politics, social issues, and medical claims spreading widely. Social media companies employ content moderators to combat this, but the high volume of information makes it hard to spot fake news effectively. Consequently, developing methods to identify and verify social media claims automatically has become a critical NLP research challenge.

Most current claim detection research concentrates on identifying claims at the sentence level, neglecting the precise details of exact claim spans. Recently, advancements in this field have shifted from broad sentence-level models to more detailed identification of specific claim spans. After determining if a post contains a claim, the next step is to pinpoint the exact span of the claim within the post. Accurately identifying the specific words or phrases forming the claim is essential, as this precision is crucial for fact-checking accuracy.

6.1.2 Proposed Approach

Initially, we gathered a dataset consisting of approximately 750 training samples in English, Hindi, Bengali, and CodeMixed languages. The input for each sample is a text document, and the output is a string or an array of strings representing the claim span(s). We divided this data into training and testing subsets. For our task, we employed transfer learning techniques using XLM-RoBERTa, MuRIL, and BERT Multilingual models. We began by tokenizing the sentences. Following tokenization, each token was labeled with B-I-O tags. These tagged tokens were then fed into the model, which produced the predicted claim span(s).

6.1.3 Implementation

6.1.3.1 Tokenization

Tokenization involves breaking down a text into smaller units known as tokens. In this study, we explored three primary approaches:

- **XLM-RoBERTa (Robustly Optimized BERT Pretraining Approach):** This model leverages a robust pretraining method to enhance BERT's performance across various languages.
- **MuRIL (Multilingual Representations for Indian Languages):** Specifically designed to handle the intricacies of Indian languages, this model offers multilingual capabilities.
- **BERT (Bidirectional Encoder Representations from Transformers) Multilingual:** A model that captures context from both directions in a sentence, enhancing its ability to understand multilingual text.

For each of these models, we utilized distinct pre-trained tokenizers tailored to their respective architectures.

6.1.3.2 B-I-O Tagging

After tokenization, each token must be assigned a B-I-O tag, where B stands for the Beginning of a Claim, I indicates the Inside of a Claim, and O signifies the Outside of a Claim. The use of return_offsets_mapping=True in the tokenizer configuration allows us to retrieve the start index and the end index (plus one) for each token within the original text.

Additionally, the start and end indices of each claim span within the original text are calculated and recorded. This enables us to determine which tokens correspond to which parts of the claim. When the offset mapping of a token falls within the range of the start and end indices of a claim span, the appropriate B-I-O tagging is applied to that token. This process ensures that each token is accurately labeled according to its position within or outside the claim spans.

6.1.3.3 Model Architecture

To accomplish our task, we utilized transfer learning methods with models such as XLM-RoBERTa, MuRIL, and BERT Multilingual. Initially, we performed tokenization on the sentences. After this step, each token was annotated with B-I-O tags. The annotated tokens were subsequently input into the model, which generated the predicted claim span(s). Figure 6.1 shows an abstract overview of our model.

6.1.4 Experimental Results

Among the three models, XLM-RoBERTa demonstrated the best performance. The other models also yielded good results. We evaluated their performance us-

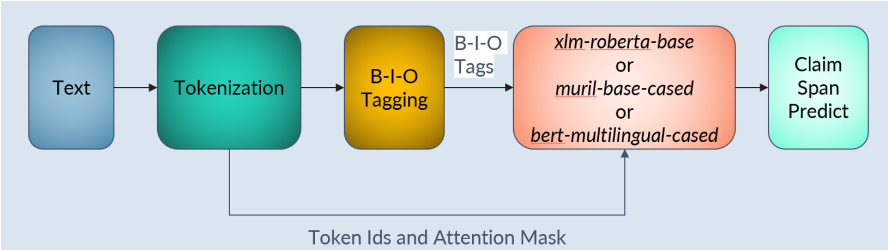


Figure 6.1: Architecture of Claim Span Identification Model

ing Accuracy and F1-Score metrics. After 4 epochs, XLM-RoBERTa reached an accuracy of 81% and an F1-Score of 54%. The performance metrics for the other models are detailed in Table 6.1.

Epoch	xlm-roberta-base		muril-base-cased		bert-base-multilingual-cased	
	Acc.	F1-Score	Acc.	F1-Score	Acc.	F1-Score
1	0.75	0.50	0.50	0.26	0.78	0.52
2	0.80	0.54	0.72	0.48	0.79	0.53
3	0.81	0.54	0.74	0.50	0.80	0.53
4	**0.81**	**0.54**	0.75	0.50	0.80	0.53

Table 6.1: Results on Test Split of the Dataset

6.1.5 Error Analysis

Although the evaluation metrics indicate that XLM-RoBERTa performs the best overall, this is not always consistent. Table 6.2 presents three examples to highlight the strengths and weaknesses of our models. For the first sentence, XLM-RoBERTa achieved perfect results, whereas MuRIL and BERT-Multilingual included a few extra words at the beginning. In the second sentence, all models performed poorly, capturing more words than the actual claim span. For the third sentence, the BERT-Multilingual model performed the best, accurately identifying the claim span, while XLM-RoBERTa and MuRIL captured more than the necessary span.

