

MULTILINGUAL IDENTIFICATION AND MITIGATION OF BIAS AND CLICKBAITY CONTENT

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

Master of Science
in
Computational Linguistics
by Research

by

ANUBHAV SHARMA

2018114007

`anubhav.sharma@research.iiit.ac.in`



International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2023

Copyright © Anubhav Sharma, 2023
All Rights Reserved

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “**Multilingual Identification and Mitigation of Bias and Clickbaity Content**” by **Anubhav Sharma**, has been carried out under our supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. Radhika Mamidi

To my beloved nanis, who couldn't live to see the light of this day. Both of you remain in my thoughts and prayers incessantly.

Acknowledgments

This endeavor has been a collaborative effort, and while I may appear as the public face of my research, I humbly extend my heartfelt gratitude to every single individual who has played a role in its progress. Without your invaluable contributions, I would not have reached this stage of accomplishment. I am deeply appreciative of your steadfast support, which has made for a truly enriching "*Anubhav*".

I express my sincere gratitude to Dr. Radhika Mamidi, my advisor, for her unwavering guidance and support throughout my research journey. Her kindness and assistance extended beyond academics, and she provided solace during difficult times. I am grateful for her motherly care and constant presence. Additionally, I would like to thank Dr. Vasudeva Verma, my co-advisor, for his invaluable mentorship and provision of resources that facilitated my research endeavors for over three years. I greatly admire your ability to ideate and execute ideas. Furthermore, I extend my thanks to Dr. Manish Gupta, my mentor/co-advisor, for his ceaseless guidance during my research cycle. Your work ethics and dedication to the field are an inspiration, and I appreciate the constructive criticism you provided that made me strive harder.

As it is said that research is the product of constructive thinking, I am grateful to have had good company, and I thank Hanuman Jee for blessing me with it. Firstly, I would like to express my gratitude to you, Tushar Sir (fondly called TA), as words cannot fully convey how much you have influenced my mindset and how I look up to you as an elder brother. Our research journey began at the same time and progressed in tandem. Your work ethics and mentality towards work are admirable. Secondly, Sagar, thinking of you as a brother, I express my heartfelt gratitude for everything. You are truly a gem of a person, and I cannot stress enough how our chance encounter can be described as serendipity. I wish I had your warmth and helpfulness. I just wish you loved me more than Sebaschion. I would like to thank my research mate Ankita. "Generally"(the way you say it) it was great to work with you and our interactions were always fruitful. I would like to express my gratitude to Pavan for our engaging discussions and teasing sessions filled with laughter, which I will deeply miss. I am also thankful to Bhavyajeet and Shivprasad for their unwavering support and for being wonderful teammates. Furthermore, I extend my appreciation to my other fellow members in the IRE Lab, Sachan, Padhi, ted and Sumba, for being amazing colleagues.

I was fortunate to have supportive friends during my personal journey, and I am immensely grateful to Muskan for being a constant source of support. Spending time with her taught me valuable life lessons, and her "just bow down and work" philosophy is something I aspire to execute perfectly. Throughout my college life, I was blessed with great friends who formed the F.R.I.E.N.D.S - Anishka, Bakka Cha, Prerna, Haathi, Rads, Vinashak, Chutki, Nik Pa, and Ruttu. You are my lifelong friends, and the memories we shared will stay with me forever. I was also part of another close group of friends, the Landoore Mitr Mandal - Akshi Paa, Sajji Pa, Jashn Pa, Kushagra, Momo Boi, and Subodh. You are some of the finest people I have ever met, and I thank you for always being there for me. I also extend my heartfelt thanks to Daga, Hari, and Saboo for making me feel at home wherever I went. Conversations with you were always enriching, and I miss you dearly. Daga, thank you for being my forever project partner and my go-to person for soya chaap. I am also grateful to Adarsh and Joseph for being such wonderful roommates, making my stay away from home much easier. I would like to thank Srinath, Dolton, Sudhansh, Tanvi, and Manas for bringing me closer to music. I have learned a lot from you guys, and I will miss our endless jam sessions. Making friends from the same batch is common, but having the opportunity to befriend seniors and juniors was unexpected. I express my gratitude to Manan Sir for always guiding me through difficult times. Additionally, I am grateful for the friendships that I have formed with Yash, Abhishek, Sancheti, Patel, and Bittu in my junior batches. Thank you for the time we shared together, it was truly memorable. I want to express my heartfelt gratitude to my friends from Prithvi House. Working with you guys has been an absolute pleasure. Bhaiya, Eshaan, Aryan Singh, Sriram, Tanvi, Nikhil, and Sethi, I will always cherish the memories we made together. Prithvi truly rules! I also want to thank Pranali, Divya, and Anmol for always being there for me, no matter what. Your unwavering support and kindness mean the world to me. To my friends from UG2 - Deepti, Hardik, Brahad, Krishna, and Vinkesh - I wish you all the best in your future endeavors. Thank you for the memories we shared together. But, the people who hold a special place in my heart are the people from my GYM. Khushi, you are the kindest and sweetest soul I know. Your quirkiness and cuteness are what I expect from a little sister. Thank you for always being there for me. Vaibhav, you are an immensely talented person who would do anything to make his people feel comfortable. Sarthak, you are the strongest person I know with an insatiable curiosity and an infectious sense of humor. Samarth, you have a soothing presence, and your humor always lifts my spirits. I will always cherish the moments we shared together, and I will miss you all dearly. I would also like to thank Kushal and Gautam, without you guys I wouldn't be here and that year in Allen that we spent was truly memorable. Lastly, I would also like to thank my closest friends of all - Krit, Momo, Steff, Deepika and Vinay, you guys are truly my safe place and thank you for bearing with me throughout the years.

Throughout my life, I have been blessed with many supportive friends, but I cannot forget the sacrifices and love that my parents have shown me. They have dedicated their entire lives to me, and I am grateful beyond words. Thank you, Mumma and Papa, for your unconditional love and support. I hope to make you both proud one day. I am also grateful for the endless blessings from my maa and baba. I would like

to extend my gratitude to Shivam Bhaiya, Mayank Bhaiya, Dabbu Bhaiya, Chetan Bhaiya, Diksha Di, and Priya Di for being more than just cousins. Your love and support have always been there for me, and I will forever be grateful. Lastly, I want to thank each and every person who has been a part of my journey. It has been a wild ride, but with the support of my loved ones, I was able to get through it all.

”Its about the damn time.”

Abstract

Handling fine-grained subtleties in text remains a challenge, despite the advances in the understanding and generation of textual content. The difficulties in the identification and segregation of such subtleties arises from the difficulty in sufficient human comprehension of these aspects and their implicit presence in the data used to train NLP models in understanding and generation tasks. In this thesis, we primarily focus on two types of such content: bias and clickbaits. Textual bias, which is implicitly manifested across corpora, arises from individual inclinations, perspectives and interpretations of facts and serves to distort the understanding of the information from the discourse by imposing subjective opinions. The presence of such affective content is a nuisance for encyclopedic platforms like Wikipedia that serve to provide knowledge from a neutral point of view and are used as a reliable source worldwide. Clickbaits are another form of malicious content that serve to distract user attention on social media websites. They usually work by luring a user to click on linked articles which often contain only trivial to no useful informational content in contrast to the intensity suggested by the “bait”, and result in unnecessary user frustration and potential masking of helpful information by diverting attention that could be paid to websites with objective information.

While bias identification and mitigation have been well studied problems in NLP on the English language and there have been multiple definitions of bias from the perspectives of different corpora and objectives, we fix our definition from perspective of the neutral point of view on Wikipedia. Also, we seek to study the bias problem on a multilingual scale, dealing with native languages from the South Asian subcontinent as a study on low resource languages in conjunction with English. Prior to taking up these multilingual problems on the Wikipedia domain, we precede the same with a study of three different problems in this setting. In our first multilingual problem, we propose a novel architecture for fine-grained categorization of entities on a Wikipedia-based dataset of 30 languages. Following this, we set out to study two dual problems in a cross-lingual setting. In the first of these two, we aim at content enrichment in low resource languages making use of factual information from structured knowledge bases in English and explain the creation of a novel dataset for this study. In the other problem, we propose two methods for extraction of English facts from unstructured content in low resource languages with the aim of adequate utilization of the knowledge contained in such languages. A study of these three problems helped lay a solid foundation for the study of more subtle problems like bias on a multilingual scale.

As a part of our study on bias identification and mitigation, we present several attempts at creating a sizeable multilingual, parallel dataset making use of edit tags on Wikipedia. We curate and propose our dataset for study which is based on translation from existing datasets in English. Following dataset creation, we present our modeling of the two problems of bias detection as a classification problem and bias mitigation as a style transfer problem through extensive experiments carried out in monolingual and multilingual settings. With the existing methods for evaluation of textual debiasing not sufficient for the purpose, we design an evaluation strategy combining traditional generation-based metrics with two additional metrics measuring the percentage of change and bias classification accuracy on the generated output. We also present several directions for extending this problem, following an analysis of our results.

We approach the problem of mitigating the effects induced by a clickbait by looking at generation of “spoilers”, or short pieces of content to satisfy the curiosity generated by the clickbait. We attach the problem as a 2-stage pipeline, where the first stage predicts the type of spoiler that will be generated, and in the second stage, a spoiler type-specific spoiler generator extracts the necessary content from the article. We propose a novel Information Condensation-based modeling approach to tackle this problem, where we add filtering to the article associate with the clickbait, which helps wade out a lot of the potentially unnecessary information from the article. The article with condensed information is then used for the 2-stage problem. Our experiments reveal the merit of a contrastive learning-based method to design the filtering model, as opposed to simpler classification-based methods. We achieve SoTA results on the problem, and present an extensive analysis of our techniques used.

Contents

Chapter	Page
1 Introduction	1
1.1 Biases in Text	1
1.2 On Clickbaity Content	2
1.3 Textual Corpora for Study	3
1.3.1 Wikipedia	4
1.3.2 Social Media	4
1.4 Multilinguality in NLP	5
1.4.1 Multilingual Models	6
1.5 Thesis Contributions	6
1.6 Organization of This Thesis	8
2 An Overview of Related Works	9
2.1 Task: Multilingual Setting for NLP tasks	9
2.1.1 Fine Grained Entity Categorisation	9
2.1.2 Automated Text to Fact Generation	10
2.1.3 Text to Fact Extraction	10
2.2 Multilingual Bias Detection and Mitigation	11
2.2.1 Dataset	11
2.2.2 Detection and Mitigation Approaches	12
2.3 Task: Clickbait Spoiling	13
2.3.1 Spoiler Type Classification	14
2.3.2 Clickbait Generation	14
2.3.3 Paragraph-based Information Retrieval Techniques	15
3 Multilingual Fine-grained Entity Categorization in Wikipedia Text	16
3.1 Multilingual Entity Categorization: Motivation	16
3.2 Introduction and Approaches to Entity Categorization	17
3.3 About the Dataset	19
3.3.1 Original Dataset	19
3.3.2 Preprocessing	19
3.3.3 Taxonomy	19
3.3.4 Multilingual Dataset Creation	19
3.4 Baseline Approaches for Entity-Type Classification	20
3.4.1 MPAD	21
3.4.2 DTMT	21

3.4.3	MAGNET	21
3.4.4	NER-based Model	22
3.4.5	Multilingual Models	22
3.5	Proposed Method - RNN_GNN_XLM-R	22
3.5.1	Overall Architecture	22
3.5.2	Contextual Representation using RNN	24
3.5.3	GAT for Capturing Label Relatedness	25
3.5.4	Final Scoring	25
3.6	Experiments & Results	26
3.6.1	Evaluation	26
3.6.2	Monolingual Results for Hindi	26
3.6.3	Multilingual Results	28
3.7	Conclusion	29
4	A Cross-lingual Study in Fact-to-Text and Text-to-Fact Problems	30
4.1	On Cross-lingual Fact-to-Text Generation (XF2T)	30
4.2	XAlign: A Cross-lingual Dataset for XF2T	31
4.2.1	Data Collection and Cleaning	32
4.2.2	Automatic Pipeline for Fact Alignment	33
4.2.2.1	Stage 1: Candidate Generation	33
4.2.2.2	Stage 2: Candidate Selection	34
4.2.3	Manual Annotation Process for Evaluation Dataset	34
4.2.3.1	Annotation Guidelines	35
4.2.3.2	Annotation Instructions	35
4.2.4	Statistics of the XAlign Dataset	36
4.3	Experiments on XF2T	38
4.3.1	Input Representation: Structure-Aware Input Encoding	38
4.3.2	Fact-Aware Embeddings	39
4.3.3	Results on the XF2T Task	39
4.4	Cross-lingual Fact Extraction for Knowledge Graph Enrichment	40
4.4.1	On Knowledge Graphs	40
4.4.2	Fact Extraction v/s Fact Linking	40
4.4.3	On Cross-lingual Fact Extraction (CLFE)	41
4.5	Methods for CLFE	41
4.5.1	Tail Extraction and Relation Classification (TERC)	41
4.5.2	End to End Generative extraction	42
4.5.3	Results on CLFE Experiments	43
4.6	Conclusion	43
5	Bias Detection and Mitigation in a Multilingual Setting	45
5.1	Why Bias Mitigation for Wikipedia?	45
5.2	Dataset for Bias Identification and Mitigation	46
5.2.1	Necessity of an Appropriate Dataset for Study	46
5.2.2	Our Dataset	46
5.3	Attempts at Creating a Multilingual Bias Mitigation Dataset	48
5.4	Methodology	49

5.4.1	Bias Identification: Classification Approach	49
5.4.2	Bias Mitigation: Style Transfer Approach	49
5.5	Results & Analysis	50
5.5.1	Classification-based Experiments	51
5.5.2	Style Transfer-based Experiments	55
5.6	Directions for Future Work	57
6	A Two-Stage Approach to Clickbait Spoiling	59
6.1	Introduction to Clickbait Spoiling	59
6.2	Information Condensation for Clickbait Spoiling: Motivation and Highlights	60
6.2.1	Motivation & Introduction to Information Condensation	60
6.2.2	Highlights of our Approach	61
6.3	About the Data	61
6.3.1	WCS Corpus	62
6.3.2	Pw Dataset	62
6.3.3	Auxiliary Datasets	62
6.4	Methodology	63
6.4.1	Oracle Experiments	63
6.4.2	Paragraph-wise Filtering	63
6.4.3	Classification Paradigm	64
6.4.3.1	Vanilla Classification	64
6.4.3.2	ASNQ Pretraining	64
6.4.4	Contrastive Learning Paradigm	66
6.4.4.1	Modeling	66
6.4.4.2	Data Feeding Techniques	66
6.4.5	Inferencing for Paragraph Filtering	66
6.4.6	Spoiler Type Classification	66
6.4.7	Spoiler Generation	67
6.5	Experimentation	67
6.5.1	Architectural Specifics	68
6.5.1.1	Models used for classification	68
6.5.1.2	Models used for contrastive learning	68
6.5.2	Training	68
6.5.2.1	Classification based experiments	68
6.5.2.2	Contrastive learning based experiments	70
6.5.2.3	Extractive QA based experiments	70
6.5.3	Evaluation	70
6.5.3.1	Metrics for Classification	70
6.5.3.2	Metrics for Ranking	71
6.5.3.3	Metrics for Spoiler Generation	71
6.6	Results	71
6.6.1	Paragraph-wise Filtering	71
6.6.2	Spoiler Type Classification	72
6.6.3	Spoiler Generation	73
6.7	Analysis	73
6.7.1	Differences based on Spoiler Type	73

6.7.2	Spoiler Loss Due to Chosen k	74
6.7.3	Performance Variation with Filtering	74
6.7.4	Qualitative Analysis	76
6.7.4.1	Examples for Phrase Spoiler Extraction	76
6.7.4.2	Examples for Passage Spoiler Extraction	76
6.7.4.3	Output v/s Expected Overlap Comparison	77
6.8	Conclusion	78
7	Conclusion	81
7.1	Inferences from the Problems for Multilingual Study	81
7.1.1	Fine-grained Entity Categorization	81
7.1.2	Cross-lingual Fact-to-Text Generation	82
7.1.3	Cross-lingual Fact Extraction	83
7.2	Inferences from the Study on Bias Identification & Mitigation	83
7.3	Inferences from the Study on Clickbait Spoiling	84
	Bibliography	86

List of Figures

Figure	Page
2.1 Description of the task of ClickBait Spoiling: For each tweet that serves as a clickbait, the spoiler is also displayed.	13
3.1 Architecture of our proposed system, RNN_GNN_XLM-R.	23
4.1 Examples of cross-lingual fact-to-text (XF2T) generation from facts in English to sentences in English or LR languages.	31
4.2 Adopted 2-stage pipeline for automatic fact alignment in creating the XAlign dataset followed by XF2T generation.	33
4.3 Fact count distribution across languages in the XAlign Dataset.	37
4.4 Fact count distribution across data subsets in the XAlign Dataset.	38
4.5 The input given to the encoder of mT5 includes English facts, along with token and position embeddings. In addition to that, role embeddings are used to indicate the specific roles of different tokens in the facts.	39
4.6 Pipeline Architecture for the TERC method, consisting of tail extraction from translated sentences followed by relation classification.	41
4.7 End to end generative pipeline for CLFE, which simultaneously extracts the entities and the relations present from the given sentence.	42
5.1 An example illustration of debiasing in different languages.	50
5.2 Metrics for evaluating debiasing: use of classification module for evaluation generated output from the style transfer module.	51
6.1 In this particular instance, the red text represents the spoiler, categorized as “passage.” The dotted arrow connecting the two tasks represents their interdependence and serves as a motivation to have a cohesive pipeline for spoiling clickbait.	60
6.2 Model architecture for contrastive learning from clickbait-paragraph text sequence. . .	67
6.3 Experiment flow for our entire system. The red cross represents the termination of an approach due to inferior performance compared to the best-performing one, whose flow is marked with bold arrows.	69
6.4 Performance analysis of the best model on spoiler type classification with confusion matrix (left) and bar chart for type-wise classification metrics (right).	76
6.5 Plot of percentage loss of spoiler containing paragraphs in an article for different values of k.	77
6.6 Variation in Corpus BLEU score of rb-l and rb-l-squad models for different values of k. . .	78

List of Tables

Table	Page
3.1 Examples of entity categorization in four different languages.	18
3.2 Languages selected for creation of multi-lingual dataset along with training data statistics, using SVO order.	20
3.3 Languages selected for creation of multi-lingual dataset along with training data statistics, using SOV order.	20
3.4 The F1-score reflected from the task leaderboard submission for Hindi evaluation dataset.	26
3.5 Micro-averaged F1 score comparison across various models on the leaderboard test set. HUKB, PribL and uomfj were the other top performing teams on the task challenge leaderboard. mBERT-base and XLM-R-base are our other baselines. RNN_GNN_XLM-R is our best proposed model. Best results for each language are highlighted in bold. . . .	27
3.6 Micro-averaged F1 score comparison across various models on the official evaluation dataset for 6 languages. ousia, uomfj, LIAT and PribL are the other top performing teams on the task challenge leaderboard. RH312 (i.e., RNN_GNN_XLM-R) is our best proposed model. Best results for each language are highlighted in bold.	28
4.1 Basic Statistics of XAlign. $ I $ = # instances, $ T $ = avg/min/max word count, $ F $ = avg/min/max fact count, $ V $ = Vocabulary size, κ = Kappa score, $ A $ = # annotators . .	36
4.2 XF2T scores on the XAlign test set using mT5 with fact-aware embeddings.	40
4.3 Precision, recall and F1 scores of various methods applied on all languages in the test set.	43
5.1 The table contains all details for both datasets. We create a similar parallel dataset for each language in the multilingual versions.	47
5.2 Classification results over multilingual experiments on the mWNC dataset using the MuRIL model.	52
5.3 Classification results over multilingual experiments on the mWNC dataset using the InfoXLM model.	52
5.4 Classification results over monolingual experiments on the mWNC dataset using the MuRIL model.	52
5.5 Classification results over monolingual experiments on the mWNC dataset using the InfoXLM model.	53
5.6 Classification results over multilingual experiments on the mWIKIBIAS dataset using the MuRIL model.	53
5.7 Classification results over multilingual experiments on the mWIKIBIAS dataset using the InfoXLM model.	53

5.8	Classification results over monolingual experiments on the mWIKIBIAS dataset using the MuRIL model.	54
5.9	Classification results over monolingual experiments on the mWIKIBIAS dataset using the InfoXLM model.	54
5.10	Debiasing results on the mWNC dataset.	56
5.11	Debiasing results on the mWIKIBIAS dataset.	56
6.1	Statistics of the given WCS Corpus for the main task and the derived Pw Dataset. . . .	62
6.2	Statistics of the ASNQ dataset, for the number of training and validation (dev) samples for each type of annotation.	64
6.3	Label-wise statistics detailing the construction of the v1 and v2 subsets of ASNQ for our experimentation. Label 1 represents easy negatives, label 2 represents hard positives, label 3 represents easy positives and label 4 represents easy positives.	65
6.4	Results for the experiments on paragraph-wise filtering on the dev set of the Pw dataset. F1, Recall and Precision are computed in macro forms.	72
6.5	Results on spoiler type classification task for dev set. Balanced Accuracy is the same as macro-Recall.	73
6.6	Results for phrase spoiler generation (extraction) on the dev set (325 samples).	74
6.7	Results for passage spoiler generation (extraction) on the dev set (322 samples).	75
6.8	Type-wise differences in Mean Rank for the two best-performing models on the Pw task.	75
6.9	Some examples produced by the best performing model (Filtered-phrase: rb-l-squad) for phrase spoiler type extraction.	79
6.10	Some examples produced by the best performing model (WCS-passage: rb-l-squad) for passage spoiler type extraction.	80
6.11	% overlap statistics for phrase and passage spoiler extraction between the output (O) and expected (E) spans by the best models on each.	80

Chapter 1

Introduction

In this thesis, we primarily focus on two aspects in text: biases and clickbaity content.

As a precursor to our multilingual work in the bias domain, we study several other multilingual problems starting with multilingual entity categorization in Wikipedia articles. We also work on generating objective text in low resource South Asian languages. The selection of these languages is based on the low formal digital content [25] and poor performance of existing translation modules [23] in comparison to European languages. We curate a dataset for cross-lingual fact-to-text generation and work on techniques to improve the content quality over translation in the fact-to-text generation pipeline. Additionally, we propose a problem for utilizing information available in low resource languages for knowledge graph enrichment.

Following a detailed study of these multilingual problems, we move on to one of our primary problem of bias identification and mitigation. Here, we curate a multilingual dataset for studying this problem and survey several approaches on the same. This follows a study on our second problem which revolves around mitigating the unnecessary, undesirable attractiveness in clickbaity content by spoiling the same. Working in a monolingual setting, we research on the newly introduced problem, and propose a SoTA approach on the task.

This chapter introduces the characteristics of biased and clickbaity texts, discusses the nature of content commonly encountered in them, and explores two types of textual corpora that the work has been done on - Wikipedia and Social Media - for sourcing datasets for experimentation. This follows an overview of multilinguality in NLP. The chapter concludes by a discussion on the primary contributions of this thesis and its organization.

1.1 Biases in Text

Written communication or discourse typically tends to be manifested with some form of bias present in it. Textual bias refers to any instance in which the text involves an unfair or disproportionate effect that affects the way in which the message is conveyed and interpreted. The bias can be intentional or unintentional, as an author may deliberately use language or make use of other techniques to sway

the reader's opinion, or it may happen that the bias gets induced due to the author's personal beliefs, experiences, or assumptions, without a deliberate attempt made to include the same.

Bias can be found in any form of written communication, from news stories to research papers to social media postings and personal blogs. This makes it important for readers to be aware of the possibility of bias and to critically evaluate the information presented to ensure that the message being understood is accurate, balanced, and free from undue influence. Readers can obtain a more thorough awareness of the problems at hand and make more informed judgements based on the information offered by being aware of textual bias and critically analysing texts.

Biases can exist in text in a variety of ways. Here are some examples:

1. Word choice

The use of certain words or phrases can reveal the author's bias. For instance, using words with positive or negative connotations can influence the reader's perception of a particular issue.

2. Tone

The tone of the text can also reveal bias. For instance, if a text is written in a sarcastic or dismissive tone, it can indicate a bias against the subject matter or the people involved.

3. Selective use of information

Bias can also be present in a text when the author selectively uses information that supports their viewpoint, while ignoring information that contradicts it.

4. Stereotypes and assumptions

Bias can also be present in text when the author makes assumptions or perpetuates stereotypes about a particular group of people.

5. Structural biases

Text can also be biased due to structural factors, such as the lack of diversity in the authorship or editorial process, which can lead to certain perspectives being overrepresented or underrepresented.

It is important to be aware of these biases and critically evaluate the information presented in a text to ensure a more balanced understanding of the topic.

1.2 On Clickbait Content

A clickbait is a type of content, such as a headline or thumbnail image, that is designed to attract attention and encourage people to click through to a particular web page or article. Clickbaits often use sensational or exaggerated language or images to create curiosity or a sense of urgency, without providing much substantial information. The ultimate goal of a clickbait is to generate more traffic, engagement, and revenue for the website or platform hosting the content.

There are many different types of clickbait which appear in text, but here are some common examples:

1. **Sensational headlines**

These are headlines that use exaggerated or hyperbolic language to grab the reader’s attention, often by playing on emotions like fear, anger, or curiosity.

2. **Outrageous claims**

Similar to sensational headlines, these clickbait articles make outrageous or unbelievable claims in order to get people to click through and read the story.

3. **False promises**

Some clickbait articles use false promises to entice readers, such as “You won’t believe what happens next!” or “This one simple trick will change your life forever!”

4. **Salacious content**

Clickbait articles may also feature sexually suggestive or provocative content, often accompanied by racy images or headlines designed to shock or titillate readers.

5. **Celebrity gossip**

Many clickbait articles are centered around celebrity news and gossip, often featuring misleading or sensational headlines to draw in readers.

6. **Lists and rankings**

Clickbait articles in the form of lists or rankings are also popular, often featuring eye-catching headlines like “The top 10 most shocking moments in history” or “The 5 best ways to lose weight fast.”

Clickbait articles often have low-quality content that is poorly written or researched, or contains little actual information. They may also have pop-up ads or other forms of aggressive advertising that disrupts the user experience. Hence, mitigation of the unnecessary frustration caused due to the clickbaity content is necessary to improve user experience on the web.

1.3 **Textual Corpora for Study**

We distinctively make use of two types of textual corpora in this thesis: **(1)** the Wikipedia corpus, which is extensively used for our problems of bias identification and mitigation, as well as for studying the prior multilingual problems of entity categorization, fact-to-text and text-to-fact problems; **(2)** the social media corpus, in which we make use of a subset of posts found on the social media websites Reddit, Facebook and Twitter for experimenting on clickbait spoiling. In this section, we provide an introduction to these types of corpora by discussing the nature of content encountered in them.

1.3.1 Wikipedia

The Wikipedia corpus is a large, multilingual, and collaboratively edited encyclopedia containing millions of articles on a wide range of topics. It is considered to be one of the largest and most comprehensive collections of human knowledge available in digital form, covering everything from history and science to current events and pop culture.

