A Framework for Non-factoid Question Answering in Indic Languages

Submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

By

Ritwik Mishra
PhD19013

Under the supervision of

Prof. Rajiv Ratn Shah

and

Prof. Ponnurangam Kumaraguru

Department of Computer Science & Engineering
Indraprastha Institute of Information Technology Delhi
New Delhi - 110020, India
May 2025

Dedicated to my homeland
India, that is भारत

Certificate

This is to certify that the work presented in this thesis, entitled **A Framework for Non-factoid Question Answering in Indic Languages**, has been submitted to fulfill the requirements for the PhD program at Indraprastha Institute of Information Technology Delhi (IIIT Delhi). To the best of my knowledge and belief, this thesis does not contain any material previously published or authored by another individual, except where proper citations have been provided.

Prof. Rajiv Ratn Shah
Associate Professor
Indraprastha Institute of Information Technology Delhi (IIIT Delhi)
New Delhi 110020, India

Prof. Ponnurangam Kumaraguru
Professor
International Institute of Information Technology Hyderabad (IIIT Hyderabad)
Telangana 500032, India

Declaration

I hereby declare that the work presented in this thesis is an authentic record of my own efforts, conducted under the supervision and guidance of my supervisor during the period from September 2019 to May 2025. This work has not been submitted to any other institution for any degree or diploma. I have adhered to the institute's guidelines for thesis preparation and complied with ethical norms throughout the writing process. Furthermore, whenever materials such as data, models, figures, or text from other sources have been utilized, due credit has been given through proper citations in the thesis, along with detailed references. Permissions from copyright owners have been obtained where required.

Date:

Ritwik Mishra
PhD19013
IIIT Delhi

Acknowledgments

My journey into the world of Computer Science (CS) and programming began in the 11th standard (Junior year). It was catalyzed by my classmates, Utkarsh and Moksh, who refused to share their code for printing a triangle of asterisks of height n. Determined to solve it independently, I discovered that programming encompassed the very elements I appreciated in science—logic, mathematics, and development. As my undergraduate studies in CS progressed, it became evident that the breadth of the field was immense, and the limited duration of the program was insufficient to thoroughly explore and appreciate every subject. Lectures by Dr. Kundan Kumar Chandan were particularly inspiring and deepened my interest in the discipline. The 2014 film *The Imitation Game*, based on the life of Alan Turing, sparked a keen interest in network security and cryptanalysis. However, the prospect of pursuing this domain was curtailed by the retirement of Prof. Karmeshu, whose mentorship I had aspired to seek. Around this time, I became increasingly intrigued by the parallels between natural human languages and cryptographic systems—both facilitating communication while remaining opaque to outsiders. This fascination guided my decision to pursue a master's degree in Natural Language Processing (NLP) and Artificial Intelligence (AI) under the mentorship of Dr. Tirthankar Gayen, whose guidance instilled in me the value of wholehearted commitment to any endeavor. Motivated by a growing curiosity in NLP, I aimed to develop a deep expertise in the area. I am sincerely grateful to Prof. Karmeshu and my friends—Prateek, Ranjodh, and Abhineet—for encouraging me to pursue doctoral studies at institutions with greater resources and stronger reputation in AI.

My time at IIIT-Delhi has been marked by access to exceptional facilities, resources, and opportunities, making it an ideal environment for doctoral research in AI. I remain deeply indebted to my doctoral advisors, Prof. Rajiv Ratn Shah and Prof. Ponnurangam Kumaraguru, for accepting me as their student and for their unwavering support across diverse research projects. I am also thankful to ACM-India and the Anveshan Setu Fellowship program for the opportunity to work under the guidance of Prof. Pushpak Bhattacharya, whose mentorship greatly enhanced my skills in academic writing. Heartfelt gratitude goes to my friends in the MIDAS lab and Precog lab—Avinash, Hitkul, Shivangi, Hemant, Mann, Prashant, Hanuma, Raaj, Adarsh, Anam, Mohit, Nidhi, Neha, AT, Aman, Yaman, Ritika, Rohan, and Rishabh. They endured many of my long intellectual discussions and taught

me the art of patience and listening. I am deeply indebted to our *guru-maa*, Dr. Rushali, for her emotional mentorship. I am also grateful to Rama for his kind affirmation that *Ritwik bhaiya is a good boy.* Finally, I appreciate Binu and Nikita for nurturing a warm and collegial environment.

I am equally grateful to the friends I met during my undergraduate and postgraduate studies, whose willingness to engage with my persistent questions and opinions continues to this day. Above all, I express my deepest gratitude to my family. My academic journey would have been impossible without their selfless support. My parents bore many personal hardships while shielding me from them, and my sisters assumed additional responsibilities during critical phases of my research. The collective strength of my joint family provided the foundation on which I could stand, walk, and eventually run. And finally, to God—thank you for giving me such incredible people and life-changing opportunities. I will always remain indebted to you. Let me know if you need anything :)