6.1.6 Discussion

The dataset we utilized is relatively small, containing only about 750 samples. To thoroughly validate the overall performance of our models, a larger dataset is necessary. Additionally, we have not explored other potentially effective models such as GPT or BERT-large. Further, we need to perform more hyperparameter tuning to enhance the models' performance. Our current training is based solely on social media data; in the future, we plan to extend our training to other types of

Original Span	**xlm-roberta**	**muril**	**bert-multilingual**
'Under which provision you got re elected as RS member inspite of getting defeated in #BengalElection2021	'Under which provision you got re elected as RS member inspite of getting defeated in #BengalElection2021	'@ swapan55 Under which provision you got re elected as RS member inspite of getting defeated in	'@', 'Under which provision you got re elected as RS member inspite of getting defeated in # Bengal-Election2021'
jankibaat1 and Pardip would continue to bark for the next 6 months to make that a reality!	First BJPee would establish a fake theory! Then Low Level Dallals, @jankibaat1 and Pardip would continue to bark for the next 6 months to make that a reality	First BJPee would establish a fake theory! Then Low Level Dallals, @ jankibaat1 and Pardip would continue to bark for the next 6 months to make that a reality	First BJPee would establish a fake theory! Then Low Level Dallals, @ jankibaat1 and Pardip would continue to bark for the next 6 months to make that a reality!
they have spent huge amount in #Bengal-Election2021 for BJP	not expect that #JIO will solve your problems', 'they are making fool to the customers as they have spent huge amount in #Bengal-Election2021 for BJP	##jio Do not expect that # JIO will solve your problems.. they are making fool to the customers as they have spent huge amount in	have spent huge amount in # Bengal-Election2021 for BJP

Table 6.2: Error Analysis of Claim Span Identification Task

texts, such as news articles and online blogs, to evaluate and improve the models' versatility and robustness across various domains.

6.2 Sentiment Span Identification

Span Identification involves pinpointing specific sections within a text that match particular types of information, such as claims or sentiments. Our research delved into under-explored languages, addressing the need to enhance NLP resources and tools for these languages. By incorporating Hindi and CodeMixa blend of English and other languages frequently used in informal communication in Indiawe aimed to develop models that are more inclusive and adept at understanding a broader spectrum of linguistic nuances.

Sentiment span identification entails identifying parts of the text that convey emotions or opinions, which is particularly valuable for sentiment analysis in areas like customer feedback, social media monitoring, and market research.

6.2.1 Problem Statement

As previously mentioned, the rise of social media has highlighted the need for claim span identification. Similarly, businesses now also require Sentiment Span Identification. With countless tweets posted every second, it's challenging

to determine whether the sentiment of a tweet will positively impact a brand by going viral or harm profits due to its negative tone. Accurately capturing sentiment in language is crucial in an era where decisions and reactions are made rapidly. However, identifying the specific words that convey this sentiment is a complex issue.

Businesses need to understand which parts of a tweet contribute to its overall sentiment to respond appropriately and protect their brand's reputation. Positive sentiments can enhance brand image and increase engagement, while negative sentiments can lead to crises that require swift management. The challenge lies in pinpointing the exact words or phrases within tweets that drive these sentiments.

To address this, machine learning tools can be employed to analyze large volumes of social media data. These tools can help identify the specific words or phrases that indicate a positive, negative, or neutral sentiment. By doing so, companies can gain deeper insights into public perception and tailor their responses more effectively. Developing robust algorithms for sentiment span identification not only aids in understanding social media dynamics but also empowers businesses to make informed decisions in real time.

6.2.2 Proposed Approach

For this task, we will employ SpanBERT[1], a specialized version of BERT designed for span prediction, as our transformer model. After gathering our dataset and dividing it into training and testing sets, we proceed with tokenizing the text. To tackle the problem, we can frame the tweet as the context, the sentiment as the question, and the selected portion of the tweet as the answer. We then create an object for the input data, which involves tokenizing both the text and the selected text, identifying the indices of the selected text within the main text, and preparing the data for input into SpanBERT. We apply the same process to the test dataset. Next, we build the model using the SpanBERT transformer, adding additional layers on top. The model outputs the start and end indices of the sentiment span.

6.2.3 Implementation

6.2.3.1 Data Preprocessing

We have observed that for neutral tweets, the "selected text" usually corresponds to the entire tweet. Based on this observation, we will proceed with the assumption that for neutral tweets, the selected text is always the entire tweet.

[1]https://github.com/facebookresearch/SpanBERT

Therefore, we will focus on predicting the selected text only for tweets with positive or negative sentiments. This approach simplifies the task by narrowing down the scope of predictions to just those tweets that express a clear sentiment.

6.2.3.2 Tokenization

We'll break down the text into tokens, effectively converting it into a sequence of numerical representations. This process, known as tokenization, assigns a unique numerical value to each token, enabling the text to be processed by machine learning algorithms. For this task, we will use the tokenizer provided by SpanBERT, utilizing the BertTokenizerFast's pretrained function.

6.2.3.3 Input Preparation

We took our inspiration from the example provided on the Keras website[2] that deals with the SQUAD dataset, which is renowned for text extraction tasks. In this dataset, each instance contains a "context" (typically a paragraph), a "question," and an "answer" (a segment of text extracted from the context). In this scenario, the tweet can be regarded as the context, the sentiment as the question, and the selected text of the tweet as the answer. An object is instantiated to handle the input data, which involves tokenizing both the text and the selected text, identifying the indices of the selected text within the text, and preparing the data for input into the model. The same process is applied to the testing data, which comprises only the text and the sentiment of the tweet.

6.2.3.4 Creating and Training the Model

The process begins by loading a pre-trained SpanBERT model from a specified directory. It defines three types of input data: tokenized text, token types, and attention masks, each with a predefined shape. These inputs are processed by the SpanBERT model to produce embeddings, which are rich numerical representations of the input data. The embeddings are further processed to predict the start and end positions of an answer within the text. This involves applying dropout for regularization, followed by convolutional layers and activation functions to convert the embeddings into logits, which are raw prediction scores. The logits are then flattened and transformed into probability distributions using the softmax activation function, providing the probabilities of each position being the start or end of the answer. Figure 6.2 gives a brief overview of our model architecture. A Keras model is then created with these inputs and outputs. The model is compiled using a Sparse Categorical Crossentropy loss function and the Adam optimizer with a

[2]https://keras.io/examples/nlp/text_extraction_with_bert/

learning rate of 5e-5, preparing it for training. Table 6.3 displays the parameters chosen for this task.