The corpus is maintained by a community of volunteer editors who create and edit articles using a set of guidelines and policies. These editors are encouraged to use reliable sources and to write from a neutral point of view, which means that articles should be written without bias or preference for any particular point of view. It is necessary to avoid bias in Wikipedia content because of Wikipedia being a widely used source of information that has a significant impact on how people perceive and understand various topics. Biased content can lead to misinformation, which can have negative consequences in areas such as education, public policy, and social justice. Moreover, Wikipedia is meant to be a neutral and reliable source of information, and biased content goes against these principles.

The articles in the corpus are written in a variety of languages, with English being the most common. However, there are also articles available in many other languages, including Chinese, Spanish, Arabic, and Russian. This makes the corpus a valuable resource for researchers and developers who are interested in natural language processing and machine learning applications.

In addition to the articles themselves, the Wikipedia corpus also includes metadata such as revision histories, user comments, and other information that can be used to analyze the content and structure of the corpus. This metadata can be particularly useful for understanding how articles are created and edited over time, and for identifying patterns and trends in the content. Pertaining to this thesis, such metadata helps identify the sentences which were post-edited explicitly due to the presence of bias, and thus analyze the cases of debiasing edits in greater detail.

Overall, the Wikipedia corpus is a rich and diverse source of information that can be used for a wide range of research and development projects. Its collaborative nature and wide range of topics make it an invaluable resource for anyone working in the field of natural language processing or machine learning.

1.3.2 Social Media

The nature of content on social media is diverse and can vary greatly depending on the platform and the user. Social media platforms allow users to share text, images, videos, links, and other multimedia content with others in their network or with the public.

Some common types of content on social media include personal updates, news articles, memes, videos, product promotions, reviews, and opinion pieces. Social media can also be used for networking and communication, with users engaging in conversations and debates about various topics.

The tone and style of social media content can also vary greatly, with some users adopting a more casual and conversational tone, while others may use more formal or professional language. Additionally,

the content on social media can be influenced by various factors, such as current events, cultural trends, and personal interests of the users.

As a part of this thesis, we deal with clickbaity posts that are found on Reddit, Facebook and Twitter. Of these, Reddit is a website for user-generated content categorized into subreddits. Facebook is a social network with personalized content from users' networks, including news articles and political commentary. Twitter is a real-time microblogging platform for short messages, often used by public figures, politicians, and journalists to share opinions on current events.

There is no one type of page that exclusively posts clickbaity content on social media, as clickbait can be found across a variety of pages and topics. However, some pages that are known to frequently post clickbaity content include sensational news outlets, celebrity gossip blogs, and viral content pages that rely on attention-grabbing headlines and images to attract clicks and engagement. Additionally, some pages may use clickbait as a marketing strategy to drive traffic to their websites or social media profiles, often by promising exclusive deals or insider information in their headlines.

1.4 Multilinguality in NLP

Multilinguality in NLP refers to the ability of natural language processing systems to handle and process multiple languages. It involves developing models and techniques that can effectively handle text data from multiple languages.

The importance of multilinguality in NLP stems from the fact that there are thousands of languages spoken around the world, each with their own unique characteristics and nuances. In order for NLP systems to be truly effective in a global context, they must be able to process and understand text data from a wide variety of languages.

There are several challenges involved in developing multilingual NLP systems. One major challenge is the fact that different languages have different structures and grammatical rules, making it difficult to develop a one-size-fits-all approach. Another challenge is the lack of high-quality language resources for many languages, such as large-scale annotated corpora and pre-trained language models.

Despite these challenges, significant progress has been made in recent years towards developing multilingual NLP systems. One approach is to develop multilingual models that can process text in multiple languages, using shared representations to capture commonalities across different languages. Another approach is to develop language-specific models that are trained on data from a specific language, but can be adapted to handle other languages through transfer learning.

Multilinguality in NLP has numerous applications, including machine translation, sentiment analysis, text classification, and named entity recognition, among others. It enables businesses, organizations, and individuals to communicate and understand each other across different languages, making NLP more accessible and useful in a global context.

1.4.1 Multilingual Models

There have been several recent models for multilinguality in NLP, some of which are:

1. **mBERT** [46]

Multilingual BERT (Bidirectional Encoder Representations from Transformers) is a multilingual pre-training language model that has been trained on Wikipedia texts from 104 languages. It uses a single model to handle multiple languages and is able to achieve state-of-the-art results on various NLP tasks.

2. **XLM** [30]

The Cross-lingual Language Model (XLM) is a multilingual pre-training language model that has been trained on parallel texts in multiple languages. It uses a cross-lingual encoder that can map the same semantic meaning across different languages, enabling it to transfer knowledge from one language to another.

3. **mT5** [61]

Multilingual T5 is a variant of T5 (Text-to-Text Transfer Transformer) model that has been trained on a large corpus of texts in 101 languages. It is a unified model that can perform various NLP tasks in multiple languages.

4. **LASER** [14]

Language-Agnostic SEntence Representations is a multilingual encoder-decoder model that has been trained on parallel texts in multiple languages. It uses a combination of BiLSTMs and Transformer models to encode sentence-level representations in a language-agnostic manner, enabling cross-lingual transfer learning.

5. **MASS** [56]

Masked Sequence-to-Sequence Pre-training is a multilingual pre-training approach that has been trained on monolingual and parallel texts in multiple languages. It uses a masked sequence-to-sequence model to learn cross-lingual representations, enabling it to perform various NLP tasks in multiple languages.

These models have demonstrated the effectiveness of multilingual pre-training for various NLP tasks, and they have paved the way for developing more sophisticated multilingual models in the future.

1.5 Thesis Contributions

In this thesis, we aim to investigate methods for detecting and mitigating bias in text written in South Asian languages and also explore the use of spoilers as a potential approach to address clickbait content. Through a comprehensive analysis, we strive to contribute to the field of natural language

processing by proposing effective strategies to identify and neutralize bias in text, while also exploring innovative techniques to tackle sensationalized content. Moreover, we aim to explore the challenges and opportunities in the multilingual space. Specifically, we delve into problems related to fine-grained entity categorization in low-resource languages and generate multilingual information. Our focus is on developing effective strategies for addressing these challenges and producing high-quality multilingual content. We can summarize the major contributions of our work through the following points:

1. Developing a pipeline for **Detecting and Mitigating Bias** in low-resource South Asian Languages:

- We explored various state-of-the-art pipelines to curate a multilingual parallel corpus that can help in detecting and mitigating bias.
- We curated a multilingual parallel corpus in 8 South Asian languages.
- We studied and experimented on the tasks of bias detection as a classification problem and bias mitigation as a style transfer problem and reported the SoTA results obtained.

2. Mitigating the effects induced by clickbaity content through **Clickbait Spoiling**:

- We examined the task of clickbait spoiling as a two-stage approach involving spoiler type identification and spoiler generation of the identified type.
- We proposed a novel information condensation-based pipeline for approaching the problem, which involved reducing the amount of unnecessary information in the article for generating the spoilers.
- We compared several methods for achieving the information condensation, and showed the merit in our proposed approach making use of contrastive learning, which achieved SoTA results on the problem.
- We presented a comprehensive analysis of our approach and brought the challenges involved, which could be tackled in future scope.

3. Solving **Multilingual NLP Problems** of fine-grained entity categorization, multilingual fact-to-text generation, and text-to-fact extraction:

- We explored the multilingual task of fine-grained entity categorization in Wikipedia articles across 30 languages and curated a highly performant model across all the languages.
- We formulated a new fact-to-text task for low resource languages by investigating challenges in creating a cross-lingual dataset with high-quality semantic overlap between facts and sentences, using Wikimedia resources.
- We proposed and provided strong baselines and approaches for the task of Cross-lingual Fact Extraction (CLFE) for 7 Low-Resource South Asian languages and English, aiming to extract English triples from text written in these languages.

1.6 Organization of This Thesis

This thesis is divided into seven chapters, which are organized as follows:

1. **Chapter 1** (this chapter) establishes the background for the problems we solve in this thesis by discussing the nature of bias and clickbaity content, nature of textual corpora such as Wikipedia and social media, along with a discussion of multilinguality in NLP.
2. **Chapter 2** provides an overview of relevant works to the multilingual tasks that we study, along with an overview of the literature relevant to the problems of bias detection and mitigation, and clickbait spoiling.
3. **Chapter 3** studies the problem of multilingual entity categorization by comparing several prior methods on the task, and shows the better performance of our proposed approach over a Wikipedia-based dataset of 30 languages.
4. **Chapter 4** explores the cross-lingual problems of fact to text generation for content enrichment in low resource languages and text to fact extraction for enhancement of knowledge bases, and explains the curation of a suitable dataset through automatic and human annotation-based pipeline.
5. **Chapter 5** explains our study of multilingual bias identification and mitigation by discussing several attempts at the creation of a suitable dataset, and presents extensive results obtained on the two problems based on a dataset we curate.
6. **Chapter 6** proposes a novel pipeline for the less-studied task of clickbait spoiling as a 2-stage process, in which we establish SoTA results based on our contrastive learning-based approach for condensing information in the articles associated with a clickbait, and present a detailed study on the working of the same.
7. **Chapter 7** concludes the thesis by summarizing our findings and provides a few directions for future work.

Chapter 2

An Overview of Related Works

This thesis primary looks at the the problems of handling biases (multilingual bias identification and mitigation) and clickbaity content (proposing a 2-stage approach to clickbait spoiling) in text. However, considering our focus on multilinguality for the bias-related problems, we also we also seek to study several other problems to study multilinguality in NLP, as discussed in Chapter 1. Hence, we also present a study of the other problems we experiment with an objective to study natural language understanding and generation in a multilingual setting. The additional multilingual problems that we study include fine-grained entity categorization, cross-lingual data-to-text generation and text-to-data extraction. We first discuss some of the related works to our contributions towards these problems. Following this, we proceed for a literature review on the problems of bias identification and mitigation in which we review some of the relevant datasets and methods. In the final section, we establish the motivation for clickbait spoiling by a discussion of the prior works on understanding and categorization of clickbait in text. Besides an overview of works on spoiling of clickbaits, we also discuss an interesting thread on clickbait generation and the paradigm of paragraph-based information retrieval techniques for clickbait spoiler extraction.

2.1 Task: Multilingual Setting for NLP tasks

2.1.1 Fine Grained Entity Categorisation

[41] represented documents as word co-occurrence networks and processed graph-structured data with the very effective Graph Neural Network-based framework. They demonstrate the approach’s competitiveness against other State of the Art models on ten typical classification datasets.

[42] uses a graph attention network-based model for capturing the hierarchical substructure of dependencies among the labels, for which it uses a correlation feature matrix. To allow end-to-end training, the resulting classifiers are applied to sentence feature vectors acquired from the text feature extraction network (BiLSTM) [22].

In [38], they propose a method to enhance Recurrent Neural Network (RNN)-based Neural Machine Translation (NMT) by increasing the depth of transition between consecutive hidden states. They introduce a new architecture called DTMT, which stands for Deep Transition RNN-based Architecture for Neural Machine Translation. The DTMT architecture includes three deep transition modules that are added to the encoder and decoder RNN modules of the shallow RNMT architecture [9]. These deep transition modules aim to improve the non-linear transformation that occurs between successive hidden states in the NMT model.

2.1.2 Automated Text to Fact Generation

In [31], the authors propose a statistical generation model that is dependent on a Wikipedia infobox. The main focus of this model is to generate the opening sentence of a biography, which requires selecting from numerous potential fields in order to produce an appropriate output. The authors' model leverages both global and local conditioning based on the structured data. Global conditioning involves summarizing all the information about a person to capture high-level themes, such as whether the biography is about a scientist or an artist. On the other hand, local conditioning describes how the previously generated tokens relate to the information in the infobox, allowing for more context-aware generation.

In the [13] approach, the authors create a dataset called T-REx, which is designed for large-scale alignment between free text documents and Knowledge Base (KB) triples. T-REx includes 3.09 million Wikipedia abstracts that are aligned with 11 million Wikidata triples, covering over 600 different unique Wikidata predicates. This dataset provides a comprehensive alignment between the text documents and KB triples, allowing for effective training and evaluation of models that require such alignment for tasks like information extraction, entity linking, and relation extraction.

In the approach [1], the authors tackle the task of verbalizing the entire English Wikidata Knowledge Graph (KG) [59], and discuss the unique challenges associated with this broad, open-domain, and large-scale verbalization process. They demonstrate that converting a comprehensive and encyclopedic KG like Wikidata into natural text enables seamless integration with existing language models, providing a novel approach to integrate structured KGs with natural language corpora. The authors introduce datasets such as Text-KG aligned corpora (TeKGen) and Knowledge Enhanced Language Model (KeLM) as major contributions of their work.

2.1.3 Text to Fact Extraction

In [27], the paper presents three main contributions. They introduce the task of Multilingual Fact Linking (MFL), which is aimed at linking Knowledge Graph (KG) facts with their mentions in text, particularly when there is a language mismatch between the fact label and text. Then the authors introduce INDICLINK, which is a new evaluation dataset for MFL in English and six Indian languages. This is the first dataset of its kind for these languages. Lastly, they propose a new method called REFCOG, which

is a Retrieval+Generation model designed for MFL. The REFCOG model outperforms the standard Retrieval+Re-Ranking model for MFL.

In [2], the paper proposes a new approach for extracting informative and broad coverage triples from sentences. Instead of relying on a large pattern set, the authors suggest pre-processing the sentence using linguistically motivated techniques to create coherent clauses that are logically entailed by the original sentence and easy to segment into open Information Extraction (IE) triples. The approach involves two stages: first, a classifier is used to split the sentence into shorter utterances, and then natural logic is used to further shorten these utterances while preserving the necessary context. By doing so, the authors aim to reduce the burden of extracting triples from a large pattern set while still maintaining a high level of accuracy.

In [65], the authors describe the problem of extracting entities and their relations from unstructured text has two subtasks - named entity recognition and relation extraction. Prior research had suggested that joint models could improve accuracy by better capturing the interactions between entities and relations. However, in this paper, the authors propose a simple approach that utilizes two independently trained encoders built on top of deep pre-trained language models. These models are referred to as the entity model and relation model, respectively, and the relation model only relies on the entity model for input features. The entity model builds on span-level representations, while the relation model builds on contextual representations specific to a given pair of spans. Despite its simplicity, this pipelined approach outperforms all previous joint models on three standard benchmarks using the same pre-trained encoders.

2.2 Multilingual Bias Detection and Mitigation

Despite bias being so important for the operation of large sources of free and fair information, its detection and mitigation in natural language processing is predominantly focused on English language text. Although some efforts have been made to apply these techniques to other languages, most of the research is limited to English due to the availability of resources and data. This creates a significant gap in the ability to identify and mitigate bias in other languages, underscoring the necessity for more multilingual research in this area.

2.2.1 Dataset

Every effort was made to detect each kind of bias, especially in the case of encyclopedic knowledge sources like Wikipedia, using custom datasets initially. With the introduction of [49], the data was then standardized as one of the datasets. In their research, the authors created the first parallel corpus of biased language by collecting 180,000 sentence pairs from Wikipedia edits. These edits removed various framings, presuppositions, and attitudes from biased sentences. Furthermore, they proposed two strong encoder-decoder baselines for the task. The first system, called CONCURRENT, uses a BERT encoder to identify subjective words during the generation process. The second algorithm, MODULAR, separates

the steps and uses a BERT-based classifier to identify problematic words and a novel join embedding to edit the hidden states of the encoder. The authors conducted large-scale human evaluations across four domains, including encyclopedias, news headlines, books, and political speeches. These evaluations suggest that their algorithms are a first step towards the automatic identification and reduction of bias.

The other relevant dataset is the WIKIBIAS corpus, as created through the approach outlined in [64]. The corpus is created by first extracting Wikipedia revisions where editors provide justifications for Neutral Point of View (NPOV), which is a standard for unbiased writing. This automatically labeled data is then further annotated with fine-grained bias types at the span-level through a two-stage human annotation methodology, resulting in the creation of clean ground truth. This approach differs from previous works that only annotated on the sentence-level. The authors highlight the importance of this dataset for improving the identification and reduction of bias in natural language processing.

Expanding on the work done on the WIKIBIAS corpus, the authors conduct a comprehensive analysis of subjectivity bias in text through three sub-tasks: bias classification, tagging biased segments, and neutralizing biased text. The study reveals that current state-of-the-art models have difficulty detecting multi-span biases, even though their performance is reasonable, indicating that the WIKIBIAS dataset can serve as a useful benchmark. Additionally, the authors demonstrate that models trained on the WIKIBIAS corpus can generalize well to various domains, including news and political speeches.

2.2.2 Detection and Mitigation Approaches

There were multiple monolingual approaches for bias detection and mitigation that were explored. In [52], the authors conducted an analysis of biased language using Wikipedia’s edits and identified two significant classes of bias-driven edits. The first class, framing bias, is characterized by subjective words or phrases associated with a particular point of view. The second class, epistemological bias, is related to linguistic features that subtly focus on the believability of a proposition, often through presupposition. The authors provide a comprehensive analysis of these two classes of bias-driven edits and the underlying linguistic features to better understand the nature of biased language in natural language processing.

The authors in [49] elucidate their innovative dual-pronged approach, comprising of two distinct algorithms. The initial algorithm, dubbed MODULAR, encompasses two sequential stages: BERT-based detection and LSTM-based editing. The former stage entails leveraging a pre-trained BERT model to discern problematic lexemes, while the latter stage entails training a distinct LSTM model to manipulate the hidden states of the encoder based on the identified problematic words. The second algorithm, known as CONCURRENT, employs an encoder-decoder neural network that harnesses BERT as the encoder and incorporates an attentional LSTM with copy and coverage mechanisms as the decoder. This system facilitates the direct generation of a neutralized rendition of the problematic source text, thereby rendering it more convenient to operate and train.

In [64], multiple binary classifiers were trained using different data splits in the study. The first classifier was trained using only human-annotated WIKIBIAS-MANUAL data, while the second one was trained on WIKIBIAS-AUTO data. Two methods from previous research were also tested to enhance



Clickbait tweet	Spoiler
 Lifehacker  @lifehacker How to keep your workout clothes from stinking: lifehack.kr/57Y0uEZ	"washing [them]"
 New York Post  @nypost Just how safe are NYC's water fountains? nyp.st/2yHSGnr	"The Post independently tested eight water fountains in New York City's most frequented parks, and found that all met or exceeded the state's guidelines for water quality."
 CNBC  @CNBC A Harvard nutritionist and brain expert says she avoids these 5 foods that "weaken memory and focus." (via @CNBCMakeIt) cnb.cx/2TG6zeX	"1. Added sugar" [...] "2. Fried foods" [...] "3. High-glycemic-load carbohydrates" [...] "4. Alcohol" [...] "5. Nitrates" [...]

Figure 2.1: Description of the task of ClickBait Spoiling: For each tweet that serves as a clickbait, the spoiler is also displayed.

the performance of classifiers with noisy labels. One of them involved finetuning the model trained on noisy labels further by incorporating clean data, while the other involved training on a filtered version of WIKIBIAS-AUTO data with the top-5% and top-10% of automatically labeled “biased” instances with the lowest possibility removed. The latter method used a classifier trained on the original WIKIBIAS-AUTO data.

2.3 Task: Clickbait Spoiling

Most studies on clickbait operate under the assumption that it is a data-driven strategy used to optimize social media posts by exploiting the “curiosity gap” concept introduced in the paper [36]. This idea was also shared by Peter Koechley, CEO of Upworthy [26], a platform that played a significant role in popularizing clickbait on Facebook. The success of Upworthy and other similar platforms prompted Facebook to revise its news recommendation algorithms twice to reduce the amount of clickbait content [45]. In [53], the paper aims to address the problem of clickbait posts on social media and their impact on users and collect a large dataset of 1.67 million Facebook posts from over 150 U.S. based media organizations to analyze clickbait practices with their impact.

2.3.1 Spoiler Type Classification

In their first approach of such kind [7], the authors propose a comprehensive solution to address clickbait in media outlets. It starts with building a classifier that automatically detects whether a headline is clickbait or not. Then, it explores ways to block certain clickbaits from appearing in different websites. To address the variation in readers' preferences, personalized classifiers are developed to predict whether a particular clickbait should be blocked based on the reader's previous block and click history. Such integrated pipelines aimed to discourage media outlets from relying on clickbait to attract visitors to their sites.

In the best performing approach of [66] in the shared task organised by [48], the author utilized a self-attentive network to tackle the problem of annotation distribution prediction. The self-attentive RNN applied a token-level attention mechanism over the hidden states generated by the RNN to infer the levels of importance of tweet tokens in predicting the annotation distribution. This approach is different from regular external-attentive networks, as the self-attentive network does not require any external information to learn the context vector. Instead, the inferred information can be used as the tokens' weights when aggregating their hidden states into the vector representation of the tweet.

2.3.2 Clickbait Generation

In certain approaches, the task of clickbait generation was experimented with before the task of Clickbait Spoiler Generation. [55] introduces a new method for generating stylized headlines to identify clickbait from original documents. The method consists of two learning components: a generator component that includes a document auto-encoder and a headline generator, which generates headlines with specific styles based on the latent content representations and style vectors. The discriminators are used to regulate the generation process by incorporating necessary constraints. A pair discriminator preserves the relationships between documents and their correspondent headlines, while a style discriminator maximizes the differentiability of styles for original and generated headlines. Lastly, a transfer discriminator is introduced for adversarial training to ensure that the generated headlines and transferred headlines with the same styles are similar in distribution. The overall goal is to deceive the transfer discriminator so that it cannot distinguish between original and generated headlines easily.

In [60], the authors propose a first-ever model for sensational headline generation that utilizes reinforcement learning techniques and they develop a distant supervision strategy for training a sensationalism scorer as a reward function without relying on human-annotated data. They also introduce a novel loss function called "Auto-tuned Reinforcement Learning" that balances between Maximum Likelihood Estimation (MLE) and Reinforcement Learning (RL) using dynamic weights. It allows for a more precise optimization process.

2.3.3 Paragraph-based Information Retrieval Techniques

Spoiler extraction is a specific application of the passage retrieval task. Instead of retrieving informative paragraphs or sentences, spoiler extraction involves extracting specific paragraphs that reveal the plot or key events of a movie, book, or TV show. Therefore, spoiler extraction is an important aspect of the paragraph retrieval task, and some of the paragraph retrieval-based approaches can be used for the task of spoiler extraction.

In [33], the survey discusses the task of text ranking, which involves generating an ordered list of texts in response to a query. While most commonly used for search, it is also relevant in many natural language processing applications. The survey provides an overview of text ranking with neural network architectures, specifically transformers, such as BERT. The combination of transformers and self-supervised pretraining has revolutionized the fields of natural language processing and information retrieval. The survey covers two categories: transformer models that perform re-ranking and learned dense representations that perform ranking directly. The survey also discusses techniques for handling long documents and the tradeoff between effectiveness and efficiency.

Chapter 3

Multilingual Fine-grained Entity Categorization in Wikipedia Text

In this chapter, we venture into the multilingual setting, dealing with categorization of Wikipedia entities. With this study, we gain an understanding of methods for multilingual understanding and knowledge representation in NLP. The goal of entity categorization - the problem that we deal with in this chapter - is establishment of a unified ontology, which can result in the availability of large, unified knowledge bases which make use of unharnessed knowledge from the corpora in several low resource languages. Such structured knowledge bases across languages can help improve the performance of NLP methods in many other tasks that fall under the generic umbrella of information access. In this study, we deal with the Wikipedia data from 30 different languages for classifying the Wikipedia entities in those languages into 219 fine-grained categories. Graph-based approaches form a major focus of our modeling, where we experiment with several monolingual baselines and propose a Graph Neural Network (GNN)-based multilingual architecture making use of XLM-RoBERTa and bi-GRU for modeling, which decisively improved the performance against the baselines. This work was done as a part of the NTCIR-15 Shinra 2020-ML Classification Task [54], where our proposed method ranked second in the overall evaluation.

3.1 Multilingual Entity Categorization: Motivation

Wikipedia, being one of the largest knowledge sources on the internet, provides a rich source of structured and unstructured data for many NLP tasks. However, the challenge is to extract and structure the knowledge in a way that can be used for inference and reasoning. Entity-centric articles on Wikipedia provide a wealth of information on various topics, ranging from people, organizations, events, and locations to abstract concepts.

To get the most of this information, researchers and developers have created various resources making use of the large amount of knowledge in the Wikipedia corpus, such as DBpedia, Wikidata, Freebase, YAGO, and others. These resources aim to structure and organize the knowledge from Wikipedia for various NLP applications, such as named entity recognition, entity linking, question answering, and knowledge graph construction.

However, the current structured knowledge bases are primarily created through bottom-up crowdsourcing, which can lead to inconsistencies in the structure of the knowledge base. For instance, different editors may use different naming conventions or have different opinions on the relationships between entities, leading to duplicate or conflicting entries in the knowledge base. This method is as well very expensive, which hinders the scalability to larger amounts of data. Enhancement to low resource languages with fewer speakers and limited knowledge of common languages such as English makes the extension of the knowledge bases to such languages even more difficult. Besides this, adaptation of the knowledge bases to the changes in corpora over time is a formidable challenge.

To address these issues, researchers have proposed various methods for automatically extracting structured knowledge from Wikipedia. These methods include entity linking, which involves linking the entities mentioned in text to the corresponding entries in the knowledge base, and entity extraction, which involves identifying and extracting the entities from the text. By leveraging the rich information available in Wikipedia, these methods can create structured knowledge bases that are more consistent and reliable than those created through crowdsourcing.

The NTCIR-15 Shinra 2020-ML Classification Task aims to address the challenges in structuring knowledge in Wikipedia and creating a knowledge base that is accurate and consistent. This task involves the classification of 30 language-specific Wikipedia entities into 219 categories based on the Extended Named Entity (ENE) ontology. ENE is a four-layered ontology that covers names, time, numbers, and concepts. The Shinra ML Task uses categorized Japanese Wikipedia pages and inter-language links to the corresponding pages in other languages to classify the entities. The goal is to create a structured knowledge base that can be used for various natural language processing (NLP) tasks. This task thus attracted interest from researchers in the NLP community and led to the development of several approaches to tackle the problem. The problem is motivated to facilitate the creation of structured knowledge bases using information present in the Wikipedia articles across different languages. The creation of a structured knowledge base from Wikipedia can benefit a wide range of applications, such as information retrieval, question answering, and natural language understanding.

3.2 Introduction and Approaches to Entity Categorization

In Table 3.1, we can find a few examples of fine-grained categorization of the entity specified in different languages, to a predefined type. As explained in Section 3.1, we perform such entity categorization over a dataset consisting of 30 languages, constituting 219 predefined categories.

Several Transformer-based models that can perform various NLP tasks across multiple languages have been proposed recently, including translation, summarization, question generation, and sentiment analysis. In our work, we focus on multi-lingual text categorization, specifically on the classification of Wikipedia entities into various categories, using the Transformer architecture as a backbone. Based on pre-existing literature, multilingual transformer-based models such as mBERT, XLM, XLM-R, Unicoder, and InfoXLM are relevant to their problem, as these models are capable of performing multi-lingual text

Language	Entity Name	English Translation	Fine-grained category
Japanese	東京都	Tokyo	Province
Hindi	प्रशांति निलयम	Prasanthi Nilayam	Worship_Place
Russian	ЗВОНЯЩИЙ в полночь	Caller at midnight	Broadcast_Program
Spanish	Sala de cri- sis de la Ca- sa Blanca	White House Crisis Room	Facility_Part
French	Sneaker Pimps	Sneaker Pimps	Show_Organization

Table 3.1: Examples of entity categorization in four different languages.

classification. At the time of its implementation, this work was one of the first to use transformer-based models for multi-lingual categorization of Wikipedia entities. By leveraging these models, we achieved higher accuracy in entity classification while also being able to process text in multiple languages.