Abstract

We address the development of explainable Question Answering (QA) systems for Indic languages, focusing on the unique challenges posed by resource scarcity and the complexities of multilingual processing. The research begins by categorizing QA systems based on context, domain, conversational requirements, and answer types, emphasizing the importance of text-based QA for cognitive development. A comprehensive literature review highlights advances in factoid and non-factoid QA, the rise of Transformer-based models, and the critical role of retrieval mechanisms for handling extended contexts. Our work also identifies significant gaps in resources for Indic languages, particularly for non-factoid QA, and underscores the necessity for efficient, explainable, and retrieval-augmented models. To address the lack of structured knowledge extraction tools for low-resource languages, the thesis introduces IndIE, an Open Information Extraction (OIE) system designed for Hindi. IndIE employs a multilingual pretrained transformer, fine-tuned on chunk-annotated data from English and five Indic languages, to generate triples from unstructured sentences. In sequence labeling tasks (like chunking), it was found that the mean of subword token embeddings is more beneficial than other approaches. The system leverages chunk tagging and Merged-Phrase Dependency Trees, achieving a 0.51 F1-score on a benchmark of 112 Hindi sentences and producing more granular triples than existing multilingual approaches. The underlying methodology demonstrates potential for extension to Urdu, Tamil, and Telugu, given the generalizability of the chunker and the language-agnostic nature of the triple extraction rules.

Recognizing the challenge of resolving references to the same entity across text, the thesis present TransMuCoRes, a multilingual coreference resolution dataset spanning 31 South Asian languages. Using automated translation and word alignment, TransMuCoRes fills a critical resource gap for coreference tasks in these languages. Two coreference models, trained on a combination of TransMuCoRes and manually annotated Hindi data, achieve LEA F1 and CoNLL F1 scores of 64 and 68, respectively, on a Hindi test set. The work also critiques current evaluation metrics, advocating for improved measures to handle split antecedents. Building on these foundational tools, the thesis introduces MuNfQuAD, a multilingual non-factoid QA dataset comprising over 578K question-answer pairs across 38 languages, including numerous low-resource languages. Questions are derived from in-

terrogative sub-headings in BBC news articles, with corresponding paragraphs serving as silver-standard answers. Manual annotation of 790 pairs reveals that 98% of questions are answerable using the provided context. An Answer Paragraph Selection (APS) model, fine-tuned on this dataset, achieves 80% accuracy and 72% macro F1 on the test set, and 72% accuracy and 66% macro F1 on the golden set, outperforming baseline methods and demonstrating effective context reduction.

The thesis further investigates explainability in QA and related tasks. Through experiments on the HateXplain benchmark, it compares three post-hoc interpretability methods for transformer-based encoders in hate speech detection. Notably, Layerwise Relevance Propagation (LRP) underperforms, sometimes even less informative than random rationale generation, due to its tendency to focus on initial tokens. This finding highlights the limitations of LRP for explaining fine-tuned transformer predictions. To enhance QA performance for long contexts, especially in Indic languages, the thesis explores various context-shortening strategies based on OIE, coreference resolution, and APS. Experiments with three popular Large Language Models (LLMs) on Hindi, Tamil, Telugu, and Urdu show that these techniques improve semantic scores by an average of 4% and token-level scores by 47% without fine-tuning, and by 2% with fine-tuning, while also reducing computational demands. Explainability analyses using LIME and SHAP indicate that APS-selected paragraphs concentrate model attention on relevant tokens. However, the study notes persistent challenges for LLMs in non-factoid QA requiring reasoning, and finds that verbalizing OIE triples does not further enhance performance.

As a retrospective epilogue of the thesis, we also present a Hindi chatbot for maternal and child health queries. Using a curated FAQ database and an ensemble of rule-based, embedding-based, and paraphrasing classifiers, the system covers 80% of user queries and retrieves at least one relevant answer in the top three suggestions for 70% of cases. Collectively, this work advances the state of explainable QA for Indic languages by developing novel resources, tools, and evaluation frameworks, and by demonstrating the effectiveness of context-shortening and interpretability techniques in low-resource, multilingual settings. Future work in QA systems for Indic languages includes expanding benchmarks like Hindi-BenchIE to other low-resource languages for standardized evaluation of triple extraction methods, thus advancing multilingual OIE. The release of TransMuCoRes checkpoints offers a baseline for multilingual coreference resolution research. Using APS models as reward models for LLM alignment may improve answer accuracy for complex queries. Additional directions involve deploying chatbots in real-world settings, refining OIE and coreference models, expanding multilingual QA datasets, and enhancing explainability. Evaluating systems on longer contexts and integrating advanced alignment strategies will foster robust, transparent QA frameworks for Indic languages.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
APS	Answer Paragraph Selection
ASHA	Accredited Social Health Activists
FAQs	Frequently Asked Questions
FFNN	Feed-Forward Neural Network
IVC	Indus Valley Civilization
IVR	Indigenous Language Revitalization
LIME	Local Interpretable Model-agnostic Explanations
LLM	Large Language Model
LRP	Layer-wise Relevance Propagation
mAP	Mean Average Precision
MRC	Machine Reading Comprehension
MRR	Mean Reciprocal Rank
nDCG	normalized Discounted Cumulative Gain
NLP	Natural Language Processing
OIE	Open Information Extraction
POS	Part-Of-Speech
QuAD	Question-Answering dataset
RAG	Retrieval-Augmented Generation
RNN	Recurrent Neural Networks
SHAP	SHapley Additive exPlanations
SPC	Sentence Pair Classification
SR	Success Rate
XAI	Explainable Artificial Intelligence

Chapter 1

Introduction

Question-Answering (QA) has long played a crucial role in facilitating knowledge acquisition for humans. The advancement of QA systems has significantly impacted information retrieval and comprehension, making them a fundamental component of artificial intelligence research. This work aims to trace the evolution of QA systems, emphasizing their growing importance and transformative potential. To ensure a comprehensive understanding, this chapter introduces key terminologies frequently used in the domain of automated question-answering. Additionally, we articulate the rationale for selecting the specific modality addressed in this study, with a particular emphasis on the challenges and opportunities associated with Indic languages. Furthermore, this chapter provides an overview of the thesis structure, delineating its core contributions. A summary of the research outcomes, including key publications resulting from this work, is also presented.