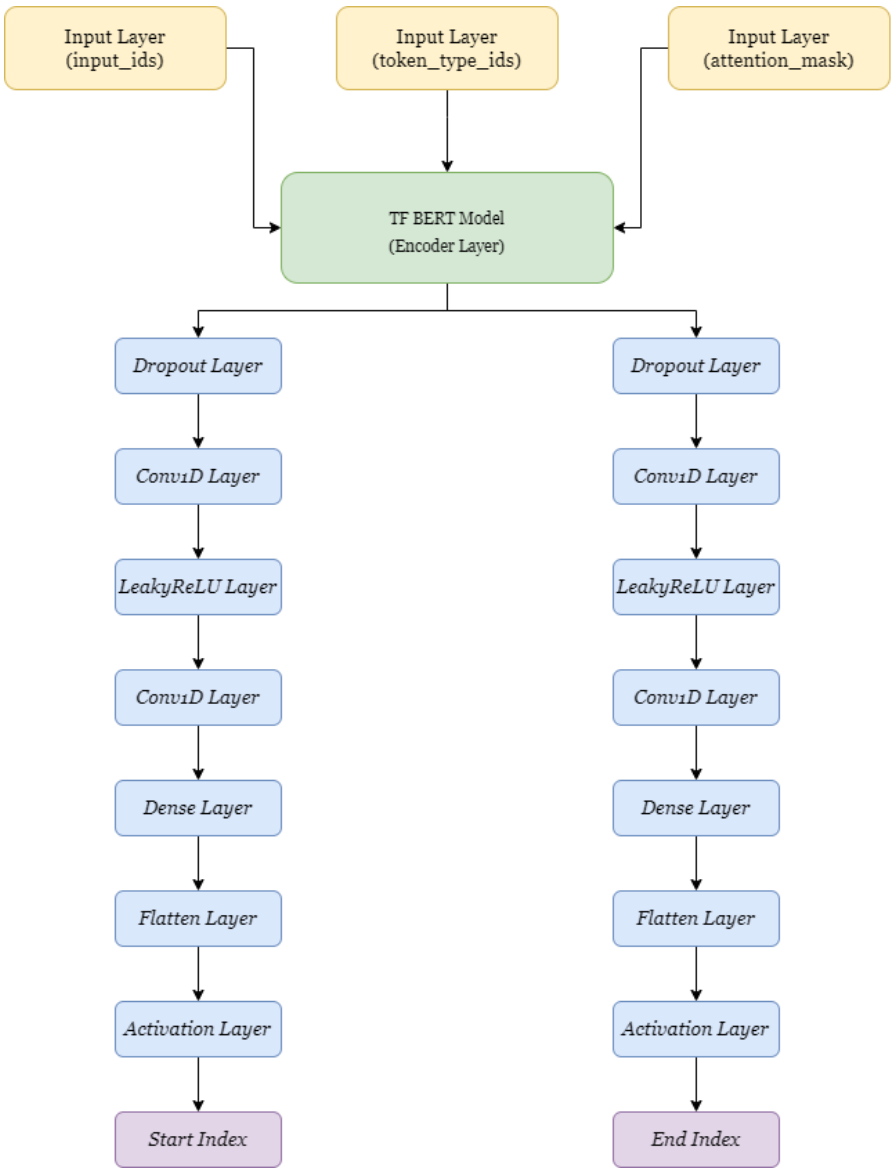


Figure 6.2: Architecture of Sentiment Span Identification Model

6.2.3.5 Fitting the Model

Next, the model undergoes training using the processed input and output data through the fit() function. Training spans across 5 epochs, with a batch size set to 8. Throughout the training process, validation data and labels are employed for validation purposes, ensuring the model's performance is assessed and refined as it learns. Table 6.3 shows the details of hyper-parameters used during training.

Table 6.3: Parameters used to train the model.

Parameters	Value
Dropout	0.3
Loss function	SparseCategoricalCrossentropy
Optimizer	Adam
Learning rate	5e-5
Epoch	5
Batch size	8

6.2.3.6 Prediction using the Model

Every Tweet object includes the character-level offsets for every token within its input text. We utilize these offsets to retrieve the text span that aligns with the tokens positioned between our predicted start and end tokens. Additionally, all the actual answers are available within each Tweet object. We compute the percentage of data points where the text span derived from model predictions aligns with one of the actual answers, serving as a measure of the model's accuracy.

6.2.4 Experimental Results and Error Analysis

Following each epoch, we assessed the exact match score. Upon completing 5 epochs, our model achieved an accuracy of 0.55. The exact match scores for each epoch are detailed in Table 6.4 for reference.

Epoch	Exact Match Score
1	0.11
2	0.27
3	0.39
4	0.57
5	0.55

Table 6.4: Exact match score of Sentiment Span Identification Model

Table 6.5 illustrates several instances where our model made incorrect predictions. This analysis provides insights into the limitations of our model. In the Sentence S_1, our model's prediction was partially incorrect, which is understandable given the difficulty of accurately predicting such cases where only a few words differ. Similarly, in the Sentence S_2, our model predicted one or two extra words as part of the span, demonstrating a common challenge in prediction tasks. Interestingly, the Sentence S_3 was a neutral sentence, where most instances in our dataset have the entire sentence as the span. However, our model may have misinterpreted it as a negative sentence and made predictions accordingly.