Initially, we conducted experiments to categorize entities in Hindi using various models such as MPAD, MAGNET, DTMT, and a named entity recognition-based model. Following this, we tested the performance of multi-lingual models like mBERT and XLM-R on entity categorization for 29 languages, excluding Greek due to differences in the input dataset format. However, the results were not satisfactory. Hence, we proposed a novel multi-lingual approach RNN_GNN_XLM-R, which resulted in the best-reported results.

We explored different models to categorize entities in Hindi and then extended their experimentation to multi-lingual models to categorize entities in the remaining languages. The use of mBERT and XLM-R did not lead to satisfactory results, indicating the need for a new approach. Our proposed RNN_GNN_XLM-R approach, however, outperformed other models in entity categorization. The details of this approach and the reasons for its effectiveness can be found in Section 3.5.

Our model achieved a micro-F1 average score of 73.7 across 30 languages. In comparison to other teams that participated in the NTCIR-15 Shinra 2020-ML Classification Task, the model’s performance was ranked second in the overall evaluation. While this task aimed to classify Wikipedia entities into 219 categories defined in Extended Named Entity (ENE) using categorized Japanese Wikipedia pages and inter-language links to corresponding pages in other languages, it provided a benchmark to evaluate models that classify entities across multiple languages.

3.3 About the Dataset

In this section, we describe the dataset provided for the task and the preprocessing applied prior to experimentation. We also brief on the taxonomy of categories in the dataset before elaborating on our multilingual dataset based on SVO and SOV languages.

3.3.1 Original Dataset

We used a dataset that included over 5 million Wikipedia entity pages, which were categorized into 219 different categories across 30 different languages. This dataset was provided as part of the SHINRA2020-ML Task. More information about it can be found in the task overview paper [54].

3.3.2 Preprocessing

For our experimentation, we rely on the Wikipedia Cirrus Dump available for each of the 30 languages. We use this dataset for training, validation, and testing of our models. Our focus is on the “text” section of the dump which we treat as the input to our models. Before we use the data, we preprocess it by masking numerical values, removing white spaces and punctuation characters. Additionally, we remove any characters of other languages from the “text” field, as well as any hyperlinks present in the data. This preprocessing step ensures that our models only operate on text data in the language of interest and improves the quality of the input data.

3.3.3 Taxonomy

The dataset has been hierarchically annotated into four levels, where each data sample can have multiple labels belonging to any of the four levels. The five topmost categories, namely Name, Timex, Numex, Concept, and Ignored, constitute the first level. The sub-categories of Name, Timex, and Numex are further divided into three more levels, while the last two topmost categories are the leaf nodes in the type hierarchy.

3.3.4 Multilingual Dataset Creation

We created a multi-lingual dataset by carefully selecting 3.8 million entity pages from a subset of 18 languages out of the 30 available. We chose the subset based on two criteria: syntactic word order and the availability of training data in each language. To ensure a diverse set of languages, we used syntactic word orders, specifically subject-verb-object (SVO) and subject-object-verb (SOV), as the primary criteria for selection. We believe this choice covers most of the world’s spoken languages. Additionally, we considered the number of training data available in each language as a secondary criterion.

We then randomly shuffled the obtained multi-lingual dataset without replacement to ensure a proper mix of different languages in the training batch. To create the training and validation datasets, we used a

Languages (SVO Order)	Training Data
Chinese	267,107
English	439,354
French	318,828
Finnish	144,750
Italian	270,295
Polish	225,552
Portuguese	217,896
Russian	253,012
Spanish	257,835
Swedish	180,948
Vietnamese	116,280

Table 3.2: Languages selected for creation of multi-lingual dataset along with training data statistics, using SVO order.

Languages (SOV Order)	Training Data
Dutch	199,983
German	274,732
Hindi	30,547
Hungarian	120,295
Korean	190,807
Persian	169,053
Turkish	111,592

Table 3.3: Languages selected for creation of multi-lingual dataset along with training data statistics, using SOV order.

stratified split ratio of 80:20. The languages selected for the creation of the multi-lingual datasets using SVO order are listed in Table 3.2 and using SOV order in Table 3.3.

3.4 Baseline Approaches for Entity-Type Classification

As part of our experiments, we have tested several state-of-the-art models to serve as our baselines for the task. To account for the fact that a majority (approximately 98%) of the entity pages, particularly in Hindi, have only one label, we first evaluated the Message Passing Attention Networks for Document Understanding (MPAD) [41]. This approach is known to be highly effective in single label document

classification. Additionally, we explored multi-label text classification techniques such as the Attention-based Graph Neural Network (MAGNET) [42].

Given the entity-centric nature of the task, we also designed a model that is specific to entities, using a combination of CNNs and RNNs. However, as all of these methods utilized Gated Recurrent Units (GRUs), we switched to the DTMT encoder approach to further improve the effectiveness of our RNN models.

3.4.1 MPAD

The Message Passing Attention Networks for Document Understanding (MPAD) [41] approach is used to represent documents in the form of word co-occurrence networks. In this method, GNNs are used along with an improved COMBINE step to create an intermediate representation for each node in the graph. These intermediate node representations are then pooled in the READOUT step to create a representation for the entire document. MPAD exhibits excellent performance on five out of ten standard text classification datasets and delivers competitive results with the current state-of-the-art on the remaining datasets. It should be noted that the creators of MPAD also designed hierarchical variants of the method to capture document hierarchy, but we did not employ them in this particular sub-task.

3.4.2 DTMT

In their work, [38] proposed a novel approach to improve the RNN-based Neural Machine Translation (NMT) by increasing the depth of the transition between consecutive hidden states. They introduced a new architecture called Deep Transition RNN-based Architecture for Neural Machine Translation (DTMT), which uses multiple non-linear transformations to reinforce the hidden-to-hidden transition and capture linear transformation path to prevent the issue of gradient vanishing. In our task, we leveraged the bi-directional encoder from this architecture and utilized the final hidden state of the last encoder token as input to a linear layer for sentence classification into multiple labels. This allowed us to benefit from the enhanced transition depth and the improved gradient flow in the DTMT architecture.

3.4.3 MAGNET

The Attention-based Graph Neural Network (MAGNET) proposed by [42] is a multi-label text classification model that leverages GNNs to capture the relationships between labels and classifiers. In this approach, each label is treated as a node in a graph, and the GNNs are used to implicitly learn the dependencies among them. MAGNET uses both the feature matrix and correlation matrix obtained from the GNNs to better capture these dependencies. This model has been shown to achieve state-of-the-art performance on several multi-label classification datasets, making it a promising approach for this type of task.

3.4.4 NER-based Model

As part of our task, we also explored a model that relied on the named entities contained within Wikipedia pages. To achieve this, we extracted the named entities present in the “text” field of the Wikipedia dump, along with their corresponding entity types. To combine this information with the text of each article, we utilized a CNN-RNN architecture. We first trained embedding layers for both the named entities and their entity types. Additionally, we fine-tuned the 300D fastText [6] word embeddings for our input text. As a result, each input text token was represented as a 900D vector, comprising a 300D fastText embedding, a 300D entity embedding, and a 300D entity type embedding. We then transformed this vector using three CNN layers to obtain a 100D representation for each token. These representations were subsequently passed through an LSTM, with the final layer connected to an output softmax layer for the final prediction.

3.4.5 Multilingual Models

We also evaluated the performance of two widely-used pre-trained multi-lingual Transformer architectures: mBERT-base [46] and XLM-R-base [30]. These models have 12 layers and are trained on a large amount of multi-lingual data. In our experiments, we fine-tuned these models on our multi-lingual dataset described in Section 3.3.4. We used the vector representation of the special [CLS] token from the last layer of the models, which captures the overall information of the input sequence, and connected it to a linear classification head to make the final prediction for each sample. By training on our dataset, we aim to optimize the model for the specific task at hand and evaluate its performance on our evaluation metrics.

3.5 Proposed Method - RNN_GNN_XLM-R

We proposed a novel architecture, “RNN_GNN_XLM-R”, which incorporates RNNs, Graph Neural Networks (GNNs), and Transformer modules. In this section, we will provide a clear description of the approach.

3.5.1 Overall Architecture

The overall architecture of our proposed model is illustrated in Figure 3.1.

In our architecture, the representation of each Wikipedia article is denoted as a sequences of words of length m , $D' = [b_1, b_2, \dots, b_m]$, where b_i represents the i^{th} word. The next step is to obtain input sub-word token by using a sentence-piece tokenizer on document D' , producing $D = [x_1, x_2, \dots, x_n]$ where x_i indicates the i^{th} subword. To obtain feature representation of each sub-word vector, the document D is fed into XLM-Roberta and the output from the last layer is used, resulting in $U = [u_1, u_2, \dots, u_n]$. To further enhance contextual representation, the input representation U is processed by a K -layer attentional bidirectional-GRU, and the output vectors from both directions are concatenated. Similar to the approach

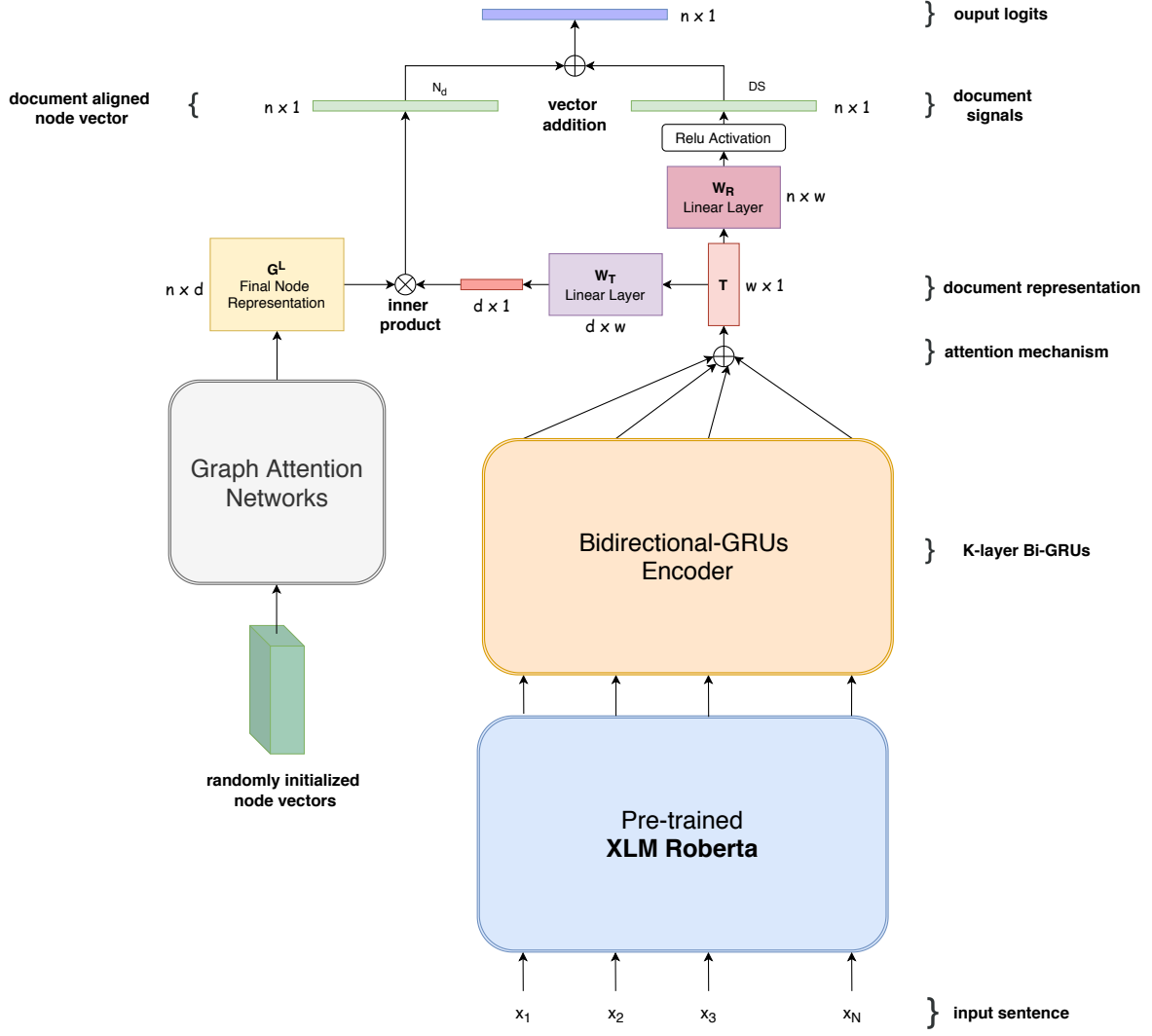


Figure 3.1: Architecture of our proposed system, RNN_GNN_XLM-R.

in [42], our model uses Graph Attention Networks (GAT) [58] to learn the relationships among the labels. In order to retain document level signals for classification, we apply a nonlinear transformation with ReLU activation to the document vector T . The final score for each label is calculated by adding it to the node representation and then connecting it to an output softmax layer.

3.5.2 Contextual Representation using RNN

The input document D is passed on to the XLM-Roberta model, which generates a feature representation of each sub-word vector. This results in a vector set U that contains n sub-word vectors denoted by u_1, u_2, \dots, u_n . The dimension of each sub-word vector is represented by w , which is determined by the output of the last layer of the XLM-Roberta model. We can represent this as follows:

$$U = \text{XLMRoberta}(D) \in \mathcal{R}^{n \times w} \quad (3.1)$$

Here, $u_i \in \mathcal{R}^w$ and w is the dimension of vector representation from last layer of the XLM-Roberta model.

The input representation U is next utilized to obtain an enhanced contextual representation through a K -layer attentional bidirectional-GRU, and the output vectors from both directions are concatenated. The transformation equations for the first layer of GRU can be expressed as follows:

$$\vec{h}_i^0 = \text{GRU}(u_i, \vec{h}_{i-1}^0) \quad (3.2)$$

$$\overleftarrow{h}_i^0 = \text{GRU}(u_i, \overleftarrow{h}_{i+1}^0) \quad (3.3)$$

$$h_i^0 = [\vec{h}_i^0, \overleftarrow{h}_i^0] \quad (3.4)$$

Here, u_i is the output vector from the last layer of XLM-Roberta. The equations used to transform the k^{th} layer (excluding the first layer) are given as follows:

$$\vec{h}_i^k = \text{GRU}(h_i^{k-1}, \vec{h}_{i-1}^k) \quad (3.5)$$

$$\overleftarrow{h}_i^k = \text{GRU}(h_i^{k-1}, \overleftarrow{h}_{i+1}^k) \quad (3.6)$$

$$h_i^k = [\vec{h}_i^k, \overleftarrow{h}_i^k] \quad (3.7)$$

In order to create the document's representation, attention mechanisms are utilized to combine the hidden states from the final layer, which is denoted as $k = K$, in the following manner:

$$p_i = \tanh(W_p h_i^K) \quad (3.8)$$

$$a_i = \frac{\exp(Q p_i)}{\sum_j \exp(Q p_j)} \quad (3.9)$$

$$T = \sum_i a_i h_i^K \quad (3.10)$$

Here, W_p and Q are trainable weights in the network and $T \in \mathcal{R}^w$ is the document representation vector. The attention mechanism helps the model to give more weightage to the significant words that play a crucial role in document classification, and therefore it can build a more effective representation of the document.

3.5.3 GAT for Capturing Label Relatedness

Our approach, similar to the method proposed in [42], utilizes Graph Attention Networks (GAT) [58] to learn the relationship between labels. Specifically, we represent each of the 219 labels from the ENE (version 8) set as a node in a graph G . The connectivity between the nodes is determined by the adjacency matrix A , which is initialized using the Xavier initialization [18] method. The adjacency matrix is also trained alongside the model to capture the correlation between different classification labels. In this way, the model can focus on the most relevant labels for a given document and build a better classification representation. The process can be represented as:

$$G^k = \text{GAT}(G^{k-1}, A) \quad (3.11)$$

Here, $G^k \in \mathcal{R}^{n \times d}$ is combined vector representation for n nodes (here $n=219$) of the graph at step k and $A \in \mathcal{R}^{n \times n}$ is the adjacency matrix.

We set the GAT to run for L iterations, and at the end, to obtain a node vector (N_d) aligned with the underlying document D , we combine the vector representation of nodes from the last step (i.e., $k = L$) of GAT by linearly transforming the document representation T and multiplying it with the final node representation $G^L \in \mathcal{R}^{n \times d}$.

3.5.4 Final Scoring

In order to maintain the signals at the document level for classification purposes, we apply a ReLU activation function to the document vector T . The final score for each label is generated by adding the node representation to the transformed document vector, and then passing it through an output softmax layer. This allows us to obtain a probability distribution over the labels for the given document.

$$DS = \text{ReLU}(W_R T) \quad (3.12)$$

$$\text{logits} = N_d + DS \quad (3.13)$$

Here, $W_R \in \mathcal{R}^{n \times w}$ are learnable parameters.

The training process of the model was conducted in an end-to-end manner, utilizing categorical cross-entropy loss on the multi-lingual dataset that was introduced in Section 3.3.4.

Methods	Training Data	Micro-Averaged F1-score
MPAD	Mono-lingual	59.5
DTMT	Mono-lingual	61.9
MAGNET	Mono-lingual	61.3
NER based model	Mono-lingual	61.1
mBERT-base	Multi-lingual	66.5
XLM Roberta	Multi-lingual	68.1
RNN_GNN_XLM-R	Multi-lingual	72.1

Table 3.4: The F1-score reflected from the task leaderboard submission for Hindi evaluation dataset.

3.6 Experiments & Results

3.6.1 Evaluation

The objective of the model is to accurately classify each page into one or more of the 219 taxonomy categories. The model only receives a score if it accurately predicts the category, and it is evaluated using the micro-averaged F1 measure, which is the harmonic mean of micro-averaged precision and micro-averaged recall.

In order to assess the effectiveness of our proposed model, RNN_GNN_XLM-R, we conducted experiments in addition to the baseline models described in Sections 3.4.1 through 3.4.5. The experiments were conducted on the Shinra2020 ML Task leaderboard dataset, which contained 2000 data samples with a distribution that was not related to the training dataset. The evaluation of the models was performed using the micro-averaged F1 metric, which was provided by the organizers of the task. This metric takes into account both the precision and recall of the model’s predictions, and produces a single score that reflects the overall performance of the model on the dataset. Results from the leaderboard are described in Table 3.4 on the Hindi language for the monolingual methods and in Table 3.5 for the multilingual methods on all the languages.

In the final evaluation stage, we were required to submit our model predictions on the complete test set, which could be in any number of languages. The statistics of the entire test set were undisclosed to the participants. The organizers evaluated the model’s performance using the micro-averaged F1 metric. The results on the entire test set are reported in Table 3.6.

3.6.2 Monolingual Results for Hindi

In order to evaluate the performance of our models on the Hindi dataset, we experimented with all the approaches discussed in Sections 3.4 and 3.5. We trained the models on the Hindi language dataset exclusively. We applied the MPAD approach by selecting only the data points with a single label and tested the model on them. On the other hand, for the MAGNET, NER-based model, and DTMT

Language	Models					
	HUKB	PribL	uomfj	mBERT-base	XLM-R-base	RNN_GNN_XLM-R
English (en)	51.2	73.9	74.5	72.1	74.4	76.0
Spanish (es)	56.7	75.1	73.2	70.7	73.1	75.4
French (fr)	48.9	73.5	70.4	73.3	73.1	74.3
German (de)	61.1	75.8	71.3	72.5	70.8	75.3
Chinese (zh)	60.1	75.4	72.5	70.5	73.0	75.8
Russian (ru)	55.0	74.5	69.8	70.9	70.3	73.4
Portugese (pt)	51.9	71.0	69.5	69.5	68.9	72.4
Italian (it)	49.0	73.4	72.4	70.0	71.9	74.3
Arabic (ar)	51.0	70.7	70.2	67.1	69.6	72.3
Indonesian (id)	54.0	-	71.2	72.9	74.4	74.5
Turkish (tr)	58.2	73.2	69.5	72.6	73.9	73.9
Dutch (nl)	60.3	73.8	70.8	72.3	72.9	72.9
Polish (pl)	61.6	76.6	73.5	71.8	74.2	75.1
Persian (fa)	60.9	-	73.8	70.0	72.9	73.9
Swedish (sv)	59.6	-	69.7	70.9	69.9	73.4
Vietnamese (vi)	61.7	72.2	72.1	74.4	75.4	74.7
Korean (ko)	53.8	74.6	72.3	71.2	71.2	70.8
Hebrew (he)	52.2	-	68.6	68.9	68.0	72.1
Romanian (ro)	53.2	-	69.8	72.3	71.7	75.0
Norwegian (no)	50.2	71.7	70.8	68.5	70.5	72.2
Czech (cs)	53.3	69.2	68.1	68.0	69.5	72.2
Ukrainian (uk)	58.2	69.6	70.5	70.5	70.0	71.5
Hindi (hi)	44.1	60.5	65.9	66.5	68.1	72.1
Finnish (fi)	51.9	-	73.1	73.2	73.0	72.9
Hungarian (hu)	54.9	-	71.4	71.7	72.8	74.6
Danish (da)	52.7	72.2	73.2	70.4	71.4	73.8
Thai (th)	60.3	-	50.3	66.8	70.9	75.2
Catalan (ca)	43.9	-	71.6	72.1	70.9	71.9
Bulgarian (bg)	56.7	-	74.7	75.2	73.9	76.7
Average	54.7	72.5	70.5	70.9	71.7	73.7

Table 3.5: Micro-averaged F1 score comparison across various models on the leaderboard test set. HUKB, PribL and uomfj were the other top performing teams on the task challenge leaderboard. mBERT-base and XLM-R-base are our other baselines. RNN_GNN_XLM-R is our best proposed model. Best results for each language are highlighted in bold.

Languages	Teams				
	ousia	uomfj	LIAT	PribL	RH312 (ours)
Bulgarian	-	83.07	75.2	-	82.13
French	81.01	78.21	76.88	78.52	80.31
Hindi	69.75	66.67	16.49	-	71.7
Indonesian	-	78.51	72.44	-	77.55
Thai	76.36	65.02	49.58	-	76.77
Turkish	-	84.85	77.19	84.36	83.28

Table 3.6: Micro-averaged F1 score comparison across various models on the official evaluation dataset for 6 languages. ousia, uomfj, LIAT and PribL are the other top performing teams on the task challenge leaderboard. RH312 (i.e., RNN_GNN_XLM-R) is our best proposed model. Best results for each language are highlighted in bold.

encoder, we used all the data points during testing. Table 3.4 presents the results obtained only for the Hindi language from our Shinra leaderboard submission. We also evaluated the performance of the mBERT-base and XLM Roberta models trained on the multi-lingual training dataset on the Hindi leaderboard dataset.

The table validates that Transformer-based approaches perform better than RNN-based approaches. It is also observed that MAGNET yields better results than MPAD. DTMT, which employs an improved RNN encoder, outperforms traditional RNN encoder models like MPAD and MAGNET. Surprisingly, the entity-aware NER based model does not perform well. It is expected that multi-lingual training will produce better results than mono-lingual training, and this is confirmed by the results. Our proposed model, RNN_GNN_XLM-R, is superior to other transformer-based models, and this is attributed to its ability to leverage correlations between class labels and text semantics to improve predictions.

3.6.3 Multilingual Results

We retrieved the official leaderboard results from Shinra [54] and added our proposed system, RNN_GNN_XLM-R, and multi-lingual baseline models to the table, along with some other top-performing participant models. Our RNN_GNN_XLM-R model was trained on 18 languages, and we performed zero-shot inference on 11 additional languages, namely Arabic, Bulgarian, Catalan, Czech, Danish, Hebrew, Indonesian, Norwegian, Romanian, Ukrainian, and Thai. Table 3.6 displays the official leaderboard results, which were published on September 30th, 2020.

Our team has submitted the model predictions for six languages to the Shinra evaluation system. and have thus received official results for these six languages. The official evaluation results include the best micro-averaged F1-score obtained by our model as well as the scores achieved by some of the other competitive models in their respective languages. The performance of our proposed model and the

baselines are evaluated using the micro-averaged F1 metric. The official evaluation results are presented in Table 3.6 for the six languages.

3.7 Conclusion

In this work, we delved into the problem of fine-grained entity type classification across multiple languages. In order to tackle this issue, we explored the effectiveness of several cutting-edge monolingual RNN encoders and multi-lingual Transformer-based models. Furthermore, we proposed a unique adaptation of the XLM-R model, where we augmented the XLM-R base with RNN and GNN modules. We carried out extensive experiments with these models on the datasets provided by the NTCIR-15 Shinra2020 ML task organizers, and evaluated the effectiveness of different models on the Hindi dataset as well as on datasets for six different languages.

In light of the experiments and evaluations, we found our proposed model RNN_GNN_XLM-R to outperform the other methods that we experimented with. This was attributed to a combination of factors, including the ability of XLM Roberta to extract rich semantics from the document, the attentional-RNN module’s capacity to enhance contextual information, and the GNN module’s ability to learn classification label correlations, ultimately resulting in improved generalization. Furthermore, the leaderboard and final evaluation results demonstrate the effectiveness of our models. Moreover, the fact that our model secured either the second or third position in most languages on the leaderboard validates the lack of language dependence of the models as well.

Chapter 4

A Cross-lingual Study in Fact-to-Text and Text-to-Fact Problems

Following a multilingual study in fine-grained entity categorization in Chapter 3, we move on to multilingual generative problems in this chapter. With the abundance of information existing in the knowledge bases of high resource languages such as English, one can leverage methods for enrichment of content in low resource languages, converting the English facts to informational texts in those languages. Wikipedia being an ideal platform for this study, we explore two cross-lingual problems: (1) generating content from facts in English, and (2) extracting facts in English from texts in different languages. We first explain the creation of a dataset aligning facts in Wikidata to the sentences in the Wikipedia corpus, which we call the XAlign dataset, and explain some of our experimental approaches on the problems proposed. The work carried out in this chapter establishes a solid base for studying the problems of multilingual bias identification and mitigation that we present in Chapter 5, as we work on a subset of languages we study in this work while continuing to work on the Wikipedia domain, and study the performance of applicable NLP models and tools for working in a multilingual setting.

4.1 On Cross-lingual Fact-to-Text Generation (XF2T)

Fact-to-text (F2T) generation is a fundamental task in natural language generation that involves converting structured data, such as fact triples, into natural language sentences. The generation of natural language from structured data has various practical applications, including automated dialogue systems and question-answering. Fact-to-text systems are crucial in these downstream applications, as they facilitate the transformation of structured data into human-readable text. The conversion of structured data into natural language sentences is a challenging task that requires significant research and development to ensure that the generated text is grammatically correct, semantically meaningful, and coherent. Therefore, F2T systems have garnered significant interest in recent years, and there is a growing body of research focused on developing more effective F2T models.