Unicorn bull seal [14].

The ability to ask questions is considered one of the highest cognitive faculties in humans, distinguishing us from other animals. There have been significant efforts to communicate with various great apes using sign language [15]. Through extended interactions, humans have successfully engaged in conversations with apes, eliciting responses to questions regarding their dietary habits [16], behaviors [17], and even abstract concepts such as death [15]. In some instances, apes have formulated instructions (e.g., requests such as "give me *Brazil nuts*" or "bring my leash") in a manner resembling queries [18]. However, to date, there is no documented evidence of an ape independently posing an information-seeking question to a human interlocutor [19, 20].

During the Second World War, humanity endured numerous harrowing experiences. However, one notable outcome of this period was the inception of the field of Natural Language Processing (NLP), primarily aimed at developing mechanical methods for translating Russian to English [21, 22]. Within less than two decades, QA emerged as a subfield of NLP, focusing on the automatic generation or extraction of answers to posed questions [23]. The earliest QA system was designed to answer questions specifically related to baseball games. It processed questions read from punch cards, and representing them in the form of attribute-value tuples. Missing attributes were retrieved from associated data dictionaries. For instance, a punch card containing the question '*Where did the Yankees play on September 10 in 1960?*' is depicted in Figure 1.1. The corresponding extracted attributes and their values would be as follows:

```
Place = ?                    Team = Yankees
Month = September            Day = 10
Year = 1960
```

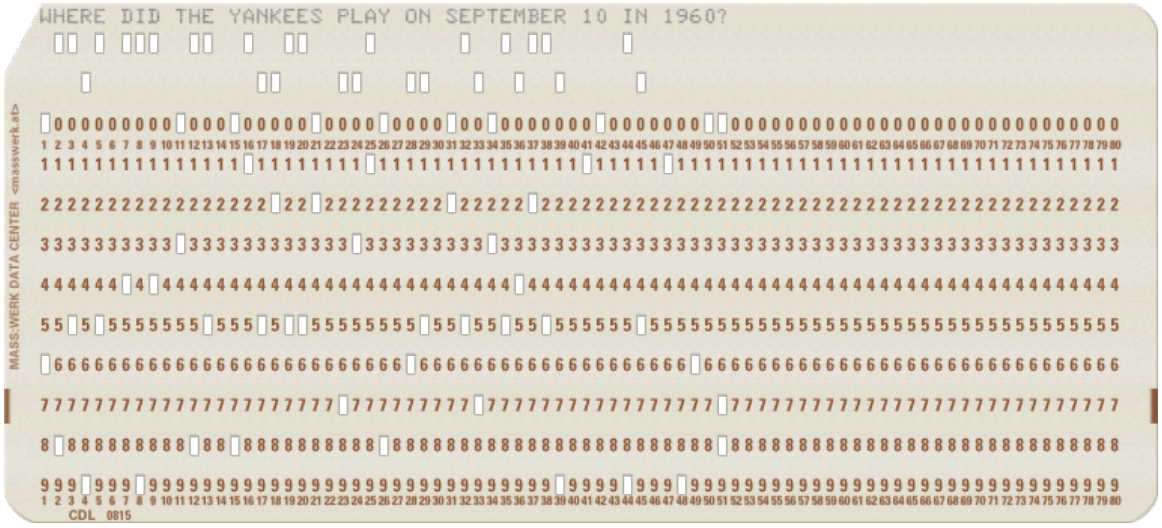


Figure 1.1: An image of a punch card carrying the question "*Where did the Yankees play on September 10 in 1960?*". Image generated by `https://www.masswerk.at/keypunch/`.

Alan Turing identified the efficacy of the QA methodology as a means to assess whether 'machines can think' [24]. The proposed evaluation, termed *The Imitation Game*, involved a human evaluator interacting with both a machine and a human through a series of questions to determine which participant was the machine. This test later became widely known as *the Turing Test*. Recent advancements in the fields of NLP and Artificial Intelligence (AI) have enabled the development of models that have successfully passed this *Turing Test* [25]. The study of QA tasks holds significant importance, as learning through questioning has been shown to effectively enable models to grasp complex nuances across various modalities, including text [26], video [27], and audio [28]. In this work, we have focused specifically on the text modality.

1.1 Modality

Our motivation for selecting the text modality arises from the recognition that written text has played a fundamental role in the advancement of human knowledge. In the biological domain, information is primarily transmitted across generations through genetic inheritance. While parents can impart limited knowledge to their offspring during their lifetime, ancestral information is predominantly passed down through genetic mutations. For instance, voles (a species of rodent) inhabiting prairie environments experience lower food availability and population density compared to their counterparts in mountainous regions [29]. Consequently, the vasopressin receptor gene in prairie voles has evolved to be longer than that in mountain voles, leading to strict monogamous behavior in the former [30]. However, humans have circumvented the reliance on genetic mutations for intergenerational knowledge transfer by developing a system commonly referred to as 'writing' [1].