ID	Original Sentence	Original Span	Predicted Span
S_1	I really really like the song Love Story by Taylor Swift	really like	like
S_2	i want to go to music tonight but i lost my voice.	lost	lost my voice
S_3	oh Marly, I'm so sorry!! I hope you find her soon!! <3 <3	oh Marly, I'm so sorry!! I hope you find her soon!! <3 <3	so sorry

Table 6.5: Error Analysis of Sentiment Span Identification Task

6.2.5 Discussion

In the sentiment span identification task, our model demonstrated a moderate level of accuracy, with an exact match score of 0.55 after five epochs. Despite this, several shortcomings were observed. The model struggled with partial matches, often predicting slightly more or fewer words than the actual sentiment span, particularly in complex sentences where the sentiment is not clearly delineated. Additionally, the model occasionally misclassified neutral sentences, indicating a possible or negative bias or limitation in distinguishing between sentiment types. These errors highlight the need for a larger and more diverse dataset to better train and validate the model.

Future work should focus on expanding the dataset to include a wider range of sentence structures and sentiment expressions. Incorporating more advanced models, such as GPT or BERT-large, could also enhance performance. Furthermore, extensive hyperparameter tuning and exploring different preprocessing techniques could help refine the models accuracy. Applying the model to other types of text, such as news articles or online blogs, could also test its versatility and improve its robustness across various contexts. Addressing these areas will be crucial for developing a more accurate and reliable sentiment span identification system.

CHAPTER 7

Conclusion

In this thesis, we have delved into three critical areas of Natural Language Processing (NLP): Claim Detection, Multi-task Learning, and Span Identification. Our comprehensive exploration and experimentation demonstrate the considerable potential of machine learning (ML) and deep learning (DL) techniques in tackling the unique challenges posed by multilingual, informal, and dynamic social media content.

Social media, with its evolving and diverse nature, necessitates robust and adaptable NLP models. Our research underscores the importance of these models in maintaining the relevance and accuracy of claim-related tasks, which are crucial for various business applications. Specifically, we addressed claim checkworthiness and classification, recognizing that a single tweet can encompass multiple types of claims. To this end, we developed a multi-label classifier capable of handling such complexity effectively.

In the realm of Multi-task Learning, our work extends beyond the prevalent focus on homogeneous task environments. By exploring multi-task learning with dissimilar tasks, we identified the challenges and potential solutions, such as the long-head approach and layer fusion, to enhance performance. This exploration is vital as it broadens the understanding and applicability of multi-task learning in more varied and realistic settings.

For Span Identification, our research ventured into less-explored languages like Hindi and CodeMix, alongside English, with a particular focus on the Indian context. We tackled both claim and sentiment span identification, reflecting their growing significance in business applications. This multilingual and multi-context approach highlights the necessity for NLP models that can operate effectively across different languages and cultural contexts.

Throughout our research, we employed a variety of evaluation metrics, includ-

ing accuracy, precision, recall, F1-score, and confusion matrices, to thoroughly assess the performance of our models. These metrics provided a comprehensive understanding of the strengths and areas for improvement in our approaches.

In conclusion, our thesis contributes to the field of NLP by addressing significant gaps and challenges in claim detection, multi-task learning, and span identification. As social media continues to evolve, the development of adaptable and robust NLP models remains crucial. Our findings not only advance the current state of research but also lay the groundwork for future studies aimed at enhancing the capabilities of NLP systems in increasingly complex and diverse environments.

7.1 Future Work

While this thesis makes significant contributions to the field of Natural Language Processing in multilingual social media text, there are numerous opportunities for future research and enhancement:

- **Multimodal Integration:** Expand the research to include image and video data from tweets, as these modalities can provide additional context and valuable information for various NLP tasks.
- **Cross-language Models:** Develop models that can handle claim detection, multi-task learning, and span identification across multiple languages without requiring language-specific training.
- **Real-time Processing:** Modify the models to enable real-time claim detection or span identification, addressing the dynamic nature of social media and the necessity for swift responses to emerging claims.
- **Detailed Classification:** Investigate fine-grained claim detection to differentiate between various types of claims, such as factual, opinion-based, and controversial statements.
- **User and Contextual Analysis:** Incorporate user profiles and contextual information to enhance classification accuracy by considering user credibility and historical tweet behavior.
- **Ethical and Bias Assessment:** Assess the ethical implications and potential biases in automated task completion, and develop strategies to mitigate biases and ensure fairness.
- **Scalability and Generalizability:** Test the scalability of the models to manage large-scale social media data and generalize the approaches to other languages and platforms.

- **Class Imbalance Resolution:** Implement strategies to address and rectify the class imbalances within the dataset used for multi-task learning, ensuring a more balanced representation of different classes.
- **Transformer-based Architectures:** In future work, we want to investigate and incorporate transformer-based architectures to leverage their advanced capabilities for improved performance in multi-task learning scenarios.
- **Dataset Expansion and Additional Tasks:** Increase the size and diversity of the dataset to enhance the model's ability to generalize and perform effectively across a wider range of inputs. Introduce new, similar classification tasks to the model, broadening its scope and improving its versatility in handling various types of data.
- **Diversified Training Data:** Expand the training data beyond social media to include other types of texts, such as news articles and online blogs, to enhance the model's applicability and robustness across different domains.

List of Publications

Published

1. **S. S. Das**, S. Choudhury, P. Saha, and D. Das, VaxTweetClassifier: BERT for Dealing with Vaccination Tweets, 2021. Forum for Information Retrieval Evaluation, December 15-18, 2023, India.

Accepted

- S. Choudhury, P. Saha, S. Ray, **S. S. Das**, and D. Das, "AlphaIntellect at SemEval-2024 Task 6: Detection of Hallucinations in Generated Text", 2024.
- P. Pal, **S. S. Das**, D. Das, and A. K. Kalya, "Decoding Multi-Task Learning: A Comparative Exploration for Similar and Dissimilar Task", 2024.

Communicated (under review)

1. P. Pal, **S. S. Das**, D. Das, and A. K. Kalya, "Does Multi-task Learning with Infusion perform better than Simple Multi-task Learning? A Comparative Exploration using Similar and Dissimilar Tasks", 2024.