However, most F2T systems are limited to English and not available for low-resource (LR) languages. This is primarily because of the lack of training data in these languages. For instance, the number of Wikidata entries for person entities in LR languages is much smaller than in English. In this thesis, we

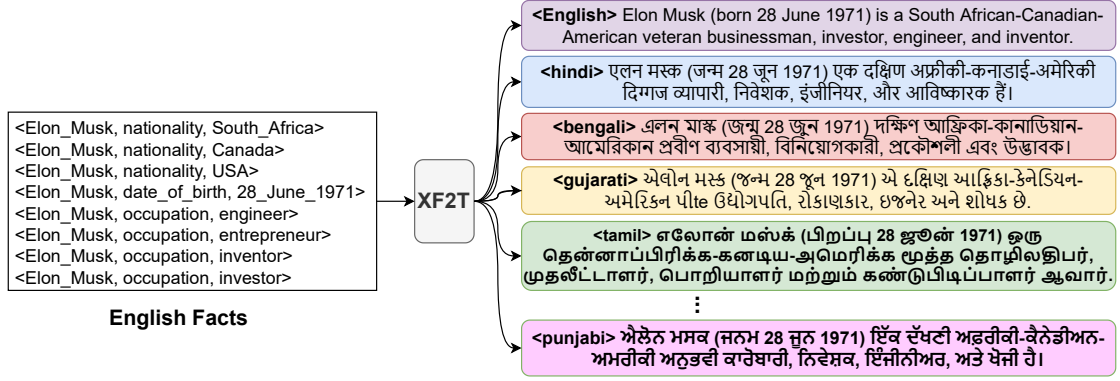


Figure 4.1: Examples of cross-lingual fact-to-text (XF2T) generation from facts in English to sentences in English or LR languages.

specially focus on a set of 11 LR languages native to the region of South Asia. Besides lack of sufficient content on Wikipedia as compared to English, these languages also suffer from data sparsity in that the average number of facts per entity in the Wikidata for these languages is almost half that of English, at 11.3 versus 22.8. Due to the data sparsity problem, monolingual F2T for LR languages is challenging. Therefore, we proposed a new task called cross-lingual F2T generation (XF2T), where we took a set of English facts as input and generated a sentence that captures the fact-semantics in the specified LR language. Here, we refer to a “fact” as a <subject, relation, object> triplet. Figure 4.1 shows an example of cross-lingual generation of content from English facts.

Training a cross-lingual F2T (XF2T) system poses a significant obstacle due to the requirement of aligned data, where English facts are semantically equivalent to LR language text. Creating such an aligned dataset can be a daunting task, as it requires extensive human annotation and is not easily scalable. Manual annotation is a time-consuming and laborious process that involves aligning every English fact with its corresponding text in the LR language, and ensuring that the alignment is semantically equivalent. Moreover, the creation of aligned data is highly dependent on the availability of bilingual annotators, which is not always feasible for low-resource languages. Hence, the development of an XF2T system is highly challenging due to the scarcity of aligned data for LR languages.

4.2 XAlign: A Cross-lingual Dataset for XF2T

The problem of XF2T we discussed earlier lacked a suitable dataset for experimentation. Hence, we first worked on creating a dataset for cross-lingual F2T generation, called XAlign, which contains high-quality pairs of English facts and semantically equivalent LR language text. The dataset includes a total of 0.55 million pairs, which has automatically aligned and human-aligned pairs for training and evaluation respectively. The dataset covers 12 different languages, including English and 11 LR South Asian languages. This section describes our methodology for data collection and cleaning followed by

the pipeline we adopted for automatically aligning the facts to sentences from English to LR languages. The languages in our dataset include English (en), Assamese (as), Bengali (bn), Gujarati (gu), Hindi (hi), Kannada (kn), Malayalam (ml), Marathi (mr), Oriya (or), Punjabi (pa), Tamil (ta) and Telugu (te). Next, we detail the annotation process for generating the gold data and brief the statistics of our created XAlign dataset.

4.2.1 Data Collection and Cleaning

To begin our work, we collected a list of 95018 person entities from Wikidata that are linked to a corresponding Wikipedia page in at least one of our 11 low-resource (LR) languages. We did this to create a dataset, which we denote as D , where each instance, d_i , is a tuple containing the entity ID, English Wikidata facts, the LR language, and the LR-language Wikipedia URL for the entityID. This dataset forms the foundation of our work on cross-lingual fact-to-text generation. To collect the facts for all the entities in our dataset D , we utilized the WikiData API¹ to extract the necessary information from the 20201221 WikiData dump for all the 12 languages. This allowed us to gather a comprehensive set of facts for each entity in multiple languages. To extract the most useful factual information from the WikiData dump, we limited ourselves to certain Wikidata property types: WikibaseItem, Time, Quantity, Monolingualtext. For each fact triple, we also kept any additional information that supports it as a fact qualifier. As a result, we obtained approximately 0.55 million data instances for all 12 languages.

To extract text from the 20210520 Wikipedia xml dump for each language, we used a tool called Wikiextractor [3]. Then, we split this text into sentences using a library called IndicNLP [29], which is specifically designed for processing text in South Asian languages. However, we also had to add a few extra rules to handle things like South Asian language-specific punctuation characters, sentence delimiters, and non-breaking prefixes. Overall, this allowed us to create a dataset of sentences in each of the 11 low-resource languages we were working with.

As a part of our data cleaning, we pruned out the following cases:

1. Sentences from other languages using the Polyglot language detector²
2. Sentences with less than 5 words or greater than 100 words
3. Sentences which could potentially have no factual information (sentences with no noun or verb³).

We also keep track of the section information for each selected sentence associated with a given Wikipedia URL.

¹<https://query.wikidata.org/>

²<https://polyglot.readthedocs.io/en/latest/Detection.html>

³For POS tagging, we used Stanza [50] for en, hi, mr, te, ta; LDC Bengali POS Tagger [4] for bn and ma; and [44] for gu. For other languages, we obtained a list of all entities from language-specific Wikidata, and then labeled mentions of these entities as proper nouns; for these languages we ignored the presence of verbs check.

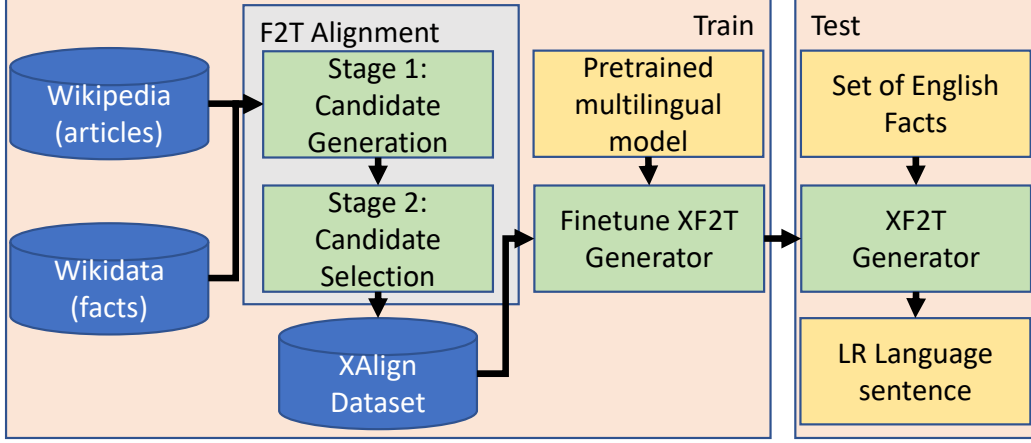


Figure 4.2: Adopted 2-stage pipeline for automatic fact alignment in creating the XAlign dataset followed by XF2T generation.

4.2.2 Automatic Pipeline for Fact Alignment

At this stage, our dataset D has a set F_{el} of English Wikidata facts and a set of Wikipedia sentences S_{el} in language l for every pair of (entity e , language l). To associate a sentence in S_{el} with a subset of facts from F_{el} , we propose a two-stage system for automatic alignment, as shown in Figure ??.

In the first stage, which we call “Candidate Generation”, we create pairs of (facts, sentence) candidates based on syntactic and semantic matches. This is done automatically using techniques such as named entity recognition, part-of-speech tagging, and dependency parsing.

In the second stage, which we call “Candidate Selection”, we use transfer learning and distant supervision methods to retain only the strongly aligned candidates. Transfer learning allows us to leverage knowledge from other aligned datasets, while distant supervision helps to overcome the challenge of limited human-annotated data. By combining these techniques, we are able to produce a high-quality aligned dataset that can be used for cross-lingual fact extraction tasks.

4.2.2.1 Stage 1: Candidate Generation

For each pair of a fact in English and a sentence in language l , we calculate a similarity score $sim(f_i, s_j)$ that measures both syntactic and semantic similarity. To achieve this, we use TFIDF to compare the sentence and fact either by translating the fact to l or the sentence to English. We also use MuRIL to compute the cosine similarity between the fact and sentence or their translations. The final score is an average of these four scores: $sim-MURIL(f_i, s_j)$, $sim-TFIDF(translate(f_i, l), s_j)$, $sim-TFIDF(f_i, translate(s_j, English))$, $sim-MuRIL(translate(f_i, l), translate(s_j, English))$. We use IndicTrans to translate sentences, and we only translate the non-entity parts of the fact while retaining the labels of the entities for which Wikidata multilingual labels are available in l . We keep a sentence if the most

similar fact has $\text{sim}(f_i, s_j) > \tau$, and we retain up to the top- K most similar facts for each sentence. We set $\tau = 0.65$ and $K = 10$ empirically for our experiments.

4.2.2.2 Stage 2: Candidate Selection

The first stage produces sentences associated with a maximum of K facts for each entity and language pair. We use two techniques to filter out weakly aligned (fact, sentence) pairs: NLI transfer learning and distant supervision from an English-only F2T dataset. Both methods take the input in “sentence<SEP>subject|predicate|object” format.

1. Transfer learning from NLI

In NLI, we predict whether a hypothesis entails, contradicts, or is neutral to a given premise. Fact to sentence alignment is similar to NLI, where the fact is like the premise and the sentence is like the hypothesis. So, we use NLI models like XLM-R, mT5, and MuRIL⁴ to infer (fact, sentence) alignment. We use their finetuned checkpoints from Huggingface’s Xtreme-XNLI, except for MuRIL which we finetuned for en, hi, and ur only. If the model predicts entailment, we consider the (fact, sentence) pair to be aligned. We evaluate the accuracy of each model by selecting a subset of facts from the K candidate facts for each sentence, and comparing them with the golden fact list.

2. Distant supervision

We created a binary classifier to predict whether an (English fact, LR language sentence) pair is strongly aligned using the Knowledge Enhanced Language Modeling (KELM) [1] dataset, which contains automatically aligned English (Wikipedia sentence, Wikidata facts) pairs. For every sentence in KELM, we created a positive instance for every fact aligned with it and a negative instance by randomly selecting a sentence from the same Wikipedia page. The dataset has ~ 1.3 million instances. Since the XAlign dataset is cross-lingual but KELM is English-only, we experimented with cross-lingual, translate-test, and translate-train settings for inference on Stage 1 output. Translate-train gave the best results, so we report the results using this setting.

4.2.3 Manual Annotation Process for Evaluation Dataset

We required annotated gold data for assessing Stage 2 and XF2T output. Annotation was performed in two phases, where annotators were given (LR sentence s , K English facts) from Stage 1 and asked to identify relevant facts, following specific guidelines and instructions detailed in Sections 4.2.3.1 and 4.2.3.2. The first phase involved 8 expert annotators who labeled 60 instances per language. In phase 2, 8 annotators per language were selected from the National Register of Translators⁵ and tested using phase

⁴Since MuRIL does not support vocabulary for all XNLI languages, it was finetuned for en, hi and mr only.

⁵<https://www.ntm.org.in/languages/english/nrtadb.aspx>

1 data as a control set. Up to 4 annotators per language were shortlisted based on their Kappa score with the control set. On average, each sentence required two fact triples for verbalization.

4.2.3.1 Annotation Guidelines

The objective was to identify the relevant English facts that were mentioned in the given LR language sentence. The annotator had to select all the applicable facts that could be deduced from the provided sentence by marking the corresponding checkbox. Additionally, it was necessary to specify whether the chosen set of facts covered the semantic information of the sentence partially or completely.

When a question was selected, the user was presented with a sentence in low resource (LR) language and a list of English facts. It was advised to read the LR sentence carefully and not rely solely on the provided English translated sentence as it may not always be accurate. The user was required to select the facts that could be inferred from the given sentence by selecting the checkbox against them. If the sentence was incorrect or incomplete, the user needed to mention the reason in the textbox at the bottom.

4.2.3.2 Annotation Instructions

1. Exact fact matching

Information in the sentences needed to be matched exactly (with few exceptions mentioned later, which had to also be followed strictly).

2. Implied information in facts

- (a) Facts that were related to language inference and did not require external knowledge were identified and marked.
- (b) If the fact or information required information or context from the external world, it was not marked.

3. Redundant facts

The facts that contained redundant information were not marked.

4. Abbreviations

If a part of the sentence was abbreviated in the facts or a part of the fact was abbreviated in the sentence, it was not considered.

5. Fact generalisation

- (a) If specific information was present in the sentence but there was no exact match in the fact list, then appropriate synonyms were selected.
- (b) If a fact contained more specific terms than the term present in the sentence, it was considered for annotation.

Lang	V	Train + Validation Splits			Manually Labeled Test Split				
		I	T	F	κ	A	I	T	F
en	103626	132584	20.2/4/86	2.2/1/10	0.74	4	470	17.5/8/61	2.7/1/7
as	27K	9K	19.23/5/99	1.6	-	1	637	16.22/5/72	2.2
bn	130684	121216	19.3/5/99	2.0/1/10	0.64	2	792	8.7/5/24	1.6/1/5
gu	34605	9031	23.4/5/99	1.8/1/10	0.50	3	530	12.7/6/31	2.1/1/6
hi	75037	56582	25.3/5/99	2.0/1/10	0.81	4	842	11.1/5/24	2.1/1/5
kn	87760	25441	19.3/5/99	1.9/1/10	0.54	4	642	10.4/6/45	2.2/1/7
ml	146K	55K	15.7/5/98	1.9	0.52	2	615	9.2/6/24	1.8
mr	49512	19408	20.4/5/94	2.2/1/10	0.61	4	736	12.7/6/40	2.1/1/8
or	28K	14K	16.88/5/99	1.7	-	2	242	13.45/7/30	2.6
pa	59K	30K	32.1/5/99	2.1	0.54	3	529	13.4/5/45	2.4
ta	121212	56707	16.7/5/97	1.8/1/10	0.76	2	656	9.5/5/24	1.9/1/8
te	60865	24344	15.6/5/97	1.7/1/10	0.56	2	734	9.7/5/30	2.2/1/6

Table 4.1: Basic Statistics of XAlign. $|I|$ = # instances, $|T|$ = avg/min/max word count, $|F|$ = avg/min/max fact count, $|V|$ = Vocabulary size, κ = Kappa score, $|A|$ = # annotators

4.2.4 Statistics of the XAlign Dataset

The statistics of the XAlign dataset for the automatically aligned train and validation splits as well as manually annotated test split can be seen in Table 4.1. We include the vocabulary count, number of instances, per-sentence word and fact count for the data along with annotation information such as number of annotators and the Kohen’s Kappa score per language. English, naturally has the most number of instances, but is closely followed by Bengali. The least number of instances were found for the Assamese language, which also had the least average fact count per sentence. The Malayalam language had the highest vocabulary size of 146k. Punjabi language has the longest sentences in the dataset, with an average of 32 words per sentence, which is distinctively higher than the second highest language Hindi, which has an average of 25.3 words per sentence. Telugu language turns out to have the shortest sentences at an average of 15.6 words per sentence, less than half of that for the Punjabi language. We were unable to report the Kohen’s Kappa score indicating the inter-annotator agreement for the Assamese language, since we had the availability of only one annotator, while for the Oriya language, there were no redundant (i.e. fact alignments marked in common by more than annotator) facts aligned.

Figure 4.3 visualizes the fact distribution statistics across all the 12 languages in our dataset. We bucketed the fact-aligned sentences in our dataset based on the number of facts they are aligned with in buckets of 1 to 4, and a common bucket for the ones aligned with 5 or more number of facts, and plotted their proportion in the overall dataset for each language. As can be seen almost all the languages exceed the 40% mark for the bucket corresponding to one fact aligned. The proportion decreases exponentially for increased number of facts, indicating that a large number of sentences ($\sim 70\%$) are aligned with only

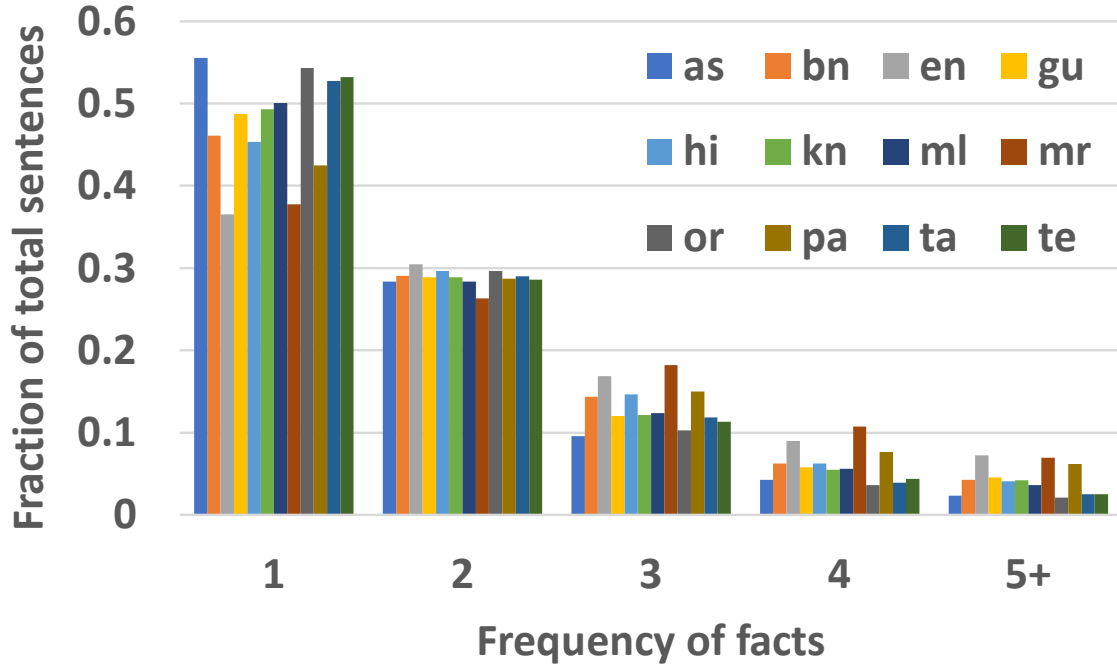


Figure 4.3: Fact count distribution across languages in the XAlign Dataset.

up to 2 facts. We can also observe spikes in the proportion of sentences for higher fact order buckets for the Marathi language followed by the English and Punjabi languages. These languages naturally also have dips in the initial buckets corresponding to low fact order buckets, indicating the nature of the sentences in languages to have relatively more number of facts aligned. We can find Assamese followed by Oriya, Tamil and Telugu to have higher sentences in the lowest fact order bucket, indicating the relatively less association of sentences with facts in these language-based data splits.

In a similar way as we did for language-based data splits, we visualize the fact count distribution for the training, validation and test splits in Figure 4.4. The nature of fact count distribution is expected to be the same for the training and validation splits because of their processing through the same automated 2-stage pipeline for fact alignment. We can observe $\sim 45\%$ of the sentences to be associated with only one fact, and $\sim 30\%$ of the sentences to be associated with 2 facts per sentence. The proportion of sentences with more than 5 facts aligned is very low, $\sim 4\%$ of the total sentences. We can observe minor differences against the test split, which was annotated by humans for fact alignment, however, the distribution is still consistent with the train and validation splits. The annotated sentences tend to have a slightly higher number of facts aligned against the automated ones in the training and validation splits, indicating a small amount of recall deficiency in the adopted technique.

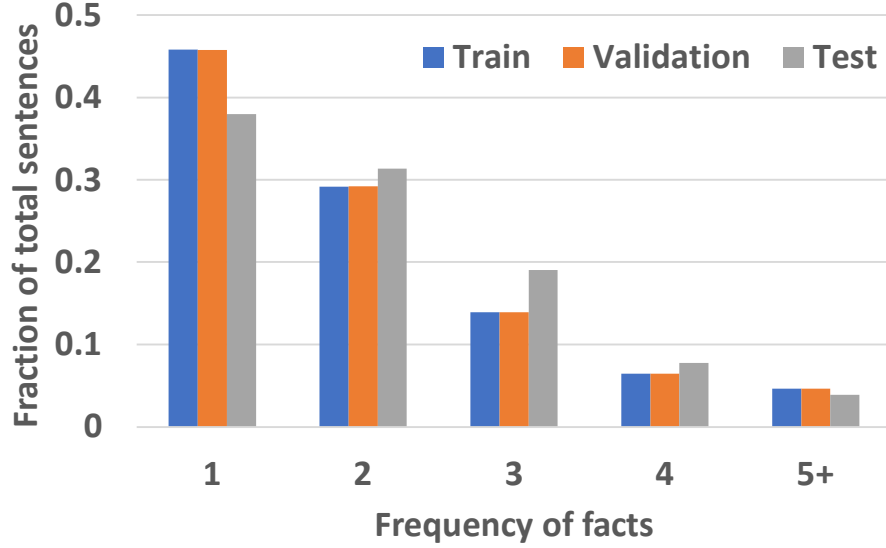


Figure 4.4: Fact count distribution across data subsets in the XAlign Dataset.

4.3 Experiments on XF2T

In this section, we discuss an approach we adopted for multilingual generation of text in our set of 12 languages from English facts as a part of our cross-lingual fact-to-text (XF2T) generation problem. First, we explain the input representation for XF2T followed by a discussion on fact-aware embeddings, which was performed to augment the model for XF2T generation.

4.3.1 Input Representation: Structure-Aware Input Encoding

An input instance is comprised of a section title t and multiple facts $F = \{f_1, f_2, \dots, f_n\}$, where each fact f_i contains a subject s_i , relation r_i , object o_i , and m qualifiers $Q = \{q_1, q_2, \dots, q_m\}$. Each qualifier q_j provides additional information about the fact and is linked to the fact through a fact-level property called qualifier relation qr_j . For instance, the fact “Narendra Modi was the Chief Minister of Gujarat from 7 October 2001 to 22 May 2014, preceded by Keshubhai Patel and succeeded by Anandiben Patel” has a subject of “Narendra Modi”, relation of “position held”, object of “Chief Minister of Gujarat”, and four qualifiers: (1) q_1 =“7 October 2001”, qr_1 =“start time”, (2) q_2 =“22 May 2014”, qr_2 =“end time”, (3) q_3 =“Keshubhai Patel”, qr_3 =“replaces”, and (4) q_4 =“Anandiben Patel”, qr_4 =“replaced by”.

Each fact f_i is encoded as a string and all fact strings in F are concatenated to form the input. The string representation of a fact f_i is “ $\langle S \rangle s_i \langle R \rangle r_i \langle O \rangle o_i \langle R \rangle qr_{i_1} \langle O \rangle q_{i_1} \langle R \rangle qr_{i_2} \langle O \rangle q_{i_2} \dots \langle R \rangle qr_{i_m} \langle O \rangle q_{i_m}$ ”, where $\langle S \rangle$, $\langle R \rangle$, and $\langle O \rangle$ are special tokens. The input with n facts is obtained by concatenating the fact strings and section title as follows: “generate [language] $f_1 f_2 \dots f_n \langle T \rangle [t]$ ”, where “[language]” represents one of the 12 languages, $\langle T \rangle$ is the section title delimiter token, and t is the section title.

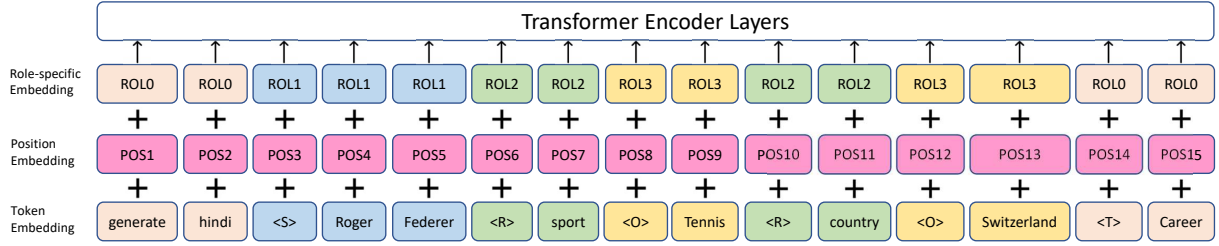


Figure 4.5: The input given to the encoder of mT5 includes English facts, along with token and position embeddings. In addition to that, role embeddings are used to indicate the specific roles of different tokens in the facts.

4.3.2 Fact-Aware Embeddings

We make use of the multilingual mT5 model for XF2T generation. The input for the mT5 model includes both token embeddings and position embeddings. However, for the XF2T model, the input is - specifically - a collection of facts. Each fact consists of separate units that are semantically distinct and play different roles, such as subject, relation, object, and qualifier. So, to enhance the XF2T model, we incorporate specific role embeddings that are aware of the input facts. These embeddings use four role IDs to represent the different roles: 1 for subject, 2 for relation and qualifier relation, 3 for object and qualifier tokens, and 0 for everything else. This is illustrated in Figure 4.5. By explicitly indicating the role played by each token in the input facts, we aim to improve the XF2T model’s performance.

We conducted further experiments to attempt to improve the performance of the model. One of the experiments involved using separate role embeddings for qualifier relation and qualifier, in addition to the fact-aware role embeddings. Another experiment involved adding fact Id embeddings to the input, such that each of the K facts in the input had a unique fact Id, and all tokens corresponding to a fact received the same fact Id embedding. However, these experiments did not result in better performance of the model, and hence were not pursued further.

4.3.3 Results on the XF2T Task

We show the results for an mT5 model using fact-aware embeddings in Table 4.2 across all the languages, along with the averaged scores. For evaluation, we make use of the usual text generation metrics BLEU [43], METEOR [5] and chrF++ [47].

We find that our system struggles on the Assamese language as well as on the dravidian languages Kannada, Tamil and Telugu. The best performances are achieved on Bengali, English and Hindi, which also reflect from the higher number of fact aligned instances available in our training data as can be observed in Section 4.2.4. The Oriya language, even with lesser fact-aligned sentences for training, has a relatively high performance on the metrics. A probable reason could be the presence of relatively shorter sentences for the language, which was an average of 16.88 words per sentence.

Lang.	BLEU	METEOR	chrF++
en	48.29	70.75	65.42
as	12.16	31.61	36.44
bn	49.48	73.03	76.19
gu	23.27	50.00	50.64
hi	42.72	67.49	68.03
kn	11.57	33.44	46.66
ml	29.04	57.15	57.60
mr	29.06	55.40	57.97
or	41.75	63.77	67.96
pa	28.65	55.19	53.38
ta	19.07	43.65	56.01
te	16.21	42.14	51.25
Avg.	29.27	53.64	57.30

Table 4.2: XF2T scores on the XAlign test set using mT5 with fact-aware embeddings.

4.4 Cross-lingual Fact Extraction for Knowledge Graph Enrichment

4.4.1 On Knowledge Graphs

Knowledge graphs are structured sources of information. They are being actively developed and used for various applications, such as text generation and question answering [28] [57]. Wikidata [59], which has over 99 million entities, is one of the largest publicly available knowledge graphs. A knowledge graph consists of interconnected facts, where a fact is a set of three values: a subject entity denoted as h , a semantic relation as r , and a tail entity as t . This set of values are usually written as a triplet in the format $\langle h, r, t \rangle$.