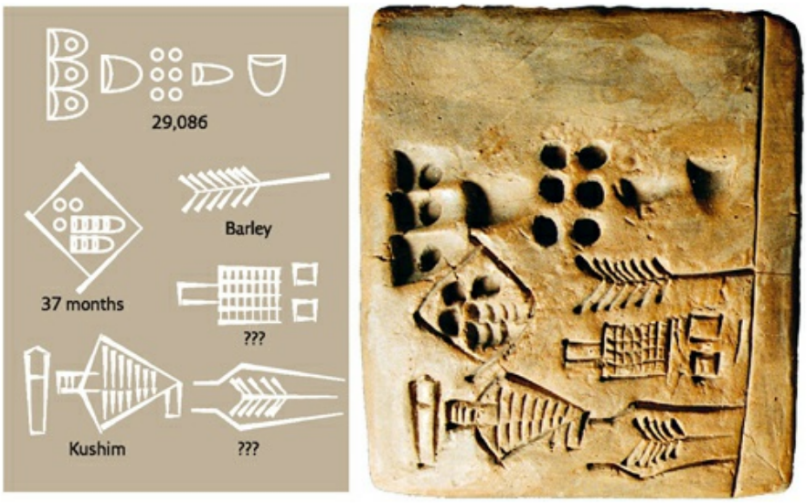


Figure 1.2: A tablet carved in clay (right) discovered from Uruk, Mesopotamia (now Iraq) belonging to the time period of 3400-3000 BCE. The tablet is an accounting record of food grain (left) [1].

Figure 1.2 illustrates one of the earliest records of human-written information. The written text modality facilitated the creation of manuscripts, books, and other forms of recorded knowledge, significantly accelerating the progress of human understanding. Therefore, this work focuses on the text modality for QA systems. QA systems based on textual data can be categorized into different groups based on their distinct features.

1.2 QA Features

Early text-based QA systems relied on hand-crafted rules to recognize patterns [31]. However, modern QA systems leverage models trained on labeled Question-Answering datasets (QuADs). The specific characteristics of any QA system are inherently influenced by the QuAD on which it has been trained [32, 33]. Figure 1.3 presents some common features of QA tasks within the text modality. The key characteristics of QuADs are outlined in the following subsections.

1.2.1 Domain

In a given QuAD, questions may pertain to a specific subject, field, or domain. Such datasets are categorized as closed-domain QuADs, whereas datasets containing questions derived

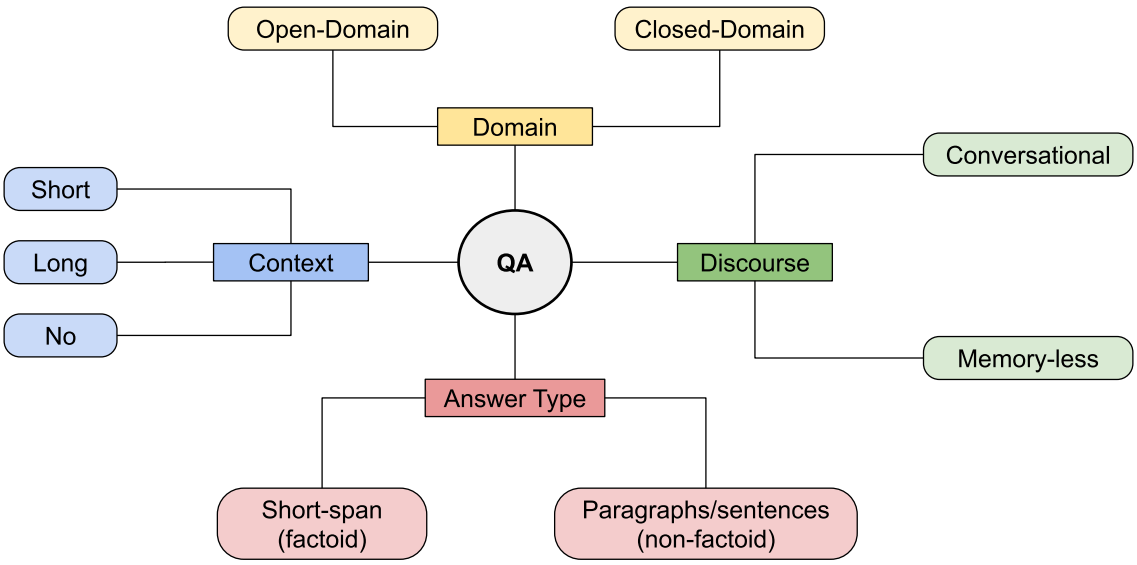


Figure 1.3: Common features of different Question-Answering (QA) systems, categorized based on key aspects such as domain, context length, discourse style, and answer type. QA systems can be either open-domain or closed-domain, depending on their scope of knowledge. They may also vary in context usage, ranging from no context to short or long contexts. Additionally, discourse styles can be conversational or memory-less, influencing how follow-up questions are handled. Lastly, QA systems produce different types of answers, either short-span factoid responses or longer, non-factoid paragraph-style answers.

from open-web sources or news portals are classified as open-domain QuADs. The advantage of closed-domain QA systems is their ability to provide reliable answers using a structured knowledge base [34]. Conversely, Large Language Models (LLMs) pre-trained on extensive text corpora from the web are better suited for handling open-domain question-answering tasks [35]. Examples of closed-domain QA systems include those designed for baseball [23] and healthcare [36]. In contrast, QA systems trained on open-domain QuADs, such as SQuAD [37], are classified as open-domain QA systems.