Bibliography

[1] P. Chowdhary, "Natural language processing," in Apr. 2020, pp. 603–649, ISBN: 978-81-322-3970-3. DOI: 10.1007/978-81-322-3972-7_19.

[2] R. Lee, "Part-of-speech (pos) tagging," in Nov. 2023, pp. 43–66, ISBN: 978-981-99-1998-7. DOI: 10.1007/978-981-99-1999-4_3.

[3] L. S. A. Alireza Mansouri and A. Mamat, "Named entity recognition approaches," *International Journal of Computer Science and Network Security*, vol. 8, pp. 339–334, 2008.

[4] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, 2019, ISSN: 2078-2489. DOI: 10.3390/info10040150.

[5] R. Gangi Reddy, S. C. Chinthakindi, Z. Wang, *et al.*, "NewsClaims: A new benchmark for claim detection from news with attribute knowledge," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 6002–6018. DOI: 10.18653/v1/2022.emnlp-main.403.

[6] D. Pavllo, T. Piccardi, and R. West, *Quootstrap: Scalable unsupervised extraction of quotation-speaker pairs from large news corpora via bootstrapping*, 2018.

[7] P. Smeros, C. Castillo, and K. Aberer, "Scilens: Evaluating the quality of scientific news articles using social media and scientific literature indicators," in *The World Wide Web Conference*, ser. WWW '19, San Francisco, CA, USA: Association for Computing Machinery, 2019, pp. 1747–1758, ISBN: 9781450366748. DOI: 10.1145/3308558.3313657.

[8] R. Levy, Y. Bilu, D. Hershcovich, E. Aharoni, and N. Slonim, "Context dependent claim detection," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, J.

Tsujii and J. Hajic, Eds., Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 1489–1500.

[9] C. Stab, T. Miller, B. Schiller, P. Rai, and I. Gurevych, "Cross-topic argument mining from heterogeneous sources," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds., Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 3664–3674. DOI: 10.18653/v1/D18-1402.

[10] 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*, ser. KDD '17, Halifax, NS, Canada: Association for Computing Machinery, 2017, pp. 1803–1812, ISBN: 9781450348874. DOI: 10.1145/3097983.3098131.

[11] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum, "Where the truth lies: Explaining the credibility of emerging claims on the web and social media," in *Proceedings of the 26th International Conference on World Wide Web Companion*, ser. WWW '17 Companion, Perth, Australia: International World Wide Web Conferences Steering Committee, 2017, pp. 1003–1012, ISBN: 9781450349147. DOI: 10.1145/3041021.3055133.

[12] D. Zlatkova, P. Nakov, and I. Koychev, "Fact-checking meets fauxtography: Verifying claims about images," 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)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2099–2108. DOI: 10.18653/v1/D19-1216.

[13] G. Karagiannis, M. Saeed, P. Papotti, and I. Trummer, *Scrutinizer: A mixed-initiative approach to large-scale, data-driven claim verification*, 2020.

[14] P. Smeros, C. Castillo, and K. Aberer, "Sciclops: Detecting and contextualizing scientific claims for assisting manual fact-checking," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, ser. CIKM '21, Virtual Event, Queensland, Australia: Association for Computing Machinery, 2021, pp. 1692–1702, ISBN: 9781450384469. DOI: 10.1145/3459637.3482475.

[15] J. Vladika and F. Matthes, *Scientific fact-checking: A survey of resources and approaches*, 2023.

[16] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Comput. Surv.*, vol. 53, no. 5, 2020, ISSN: 0360-0300. DOI: 10.1145/3395046.

[17] G. D. S. Martino, S. Cresci, A. Barron-Cedeno, S. Yu, R. D. Pietro, and P. Nakov, *A survey on computational propaganda detection*, 2020.

[18] T. Bian, X. Xiao, T. Xu, *et al.*, "Rumor detection on social media with bi-directional graph convolutional networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 549–556, 2020. DOI: 10.1609/aaai.v34i01.5393.

[19] M. Hardalov, A. Arora, P. Nakov, and I. Augenstein, "A survey on stance detection for mis- and disinformation identification," in *Findings of the Association for Computational Linguistics: NAACL 2022*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds., Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 1259–1277. DOI: 10.18653/v1/2022.findings-naacl.94.

[20] G. K. Shahi and D. Nandini, "Fakecovid - a multilingual cross-domain fact check news dataset for covid-19," *ArXiv*, vol. abs/2006.11343, 2020.

[21] T. Hossain, R. L. Logan IV, A. Ugarte, Y. Matsubara, S. Young, and S. Singh, "COVIDLies: Detecting COVID-19 misinformation on social media," in *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, K. Verspoor, K. B. Cohen, M. Conway, *et al.*, Eds., Online: Association for Computational Linguistics, Dec. 2020. DOI: 10.18653/v1/2020.nlpcovid19-2.11.

[22] D. Antypas, J. Camacho-Collados, A. Preece, and D. Rogers, "COVID-19 and misinformation: A large-scale lexical analysis on Twitter," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, J. Kabbara, H. Lin, A. Paullada, and J. Vamvas, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 119–126. DOI: 10.18653/v1/2021.acl-srw.13.

[23] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 7370–7377, 2019. DOI: 10.1609/aaai.v33i01.33017370.

[24] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," *Proc. VLDB Endow.*, vol. 2, no. 1, pp. 718–729, 2009, ISSN: 2150-8097. DOI: 10.14778/1687627.1687709.

[25] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017.

[26] N. Reimers, B. Schiller, T. Beck, J. Daxenberger, C. Stab, and I. Gurevych, "Classification and clustering of arguments with contextualized word embeddings," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds., Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 567–578. DOI: 10.18653/v1/P19-1054.

[27] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017. DOI: 10.1609/aaai.v31i1.10488.