4.4.2 Fact Extraction v/s Fact Linking

Fact extraction [8] involves extracting factual information in a structured manner from natural language text. These structured facts may or may not be present in an existing knowledge graph, hence methods in fact extraction can help grow knowledge graphs with more facts. In contrast, fact linking aligns existing facts from a knowledge graph with natural language sentences. This problem serves to derive structured information from unstructured text and help with a better representation of the textual content, which can be further useful for downstream tasks. Although there has been extensive work such as [62], [32] on monolingual fact extraction, particularly in English, there has been less focus on cross-lingual fact extraction.

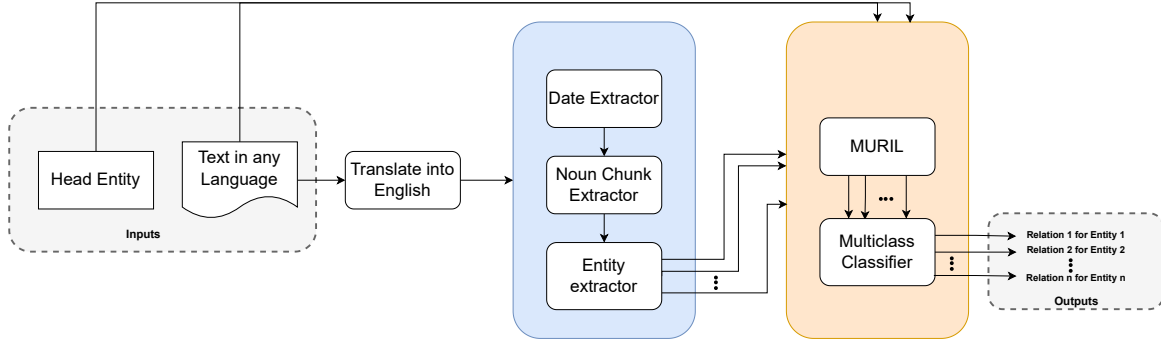


Figure 4.6: Pipeline Architecture for the TERC method, consisting of tail extraction from translated sentences followed by relation classification.

4.4.3 On Cross-lingual Fact Extraction (CLFE)

We propose a task called Cross-lingual Fact Extraction (CLFE) for 7 Low-Resource (LR) South Asian languages along with English. In this task, we aim to extract English triples directly from text written in these 8 languages. We also provide strong baselines and approaches for this task, which yield results comparable to existing state-of-the-art fact extraction pipelines for monolingual languages and significantly better than previous cross-lingual fact extraction attempts such as [62]. We discuss these methods in detail in Section 4.5. Our work makes it possible to use factual knowledge from South Asian texts to expand existing knowledge graphs, which could be useful in various downstream tasks such as fact verification and text generation. We make use of the XAlign dataset discussed in Section 4.2 for experiments on this problem.

4.5 Methods for CLFE

Our study proposes two methods for the CLFE task. The initial approach is classification-based, which involves extracting tails first and then predicting the relation between them. The second approach is generative and performs both the tasks simultaneously.

4.5.1 Tail Extraction and Relation Classification (TERC)

The TERC pipeline, illustrated in Figure 4.6, has two steps. The first step involves extracting tail entities from the source language text using IndicTrans translation to convert the text to English. Dates in the text are also extracted and normalized into a standard format, and a placeholder is used to replace them in the original text so that they don’t interfere with other entities. The spaCy [21] noun chunk extractor is then used to extract all the noun chunks, from which tail entities are selected.

To predict a relation for each tail entity, we use the pretrained MuRIL model [24] to generate a joint representation of the head entity, tail entity, and source language input text. This representation is then fed

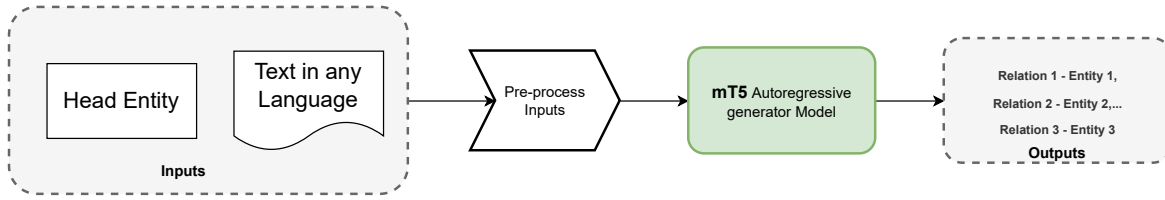


Figure 4.7: End to end generative pipeline for CLFE, which simultaneously extracts the entities and the relations present from the given sentence.

to a classifier that predicts the relation between the head and tail entities. During training, the classifier takes the tails from the ground truth as input to predict the relation, given a sentence and a $\langle h, t \rangle$ pair. To address class imbalance, we use the inverse log of class distribution as weights in the loss function, which outperforms the standard inverse class distribution and unweighted loss.

During the evaluation of the pipeline architecture, the extracted tails from the translated input text are compared with the ground truth tails to assess the performance. Once the tails are aligned, predictions are made for them, and the evaluation metrics are calculated based on the aligned tails.

4.5.2 End to End Generative extraction

We suggest an end-to-end approach to the fact extraction problem which can simultaneously extract tails and their relationships with the head entity as shown in Figure 4.7. Prior research in monolingual fact extraction has demonstrated that a model which jointly performs tail and relation extraction is more likely to perform better than a separate approach [32]. The benefit of this approach over the previously mentioned pipeline approach is that there is a bidirectional interaction between the two inter-dependent tasks of tail extraction and relation prediction, resulting in better performance for both tasks.

To address the CLFE problem, we use an end-to-end approach where the task of extracting both the tails and relations is performed jointly. Unlike the pipeline approach where tail extraction and relation prediction are independent of each other, the end-to-end approach allows for a two-way interaction between the two tasks. To accomplish this, we use mT5, an auto-regressive sequence-to-sequence model, which takes the head entity and input text as inputs and generates the relations and tails. Cross-entropy loss is used to train this model. Using a generative model enables a more generalizable and flexible approach to information extraction since the set of relations and tails are not limited to predefined classes, as in the TERC approach.

To conduct experiments on the fact extraction task, we experiment with three variations:

1. Fine-tuning the pretrained mt5 model for all languages.
2. Transliterating the input text of all languages except English to the Devanagari script to achieve script unification.
3. Training multiple cross-lingual fact extraction models, one per language.

Method	en	bn	gu	hi	kn	mr	ta	te	All languages		
	F1	F1	F1	F1	F1	F1	F1	F1	P	R	F1
TERC	50.80	41.96	40.30	50.46	42.57	44.59	52.19	43.66	40.45	53.71	46.15
E2E Cross-lingual Generative Model	76.28	75.56	72.36	86.62	68.04	77.79	82.82	71.82	74.09	81.15	77.46
E2E generation w/ script unification	74.56	75.38	72.04	83.44	70.46	77.19	85.21	72.51	78.49	76.15	77.29
Bilingual Models	76.64	78.01	67.84	86.49	63.19	71.91	83.71	70.94	79.79	71.63	75.49

Table 4.3: Precision, recall and F1 scores of various methods applied on all languages in the test set.

4.5.3 Results on CLFE Experiments

In Table 4.3, we present an overview of the outcomes obtained through various fact extraction methods discussed in Section 4.5.

The best performing approach in terms of F1 score is the open ended approach which also offers the most flexibility for possible entities and relations. Another observation is that training separate bilingual models works better than a combined model for just two languages, English and Bengali, which are the most dominant languages in the dataset, accounting for 54.44% of the training data. Overall, multilingual training is beneficial due to shared learning across the South Asian languages. Additionally, script unification, which involves transliterating input scripts to Devanagari, benefits all the Dravidian languages (te, ta, kn) in the dataset. However, one may note that the actual performance of the model might be better than what is shown because the current evaluation scheme requires a word match between predicted and actual tails, which may miss cases where they are synonymous.

4.6 Conclusion

In this chapter, we presented a cross-lingual study in the dual problems of text generation from facts and fact extraction from text. While the prior problem serves to enrich the content in low resource languages making use of large knowledge bases in high resource languages such as English, the latter problem works to utilize the pre-existing unstructured corpora in the low resource language to add information to unified knowledge bases as facts. We specifically restricted to a subset of 11 languages native to the subcontinent of South Asia for the purpose of conducting the set of experiments involved.

In the absence of a suitable data for studying these problems, the first step involved creation of a dataset. Hence, we curated the XAlign dataset, and explained the 2-stage automated pipeline for the same along with manual annotation process that worked to align facts in English to individual sentences in low resource languages. We showed the similarity in fact distribution in the dataset created using both the methods. We also visualized the fact distribution across sentences in the subsets across languages, and found majority of the languages to large proportion of sentences with lesser (up to 2) facts associated per sentence. For modeling on our proposed cross-lingual fact-to-text (XF2T) task, we proposed fact-aware embeddings to augment the multilingual mT5 model, and obtained SoTA results. We also explained

our input representation for encoding the facts as a textual stream of information, which we call as structure-aware input encoding.

We motivated our second problem by explaining the benefit of cross-lingual fact extraction for enrichment of common knowledge bases such as Wikidata. Making use of the XAlign dataset and restricting to a set of 7 South Asian languages along with English, we experimented with two primary paradigms for fact extraction. The first paradigm, based on classification, relied on entity extraction on translated sentence followed by classification for relation prediction. The second paradigm, being intuitively much simpler, made use of a generative model to directly predict the factual information as a text stream in a single step and performed decisively better than the first approach throughout. We compared three approaches on this pipeline, involving multilingual training across all the languages, multilingual training with script unification and pairwise cross-lingual models for each language text. Although the performances varied across languages for the three approaches, we found multilingual training to benefit on an aggregate set of all the languages.

Chapter 5

Bias Detection and Mitigation in a Multilingual Setting

This chapter focuses on one of the core problems of this thesis - handling biased in text, with a focus on application towards low resource languages. Our motivation stems from the fact that Wikipedia, though being a widely used source of information worldwide, is not free from subjective bias. This issue affects millions of readers, and while some work has already been done in resource-rich languages to classify and mitigate this bias, low-resource languages with large numbers of speakers have not received much attention. Therefore, in this piece of work, we propose an approach to address the dual problems of multilingual bias detection and mitigation. We conduct a thorough analysis and establish competitive baselines for our preliminary approach, which involves using classification-based models for bias detection on a multilingual dataset sourced from existing monolingual sources. For the problem of bias mitigation, we adopt the style transfer paradigm and model it using transformer-based seq2seq architectures. Additionally, we discuss several approaches that can be employed to further improve both the problems. Through this work, we aim to address the problem of subjective bias in Wikipedia data and help make information more accurate and accessible to all.

5.1 Why Bias Mitigation for Wikipedia?

Wikipedia, an online encyclopedia, is widely known for its vast and comprehensive knowledge resources. It serves as a primary source for researchers and students alike. The website enforces three fundamental content policies, which are essential for maintaining the credibility of its content. One of these policies, the Neutral Point of View (NPOV), aims to promote impartiality and fairness by prohibiting opinions and personal views from being presented as objective facts. The use of non-judgmental language is advocated to ensure that readers are provided with accurate and unbiased information. Subjective bias, which is the bias created by the use of subjective words or phrases that express a particular point of view in text, can be linked to emotions or personal beliefs. This type of bias can be problematic as it may influence the reader's perception of a topic or issue. Therefore, it is important for writers to be mindful of the language they use and strive to present information in an objective and neutral manner.

As discussed in Chapter 2, the work of bias detection and debiasing has been done primarily in English language. Hence, the primary objective of our study is to examine the detection and conversion of sentences that are in violation of the NPOV guidelines and modify them to be more neutral in South Asian languages that suffer from inadequate digital resources. Besides English (en), our research emphasizes on seven languages from South Asia, including Bengali (bn), Gujarati (gu), Hindi (hi), Kannada (kn), Marathi (mr), Tamil (ta) and Telugu (te). One of the the driving purposes of this study is to improve the quality of low resource South Asian Language Wikipedia articles and bolster its credibility as a massive source of free and fair information.

5.2 Dataset for Bias Identification and Mitigation

5.2.1 Necessity of an Appropriate Dataset for Study

To accomplish our research goals, it was imperative to procure a dataset that could provide us with consistent and substantial parallel data samples. The dataset that we sought had to maintain consistency with respect to the type of bias being addressed. Furthermore, in order for the dataset to be relevant to our study of Wikipedia, it was essential that the data be curated using the guidelines specified by Wikipedia, particularly with respect to the definitions of bias. Such guidelines would ensure that the data would be relevant and applicable to the context of Wikipedia. These requirements were necessary for us to conduct our research with accuracy and efficiency.

5.2.2 Our Dataset

As discussed in Section 5.2.1, in order to conduct our study on detecting and mitigating bias in Wikipedia articles, we needed a dataset that was not only substantial, but also consistent in terms of the type of bias being addressed. Most existing datasets related to bias are domain-specific and do not capture the breadth of topics that fall under the Neutral Point of View (NPOV) definition. Therefore, we needed to create our own datasets specifically tailored to the task at hand.

To achieve this, we used datasets curated using NPOV-related tags in the edit history of the English Wikipedia dumps, that were the Wiki Neutrality Corpus (WNC) [49] and WIKIBIAS [64] corpus, which contained over 200,000 parallel sentence structures. However, we were unable to reproduce the pipeline adopted for the creation of these datasets to our domain of low-resource South Asian languages due to the following reasons:

- **Issue with using language-dependent tags**

The revision history version of Wikipedia data can be utilized by extracting edits where the reason for the edit is tagged with a Neutral Point of View (NPOV) related tag. There are two possible ways to tag the edits: using language-dependent tags or using language-independent tags. The Wikidata knowledge graph [59], which is designed with tags in English, contains a plethora of well-defined

Dataset	Split	Biased / Unbiased Sentences
WNC	train	140k / 140k
	val	11.6k / 11.6k
	test	11.684k / 11.684k
WIKIBIAS	train	126k / 126k
	val	7k / 7k
	test	7k / 7k

Table 5.1: The table contains all details for both datasets. We create a similar parallel dataset for each language in the multilingual versions.

tags. However, the number of well-defined tags is scarce in South Asian languages, and their definitions are not clear. Therefore, using language-independent tags was a better approach for tagging edits in South Asian languages.

- **Usage of tags among users**

The usage of tags among South Asian users poses a significant challenge, as they primarily rely on the tags available in English and do not use tags in local languages as well as the language-independent tags. This approach can lead to inaccuracies in the extraction of tag-specific data, as users may not have a good understanding of the English language tags.

Thus, the reliability of tagged data seemed to be suitable only for the English languages from our set of languages for study, and hence translation of the data from English would be a better alternative to obtain more reliable and larger quantity of data. Therefore, we decided to translate the datasets into seven South Asian languages from parallel English datasets for debiasing. The South Asian languages we selected include Hindi, Marathi, Bengali, Gujarati, Tamil, Telugu, and Kannada. Using the pre-existing parallel WNC and WIKIBIAS datasets in English, we created our own datasets for further study: mWNC and mWIKIBIAS by translation. The subtle nature of bias resulted in instances of translated sentences being the same for biased and unbiased counterparts in some cases. Hence, we filtered out such sentence pairs, and used other heuristics to ensure the quality and consistency of the dataset. Details on the statistical aspects of these datasets can be found in Table 5.1.

To translate the datasets, we used the IndicTrans module. However, this module had a limitation of being unable to handle sentences with more than 100 tokens, so we selected sentences with a length cap of 80 tokens to ensure correct translation. To carry out the classification task, we obtain our dataset from the parallel corpora. We create this classification dataset by simply incorporating an additional column which specifies whether the sentence in question is biased or unbiased. We opt to use the translated dataset as is for the style transfer component of our work.

In Section 5.3, we describe several attempts made to create a parallel multilingual dataset for our problem while also discussing the translation-based method that we finally adopted.

5.3 Attempts at Creating a Multilingual Bias Mitigation Dataset

We employed several methods in our attempt to curate a new multilingual dataset for our problem:

- **Using language-independent tags for multilingual dumps (e.g. NPOV/POV).**

In an attempt to obtain more data on bias in South Asian languages, we explored popular templates used by other researchers and manually searched for templates as well. We also used a diachronic retrieval method for testing, but unfortunately, we found that the number of sentences extracted was low for South Asian languages. Specifically, we only had 116 sentences for Hindi and 1473 sentences for Italian, a high resource language. These results suggest that the method based on language-independent tags was not effective for collecting sufficient data on any language in general. Therefore, we needed to explore other approaches for obtaining larger datasets for our research.

- **Using source code tags as well as comment tags.**

We discovered that comment tags were usually not linked to NPOV-related content, but rather served other purposes, as can be seen here. As a result, we chose to focus solely on source code tags to ensure that our efforts were directed towards NPOV-related content.

- **Using domain-independent sources known to be biased (like the English Conservapedia).**

Conservapedia solely supports the English language. While domain-specific sources may exist in regional languages, the acquisition of a dataset would not be generalizable to other domains. This is due to the absence of a direct correlation between the source and the corresponding Wikipedia article. Obtaining a parallel dataset is therefore challenging.

- **Using language specific tags.**

To identify tags related to bias in languages unfamiliar to us, we tried to identify the tags that appeared in conjunction with English NPOV tags in Wikipedia articles. If the identified tags were related to bias, they were added to a seed list, and the process was repeated to obtain more tags until reaching saturation. An example of such a seed list is provided here for reference. However, our assumption that a biased article might contain both English and multilingual NPOV tags was proven wrong in this case. Although multilingual tags were present, they did not co-occur with English tags, and therefore this method was not effective for identifying them.

- **Direct translation of Wiki Neutrality and WIKIBIAS corpus into target languages (7 South Asian languages).**

As previous attempts to create a bias dataset were unsuccessful, we opted for a simpler approach to concentrate on modelling bias identification and mitigation, even if it meant working with some noise due to translation. However, we found that the translations often missed subtle nuances of the text. Also, in the process of using IndicTrans for translation, we observed that for a number of sentence pairs, both the biased and unbiased parts of the sentence were given the same translation

by the model. To address this and to obtain proper translations from the IndicTrans model, we imposed a maximum sentence length and filtered out pairs where the input and target sentences were the same. As a result, about 12% of the Wiki Neutrality corpus had to be discarded, and we could not use the same train-test split as the original authors.

5.4 Methodology

Our research heavily emphasizes the use of multilingual transformer architecture-based models in our modelling approaches. Specifically, we rely on two multilingual datasets that we curate, namely mWNC and mWIKIBIAS, to train our models. These datasets are extensively discussed in Section 5.2.2 of our this chapter, which provides a comprehensive overview of our dataset creation process. We believe that this approach is highly effective in achieving our goal of identifying and mitigating bias in multilingual text, and our research findings confirm this assertion.

5.4.1 Bias Identification: Classification Approach

In our efforts towards classification, we have implemented a method where we train classifiers based on encoder-only models such as InfoXLM [11] and MuRIL [24]. These models are used to detect the presence of bias in a given sentence. In order to tackle the problem of style transfer for debiasing, we have fine-tuned multilingual encoder-decoder transformer-based models like mT5 [61], IndicBART [12], and mT0 [40] over parallel corpus data. This approach is illustrated in Figure 5.1. The goal of our overall pipeline is to mitigate the effects of bias present in the input sentence by generating a debiased output sentence using these models. The first step of finding out if some form of bias is actually present in the given sentence is modeled by the classification problem.

5.4.2 Bias Mitigation: Style Transfer Approach

Bias mitigation, or simply text debiasing using style transfer involves using machine learning models to alter the style of a biased text to make it less biased or neutral. The process typically involves training a model on a parallel corpus of biased and unbiased text, where the biased text serves as the input and the unbiased text serves as the target. The model learns to translate the biased text into a neutral or unbiased form while preserving the underlying meaning and content of the text.

One approach is to use multilingual encoder-decoder transformer-based models, which can be fine-tuned on a parallel corpus of biased and unbiased text. In our experiments, we make use of IndicBART, mT0 and mT5 models. During training, the model learns to generate unbiased text that is similar in content and structure to the input biased text. The fine-tuned model can then be used to perform text debiasing on new biased text inputs.

To ensure a better evaluation of the debiasing process, we considered classifier accuracies instead of relying solely on content preservation metrics like BLEU scores, as can be illustrated in Figure 5.2. We

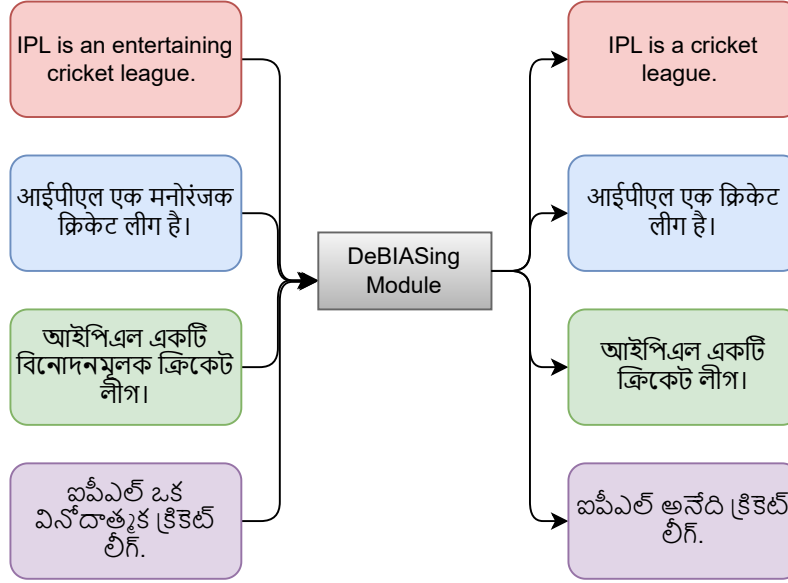


Figure 5.1: An example illustration of debiasing in different languages.

also measured the percentage of sentences that remained unchanged, considering that the debiasing may usually involve only slight changes to the input. The rationale behind this approach is that although high BLEU scores were observed for the source copy, it does not guarantee effective debiasing. During the decoding process, some words that initially had a neutral or unbiased connotation might be transformed into words with a biased connotation. Thus, it is important to consider the impact of these changes on the overall quality of the debiasing process. To this end, we calculated the average attention scores for all the words that were changed from unbiased to biased forms during decoding. This could provide a valuable metric to assess the effectiveness of the debiasing process.

The experimentation was conducted in both multilingual and monolingual settings for both the classification and style transfer tasks. The multilingual setup involved training the models on all languages together, while the monolingual setup trained the models on one language at a time. By testing in both settings, we can assess how the performance of the models varies with respect to the language they are trained on and whether a multilingual approach is more effective than a monolingual one. This allows us to evaluate the effectiveness and generalizability of our proposed methods across different languages.

5.5 Results & Analysis

We present the results for our classification as well style transfer modules in this section. The results for each experiment are given for English followed by the 7 native languages of South Asia.

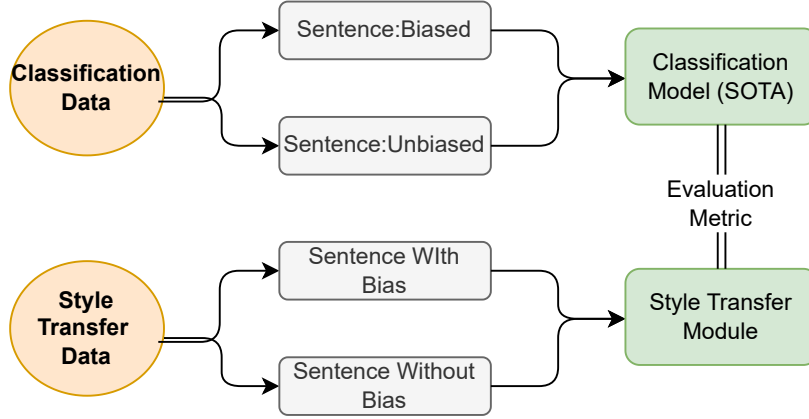


Figure 5.2: Metrics for evaluating debiasing: use of classification module for evaluation generated output from the style transfer module.

5.5.1 Classification-based Experiments

We provide detailed results of our classification-based experiments on monolingual as well as multilingual settings. For both the settings, we provide the results for experiments on two models: MuRIL and InfoXLM. The experiments are performed on our multilingual versions of the WNC and WIKIBIAS datasets, i.e. mWNC and mWIKIBIAS.

For multilingual experiments on the mWNC dataset, the results of the experiments using MuRIL can be seen in Table 5.2 and using InfoXLM can be seen in Table 5.2. Results of the monolingual counterparts of these experiments are shown in Table 5.4 and Table 5.5 respectively. In these tables, the notation $x \rightarrow y$ represents training of the model on the training data of language x , while the evaluation is conducted on the testing dataset of language y . “ED” stands for “Entire Dataset”, which corresponds to multilingual training where we concatenate the training data from all the languages to train a unified model for the classification task. In the monolingual setting, an individual model is trained per language, which is used for its corresponding evaluation.

Parallel results of our experiments performed on the mWNC dataset are also included for those on the mWIKIBIAS dataset. Table 5.6 and Table 5.7 provide results for the multilingual experiments using MuRIL and InfoXLM models respectively. In Table 5.8 and Table 5.9, we show results of the monolingual experiments using MuRIL and InfoXLM on the mWIKIBIAS dataset.

We summarize the primary observations from the results presented in Tables 5.2 to 5.9 below:

1. Monolingual versus Multilingual Models

We find the multilingual models to outperform their monolingual counterparts across experiments and languages we study. This implies that the addition of languages helps not only from the additional data perspective, but also from the point of view of an increased understanding of the task, hence a better learning. As another observation, we found that the monolingual models tend

Testing Protocol	Accuracy	Precision-macro	Recall-macro	F1-macro	MCC
ED → en	0.7166	0.7184	0.7166	0.7160	0.4350
ED → bn	0.6700	0.6757	0.6700	0.6673	0.3457
ED → gu	0.6646	0.6700	0.6646	0.662	0.3346
ED → hi	0.6748	0.6792	0.6748	0.6728	0.3539
ED → kn	0.6665	0.6723	0.6665	0.6637	0.3387
ED → mr	0.6529	0.6600	0.6529	0.6489	0.3128
ED → ta	0.6588	0.6653	0.6588	0.6553	0.3240
ED → te	0.6568	0.6629	0.6568	0.6535	0.3195

Table 5.2: Classification results over multilingual experiments on the mWNC dataset using the MuRIL model.

Testing Protocol	Accuracy	Precision-macro	Recall-macro	F1-macro	MCC
ED → en	0.7257	0.7281	0.7257	0.7250	0.4538
ED → bn	0.6568	0.6629	0.6568	0.6536	0.3197
ED → gu	0.6498	0.6547	0.6498	0.6469	0.3045
ED → hi	0.6688	0.6723	0.6688	0.6671	0.3411
ED → kn	0.6555	0.6602	0.6555	0.6529	0.3156
ED → mr	0.6430	0.6483	0.643	0.6399	0.2913
ED → ta	0.6484	0.6544	0.6485	0.6451	0.3028
ED → te	0.6471	0.6531	0.6471	0.6437	0.3002

Table 5.3: Classification results over multilingual experiments on the mWNC dataset using the InfoXLM model.