1.2.2 Discourse

Certain QA systems are designed to incorporate the context of previously asked questions and their corresponding answers to generate responses for the current query. These systems function as conversational agents, commonly referred to as chatbots. Training such QA systems requires a QuAD containing conversational data. Conversely, QA systems that process each question independently, without considering prior interactions, are characterized as memory-less discourse systems. For instance, the ChatGPT web interface operates as a conversational QA system by integrating previously asked questions and their answers with the current query [38]. Initially, this functionality did not persist across different chat sessions; however, in a recent update, ChatGPT has been enhanced to maintain a global memory across sessions, incorporating prior interactions into its responses [39]. Generally, QuADs are memory-less unless explicitly designed otherwise. Examples of conversational QuADs include CoQA [40] and TopiOCQA [41].

1.2.3 Answer Type

Certain questions can be adequately answered with just a few words or a short phrase. For instance, the question "Who was the first Prime Minister of India?" can be sufficiently answered with the phrase "Jawaharlal Nehru." In the field of Natural Language Processing, such questions are classified as factoid questions. In contrast, questions that necessitate comprehensive, multi-sentence responses are referred to as non-factoid questions. For example, the question "How did Jawaharlal Nehru become the first Prime Minister of India?" requires a detailed, explanatory answer.

In our analysis of the Natural Questions (NQ) dataset [42], which includes manual annotations distinguishing questions that cannot be answered with a factoid response, we observed clear differences between factoid (short) and non-factoid (long) answers. Out of the dataset, 78K questions had short, factoid answers, while 38K required true long, non-factoid answers. Syntactically, factoid answers were much simpler, with an average dependency tree depth of 2.6 (ranging from 1 to 15), compared to a much deeper average of 8.0 (ranging from 2 to 333) for non-factoid answers. Furthermore, the composition of Part-Of-Speech (POS) tags differed notably: in short answers, proper nouns, common nouns, and numerals (PROPN+NOUN+NUM) made up 72% of the answer tokens, whereas in long answers,

Name	Source	Domain	Context	Answer Type	Discourse	#QA
MedQuAD	Ben Abacha and Demner-Fushman 2019, [47]	Closed (Medicine)	No	Non-factoid	Memory-less	47K
LiteratureQA	Wang et al. 2021, [48]	Closed (Literature)	Short	Factoid	Memory-less	150K
ELI5	Fan et al. 2019, [49]	Open (Web)	Short	Non-factoid	Memory-less	270K
CoQA	Reddy et al. 2019, [40]	Open (Multiple)	Short	Factoid	Conversational	127K
TopiOCQA	Adlakha et al. 2022, [41]	Open (Wikipedia)	Long	Factoid	Conversational	50K
SQuAD	Rajpurkar et al. 2016, [37]	Open (Multiple)	Short	Factoid	Memory-less	100K
SearchQA	Dunn et al. 2017, [50]					140K
NaturalQA	Kwiatkowski et al. 2019, [42]	Open (Search Engine)	Long	Both	Memory-less	300K
MS MARCO	Nguyen et al. 2016, [51]					182K
TriviaQA	Joshi et al. 2017, [52]	Open (Multiple)	Long	Factoid	Memory-less	95K
NewsQA	Trischler et al. 2017, [53]					100K
HotpotQA	Yang et al. 2018, [54]					113K
NarrativeQA	Kočiskỳ et al. 2018, [55]					47K

Table 1.1: Popular Question-Answering Datasets (QuADs) with their corresponding features. The aforementioned works in the table have been collectively cited in more than 22K research papers.

these tags accounted for only 41%. These findings highlight that factoid answers tend to be short, syntactically simple, and dominated by entity or number tokens, while non-factoid answers are longer, structurally complex, and more lexically diverse.

Historically, the development of factoid QuADs has been more prominent, with well-known examples including SQuAD [37] and NaturalQA [42]. However, there has also been significant progress in the creation of non-factoid QuADs, as exemplified by works such as NLQuAD [4].

1.2.4 Context

A question posed to a QA system may be accompanied by a text or document from which the system is expected to extract or generate an answer. This differs from closed-domain QA systems, as the questions and their associated contexts can originate from any domain. An extension of QA is the task of extracting or generating answers based on a provided context, commonly referred to as Machine Reading Comprehension (MRC) [43]. The accompanying context may be either short or long. In the absence of a universally accepted definition for a long context, this study adopts a threshold of 512 tokens to classify a context as long. This threshold is primarily chosen because it aligns with the default input limit of transformer-based encoder architectures, such as BERT [44] and RoBERTa [45], which are widely utilized in QA models [46].