[28] C. T. Duong, Q. V. H. Nguyen, and K. Aberer, "Interpretable node embeddings with mincut loss," en, in *Learning and Reasoning with Graph-Structured Representations Workshop - ICML 2019*, 2019.

[29] S. Yuan and B. Yu, "Hclaime: A tool for identifying health claims in health news headlines," *Information Processing & Management*, vol. 56, no. 4, pp. 1220–1233, 2019, ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2019.03.001.

[30] A. Wührl and R. Klinger, "Claim detection in biomedical Twitter posts," in *Proceedings of the 20th Workshop on Biomedical Language Processing*, D. Demner-Fushman, K. B. Cohen, S. Ananiadou, and J. Tsujii, Eds., Online: Association for Computational Linguistics, Jun. 2021, pp. 131–142. DOI: 10.18653/v1/2021.bionlp-1.15.

[31] P. Nakov, A. Barrón-Cedeño, G. da San Martino, *et al.*, "Overview of the clef–2022 checkthat! lab on fighting the covid-19 infodemic and fake news detection," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, A. Barrón-Cedeño, G. Da San Martino, M. Degli Esposti, *et al.*, Eds., Cham: Springer International Publishing, 2022, pp. 495–520, ISBN: 978-3-031-13643-6.

[32] C. Zuo, K. Mathur, D. Kela, N. Faramarzi, and R. Banerjee, "Beyond belief: A cross-genre study on perception and validation of health information online," *International Journal of Data Science and Analytics*, vol. 13, pp. 1–16, May 2022. DOI: 10.1007/s41060-022-00310-7.

[33] C. Stahlhut, "Interactive evidence detection," en, Ph.D. dissertation, Technische Universität Darmstadt, Darmstadt, 2021, x, 157 Seiten. DOI: https://doi.org/10.26083/tuprints-00019154.

[34] C.-H. Wei, A. Allot, R. Leaman, and Z. Lu, "PubTator central: automated concept annotation for biomedical full text articles," *Nucleic Acids Research*, vol. 47, no. W1, W587–W593, May 2019, ISSN: 0305-1048. DOI: 10.1093/nar/gkz389.

[35] H.-M. Müller, K. V. Auken, Y. Li, and P. W. Sternberg, "Textpresso central: A customizable platform for searching, text mining, viewing, and curating biomedical literature," *BMC Bioinformatics*, vol. 19, 2018.

[36] A. Allot, Q. Chen, S. Kim, *et al.*, "LitSense: making sense of biomedical literature at sentence level," *Nucleic Acids Research*, vol. 47, no. W1, W594–W599, Apr. 2019, ISSN: 0305-1048. DOI: 10.1093/nar/gkz289.

[37] X. Wang, Y. Guan, W. Liu, *et al.*, "EVIDENCEMINER: Textual evidence discovery for life sciences," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, A. Celikyilmaz and T.-H. Wen, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 56–62. DOI: 10.18653/v1/2020.acl-demos.8.

[38] C. Macdonald, R. L. Santos, I. Ounis, and I. Soboroff, "Blog track research at trec," *SIGIR Forum*, vol. 44, no. 1, pp. 58–75, Aug. 2010, ISSN: 0163-5840. DOI: 10.1145/1842890.1842899.

[39] W. Zhang, C. Yu, and W. Meng, "Opinion retrieval from blogs," in *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, ser. CIKM '07, Lisbon, Portugal: Association for Computing Machinery, 2007, pp. 831–840, ISBN: 9781595938039. DOI: 10.1145/1321440.1321555.

[40] R. Mochales and M.-F. Moens, "Argumentation mining: The detection, classification and structure of arguments in text," Jan. 2009, pp. 98–107. DOI: 10.1145/1568234.1568246.

[41] E. Cabrio and S. Villata, "Combining textual entailment and argumentation theory for supporting online debates interactions," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, H. Li, C.-Y. Lin, M. Osborne, G. G. Lee, and J. C. Park, Eds., Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 208–212.

[42] Y. Guo, A. Korhonen, and T. Poibeau, "A weakly-supervised approach to argumentative zoning of scientific documents," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, R. Barzilay and M. Johnson, Eds., Edinburgh, Scotland, UK.: Association for Computational Linguistics, Jul. 2011, pp. 273–283.

[43] S. Teufel *et al.*, "Argumentative zoning: Information extraction from scientific text," Ph.D. dissertation, Citeseer.

[44] I. Dagan, B. Dolan, B. Magnini, and D. Roth, "Recognizing textual entailment: Rational, evaluation and approaches," English (US), *Natural Language Engineering*, vol. 15, no. 4, pp. i–xvii, Oct. 2009, ISSN: 1351-3249. DOI: 10.1017/S1351324909990209.

[45] R. Levy, Y. Bilu, D. Hershcovich, E. Aharoni, and N. Slonim, "Context dependent claim detection," in *International Conference on Computational Linguistics*, 2014.

[46] M. Lippi and P. Torroni, "Context-independent claim detection for argument mining," in *Proceedings of the 24th International Conference on Artificial Intelligence*, ser. IJCAI'15, Buenos Aires, Argentina: AAAI Press, 2015, pp. 185–191, ISBN: 9781577357384.

[47] A. Romanov and C. Shivade, "Lessons from natural language inference in the clinical domain," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds., Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 1586–1596. DOI: 10.18653/v1/D18-1187.

[48] M. Sadat and C. Caragea, "SciNLI: A corpus for natural language inference on scientific text," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 7399–7409. DOI: 10.18653/v1/2022.acl-long.511.

[49] J. Vladika and F. Matthes, *Sebis at semeval-2023 task 7: A joint system for natural language inference and evidence retrieval from clinical trial reports*, 2023.

[50] V. Karpukhin, B. Oguz, S. Min, *et al.*, "Dense passage retrieval for open-domain question answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational

Linguistics, Nov. 2020, pp. 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550.