Testing Protocol	Accuracy	Precision-macro	Recall-macro	F1-macro	MCC
en → en	0.6520	0.6695	0.6520	0.6428	0.3210
bn → bn	0.6526	0.6670	0.6526	0.6449	0.3193
gu → gu	0.6379	0.6571	0.6379	0.6264	0.2944
hi → hi	0.6520	0.6695	0.6520	0.6428	0.3210
kn → kn	0.6450	0.6596	0.6450	0.6366	0.3043
mr → mr	0.6352	0.6501	0.6352	0.6259	0.2849
ta → ta	0.6375	0.6493	0.6375	0.6302	0.2865
te → te	0.6355	0.6586	0.6355	0.6217	0.2934

Table 5.4: Classification results over monolingual experiments on the mWNC dataset using the MuRIL model.

Testing Protocol	Accuracy	Precision-macro	Recall-macro	F1-macro	MCC
en → en	0.7010	0.7017	0.7010	0.7008	0.4027
bn → bn	0.6274	0.6321	0.6274	0.6242	0.2595
gu → gu	0.6249	0.6309	0.6249	0.6205	0.2557
hi → hi	0.6497	0.6517	0.6497	0.6486	0.3013
kn → kn	0.6344	0.6403	0.6344	0.6305	0.2746
mr → mr	0.6191	0.6206	0.6191	0.6180	0.2398
ta → ta	0.6254	0.6294	0.6254	0.6225	0.2547
te → te	0.6202	0.6235	0.6202	0.6177	0.2437

Table 5.5: Classification results over monolingual experiments on the mWNC dataset using the InfoXLM model.

Testing Protocol	Accuracy	Precision-macro	Recall-macro	F1-macro	MCC
ED → en	0.7421	0.7495	0.7421	0.7401	0.4915
ED → bn	0.6481	0.6559	0.6481	0.6436	0.3039
ED → gu	0.6451	0.6529	0.6451	0.6406	0.2980
ED → hi	0.6541	0.6607	0.6541	0.6505	0.3147
ED → kn	0.6437	0.6535	0.6437	0.6379	0.2971
ED → mr	0.6306	0.6392	0.6306	0.6249	0.2697
ED → ta	0.6366	0.6479	0.6366	0.6295	0.2842
ED → te	0.6363	0.6468	0.6363	0.6296	0.2829

Table 5.6: Classification results over multilingual experiments on the mWIKIBIAS dataset using the MuRIL model.

Testing Protocol	Accuracy	Precision-macro	Recall-macro	F1-macro	MCC
ED → en	0.7344	0.7413	0.7344	0.7324	0.4756
ED → bn	0.6312	0.6379	0.6312	0.6267	0.2691
ED → gu	0.6249	0.6329	0.6249	0.6191	0.2577
ED → hi	0.6414	0.6487	0.6414	0.6370	0.2901
ED → kn	0.6305	0.6372	0.6305	0.6259	0.2676
ED → mr	0.6147	0.6213	0.6147	0.6094	0.2359
ED → ta	0.6228	0.6317	0.6228	0.6163	0.2544
ED → te	0.6177	0.6255	0.6177	0.6117	0.2431

Table 5.7: Classification results over multilingual experiments on the mWIKIBIAS dataset using the InfoXLM model.

Testing Protocol	Accuracy	Precision-macro	Recall-macro	F1-macro	MCC
en → en	0.7259	0.7353	0.7259	0.7232	0.4612
bn → bn	0.6308	0.6568	0.6308	0.6148	0.2864
gu → gu	0.6204	0.6363	0.6204	0.6089	0.2562
hi → hi	0.6341	0.6545	0.6341	0.6215	0.2879
kn → kn	0.6248	0.6502	0.6248	0.6278	0.2736
mr → mr	0.6097	0.6312	0.6097	0.5931	0.2399
ta → ta	0.6166	0.6369	0.6166	0.6018	0.2527
te → te	0.6184	0.6278	0.6184	0.6112	0.2460

Table 5.8: Classification results over monolingual experiments on the mWIKIBIAS dataset using the MuRIL model.

Testing Protocol	Accuracy	Precision-macro	Recall-macro	F1-macro	MCC
en → en	0.7175	0.7257	0.7175	0.7149	0.4432
bn → bn	0.6000	0.6139	0.6000	0.5874	0.2134
gu → gu	0.5970	0.6164	0.5970	0.5795	0.2125
hi → hi	0.6229	0.6330	0.6229	0.6157	0.2557
kn → kn	0.6000	0.6158	0.6000	0.5859	0.2159
mr → mr	0.5936	0.6071	0.5936	0.5804	0.2002
ta → ta	0.6016	0.6103	0.6016	0.5937	0.2117
ta → te	0.5862	0.6016	0.5862	0.5699	0.1872

Table 5.9: Classification results over monolingual experiments on the mWIKIBIAS dataset using the InfoXLM model.

to identify sentences without bias better than the multilingual versions. However, they are worse at identifying sentences with bias.

2. **MuRIL versus InfoXLM models**

The comparison of model performances across languages brings up an interesting observation. We find that across parallel experiments, i.e. the pair of experiments on the same dataset, both done either in monolingual or multilingual paradigm - but using different models, the performance on the English language is substantially higher on the experiments using InfoXLM in contrast to the experiments using MuRIL for modeling. However, looking at the South Asian languages, we find the experiments using MuRIL have a marginally better performance than the experiments using InfoXLM. MuRIL was a model pretrained specifically for South Asian languages as opposed to InfoXLM, which pretrained on general multilinguality by pretraining on a corpus of 94 languages. Considering only minor differences on South Asian languages as opposed to the larger difference on English, InfoXLM can be considered to be a better alternative for the extension of this work on more languages of the world.

3. **mWNC versus mWIKIBIAS dataset**

We did not find any significant trend for the differences in the performance of our approaches on the mWNC and mWIKIBIAS datasets. However, we observed consistency in that the languages gave similar performances across experiments on both the datasets, differing by small margins, and the relative distribution of language performances was consistent for both the datasets. This validates the consistency of our approach across different data.

4. **Higher performance of English**

English naturally had the highest performance in all the experiment charts (with one exception of the monolingual experiment on mWNC using MuRIL, where it was slightly outperformed by Bengali). This stems from the dominance of the English language in the training corpora used to pretrain the models we use. Another important factor that affects here is that the data for remaining languages, due to translation, is inherently noisy and less perfect as compared to the original curated data for English. Hence, the results on English naturally turn out to be an approximate upper bound for the performance on other languages in our datasets. Like other languages, English, too, benefited from multilingual training, signifying the benefit of multilinguality even for very high resource languages.

5.5.2 **Style Transfer-based Experiments**

We present the results of the debiasing experiments performed in the style transfer paradigm on the mWNC dataset in Table 5.10 and on the mWIKIBIAS dataset in Table 5.11. For both the datasets, the experimental results are reported on IndicBART, mT0 and mT5 models, which have been trained in monolingual as well as multilingual setting. Along with the standard evaluation metrics of BLEU

Mode	Model	% of unchanged	BLEU	METEOR	chrF	BERTScore	Classifier Acc.
Monolingual	IndicBART	56.85	53.79	70	75.68	91.05	0.61
	mT0	63.96	51.89	69.65	74.83	91.03	0.59
	mT5	29.24	50.77	67.5	73.82	89.65	0.59
Multilingual	IndicBART	44.99	53.7	70.34	76.14	91.37	0.64
	mT0	67.49	55.92	71.3	77.17	91.51	0.64
	mT5	49.51	55.64	69.82	75.91	91.3	0.69

Table 5.10: Debiasing results on the mWNC dataset.

Mode	Model	% of unchanged	BLEU	METEOR	chrF	BERTScore	Classifier Acc.
Monolingual	IndicBART	77.75	65.88	78.22	81.78	93.51	0.59
	mT0	32.26	60.08	72.96	79.49	92.57	0.55
	mT5	42.98	61.7	73.93	80.42	92.62	0.69
Multilingual	IndicBART	69.47	65.91	78.26	81.81	93.59	0.64
	mT0	60.99	62.13	75.84	81.63	93.35	0.64
	mT5	56.35	65	75.56	81.59	93.29	0.71

Table 5.11: Debiasing results on the mWIKIBIAS dataset.

[43], METEOR [5], chrF [47] and BERTScore [63] for text generation, we also measure the accuracy on our trained classifier. While the generation-based metrics measure the overlap of the produced text with respect to the reference, they would fall short of capturing the bias transfer, since debiasing would often need very less changes to be made to the input. With this, cases such as no changes at all, less than adequate changes or incorrect changes made by the style transfer module for debiasing may not be captured well by the generation-based metrics. This is where the classification-based model helps in the task: since it has been trained to explicitly distinguish between biased and unbiased sentences, a higher performance on the classifier would indicate an improvement on our primary objective of text debiasing. Besides these metrics, we also note the ratio of sentences that were left unchanged by the style transfer module, since as noted earlier, no changes made to the input sentence could by itself result in a high performance on the generation metrics because of the nature of the task.

The style transfer models utilized for debiasing tasks tend to leave a significant proportion of sentences unaltered. This can be attributed to the similarity between the original and transformed sentences, which makes it challenging for the models to identify which part of the sentence requires modification. This difficulty in identifying and changing the biased parts of the sentence results in the complexity of the debiasing task. The content preservation metrics like BLEU, METEOR, chrF, and BERTScore, which measure the similarity between the original and transformed sentences, can be misleading as they tend to provide higher scores for models that leave biased sentences unchanged. We observed a general increase

in the scores of these metrics when transitioning from monolingual to multilingual settings. We present a more detailed analysis of the results of these style transfer experiments in the discussion below.

1. **Monolingual versus Multilingual**

Across most of our metrics, we find the multilingual experiments to have an edge over their monolingual counterparts. This observation is similar to that in the classification-based experiments, however, the margin of the multilingual advantage can be judged to be lesser in comparison.

2. **IndicBART versus mT0 versus mT5**

There was no clear trend observed across the three models, as each had their advantages in different settings, against different metrics and across datasets. While mT0 performed the best on generation metrics on mWNC dataset, IndicBART performed the best on mWIKIBIAS dataset. In terms of classifier accuracy, however, the best performing models on both the datasets was multilingual mT5. Also, considering the smaller performance gap of multilingual mT5 against multilingual mT0 on the mWNC dataset on the generation metrics, the multilingual mT5 model can be rated to be the best model finetuned for the task of debiasing.

3. **% of unchanged sentences**

Although monolingual models performed generally worse than their multilingual counterparts on most metrics, surprisingly, the lowest values on the % unchanged sentences metric (the lower the better) on both the datasets were obtained on monolingual models. The monolingual mT5 achieved the least value on the mWNC dataset, while the monolingual mT0 obtained the least value on the mWIKIBIAS dataset. However, optimizing on this metric may not guarantee the achievement of our objective, as generation of a completely incorrect sentence for the given input may result in a low value of the metric.

5.6 **Directions for Future Work**

This research has contributed important findings on the task of debiasing multilingual Wikipedia text. However, there is still a need for further exploration and improvement in this area. To this end, we have identified some potential areas for future experimentation, which we plan to pursue in the near future. These include:

1. We can incorporate contextual information for identifying sentence-level bias by adding category, title, or citation information from Wikipedia to the beginning of the sentence as a prefix.
2. The bias classifier score can be utilized as a reward function for the reinforcement learning based training of the generative module in conjunction with the next-word prediction loss.
3. Sentences from Wikipedia articles are fed into our classifier, and if none of them are classified as bias positive, we consider the article to be unbiased. However, if there are any bias positive

sentences, we identify the section with the highest number of them and apply a section-specific NPOV tag there. This approach is intended to make it easier for editors to remove the biased content manually by narrowing down the tag from the article level to a specific section.

4. We can investigate the use of pre-training objectives for debiasing in addition to existing methods.
5. The impact of different types of data augmentation techniques on debiasing performance can be explored.
6. The effectiveness of debiasing in more low-resource languages can be examined.
7. We can evaluate the performance of debiasing on other types of biased language data, such as social media text.
8. More fine-grained metrics to assess the quality of debiasing can be developed. For instance, additional metrics like fluency, bias and meaning can be considered for qualitative evaluation.

By conducting the aforementioned experiments, we can hope to advance the field of debiasing multilingual text and in turn enhance its practical applications.

Chapter 6

A Two-Stage Approach to Clickbait Spoiling

The presence of clickbaity texts from social media have often been studied from their understanding point of view - with problems largely involving identification and categorization of clickbait in text. In this chapter, we study the problem of mitigating the attractiveness presented by a clickbait, by “spoiling” the clickbait. While the chapter discusses the process of spoiling in detail, we primarily intend to provide a short, satisfactory piece of text from the article the clickbait points to - which essentially provides an answer to the surprise element lured to by the clickbait. We tackle this problem of clickbait spoiling with a 2-stage pipeline: **(1)** the first stage involves identification of the type of spoiler that will be needed to spoil the clickbait **(2)** the second stage produces the clickbait by extracting the necessary span from the article pointed by the clickbait. The crux of our approach lies on our proposed technique of “Information Condensation”, which seeks to reduce the amount of unnecessary content in the article to aid both the stages.

6.1 Introduction to Clickbait Spoiling

Clickbait is an umbrella word for various strategies used to capture attention and stimulate people’s interest [39]. According to [10], instead of providing complete overviews, clickbait articles entice readers to click on links to access the full content by piquing their curiosity, which could make them a tool for disseminating harmful information like false news. The Clickbait Spoiling shared task [15] focuses on categorizing and producing short paragraphs that spoil the curiosity generated by clickbait using a spoiler. The task is based on the English language and is divided into two subtasks: **(1)** Spoiler Type Classification and **(2)** Spoiler Generation. Spoiler Type Classification aims to identify the type of spoiler required for a particular clickbait and article, while Spoiler Generation aims to generate the spoiler for a given clickbait using the spoiler type information, along with the post and article. It deals with three spoiler types: phrase, passage, and multipart. The dataset used for this task consists of manually spoiled clickbaits from various social media platforms such as Facebook, Reddit, and Twitter. The goal is to develop a pipeline that first predicts the type of spoiler before generating the spoiler. We discuss the

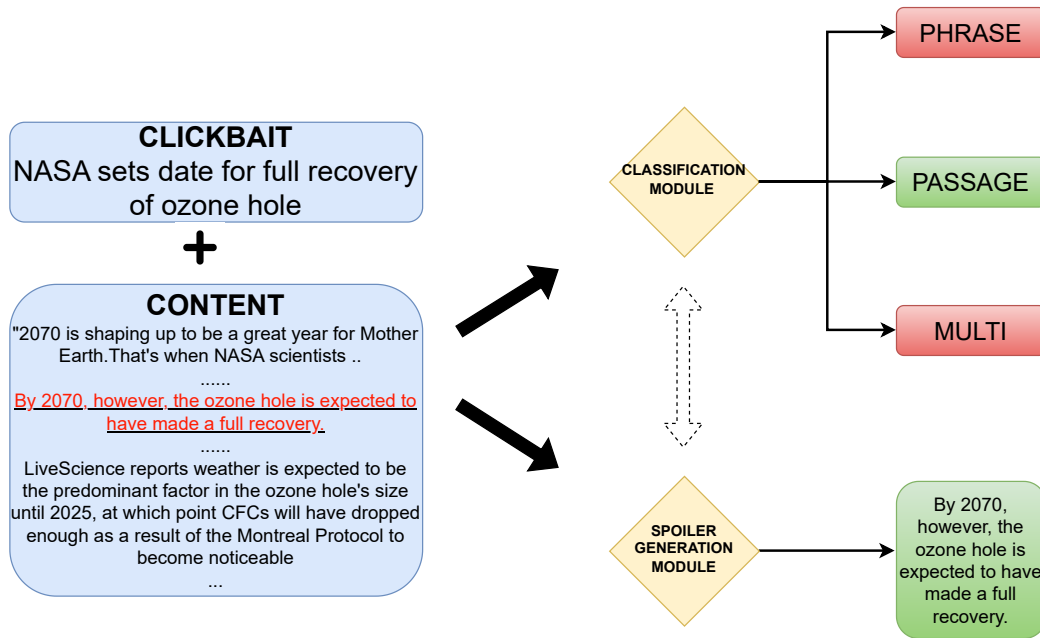


Figure 6.1: In this particular instance, the red text represents the spoiler, categorized as “passage.” The dotted arrow connecting the two tasks represents their interdependence and serves as a motivation to have a cohesive pipeline for spoiling clickbait.

differences between the three spoiler types in Section 6.3.1. In Figure 6.1, we show an example of a clickbait and article that require a “passage” type spoiler.

6.2 Information Condensation for Clickbait Spoiling: Motivation and Highlights

6.2.1 Motivation & Introduction to Information Condensation

Our approach for 2-stage clickbait spoiling proposes and explores an Information Condensation-based (IC) framework for the problem at hand. Information condensation refers to the extraction of unnecessary paragraphs from the article to provide precise information for downstream tasks. The motivation for this approach stems from the fact that the article’s size is often vast and contains diluted context, whereas the spoiler requires very specific information. To address this issue, we employ various precursor techniques before the final downstream tasks. We experiment with paragraph-wise classification with and without pretraining on datasets such as Answer Sentence Natural Questions (ASNQ) to tackle the task of Answer Sentence Selection. Additionally, we use the contrastive learning module to extract paragraphs from the article that are more likely to satisfy the clickbait curiosity, and analyze their performance to achieve

the desired Information Condensation. We also probe two different data feeding strategies, SIMPL and CONCT (further discussed in Section 6.4.4.2), and analyze their respective performances.

We model the first stage of the pipeline in a multiclass classification paradigm and the second downstream task in an extractive question answering paradigm. For the second task, an individual model is employed individual models for each spoiler type. We heavily rely on pretrained transformer architectures, such as RoBERTa [35] and DeBERTa [20], to model both the approaches.

6.2.2 Highlights of our Approach

We include a summary of the main findings and insights resulting from our experiments below.

1. Identifying whether a given paragraph contains a spoiler through paragraph-wise classification is not a trivial task to model.
2. Since we require the intermediate task of paragraph filtering to reduce information for downstream tasks while still maintaining a high likelihood of the input containing a spoiler, a high recall system is necessary for this stage.
3. A solution based on contrastive learning is very effective for achieving the required high recall with good precision in selected passages. We evaluate the performance using ranking metrics.
4. We try out two different modeling techniques while feeding input to the model - SIMPL and CONCT. SIMPL outperforms CONCT as the former enables better information exchange between the two inputs.
5. Although the filtering strategy performs well for the spoiler type classification task, it can negatively impact models that have already been trained to generate spoilers in coherent textual contexts. This is because the strategy can disrupt textual coherence and result in a loss of spoiler content.

In the first subtask, our submission achieved the highest ranking, with the best model showing a 1.71% improvement over the baselines provided for the task. In the second subtask, we obtained a BLEU score of 40.14 for extracting phrase spoilers and 36.90 for extracting passage spoilers. We evaluated our best performing method on the test set using the TIRA platform [16]. Our code is available on GitHub¹ for replication and further analysis.

6.3 About the Data

For most of our experiments, we utilized two datasets:

1. The train and validation splits of the Webis Clickbait Spoiling (WCS) Corpus 2022 [19] as released by the organizers of the task.

¹<https://github.com/anubhav-sharma13/ic-clickbait-spoiling>

2. The Paragraph-wise (Pw) Dataset which we created by extracting individual paragraphs from the WCS Corpus.

In Table 6.1, we provide the training and validation split stats for the WCS-Corpus and the Pw Dataset.

Dataset Name	Train	Dev
WCS Corpus	3200	800
Pw Dataset	48,626	12,410

Table 6.1: Statistics of the given WCS Corpus for the main task and the derived Pw Dataset.

6.3.1 WCS Corpus

The WCS Corpus contains 14 fields of information including the clickbait, article title, article content, spoiler category type, and spoiler positions. There are three types of spoilers based on their length and structure:

1. **Phrase**

A short span of an average of 2.8 words.

2. **Passage**

A longer span of an average of 24.1 words.

3. **Multipart**

A bullet-list style structure consisting of individual phrase and passage-type spoilers.

6.3.2 Pw Dataset

The Pw Dataset is a dataset that we created by using the spoiler positions in the WCS Corpus. We divided each article in the WCS Corpus into its constituent paragraphs to create the Pw Dataset. Each paragraph was assigned a binary label indicating whether it contained any part of the spoiler. A label of 1 indicates that the paragraph contains the spoiler in whole or in part, while a label of 0 indicates that it does not. However, the Pw Dataset has a significant class imbalance, with the ratio of label 1 to label 0 being 1 to 10.

6.3.3 Auxiliary Datasets

Apart from the given datasets, we use ASNQ [17] and SQuAD 2.0 [51] datasets as auxiliary sources for training. ASNQ dataset is used as a primary task for classification before finetuning on the Pw Dataset. We ensured that the ratio of positive is to negative sample in the ASNQ dataset is 1:10 (similar to Pw Dataset). Whereas, SQuAD 2.0 is used for a precursor training step prior to finetuning for spoiler generation.

6.4 Methodology

In this section, we explain our proposed method that involves filtering paragraphs to condense information, followed by finetuning for downstream tasks. Consider we have a clickbait post c and an article a associated with it, which includes the page title p_0 and n paragraphs p_1, p_2, \dots, p_n . Our objective is two-fold:

1. Identify the type of spoiler t that would spoil the clickbait.
2. Generate a spoiler s by utilizing the information about the spoiler type t .

The following sub-sections detail the methodology, starting off with the idea of Oracle experiments, which formed the motivation behind our approach.

6.4.1 Oracle Experiments

We perform oracle experiments for both downstream tasks to showcase the benefits of providing focused information to the model and to set a benchmark for the maximum possible performance in this aspect. In these experiments, we assume the presence of an oracle that selects only those paragraphs from the article that contain the spoiler. The resulting shortened article is then input to a classification model C_{oracle} for the first task and then to an extractive question-answering models $G_{oracle-phrase}$ and $G_{oracle-passage}$ for the phrase and passage spoiler extraction in the second task.

The promising outcomes obtained from the oracle experiments served as a motivation for us to develop an approach that aims to optimize the intermediate task of paragraph filtering, as elaborated in Section 6.4.2. The objective of our approach, hence, lies in achieving a performance level similar to that of the oracle experiments.

6.4.2 Paragraph-wise Filtering

The purpose of paragraph-wise filtering is to condense the information from the article into a smaller form. This is necessary not only due to the large length of the article compared to the input limits of transformer models and but also because it is advantageous to feed more focused information to the model, as demonstrated by the oracle experiments. The goal here is to select the top k paragraphs from the article using the clickbait post to guide the selection, resulting in a condensed article, a_c . The value of k should be selected to maximize recall, indicating that the filtering model should be able to rank the paragraphs containing the spoiler at high positions.

Label	S ∈ LA	SA ∈ S	# train	# dev
1	No	No	19,446,120	870,404
2	No	Yes	428,122	25,814
3	Yes	No	442,140	29,558
4	Yes	Yes	61,186	4,286

Table 6.2: Statistics of the ASNQ dataset, for the number of training and validation (dev) samples for each type of annotation.

6.4.3 Classification Paradigm

6.4.3.1 Vanilla Classification

Our first attempt on paragraph-wise filtering was modeled as a binary classification task. Here, we need to finetune a classification model $PARA_{CLF}(\theta_{CLF})$ to classify if a passage p_i contains an overlap with the spoiler s for the given clickbait c . This was an imbalanced classification problem as discussed in Section 6.3.2. We experimented with Cross Entropy, Weighted Cross Entropy and Focal Loss [34] as the objectives for directly finetuning models on the task. Once the model is trained, we infer the paragraph score $sc_i = P(p_i \cap s \neq \phi \mid c, p_i; \theta_{CLF})$ which is used in the filtering stage as discussed in Section 6.4.5.

6.4.3.2 ASNQ Pretraining

Considering the difficulty of the task, we also tried out prior conditioning of the model on the task of Answer Sentence Selection (AS2) [17]. The task of AS2 aims to identify the sentences which contain an answer to the given question from a pool of candidates, and is very similar to our setting for paragraph filtering. The Answer Sentence Natural Questions (ASNQ) [17] dataset was introduced as an effective dataset for transferring a general pretrained model on the AS2 task. In the dataset, each sample contains a question, a sentence and information on whether the sentence is a part of a long answer to the question, and whether the sentence contains a short answer to the question. Our paragraph classification model $PARA_{CLF}$ was adapted on the Pw Dataset following the transfer step on ASNQ (using the Focal loss objective), following [34]. The statistics of the ASNQ dataset are given in Table 6.2.

Out of the four types of annotations available per question-sentence pair as can be seen in the table, label 4 represents strong positives as they contain the answer in them and they are a part of the longer answer too. On the other hand, label 1 samples are strong or easy negatives, as neither of the two conditions are true in this case. Such samples are easy to identify, and contribute the major chunk of the dataset. They also fail to add much value from the point of view of model training data, since one can generate such sentences by controlled random sampling. We find the label 2 to be a relevant case for the phrase and passage type of spoilers in our problem, in which the sentence is not a part of a long answer (a long answer / spoiler perhaps doesn't exist for a given clickbait), but a short answer (spoiler) is a part

Version	Split	Label 1	Label 2	Label 3	Label 4	Total P	Total N	Total
v1	train	130,000	0	442,140	61,186	61,186	572,140	633,326
	dev	12,000	0	28,000	4,000	4,000	40,000	44,000
v2	train	220,000	25,000	440,000	35,000	60,000	600,000	660,000
	dev	31,000	2,500	29,000	4,000	4,000	40,000	44,000

Table 6.3: Label-wise statistics detailing the construction of the v1 and v2 subsets of ASNQ for our experimentation. Label 1 represents easy negatives, label 2 represents hard positives, label 3 represents easy positives and label 4 represents easy positives.

of the sentence (paragraph). Label 3 represents a class of weak positive samples, where the sentence is a part of a long answer, but the short answer is not a part of the sentence. This case can be seen closer to the multipart spoilers, since a paragraph may have some part of it being used for the multipart spoiler content, and a short answer may not really exist for the multipart spoiler case. Considering the less focus on multipart spoilers in the dataset (also considering the fact that label 3 spoilers could only very weakly approximate for paragraphs that contain content for multipart spoilers), we rather consider label 3 instances as hard (weak) negatives for our task.

Based on our analysis of the dataset, we initially sampled two versions of the dataset:

1. ASNQ v1

This contains strong (easy) positives, i.e. samples corresponding to label 4 only

2. ASNQ v2

This contains both - easy positives from label 1, and difficult positives from label 2

For both the versions, we maintained the positive : negative ratio of 1:10 as in our Pw dataset. As negatives, we first made use of the entirety of label 3 in order to have hard negatives for the task. We additionally sampled samples from label 1 data to fill up the required no. of samples to achieve the 1:10 ratio on the negatives side, for the already sampled positives. Statistics of the two versions of the ASNQ subsets can be seen in Table 6.3.

We experimented using both the versions on the paragraph-wise filtering task using the classification paradigm as explained in Section 6.4.3.1. The subset ASNQ v2 was found to be better than v1 on the modeling approaches we tried, and was selected as the dataset for pretraining the classification model prior to finetuning on the Pw dataset. Henceforth, when we refer to ASNQ pretraining, we refer to use of the v2 subset.