1.3 Non-factoid

Factoid questions, which can be answered with short spans of text, have been extensively studied in the field of QA [56]. Recent advancements in QA systems indicate that factoid questions can often be addressed with near-human performance [57]. Furthermore, Weissenborn et al. [58] demonstrated that answering factoid questions frequently relies on simple lexical heuristics between the question and its corresponding context.

In contrast, non-factoid QA has received comparatively less attention since the early stages of QA research [59]. Non-factoid QA systems are crucial for applications such as generating responses for voice assistants like Amazon Alexa [60]. Additionally, non-factoid questions are prevalent in web forums [61], and search queries requiring long descriptive answers are commonly observed in search queries on Bing [51]. Table 1.1 presents a selection of widely used English QuADs in the text modality along with their respective features. Unlike modalities such as images and videos, text-based modalities are inherently dependent on the underlying natural language and script. It has been observed that a significant proportion of available data resources are in English [33]. This study focuses on the development of QA systems and related resources for underrepresented languages within computational linguistics.

1.4 Indic Languages

India constitutes 2.4% of the world's land surface while supporting 16% of the global population and 23% of the world's languages [62, 63]. However, research suggests that the number of speakers of a language is not as significant a factor in language preservation as the natural process of intergenerational transmission [64]. Technology plays a crucial role in Indigenous Language Revitalization (IVR) by fostering a sense of pride in cultural and linguistic heritage [65]. Despite India's longstanding tradition of studying natural languages, the scarcity of computational resources for Indic languages remains an important avenue for exploration [57].

The Aṣṭādhyāyī (अष्टाध्यायी, pronounced *ash-tha-dhya-yee*), authored by the sage Pāṇini, stands as evidence that scholarly efforts to understand the structure of natural languages have been ongoing in the Indian subcontinent since the 5th century BCE. Furthermore, it continues to be an active subject of study in contemporary linguistics [66, 67, 68]. In the late 16th century, the Mughal emperor Akbar authorized language deprivation experiments, wherein mute wet nurses were instructed to raise orphaned children without linguistic exposure [69]. This initiative was driven by the pursuit of identifying an innate human language, reflecting the longstanding inquisitive tradition within the Indic subcontinent to empirically investigate the nature of human languages.

One possible approach to addressing the resource limitations of low-resource languages is to translate the input into a high-resource language, leverage existing NLP tools, and

subsequently back-translate the output into the original low-resource language. Some studies have explored this direction [70]. However, as discussed later in this thesis, this approach has several limitations. To illustrate why translation cannot serve as a permanent solution, we highlight the fundamental constraints of translation itself through the following example.

In 1989, a Malayalam-language comedy-crime-thriller film titled *Ramji Rao Speaking* was released. The film featured a scene in which the three male lead actors engage in a fistfight, during which the loincloth[1] of one character falls off without his realization. At that moment, an unknown woman enters the scene. One of the other men whispers into the ear of the man whose loincloth has just fallen, saying മുണ്ട് (*mundu*), which means 'loincloth' in Malayalam but is also colloquially used to mean 'talk.' The listener misinterprets the statement and responds, "I do not know her! You talk to her!"—resulting in a humorous situation. In 2000, the film was remade in Hindi under the title *Hera Pheri*. A direct translation of the scene into Hindi would not have been effective, as the words for 'talk' and 'loincloth' in Hindi are entirely distinct. Instead, in the Hindi adaptation, the actor whispers, उठा ले! (uthā le!, meaning 'pick up' or 'lift'), to which the half-clad character misunderstands and responds, "How can I lift her! I barely know her! Moreover, she must weigh 40-50 kilograms!" [71].

The challenges of translation extend beyond linguistic polysemy; cultural nuances embedded in language further complicate the process. For instance, consider the Hindi word जूठा (*joo-tha*), which is typically translated as *food that has been tasted by someone else*. However, within Indian cultural contexts, the term carries a deeper connotation. If a chef samples a dish to check for seasoning and spice balance, this act is referred to as चखना (tasting), not जूठा. The word जूठा bears a negative connotation, as it signifies a form of contamination where human contact (especially through saliva) renders the food or utensil impure. Historically, the term has also been employed as a tool for caste-based discrimination in certain parts of India, where individuals from lower-caste communities were deliberately given जूठा (leftover) food as an act of humiliation [72]. However, a contrasting cultural representation of this concept appears in a 1987 television adaptation of the ancient Sanskrit epic Ramayana. In one scene, Lord Ram is depicted consuming berries offered by his devoted follower, Shabri, an elderly tribal woman. In her desire to serve only the sweetest berries to Lord Ram, Shabri first tasted each berry, thereby inadvertently making them जूठा. Nonetheless, Ram willingly ate the berries, recognizing the sincerity and devotion behind her gesture. Notably, this specific incident does not appear in any of the five authoritative written versions of the Ramayana but was incorporated into the television series based on Hindi-Urdu theatrical adaptations performed in northern India [73].

Therefore, to address a question such as राम ने शबरी के जूठे बेर क्यों खाये? (Why did Ram eat Shabri's leftover berries?), any QA model must first understand the cultural nuance of the term जूठा, which does not always directly translate to "leftover". For instance, the sentence मैंने इसे जूठा किया है। (I have made it जूठा) translates to "I have made it impure." A

[1]Loincloth (*dhoti*) is a traditional men's garment in the Indian subcontinent, made of unstitched cotton, wrapped around the waist and legs, and secured by tucking or tying.

comprehensive understanding of such terms can only be achieved if QA models are trained on Indic language resources that frequently use the term. These examples underscore the fundamental limitations of translation, further emphasizing the necessity for developing dedicated tools and resources for low-resource languages.