[51] G. Tsatsaronis, G. Balikas, P. Malakasiotis, *et al.*, "An overview of the bioasq large-scale biomedical semantic indexing and question answering competition," *BMC Bioinformatics*, vol. 16, p. 138, Apr. 2015. DOI: 10.1186/s12859-015-0564-6.

[52] Q. Jin, B. Dhingra, Z. Liu, W. Cohen, and X. Lu, "PubMedQA: A dataset for biomedical research question answering," 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)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2567–2577. DOI: 10.18653/v1/D19-1259.

[53] Y. Gu, R. Tinn, H. Cheng, *et al.*, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Trans. Comput. Healthcare*, vol. 3, no. 1, Aug. 2021. DOI: 10.1145/3458754.

[54] B. J, P. C, E. J, Z. M, B. A, and H. A., "Developing a fully automated evidence synthesis tool for identifying, assessing and collating the evidence," *2021 Feb;26(1):24-27. doi:*, vol. 10. B. E. Based, Ed., 2019.

[55] G. Bai and L. Zhao, "Saliency-regularized deep multi-task learning," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '22, Washington DC, USA: Association for Computing Machinery, 2022, pp. 15–25, ISBN: 9781450393850. DOI: 10.1145/3534678.3539442.

[56] A. Maurer, M. Pontil, and B. Romera-Paredes, *The benefit of multitask representation learning*, 2016.

[57] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *J. Mach. Learn. Res.*, vol. 6, pp. 1817–1853, Dec. 2005, ISSN: 1532-4435.

[58] J. Baxter, "A model of inductive bias learning," *J. Artif. Int. Res.*, vol. 12, no. 1, pp. 149–198, Mar. 2000, ISSN: 1076-9757.

[59] S. Ben-David and R. Borbely, "Exploiting task relatedness for multiple task learning," *Lecture Notes in Computer Science*, vol. 2777, Aug. 2002. DOI: 10.1007/978-3-540-45167-9_41.

[60] G. Cavallanti, N. Cesa-Bianchi, and C. Gentile, "Linear algorithms for online multitask classification," *Journal of Machine Learning Research*, vol. 11, pp. 2901–2934, Mar. 2010.

[61] K. Lounici, M. Pontil, A. B. Tsybakov, and S. van de Geer, *Oracle inequalities and optimal inference under group sparsity*, 2010.

[62] A. Maurer, "Bounds for linear multi-task learning," *J. Mach. Learn. Res.*, vol. 7, pp. 117–139, Dec. 2006, ISSN: 1532-4435.

[63] S. Thrun and L. Pratt, "Learning to learn: Introduction and overview," in *Learning to Learn*, S. Thrun and L. Pratt, Eds. Boston, MA: Springer US, 1998, pp. 3–17, ISBN: 978-1-4615-5529-2. DOI: 10.1007/978-1-4615-5529-2_1.

[64] A. Maurer, M. Pontil, and B. Romera-Paredes, *Sparse coding for multitask and transfer learning*, 2014.

[65] A. Pentina and C. H. Lampert, *A pac-bayesian bound for lifelong learning*, 2014.

[66] X. Liu, P. He, W. Chen, and J. Gao, *Multi-task deep neural networks for natural language understanding*, 2019.

[67] S. Papay, R. Klinger, and S. Padó, "Dissecting span identification tasks with performance prediction," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 4881–4895. DOI: 10.18653/v1/2020.emnlp-main.396.

[68] N. Ding, G. Xu, Y. Chen, *et al.*, "Few-NERD: A few-shot named entity recognition dataset," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 3198–3213. DOI: 10.18653/v1/2021.acl-long.248.

[69] L. Xu, Z. Jie, W. Lu, and L. Bing, "Better feature integration for named entity recognition," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, *et al.*, Eds., Online: Association for Computational Linguistics, Jun. 2021, pp. 3457–3469. DOI: 10.18653/v1/2021.naacl-main.271.

[70] L. Derczynski, E. Nichols, M. van Erp, and N. Limsopatham, "Results of the WNUT2017 shared task on novel and emerging entity recognition," in *Proceedings of the 3rd Workshop on Noisy User-generated Text*, L. Derczynski, W. Xu, A. Ritter, and T. Baldwin, Eds., Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 140–147. DOI: 10.18653/v1/W17-4418.

[71] B. Liu, "Sentiment analysis and opinion mining," vol. 5, May 2012, ISBN: 978-3-642-19459-7. DOI: 10.2200/S00416ED1V01Y201204HLT016.

[72] I. Chalkidis, I. Androutsopoulos, and A. Michos, "Extracting contract elements," in *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, ser. ICAIL '17, London, United Kingdom: Association for Computing Machinery, 2017, pp. 19–28, ISBN: 9781450348911. DOI: 10.1145/3086512.3086515.

[73] G. Da San Martino, S. Yu, A. Barrón-Cedeño, R. Petrov, and P. Nakov, "Fine-grained analysis of propaganda in news article," 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)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5636–5646. DOI: 10.18653/v1/D19-1565.

[74] L. Cheng, L. Bing, Q. Yu, W. Lu, and L. Si, "APE: Argument pair extraction from peer review and rebuttal via multi-task learning," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 7000–7011. DOI: 10.18653/v1/2020.emnlp-main.569.

[75] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *Transactions of the Association for Computational Linguistics*, vol. 4, L. Lee, M. Johnson, and K. Toutanova, Eds., pp. 357–370, 2016. DOI: 10.1162/tacl_a_00104.

[76] P. Yang, X. Cong, Z. Sun, and X. Liu, "Enhanced language representation with label knowledge for span extraction," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds., Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4623–4635. DOI: 10.18653/v1/2021.emnlp-main.379.