6.4.4 Contrastive Learning Paradigm

6.4.4.1 Modeling

As opposed to classification, here, we adopt a ranking-based paradigm to optimize the model to rank the positive paragraphs higher among all the paragraphs in the article. For training, we implement a pairwise contrastive approach to train a model $PARA_{CONT}(\theta_{CONT})$. Given two clickbait-paragraph pairs, out of which one pair has a positive paragraph (c, p_{pos}) and the other has a negative paragraph (c, p_{neg}) , the model is trained to give a higher score sc_{pos} to the positive pair than to the negative pair sc_{neg} by at least a threshold value $threshold_{CONT}$. Thus, the objective can be given as $l_{CONT} = \max(0, sc_{neg} - sc_{pos} + threshold_{CONT})$.

At inference time, we pass a single paragraph p_i to the model in one pass to infer its corresponding score sc_i for filtering as elaborated in Section 6.4.5.

6.4.4.2 Data Feeding Techniques

In designing the model architecture for contrastive learning, we experimented with two settings for feeding the inputs to the model:

1. Concatenating the embeddings for the clickbait and paragraph separately by Siamese modeling, which we refer to as **CONCT**.
2. Passing the concatenated clickbait-paragraph text sequence as a unified input to the model, which we refer to as **SIMPL**.

The SIMPL setting performed turned out to be the better architecture as we show and discuss in Section 6.6.1. We have illustrated the SIMPL in Figure 6.2.

6.4.5 Inferencing for Paragraph Filtering

The classification model $PARA_{CLF}$ or the contrastive model $PARA_{CONT}$ is trained to give a paragraph-level score sc_i to each paragraph p_i of the given article a . We use this score to first rank the paragraphs in a followed by the selection of the top k paragraphs for an empirically suitable value of k . The value of k is fixed so as to maximize the recall of the spoiler-containing paragraphs while achieving an effective condensation of the article. The selected paragraphs are then sequentially sorted to form the condensed article a_c which can be passed to the downstream tasks.

6.4.6 Spoiler Type Classification

This is one of the two downstream tasks that are to be solved as a part of the complete problem. Here, we model a classifier model C to classify which spoiler type does a given pair of clickbait c and article a warrant. We tried out the SIMPL and CONCT strategies explained in Section 6.4.4.2 for passing the

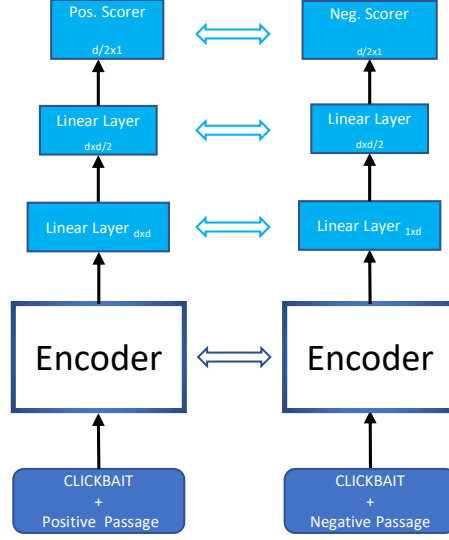


Figure 6.2: Model architecture for contrastive learning from clickbait-paragraph text sequence.

inputs a and c to the model. The CONCT strategy was tried because of the unfiltered a being particularly long, thus passing it separately would enable the model to capture more content in its input window. If paragraph-wise filtering is used, the article is passed in its condensed form a_c to the model, with which we use the SIMPL strategy. The classifier C was optimized on the cross entropy objective in a typical 3-way classification problem.

6.4.7 Spoiler Generation

In the second downstream task, we need to extract the spoiler of the given type t for an input (c, a) pair. Considering the nature of the phrase and passage spoiler generation tasks as being the extraction of single span from the article, we model the tasks in the extractive question-answering (QA) paradigm. Here, c represents the question and the article a represents the context from which we need to extract the answer s of the given t . We initially experimented with the extraction of phrase and passage spoilers using a unified QA model while not making use of the spoiler type information at the input, modeling a single extractive QA model $G_{phrase-passage}$. Considering the differences in the nature of the spoilers per category, we used the spoiler type information t to model a separate QA model per category, G_{phrase} and $G_{passage}$. The models used were also pretrained on the SQuAD 2.0 dataset [51] to benefit from task-specific alignment before finetuning on the small spoiler type-specific subsets of the WCS Corpus.

6.5 Experimentation

In this section, we explain the experimental settings we adopted for training and evaluation.

6.5.1 Architectural Specifics

6.5.1.1 Models used for classification

For the *base* encoder models, the output embedding size (d) is 768, whereas for *large* encoder models, it is 1024. The encoder output embedding is projected through a separate linear layer ($d * d$) each for the clickbait and article embeddings in the CONCT strategy following which they are concatenated to obtain a unified representation ($2d$). The projected embeddings are then passed through two linear layers ($2d * d$, $d * d/2$) and the classification layer ($d/2 * 3$). In contrast, in the SIMPL strategy, we rather use single linear layer on the encoder output embedding for classification.

6.5.1.2 Models used for contrastive learning

Similar to the classification-based experiments, we compute projections ($d * d$) for clickbait and paragraph passed to the model in the CONCT strategy followed by concatenation ($2 * d$) and application of 2 linear layers ($2d * d$, $d * d/2$) for hidden representations and a final linear layer ($d/2 * 1$) for scoring. The SIMPL strategy involved a single layer ($d * d/2$) for hidden representation on the encoder output followed by the output layer ($d/2 * 1$).

6.5.2 Training

Here, we explain the training related specification for the experiments based on classification, contrastive learning and extractive QA-based experiments. The flow of experiments has been visualised in Figure 6.3.

6.5.2.1 Classification based experiments

For the experiments specific to paragraph-wise filtering, we used the Pw Dataset, while we use the WCS Corpus (with and without filtering) for the spoiler type classification experiments. The ASNQ dataset was used for the initial transfer step following the classification on Pw Dataset (Pw adapt step) (as described in Section 6.4.3.2).

For modeling, we used transformer-based models like RoBERTa-*base* (rb-b), RoBERTa-*large* (rb-l), and DeBERTa-*base* (db-b) as the encoders. Besides the Cross Entropy (CE) objective which we commonly applied for all our experiments, we also tried out with Weighted Cross Entropy (WCE) and Focal Loss (FOC) objectives on the Pw classification task owing (Vanilla) to its imbalance in classes. We have discussed the architectural specifics of the model in Section 6.4.3. The learning rate is selected from $\{1e-5, 2e-5 \text{ and } 1e-4\}$ in combination with the number of epochs count chosen from a set of $\{10, 15 \text{ and } 25\}$. The model parameters are optimized using the AdamW [37] optimizer with default parameters. The learning rate is increased till 10% of the total training steps, followed by linear decay.

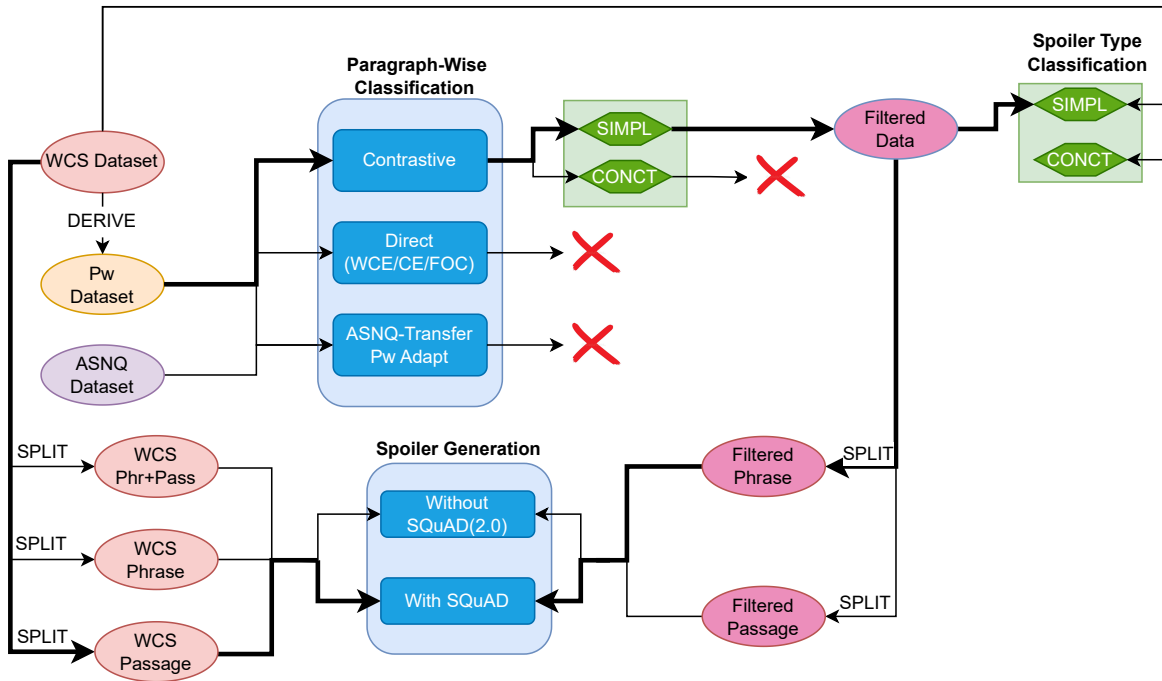


Figure 6.3: Experiment flow for our entire system. The red cross represents the termination of an approach due to inferior performance compared to the best-performing one, whose flow is marked with bold arrows.

6.5.2.2 Contrastive learning based experiments

In our pairwise contrastive learning setup, at each training step, we randomly sample a p_{neg} from all the negative paragraphs from the corresponding article for each given positive pair (c, p_{pos}) . We used 0.5 as the threshold $threshold_{CONT}$ in the margin ranking objective l_{CONT} . Architectural specifics for the models are discussed in Section 6.4.4.

6.5.2.3 Extractive QA based experiments

We employ the extractive QA paradigm for phrase and passage spoiler extraction in which we apply a linear layer on the encoder output embeddings for classifying the spoiler start and end tokens. We use DeBERTa-base (db-b) and RoBERTa-large (rb-l) as the encoder models. We also use a checkpoint of the RoBERTa-large (rb-l) finetuned on the SQuAD 2.0 dataset (rb-l-squad) to benefit from task-specific learning. As the context, we experiment with using the unfiltered a from the WCS Corpus as well as the condensed article a_c after paragraph filtering. All the models were trained for 5 epochs, with the rest of the hyperparameters being the same as used for the classification-based experiments.

The spoiler positions for each spoiler segment present in the data were given as $((st_i, st_j), (en_i, en_j))$, where st, en denote the start and end positions, and i, j denote the paragraph index and within-paragraph character offset respectively. These positions were mapped to the overall and start and end tokens indices in the tokenized article before feeding as labels to the QA model. Performing the oracle-based and filtering-based experiments involved a remapping of the paragraph indices i to suit the truncated article. An empty span was used as the training label in the small number of cases where the span was not present in the filtered data or where the span was out of bounds of the encoder model input for the complete article.

6.5.3 Evaluation

We specify the metrics used to evaluate the different objectives our models were trained on: classification for paragraph-wise filtering and spoiler type classification tasks, contrastive learning for paragraph-wise filtering task, and the spoiler generation task.

6.5.3.1 Metrics for Classification

The tasks modeled as classification were evaluated using the usual metrics of Accuracy, Precision, Recall, F1 and Matthews Correlation Coefficient (MCC). Precision, Recall and F1 for multiclass classification were calculated using the macro strategy. It should be noted that macro-Recall is the same as Balanced Accuracy (the primary metric for evaluating the task of spoiler type classification), and we thus refer to the same as Balanced Accuracy in the case of multiclass classification experiments.

6.5.3.2 Metrics for Ranking

We used ranking-based metrics of Mean Reciprocal Rank (MRR) and Mean Rank to evaluate the trained models for paragraph-based filtering, along with the classification-based metrics. Ranks given to all the positive paragraphs in an article were considered in the computation of these metrics.

6.5.3.3 Metrics for Spoiler Generation

Generation-based metrics of BLEU [43], METEOR [5] and BERTScore [63] were used along with the Exact Match metric to get an idea of the direct matches between the extracted spoilers versus the expected ones. Following the organizers, BLEU was calculated as an average of the Sentence-BLEU scores².

6.6 Results

We discuss the quantitative results on all our primary experiments in this section.

6.6.1 Paragraph-wise Filtering

The models trained on classification and contrastive learning-based metrics are evaluated using both - classification and ranking-based metrics to establish a common ground for best model selection. To evaluate the models using the ranking metrics, we use the paragraph score sc_i as discussed in Section 6.4.5. To evaluate the contrastive model using classification-based metrics, we use a threshold of 0.5 to classify the prediction of the model as a positive or a negative paragraph. Table 6.4 shows the results of the experiments under paragraph-wise filtering.

As can be seen, *rbl* model under Contrastive-SIMPL was the best performing model on most classification metrics while decisively leading the chart on the ranking metrics. This model was used to prepare the Filtered data containing condensed articles for downstream tasks.

Among the two paradigms chosen to model the problem, the contrastive learning-based paradigm turned out to be a better strategy to achieve paragraph filtering, as can be observed from the classification and ranking metrics in Table 6.4. Comparing the *base* versions of the respective models, DeBERTa outperformed its RoBERTa counterpart in the classification-based models, while underperforming on the contrastive-based models. Within the classification models, there was no clear difference observed in the performance achieved by the three vanilla models (Vanilla-CE/WCE/FOC) on the classification metrics, with Vanilla-FOC turning out to be marginally better on the ranking metrics. There was a small benefit from the ASNQ transfer step as can be observed from the performance improvement in ASNQ-Transfer→Pw-Adapt, showing the need for a better model adaptation with suited datasets as opposed to objective designing. Among the contrastive models, the SIMPL strategy clearly outperformed its CONCT

²https://www.nltk.org/_modules/nltk/translate/bleu_score.html

Experiment	Model	F1	Recall	Precision	MCC	Accuracy	MRR	Mean Rank
Vanilla-CE	rb-b	50.13	46.17	54.41	44.94	89.76	0.4612	5.20
	db-b	47.45	51.09	55.34	46.65	91.53	0.4843	4.89
Vanilla-WCE	rb-b	50.34	44.51	57.93	46.44	91.81	0.4538	5.29
	db-b	49.27	45.29	54.02	44.76	91.31	0.4841	4.88
Vanilla-FOC	rb-b	47.72	42.09	55.09	43.59	91.4	0.4613	5.23
	db-b	50.05	47.36	53.05	45.32	91.18	0.4882	4.73
ASNQ-Transfer→Pw-Adapt	db-b	52.46	46.59	60.02	48.70	92.13	0.4935	4.71
Contrastive-CONCT	rb-b	57.54	68.75	57.85	24.26	74.19	0.5120	4.67
	db-b	62.06	66.60	60.32	26.17	83.09	0.4966	5.06
Contrastive-SIMPL	rb-b	62.94	69.50	60.98	29.26	82.30	0.5816	3.98
	db-b	64.45	64.63	64.27	28.90	87.82	0.5874	3.99
	rb-l	65.11	69.56	62.97	31.86	85.09	0.6271	3.55

Table 6.4: Results for the experiments on paragraph-wise filtering on the dev set of the Pw dataset. F1, Recall and Precision are computed in macro forms.

counterpart. The reason for this can be attributed to the SIMPL strategy involving a computation of self-attention across the full clickbait-paragraph sequence over the transformer encoder pipeline leading to richer information exchange between the two pieces of input. Opposed to this in the CONCT strategy, the two representations are computed separately by the encoder. Thus, the interaction between the pair of inputs occurs only in the embedding space, leading to a weaker modeling.

6.6.2 Spoiler Type Classification

Table 6.5 shows the results of our experiments on the spoiler type classification task. On the WCS Corpus, we include experiments using the SIMPL (WCS-SIMPL) and CONCT (WCS-CONCT) strategies. On the Filtered data, we show results on the better performing SIMPL strategy (Filtered-SIMPL) for $k = 5$. Results for the Oracle-based experiments are added on top for establishing upper limits on the Filtering strategy.

For the experiments on full data, the SIMPL strategy outperformed the CONCT strategy for data feeding for reasons similar to those attributed in Section 6.6.1. Using the Filtered data with the SIMPL strategy improved the results, thus showing the benefit of information condensation for the task. Among each set of experiments, rbl outperformed its counterparts on most metrics. The best performing rbl model under Filtered-SIMPL was evaluated on the test set, which gave a Balanced Accuracy of 74.14%, being the winning entry (rank 1) on the task.

Experiment	Model	Balanced Accuracy	F1	Precision	MCC	Accuracy
<i>Oracle-SIMPL</i>	<i>db-b</i>	78.20	78.58	79.00	65.02	78.00
	<i>rb-b</i>	73.36	74.92	77.34	59.01	74.38
	<i>rb-l</i>	80.87	80.71	80.73	67.87	79.63
WCS-CONCT	db-b	68.92	69.76	71.31	53.67	70.87
	rb-b	66.09	66.98	71.93	49.54	68.37
	rb-l	67.69	69.55	74.05	53.11	70.63
WCS-SIMPL	db-b	72.32	73.33	74.71	58.21	73.88
	rb-b	67.34	67.39	68.99	50.25	68.25
	rb-l	73.38	73.40	73.47	58.01	73.50
Filtered-SIMPL	db-b	74.14	73.92	75.91	60.62	74.25
	rb-l	75.11	73.78	76.77	60.52	74.88

Table 6.5: Results on spoiler type classification task for dev set. Balanced Accuracy is the same as macro-Recall.

6.6.3 Spoiler Generation

We show the results for phrase extraction in Table 6.6 and passage spoiler extraction in Table 6.7. Like for spoiler type classification, we include results on the Oracle-based experiments followed by experiments on the WCS Corpus and the Filtered data.

As a common observation throughout the experiments, task-specific training helps - the rb-l-squad model already finetuned on the SQuAD 2.0 dataset easily outperforms its counterpart rb-l which is directly finetuned on the task. Among directly finetuned models, the rb-l model in turn outperforms db-b in virtue of larger number of pretrained parameters for modeling. On the WCS Corpus, we show the results on a unified QA model (WCS-phrase+passage) for phrase and passage spoiler extraction. The results are easily outperformed by QA models trained individually on phrase (WCS-phrase) and passage (WCS-passage) spoiler extraction objectives showing the need for spoiler-specific modeling. Using the Filtered data generally leads to an improvement in performance as compared to the full WCS Corpus for db-b and rb-l that are directly finetuned on the task. However, for rb-l-squad, we obtain only a minor improvement on phrase spoilers and a decrease in performance on passage spoilers. We analyze this behavior of rb-l-squad in a greater depth in Section 6.7.3.

6.7 Analysis

6.7.1 Differences based on Spoiler Type

Figure 6.4 displays the performance of the best performing (Filtered-SIMPL: rb-l) model for spoiler type classification for individual spoiler types.

Experiment	Model	BLEU-4	METEOR	BERTScore	Exact Match
Oracle-phrase	db-b	41.68	60.14	95.74	59.40
	rb-l	44.42	63.61	96.41	64.18
	rb-l-squad	48.41	67.90	97.20	69.55
WCS- phrase+passage	db-b	31.94	49.24	93.47	45.37
	rb-l	33.24	50.98	94.08	49.85
	rb-l-squad	35.59	54.66	94.51	53.43
WCS-phrase	db-b	32.53	50.74	93.52	47.15
	rb-l	34.93	51.66	94.55	50.15
	rb-l-squad	39.74	57.20	95.54	59.10
Filtered-phrase	db-b	32.89	48.52	93.81	47.76
	rb-l	36.18	52.27	94.55	52.54
	rb-l-squad	40.14	58.76	95.84	57.91

Table 6.6: Results for phrase spoiler generation (extraction) on the dev set (325 samples).

As can be seen from the confusion matrix, phrase and passage spoiler types are most easily confused with each other. In comparison, multipart is the least confused with the other types. The behaviour of multipart being predicted more in place of the actual class can also be reflected from the bar plot which shows the lowest precision and highest recall for the type. The highest type-wise F1 is obtained on the passage type.

6.7.2 Spoiler Loss Due to Chosen k

In Table 6.8, we compare the Mean Rank per spoiler type for the top two models for paragraph-wise filtering.

As can be seen, paragraphs containing phrase spoilers are the easiest to identify and rank higher, followed by passage and multipart. In condensing an article, we select a value of k to select from the top ranked paragraphs. We analyze the loss of spoilers resulting from different values of k in Figure 6.5.

It can be seen that articles with multipart spoilers are the most difficult candidates for achieving condensation, having highest losses throughout. Our selection of $k = 5$ stems from the observation of the decrease in spoiler loss being reduced beyond this point for all the types, thus being a suitable value for achieving high recall with an effective condensation of the article.

6.7.3 Performance Variation with Filtering

We study the variation of phrase and passage spoiler extraction for $k = 2$ to 15 across two models: rb-l directly finetuned on the task versus rb-l-squad already finetuned on SQuAD 2.0 using Figure 6.6.

Experiment	Model	BLEU-4	METEOR	BERTScore	Exact Match
Oracle-passage	db-b	57.98	73.67	94.54	20.81
	rb-l	52.89	68.30	93.97	15.53
	rb-l-squad	59.87	73.91	95.15	18.94
WCS- phrase+passage	db-b	22.66	32.41	88.72	6.83
	rb-l	24.40	34.23	89.17	8.70
	rb-l-squad	31.66	42.26	90.56	11.80
WCS-passage	db-b	25.32	36.61	89.08	7.14
	rb-l	27.62	39.33	89.58	8.38
	rb-l-squad	36.90	49.30	91.37	13.66
Filtered-passage	db-b	27.28	38.07	89.39	7.45
	rb-l	28.31	41.84	89.80	6.83
	rb-l-squad	35.02	48.53	90.97	10.25

Table 6.7: Results for passage spoiler generation (extraction) on the dev set (322 samples).

	phrase	passage	multipart
Contrastive-SIMPL: rb-l	2.78	3.80	5.36
Contrastive-SIMPL: db-b	2.97	4.29	6.24

Table 6.8: Type-wise differences in Mean Rank for the two best-performing models on the Pw task.

To conduct this analysis, we trained the models individually for each value of k and spoiler type using the hyperparameter settings in Section 6.5.2.3.

For phrase spoiler extraction on rb-l, we find that condensation helps: the model achieves high BLEU scores at lower value of k which is followed by a decreasing trend on the higher values. In case of passage spoiler extraction on rb-l, the condensation helped with a larger value of k (peaking on $k = 9$) due to significant loss of spoiler containing paragraphs in the initial values of k .

The behavior on both the spoiler types for rb-l-squad brings out differing findings. The model improves performance with increasing values of k , with peaking achieved at higher values. This can be attributed to the rb-l-squad model being already trained to handle large, coherent contexts from the SQuAD 2.0 dataset as opposed to the lack of coherence in the Filtered data which contains discretely chosen paragraphs from the article. The lack of coherence in the article, coupled with the spoiler loss due to filtering contributed to the lesser performance on lower values of k .

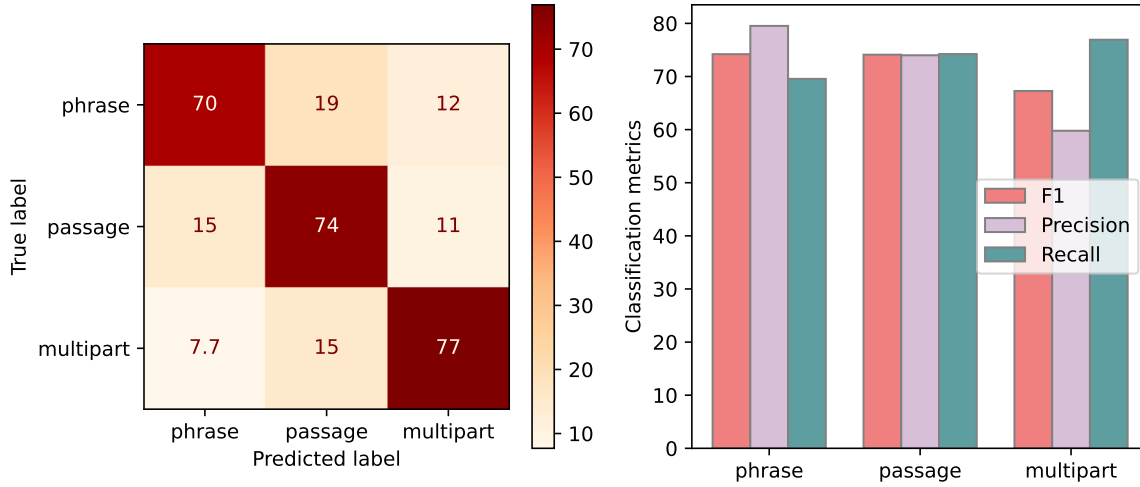


Figure 6.4: Performance analysis of the best model on spoiler type classification with confusion matrix (left) and bar chart for type-wise classification metrics (right).

6.7.4 Qualitative Analysis

6.7.4.1 Examples for Phrase Spoiler Extraction

We analyze the spoiler extraction performance of the best performing Filtered-phrase: rb-l-squad model in Table 6.9. We show four categories of outputs observed:

1. Accurately match with the expected
2. Extraction of a semantically similar but lexically different alternative
3. Output being a part of the expected
4. Expected being a part of the output

6.7.4.2 Examples for Passage Spoiler Extraction

In Table 6.10, we analyze the performance of the best performing model WCS-passage: rb-l-squad. We show four categories of outputs extracted

1. Same or almost same matching with the expected passage span.
2. The extracted span offering an alternative to the expected by either being semantically similar or offering a different perspective.

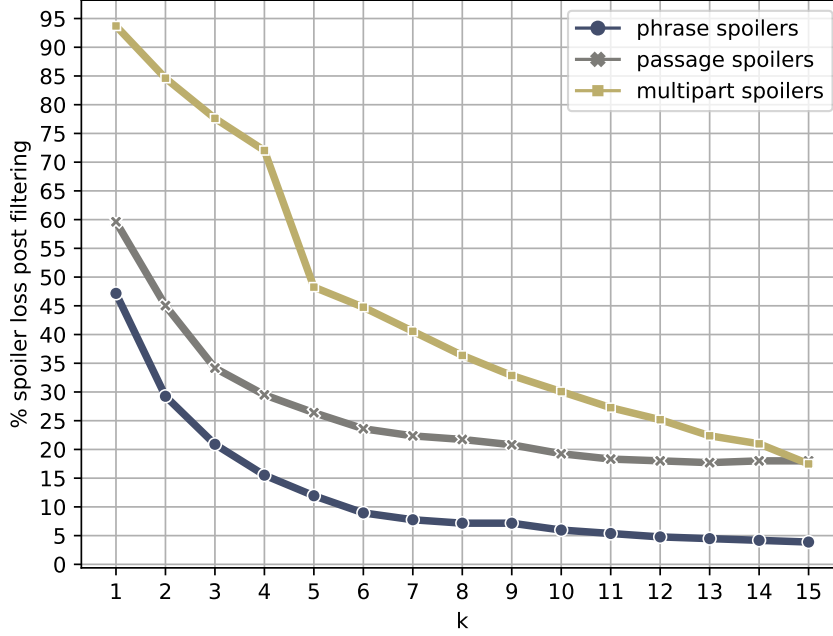


Figure 6.5: Plot of percentage loss of spoiler containing paragraphs in an article for different values of k.