Another motivation for developing tools and resources for Indic languages is that this work has been partially supported by the University Grants Commission (UGC) of India under the Junior Research Fellowship (JRF) scheme. This initiative is funded by the central government of India, which is elected through nationwide general elections held every five years. The authors believe that since nearly one billion people contribute, directly or indirectly, to supporting this research, it is imperative to work toward the development of their languages. Throughout the course of this thesis, the need to develop tools for Indic languages has become evident, particularly for problems such as Open Information Extraction and coreference resolution, which serve as essential supporting tasks in the proposed QA framework.

We hear many people saying that our languages are not rich enough in words to express our highest thinking. Gentlemen, this is no fault of the language. It is for us to develop and enrich our language.
- Mahatma Gandhi, 1917 [74]

1.5 Open Information Extraction

Open Information Extraction (OIE) aims to extract structured information, commonly represented as subject-predicate-object triples, from unstructured text without relying on a predefined ontology or schema. Unlike traditional information extraction methods that require a fixed set of relationships or entity types, OIE systems operate in an open-domain setting, enabling them to identify and extract relationships and facts from any text source. The objective is to transform free-form text into a structured format that can be efficiently processed and queried by machines. For example, from the sentence “Elon Musk founded SpaceX in 2002”, an OIE system would extract the triples (Elon Musk, founded, SpaceX) and (SpaceX, founded in, 2002). In this case, the system identifies the entities “Elon Musk” and “SpaceX” and the relationships ”founded” and ”founded in” without requiring predefined rules about companies, founders, or dates. This adaptability makes OIE valuable for tasks such as knowledge graph construction, question answering, and information retrieval from large and diverse text corpora.

1.6 Coreference Resolution

It is a NLP task that involves identifying and linking different text spans that refer to the same object or action in real-world. It aims to resolve pronouns, noun phrases, and

other referring expressions to their correct antecedents, thereby improving the coherence and understanding of a text. For example, in the sentence "Elon Musk founded SpaceX in 2002. He aimed to reduce the cost of space travel," a coreference resolution system would identify that "He" refers to "Elon Musk." Effective coreference resolution enhances various downstream NLP applications such as machine translation, information extraction, question answering, and summarization. In question answering, resolving coreferences helps the system maintain context across multiple sentences, while in summarization, it ensures that references to entities are consistent and clear. Coreference resolution is also critical in dialogue systems, where understanding user context and maintaining coherence over multiple conversational turns depends on correctly resolving references.

1.7 Problem Statement

The objective of this thesis is to develop a framework for Non-Factoid Automated QA in Indian languages by generating an explainable, query-specific short context from a longer context. This is achieved through the integration of multiple supporting tasks, including OIE, Coreference Resolution, and Information Retrieval. The problem statement has been visualized in Figure 1.4.

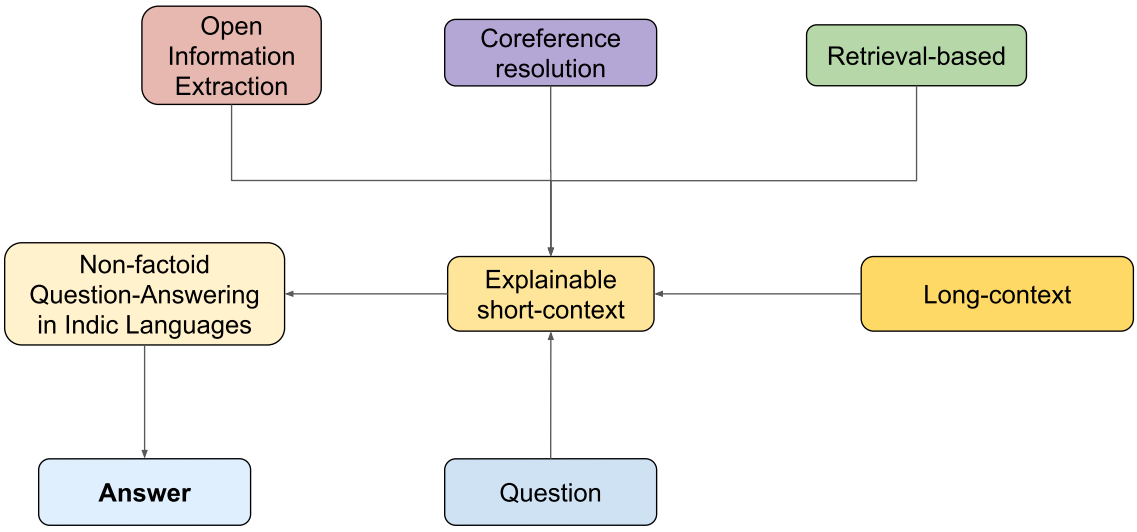


Figure 1.4: A visual representation of the problem statement, illustrating the process of developing a Non-Factoid Automated QA system for Indic languages. The framework aims to generate an explainable, query-specific short context from a longer context to improve answer retrieval.