[77] N. Tishby, F. C. Pereira, and W. Bialek, *The information bottleneck method*, 2000.

[78] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, 2022.

[79] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31st International Conference on Machine Learning*, E. P. Xing and T. Jebara, Eds., ser. Proceedings of Machine Learning Research, vol. 32, Bejing, China: PMLR, Jun. 2014, pp. 1278–1286.

[80] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, *Deep variational information bottleneck*, 2019.

[81] S. Voloshynovskiy, M. Kondah, S. Rezaeifar, O. Taran, T. Holotyak, and D. J. Rezende, *Information bottleneck through variational glasses*, 2019.

[82] M. G. Sohrab and M. Miwa, "Deep exhaustive model for nested named entity recognition," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds., Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 2843–2849. DOI: 10.18653/v1/D18-1309.

[83] C. Zheng, Y. Cai, J. Xu, H.-f. Leung, and G. Xu, "A boundary-aware neural model for nested named entity recognition," 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)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 357–366. DOI: 10.18653/v1/D19-1034.

[84] C. Tan, W. Qiu, M. Chen, R. Wang, and F. Huang, "Boundary enhanced neural span classification for nested named entity recognition," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 9016–9023, Apr. 2020. DOI: 10.1609/aaai.v34i05.6434.

[85] C. Xia, C. Zhang, T. Yang, *et al.*, "Multi-grained named entity recognition," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds., Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 1430–1440. DOI: 10.18653/v1/P19-1138.

[86] Y. Xu, H. Huang, C. Feng, and Y. Hu, "A supervised multi-head self-attention network for nested named entity recognition," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 16, pp. 14 185–14 193, May 2021. DOI: 10.1609/aaai.v35i16.17669.

[87] J. Fu, X. Huang, and P. Liu, "SpanNER: Named entity re-/recognition as span prediction," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 7183–7195. DOI: 10.18653/v1/2021.acl-long.558.

[88] F. Li, Z. Lin, M. Zhang, and D. Ji, "A span-based model for joint overlapped and discontinuous named entity recognition," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 4814–4828. DOI: 10.18653/v1/2021.acl-long.372.

[89] N. T. H. Nguyen, M. Miwa, and S. Ananiadou, "Span-based named entity recognition by generating and compressing information," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, A. Vlachos and I. Augenstein, Eds., Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 1984–1996. DOI: 10.18653/v1/2023.eacl-main.146.

[90] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '04, Seattle, WA, USA: Association for Computing Machinery, 2004, pp. 168–177, ISBN: 1581138881. DOI: 10.1145/1014052.1014073.

[91] K. N. T. Thanh, S. H. Khai, P. P. Huynh, L. P. Luc, D.-V. Nguyen, and K. N. Van, "Span detection for aspect-based sentiment analysis in Vietnamese," in *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, K. Hu, J.-B. Kim, C. Zong, and E. Chersoni, Eds., Shanghai, China: Association for Computational Lingustics, Nov. 2021, pp. 318–328.

[92] S. E. Toulmin, *The Uses of Argument*, 2nd ed. Cambridge University Press, 2003.

[93] J. Daxenberger, S. Eger, I. Habernal, C. Stab, and I. Gurevych, "What is the essence of a claim? cross-domain claim identification," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, M. Palmer, R. Hwa, and S. Riedel, Eds., Copenhagen, Denmark:

Association for Computational Linguistics, Sep. 2017, pp. 2055–2066. DOI: 10.18653/v1/D17-1218.

[94] M. Sundriyal, A. Kulkarni, V. Pulastya, M. S. Akhtar, and T. Chakraborty, "Empowering the fact-checkers! automatic identification of claim spans on Twitter," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 7701–7715. DOI: 10.18653/v1/2022.emnlp-main.525.

[95] S. Rosenthal and K. McKeown, "Detecting opinionated claims in online discussions," in *Sixth IEEE International Conference on Semantic Computing, ICSC 2012, Palermo, Italy, September 19-21, 2012*, 2012, pp. 30–37. DOI: 10.1109/ICSC.2012.59.

[96] S. Poddar, N. G. Azlaan Mustafa Samad Rajdeep Mukherjee, and S. Ghosh, "Caves: A dataset to facilitate explainable classification and summarization of concerns towards covid vaccines," 2022.

[97] S. Poddar, N. G. Azlaan Mustafa Samad Rajdeep Mukherjee, and S. Ghosh, "Caves: A dataset to facilitate explainable classification and summarization of concerns towards covid vaccines," 2022.

[98] S. Poddar, A. M. Samad, R. Mukherjee, N. Ganguly, and S. Ghosh, "Caves: A dataset to facilitate explainable classification and summarization of concerns towards covid vaccines," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 3154–3164.

[99] W. C. Maggie Phil Culliton, *Tweet sentiment extraction*, 2020.

[100] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds., Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.

[101] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional lstm networks for improved phoneme classification and recognition," in *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, W. Duch, J. Kacprzyk, E. Oja, and S. Zadrony, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 799–804, ISBN: 978-3-540-28756-8.

[102] S. Poddar, M. Basu, K. Ghosh, and S. Ghosh, "Overview of the fire 2023 track:artificial intelligence on social media (aisome)," in *Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation*, 2023.

[103] T. Jalilov, "Emoji for python," 2014.

[104] 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.

[105] R. Kundu, *Multi-Task Learning in ML: Optimization and Use Cases [Overview]*, Jul. 2023.

[106] M. S. Akhtar, D. Chauhan, D. Ghosal, S. Poria, A. Ekbal, and P. Bhattacharyya, "Multi-task learning for multi-modal emotion recognition and sentiment analysis," 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)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 370–379. DOI: 10.18653/v1/N19-1034.

[107] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv (Cornell University)*, Dec. 2014. DOI: 10.48550/arxiv.1412.6980.