3. The output and the expected spoiler spans differ in the amount of specificity of information in each other.
4. Containing very different (mostly incorrect) extraction than expected.

6.7.4.3 Output v/s Expected Overlap Comparison

We compare the % overlap between the generated and extracted spoiler spans by the best performing models on phrase and passage spoiler extraction in Table 6.11.

In phrase extraction, we observe a good examples of perfect overlap between the output and expected spans, along with an almost equal percent of the samples where the either is a subset of the other. For the samples where there was no complete overlap as well, we found good semantic similarity between the output and the expected spoilers. Refer to Table 6.9 for illustrative examples in this regard.

For passage spoiler extraction, we found a substantial number of instances where the output was a subset of the expected as compared to that for phrase, indicating that the model struggles to extract more specific information needed in the spoiler span. As can be observed in Table 6.10, we find instances where the model offers alternative spoilers which are not lexically similar. This brings up the shortcomings in the lexical matching-based evaluation metrics like BLEU as well.

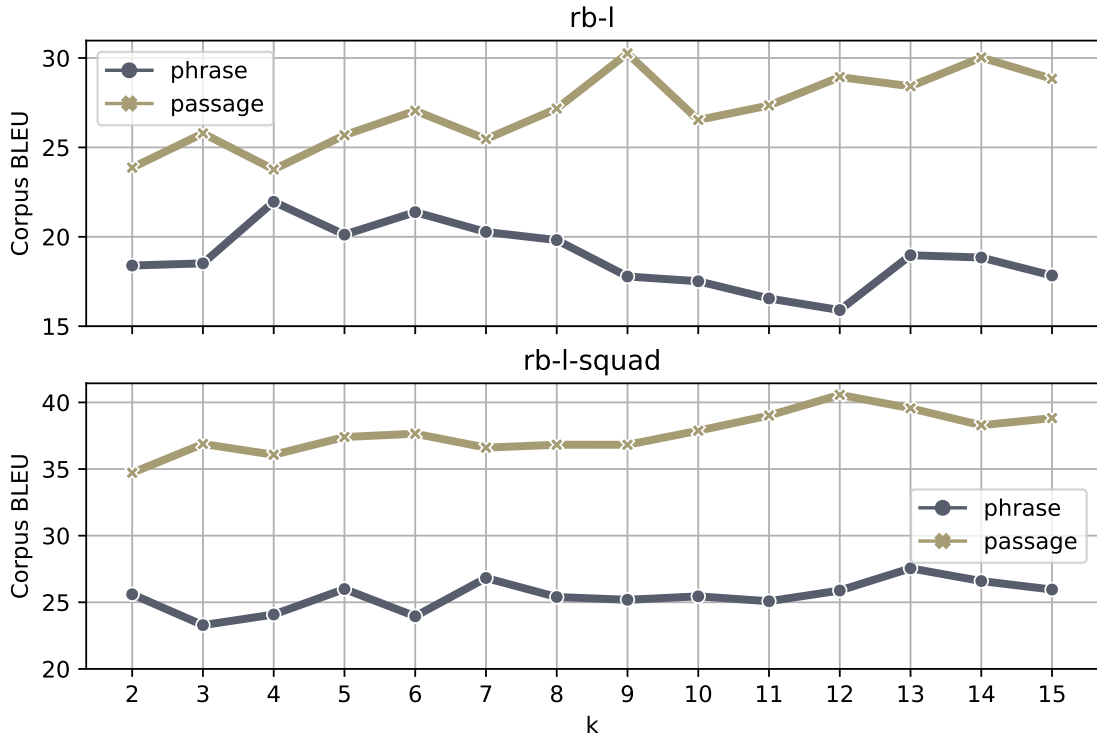


Figure 6.6: Variation in Corpus BLEU score of rb-l and rb-l-squad models for different values of k.

6.8 Conclusion

Clickbait spoiling, as opposed to clickbait detection, which frequently includes filtering out clickbait postings from users' timelines, subverts the enthusiasm aroused by clickbait by revealing the inner details in advance. Our idea of Information Condensation - providing a precise spoiler containing fragments of an article instead of the entire article, worked well on the spoiler type classification task and spoiler generation task on phrase type spoilers. We tried multiple techniques for Information Condensation, but found Contrastive learning based approach to be the most optimum and used it to filter paragraphs for both the tasks. For spoiler generation of passage type, the contrastive learning approach didn't result in an improvement due to loss of spoiler containing paragraphs and issues with textual coherence. Future work can look at the specific issue of spoiler containing paragraphs and would aim at reducing this loss (as results from Oracle experiments validates the correctness of approach). Further we can expand our experiments to generate multipart type of spoilers and analyse our findings.

	Clickbait	Output	Expected
Output == Expected (Exact)	An unlikely company is crushing America's biggest clothing stores	Amazon	Amazon
	NJ police searching for 2 men who they say tried to abduct a 10-year-old girl when she was doing this	playing hide-and-seek	playing hide-and-seek
Output == Expected (Semantically)	Apparently All Taylor Swift and Tom Hiddleston Do Is...	beach strolling	Take Walks on Beaches
	When This Guy Bought A Used Car, He Never Thought He'd Find THIS Hidden Inside.	stacks upon stacks of bills	money
Output < Expected	This is how much coffee Americans drinks every day	2.1	2.1 coffee drinks
	The cutest little European city you've probably never heard of	Lviv	Lviv, Ukraine
Output > Expected	Blocking this color light may help you sleep better	blue wavelength light	blue
	Fashion brand lets models go un-Photoshopped and makeup-free	Rag & Bone's DIY Project	Rag & Bone

Table 6.9: Some examples produced by the best performing model (Filtered-phrase: rb-1-squad) for phrase spoiler type extraction.

	Clickbait	Output	Expected
Output Expected	This chain will give you free burgers for life, but on one condition	The company wants you to change your last name to "burger."	The company wants you to change your last name to "burger."
	Dog Dies One Hour After Hiking With His Owner, Veterinarian Gives Shocking Reason Why	the plant the dog was chewing on was deadly water hemlock	the plant the dog was chewing on was deadly water hemlock
Alternative spoilers	A mother hears her dead son's heart beat	she meets the man who received his heart in a transplant	Dawn Grace lost her son four years ago, and now she meets the man who received his heart in a transplant.
	Why it's definitely worth it to get your flu shot	reduced their risk of heart attack, stroke, or other cardiovascular health problems	reduced the risk of flu-related hospitalization
Difference in specificity	Coming this fall	His untitled Daily Show companion series	Daily Show companion series will air Monday-Thursday at 11:30 p.m. ET/PT, beginning this fall
	Are Shorter Work Days Better For Your Health & Productivity?	sharply reduced absenteeism, improved productivity, enhanced creativity, and reduced turnover	An ongoing study out of Sweden seems to indicate that a shorter work day may actually result in more productivity.
Very different or incorrect generations	This is the difference between a job and a calling	work is most fulfilling when it's a calling	it's a calling because you find a deep sense of purpose and positive impact in your role
	I was really bad at sports in high school. This new study helps me understand why.	practice just doesn't matter that much	Some people are just better at sports than others

Table 6.10: Some examples produced by the best performing model (WCS-passage: rb-l-squad) for passage spoiler type extraction.

spoiler type	$O == E$	$O \in E$	$E \in O$	otherwise
phrase	57.91	9.25	10.45	22.39
passage	10.25	30.74	6.83	52.18

Table 6.11: % overlap statistics for phrase and passage spoiler extraction between the output (O) and expected (E) spans by the best models on each.

Chapter 7

Conclusion

In conclusion, this thesis focuses on handling fine-grained subtleties in text, such as bias and clickbait, which remain a challenge for NLP models despite recent advancements. The study investigates bias on a multilingual scale, with a particular focus on Wikipedia. In a separate study, we present a novel architecture for fine-grained categorization of entities on a Wikipedia-based dataset of 30 languages. The study also addresses the problem of content enrichment in low resource languages and proposes two methods for extracting English facts from unstructured content in low resource languages native to the South Asias subcontinent. Additionally, the thesis proposes several attempts at creating a sizeable multilingual, parallel dataset for studying bias detection and mitigation. The study approaches the problem of mitigating the effects of clickbait by proposing a 2-stage pipeline and a novel Information Condensation-based modeling approach. The results show that the proposed methods outperform existing techniques and provide a promising direction for future research in the field of NLP.

7.1 Inferences from the Problems for Multilingual Study

7.1.1 Fine-grained Entity Categorization

In this study, we focused on multilingual entity categorization on Wikipedia data from 30 different languages, with the goal of establishing a unified ontology that can improve the performance of NLP methods. We proposed a Graph Neural Network (GNN)-based multilingual architecture that significantly outperformed the monolingual baselines, using XLM-RoBERTa and bi-GRU for modeling. This work was part of the NTCIR-15 Shinra 2020-ML Classification Task, and our proposed method ranked second overall. In this study, we provided insights into multilingual understanding and knowledge representation in NLP that can aid the development of large, unified knowledge bases.

After conducting various experiments and evaluations, we found our proposed approach RNN_GNN_XLM-R to be superior to other methods we tested. We believe that the success of our model can be attributed to multiple factors. Firstly, XLM-RoBERTa is a powerful model that is capable of extracting rich semantic information from textual content in different languages. Secondly, the attentional-RNN module

is designed to enhance the contextual information, allowing for a more comprehensive understanding of the text. Lastly, the GNN module is useful in learning the classification label correlations and thus, aids in improved generalization. These factors together contribute to the superior performance of our proposed model. Additionally, the results of the final evaluation and leaderboard confirm the efficacy of our models. Furthermore, the fact that our models achieved either the second or third position in most of the languages on the leaderboard provides evidence of the models’ language independence.

Our findings suggest that using a combination of techniques from different domains, such as NLP and graph theory, can lead to substantial improvements in multilingual understanding and knowledge representation. The ability to categorize Wikipedia entities across multiple languages and establish a unified ontology can pave the way for the creation of large knowledge bases, which in turn can be utilized for various other tasks under the umbrella of information access.

In the future, further improvements can be made by exploring more advanced techniques, such as using pre-training language models and developing more sophisticated graph-based architectures. Overall, the results of our study are promising and highlight the potential for advancements in multilingual NLP research.

7.1.2 Cross-lingual Fact-to-Text Generation

We focused on seven low-resource (LR) languages of South Asia, which have significantly fewer entries in their Wikidata compared to English. With a view to content enrichment in these languages and considering the difficulty of monolingual fact-to-text generation due to the sparsity of factual data in these languages, we proposed a new task called cross-lingual F2T generation (XF2T), where English facts are used as input to generate sentences that capture the fact-semantics in the specified LR language. We also created a new dataset called XAlign, which consists of high-quality pairs of English facts and semantically equivalent LR language text for experimenting on our proposed task. The dataset contains 0.55 million pairs, and is split into automatically aligned and human-aligned pairs for training and evaluation respectively.

The automated pipeline for dataset creation consisted of candidate generation for generating fact-text pairs based on cosine similarity scores, followed by candidate selection for selecting fact-text pairs with a high confidence, where we made use of Xtreme-NLI and KELM datasets for transfer learning. We detailed our annotation scheme for manual generation of the dataset to ensure quality marking of facts from sentences, and found the distribution of facts in the automated and manually created datasets to be similar - indicating the success of our dataset creation. In our experiments for fact-to-text generation, we augmented the input to our multilingual mT5 model to also separately account for the type of factual information supplied at each token position in the form of fact-aware embeddings, and established benchmark results on the task.

Future work can expand upon our contributions by augmenting the data with additional languages, and experimenting on a larger multilingual scale. We noted drops in performance for generation from larger number of facts supplied at the input, which can be another interesting direction for improving

the performance. A separate, related problem that can be worked on after XF2T would be generation of paragraph-level coherent text using multiple facts at the input, and it presents exciting challenges with respect to fact ordering, coherent generation and avoidance of hallucination to the research community.

7.1.3 Cross-lingual Fact Extraction

We propose the task of Cross-lingual Fact Extraction (CLFE) for 7 LR South Asian languages and English, aiming to extract English triples directly from text in these languages. We provide strong baselines and approaches, yielding results comparable to existing fact extraction pipelines for monolingual languages and significantly better than previous cross-lingual fact extraction attempts. We experiment on this problem using the XAlign dataset. Our work enables the use of factual knowledge from South Asian texts to expand existing knowledge graphs, potentially useful in fact verification and text generation.

Specifically, we discussed two methods. The first method involved entity extraction on translated text followed by relation classification from a set of predefined relations in our dataset. The second, intuitively simpler approach involved use of the mT5 model for direct extraction of the complete factual triplet. We found the second approach to decisively outperform the former, and observed the differences in the performance of this approach on three different settings: completely multilingual modeling, script unification of the South Asian languages by transliterating to Devanagari, and training one model per cross-lingual pair. We observed different linguistic datasets to perform better on either of the settings.

This work opened the doors for further study on methods of extracting knowledge from text in low-resource languages. Future work can focus on developing approaches that can address the partially aligned nature of the dataset in future work to achieve even better results.

7.2 Inferences from the Study on Bias Identification & Mitigation

In this study, we deal with bias identification and mitigation as a multilingual problem on Wikipedia texts. Our objective is to tackle the problem of subjective bias in Wikipedia data and aid in making information more precise and widely accessible. We start with creating a multilingual dataset of 7 South Asian languages along with English. Our dataset is based on translation of existing English-based datasets of WNC and WIKIBIAS, and we propose this strategy in contrast to several other approaches we tried. We establish competitive baselines by utilizing classification-based models to detect bias on our dataset. For bias mitigation, we utilize the style transfer paradigm and model it using transformer-based seq2seq architectures. We experiment with monolingual and multilingual settings for both the problems, and observe the decisively better performance of the multilingual setting for both the problems.

We also list out approaches that could be tried out in future work on this problem. We suggest incorporating contextual information such as category, title, or citation to identify sentence-level bias. We also propose using the bias classifier score as a reward function for reinforcement learning in the generative module. Additionally, we suggest exploring pre-training objectives, data augmentation

techniques, and evaluating debiasing in low-resource languages and other types of biased language data. Finally, we propose developing more fine-grained metrics to assess the quality of debiasing, such as fluency, bias, and meaning.

7.3 Inferences from the Study on Clickbait Spoiling

This study focused on the problem of mitigating the effects of the attractive nature of content as lured by clickbaits by spoiling them. While previous research primarily focused on identifying and categorizing clickbaits, we worked on a 2-stage pipeline to spoil the clickbait by providing a satisfactory answer to the surprise element it alludes to. The first stage involved identification of the type of spoiler required for that clickbait, while the second stage involved extraction of the necessary span from the article as the “spoiler”. Our approach relied on our proposed technique of “Information Condensation” to reduce unnecessary content in the article, aiding both stages.

We proposed a high-recall system for paragraph filtering in the article to achieve information condensation needed, and found this proxy task to be rather challenging to preserve a high likelihood of the condensed article containing a spoiler. Our contrastive learning-based model was effective in achieving high recall with good precision as opposed to several experiments performed in the classification paradigm. Out of the two modeling techniques were tried for the contrastive learning model - SIMPL and CONCT - we noted the better performance of SIMPL due to better information exchange between the clickbait-paragraph input pair facilitating a better modeling. Although effective for spoiler type identification, we found the paragraph filtering strategy to have a potential for negatively affecting the models trained to generate spoilers by disrupting textual coherence, besides actual loss of spoiler content. Overall, our proposed approach established SoTA results on the newly proposed task, and our work also provided a detailed analysis and ablation study for a better interpretability.

There are several interesting directions for future work that can be taken up as a continuation of our study. Firstly, considering the rather small amount of training corpus available for spoiler generation, an investigative analysis of various extractive question-answering datasets can be conducted to adapt the models better for the downstream task. The task of multipart spoiler generation can be modeled using a recursive strategy which involves extraction of individual spans followed by their deletion from the article to extract a new span at each step. Another interesting method would be to explore generative methods for generating spans instead of extracting them as opposed to the span extraction method currently used. Exploring the synergy between the two tasks of spoiler-type classification and spoiler generation through multi-task learning can be explored to observe the benefit on either or both the tasks. With the advent of large language models having superior capabilities for content understanding and generation, future work can also look at unifying both the stages of the pipeline to directly achieve spoiler generation, thus mitigating the noise propagation from the first stage to the second.

Related Publications

1. **Anubhav Sharma***, Sagar Joshi*, Tushar Abhishek, Radhika Mamidi, Vasudeva Varma. **Billy-Batson at SemEval-2023 Task 5: An Information Condensation based System for Clickbait Spoiling**. In 17th International Workshop on Semantic Evaluation (SemEval) 2023. (*Accepted, to be published.*)
2. **Anubhav Sharma***, Ankita Maity*, Tushar Abhishek, Rudra Dhar, Radhika Mamidi, Manish Gupta, Vasudeva Varma. **Multilingual Bias Detection and Mitigation for Low Resource Languages**. In Proceedings of the Wiki Workshop 2023. (*Accepted, to be published.*)
3. Tushar Abhishek, Ayush Agarwal, **Anubhav Sharma**, Vasudeva Varma, Manish Gupta. **Rehoboam at the NTCIR-15 SHINRA2020-ML Task**. The 15th NTCIR Evaluation of Information Access Technologies 2020.
4. Tushar Abhishek, Shivprasad Sagare, Bhavyajeet Singh, **Anubhav Sharma**, Manish Gupta, Vasudeva Varma. **XAlign: Cross-lingual Fact-to-Text Alignment and Generation for Low-Resource Languages**. In Companion Proceedings of the Web Conference (WWW '22 Companion) 2022.
5. Shivprasad Sagare, Tushar Abhishek, Bhavyajeet Singh, **Anubhav Sharma**, Manish Gupta, Vasudeva Varma. **XF2T: Cross-lingual Fact-to-Text Generation for Low-Resource Languages**. arXiv 2022.
6. Bhavyajeet Singh, Siri Venkata Pavan Kumar Kandru, **Anubhav Sharma**, Vasudeva Varma. **Massively Multilingual Language Models for Cross Lingual Fact Extraction from Low Resource Indian Languages**. In Proceedings of the 19th International Conference on Natural Language Processing (ICON) 2022.

Bibliography

- [1] O. Agarwal, H. Ge, S. Shakeri, and R. Al-Rfou. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *NAACL-HLT*, pages 3554–3565, 2021.
- [2] G. Angeli, M. J. Johnson Premkumar, and C. D. Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China, July 2015. Association for Computational Linguistics.
- [3] G. Attardi. Wikiextractor. <https://github.com/attardi/wikiextractor>, 2015.
- [4] K. Bali, M. Choudhury, and P. Biswas. Indian language pos tagset: Bengali. *Linguistic Data Consortium, LDC2010T16*, 2010.
- [5] S. Banerjee and A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.
- [6] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
- [7] A. Chakraborty, B. Paranjape, S. Kakarla, and N. Ganguly. Stop clickbait: Detecting and preventing clickbaits in online news media. In *2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)*, pages 9–16. IEEE, 2016.
- [8] C. Z. Charu C. Aggarwal. Mining text data, 2012.
- [9] M. X. Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, N. Parmar, M. Schuster, Z. Chen, et al. The best of both worlds: Combining recent advances in neural machine translation. *arXiv preprint arXiv:1804.09849*, 2018.
- [10] Y. Chen, N. J. Conroy, and V. L. Rubin. Misleading online content: recognizing clickbait as” false news”. In *Proceedings of the 2015 ACM on workshop on multimodal deception detection*, pages 15–19, 2015.
- [11] Z. Chi, L. Dong, F. Wei, N. Yang, S. Singhal, W. Wang, X. Song, X.-L. Mao, H. Huang, and M. Zhou. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*:

- Human Language Technologies*, pages 3576–3588, Online, June 2021. Association for Computational Linguistics.
- [12] R. Dabre, H. Shrotriya, A. Kunchukuttan, R. Puduppully, M. Khapra, and P. Kumar. IndicBART: A pre-trained model for indic natural language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1849–1863, Dublin, Ireland, May 2022. Association for Computational Linguistics.
 - [13] H. Elsahar, P. Vougiouklis, A. Remaci, C. Gravier, J. Hare, F. Laforest, and E. Simperl. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
 - [14] F. Feng, Y. Yang, D. M. Cer, N. Arivazhagan, and W. Wang. Language-agnostic bert sentence embedding. In *Annual Meeting of the Association for Computational Linguistics*, 2020.
 - [15] M. Fröbe, T. Gollub, M. Hagen, and M. Potthast. SemEval-2023 Task 5: Clickbait Spoiling. In *17th International Workshop on Semantic Evaluation (SemEval-2023)*, 2023.
 - [16] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, and M. Potthast. Continuous Integration for Reproducible Shared Tasks with TIRA.io. In *Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023)*, Lecture Notes in Computer Science, Berlin Heidelberg New York, Apr. 2023. Springer.
 - [17] S. Garg, T. Vu, and A. Moschitti. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. In *AAAI Conference on Artificial Intelligence*, 2019.
 - [18] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org, 2010.
 - [19] M. Hagen, M. Fröbe, A. Jurk, and M. Potthast. Clickbait Spoiling via Question Answering and Passage Retrieval. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*, pages 7025–7036. Association for Computational Linguistics, May 2022.
 - [20] P. He, X. Liu, J. Gao, and W. Chen. Deberta: Decoding-enhanced bert with disentangled attention. *ArXiv*, abs/2006.03654, 2020.
 - [21] M. Honnibal and I. Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 2018.
 - [22] Z. Huang, W. Xu, and K. Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.
 - [23] A. Kandimalla, P. Lohar, S. K. Maji, and A. Way. Improving english-to-indian language neural machine translation systems. *Information*, 13(5):245, 2022.
 - [24] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, S. Gupta, S. C. B. Gali, V. Subramanian, and P. Talukdar. Muril: Multilingual representations for indian languages, 2021.

- [25] Y. Khemchandani, S. Mehtani, V. Patil, A. Awasthi, P. Talukdar, and S. Sarawagi. Exploiting language relatedness for low web-resource language model adaptation: An indic languages study. *arXiv preprint arXiv:2106.03958*, 2021.
- [26] P. Koechley. Why the title matters more than the talk, 2012.
- [27] K. Kolluru, M. Rezk, P. Verga, W. W. Cohen, and P. Talukdar. Multilingual fact linking, 2021.
- [28] R. Koncel-Kedziorski, D. Bekal, Y. Luan, M. Lapata, and H. Hajishirzi. Text Generation from Knowledge Graphs with Graph Transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2284–2293, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [29] A. Kunchukuttan. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf, 2020.
- [30] G. Lample and A. Conneau. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- [31] R. Lebrecht, D. Grangier, and M. Auli. Neural text generation from structured data with application to the biography domain. *arXiv preprint arXiv:1603.07771*, 2016.
- [32] R. Li, D. Li, J. Yang, F. Xiang, H. Ren, S. Jiang, and L. Zhang. Joint extraction of entities and relations via an entity correlated attention neural model. *Information Sciences*, 581:179–193, 2021.
- [33] J. Lin, R. Nogueira, and A. Yates. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325, 2021.
- [34] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.
- [35] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019.
- [36] G. Loewenstein. The psychology of curiosity: A review and reinterpretation. *Psychological bulletin*, 116(1):75, 1994.
- [37] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.
- [38] F. Meng and J. Zhang. Dtmr: A novel deep transition architecture for neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 224–231, 2019.
- [39] P. Mormol. ‘i urge you to see this...’. clickbait as one of the dominant features of contemporary online headlines. *Social Communication*, 5(2):1–10, 2019.
- [40] N. Muennighoff, T. Wang, L. Sutawika, A. Roberts, S. Biderman, T. L. Scao, M. S. Bari, S. Shen, Z.-X. Yong, H. Schoelkopf, X. Tang, D. Radev, A. F. Aji, K. Almubarak, S. Albanie, Z. Alyafeai, A. Webson, E. Raff, and C. Raffel. Crosslingual generalization through multitask finetuning, 2022.
- [41] G. Nikolentzos, A. J. Tixier, and M. Vazirgiannis. Message passing attention networks for document understanding. In *AAAI*, pages 8544–8551. AAAI Press, 2020.

- [42] A. Pal, M. Selvakumar, and M. Sankarasubbu. MAGNET: multi-label text classification using attention-based graph neural network. In *ICAART (2)*, pages 494–505. SCITEPRESS, 2020.
- [43] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.
- [44] C. Patel and K. Gali. Part-of-speech tagging for gujarati using conditional random fields. In *IJCNLP Workshop on NLP for Less Privileged Languages*, 2008.
- [45] A. Peysakhovich and K. Hendrix. News feed fyi: Further reducing clickbait in feed. *Facebook newsroom*, 2016.
- [46] T. Pires, E. Schlinger, and D. Garrette. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy, July 2019. Association for Computational Linguistics.
- [47] M. Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics.
- [48] M. Potthast, T. Gollub, M. Hagen, and B. Stein. The clickbait challenge 2017: Towards a regression model for clickbait strength, 2018.
- [49] R. Pryzant, R. D. Martinez, N. Dass, S. Kurohashi, D. Jurafsky, and D. Yang. Automatically neutralizing subjective bias in text. In *Proceedings of the aaai conference on artificial intelligence*, volume 34, pages 480–489, 2020.
- [50] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *ACL Demos*, 2020.
- [51] P. Rajpurkar, R. Jia, and P. Liang. Know what you don’t know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics.
- [52] M. Recasens, C. Danescu-Niculescu-Mizil, and D. Jurafsky. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1659, 2013.
- [53] M. M. U. Rony, N. Hassan, and M. Yousuf. Diving deep into clickbaits: Who use them to what extents in which topics with what effects? In *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017*, pages 232–239, 2017.
- [54] S. Sekine, M. Nomoto, K. Nakayama, A. Sumida, K. Matsuda, and M. Ando. Overview of shinra2020-ml task. In *Proceedings of the NTCIR-15 Conference*, 2020.
- [55] K. Shu, S. Wang, T. Le, D. Lee, and H. Liu. Deep headline generation for clickbait detection. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 467–476, 2018.

- [56] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, 2019.
- [57] S. Srivastava, M. Patidar, S. Chowdhury, P. Agarwal, I. Bhattacharya, and G. Shroff. Complex question answering on knowledge graphs using machine translation and multi-task learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3428–3439, Online, Apr. 2021. Association for Computational Linguistics.
- [58] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. *CoRR*, abs/1710.10903, 2017.
- [59] D. Vrandečić and M. Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, sep 2014.
- [60] P. Xu, C.-S. Wu, A. Madotto, and P. Fung. Clickbait? sensational headline generation with auto-tuned reinforcement learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3065–3075, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.
- [61] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online, June 2021. Association for Computational Linguistics.
- [62] S. Zhang, K. Duh, and B. Van Durme. MT/IE: Cross-lingual open information extraction with neural sequence-to-sequence models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 64–70, Valencia, Spain, Apr. 2017. Association for Computational Linguistics.
- [63] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.
- [64] Y. Zhong, J. Yang, W. Xu, and D. Yang. WIKIBIAS: Detecting multi-span subjective biases in language. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1799–1814, Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.
- [65] Z. Zhong and D. Chen. A frustratingly easy approach for entity and relation extraction. *arXiv preprint arXiv:2010.12812*, 2020.
- [66] Y. Zhou. Clickbait detection in tweets using self-attentive network. *arXiv preprint arXiv:1710.05364*, 2017.