1.8 Motivation

In the initial phases of our research, we observed that numerous significant works had been conducted on factoid-based QA within a small-context setting. While search engines like Google can index the open web and retrieve relevant sources for a given query, they are unable to index and process local documents. Some architectural innovations were proposed to extend the capability of existing methods to handle longer contexts; however, English data was primarily used by these models for pretraining. There was a notable scarcity of tools capable of processing long documents in Indian languages and enabling users to ask queries based on such documents. This gap led us to explore the possibility of developing a method for automatically answering questions based on long documents in Indian languages. Given the challenging nature of non-factoid questions and their prevalence in real-world scenarios, we decided to focus specifically on non-factoid QA.

As our research progressed, we identified the necessity of certain NLP techniques for building an effective framework for non-factoid question answering. Key examples include OIE and Coreference Resolution. The unavailability of such tools for Indian languages further motivated us to develop these methods ourselves. Additionally, incorporating explainability into a black-box QA system enhances its reliability, particularly in sensitive domains such as medical record queries. Therefore, the broad applicability of non-factoid QA systems and the lack of NLP resources for Indic languages were the primary motivations for pursuing this research direction.

1.9 Thesis Outline

We have structured the thesis into nine chapters, each organized to begin with a motivation outlining the significance of the chapter, followed by a description of the topics covered. Each chapter concludes with a summary that encapsulates its key points. Appendix is included at the end of the thesis, providing supplementary results related to our work. Additionally, the cover pages of the chapters feature illustrations of seals excavated from Indus Valley Civilization (IVC) sites, serving an aesthetic purpose while also representing some of the earliest known written text evidence from the Indian subcontinent.

- **Chapter 1** provides an introduction to the overarching theme of the thesis, emphasizing its significance and motivation. Additionally, it familiarizes readers with fundamental terminologies associated with QA systems, particularly in the context of non-factoid questions.

- **Chapter 2** presents a comprehensive review of the literature relevant to the core themes of this research. It examines foundational studies in question answering and its associated tasks while identifying key research gaps encountered during the course of this study.

- **Chapter 3** details the development of a triple extraction method for unstructured text in Indian languages. It highlights the limited research efforts in this area and critically evaluates the shortcomings of existing methods based on empirical observations. A comparative analysis is conducted between baseline approaches and the proposed method, which leverages dependency relations and handcrafted rules. The chapter concludes by summarizing key findings and outlining directions for further research.
- **Chapter 4** addresses the scarcity of resources for automatic coreference resolution targeting South Asian languages. It introduces a multilingual dataset constructed by automatically translating and aligning widely used English-language resources. Furthermore, it discusses the limitations of existing coreference evaluation metrics for multilingual datasets. The chapter concludes by summarizing the findings and providing motivation for subsequent research.
- **Chapter 5** examines the scarcity of multilingual Question-Answering Datasets and emphasizes the need for resources tailored to non-factoid question answering. It describes an automated approach for constructing a non-factoid QuAD through heuristic-based web scraping from a news portal. The performance of baseline models is evaluated and compared with the proposed Answer Paragraph Selection (APS) model, which is fine-tuned on the newly developed dataset. The chapter concludes by summarizing key findings and outlining directions for subsequent research.
- **Chapter 6** investigates the effectiveness of various post-hoc Explainable AI (XAI) methods by benchmarking them on a text classification task using manually annotated rationales. It provides a review of the existing literature comparing different XAI techniques while highlighting the lack of studies that evaluate these methods on the given benchmark. The chapter concludes with a discussion of the findings and potential avenues for further research.
- **Chapter 7** evaluates the state of QA systems in Indic languages following the advent of multilingual LLMs. It synthesizes the methodologies and resources introduced in previous chapters to assess the most effective techniques for enabling automated non-factoid question answering in Indic languages using various LLMs.
- **Chapter 8** offers retrospective epilogue to the thesis. It outlines the initial work undertaken in developing a chatbot designed to assist rural healthcare workers by answering queries related to maternal health. The chapter discusses the challenges inherent in this domain and explores a potential solution based on the available resources. It concludes by summarizing key findings and providing a motivation for subsequent work.
- **Chapter 9** concludes the thesis by discussing the potential applications of the research findings. It also acknowledges the limitations of the study and explores possible

directions for future work to further advance the field.

1.10 Contributions

Our research endeavors aim to push the boundaries of existing knowledge and explore new frontiers within the domain. To systematically address the overarching research problem outlined in this thesis, we have identified and formulated several subproblems, each targeting a specific aspect of the broader challenge. The methodologies employed in our investigations were carefully designed, drawing inspiration from an extensive review of existing literature and informed by prior empirical findings. These methodological choices were guided by the need to bridge identified research gaps and enhance the effectiveness of the proposed solutions. The key contributions of our research efforts can be summarized as follows:

1. We propose a greedy algorithm for extracting triples from unstructured text in four Indic languages, leveraging dependency parsing and chunking. Our study highlights the most effective strategies for obtaining token-level embeddings from transformer encoders in sequence labeling tasks. Furthermore, we introduce the first automated evaluation method for triple extraction in Hindi.

2. A large-scale multilingual coreference resolution dataset has been released, constructed by translating and aligning data from widely used English-language coreference resolution datasets. Additionally, we introduce the first-ever publicly available model checkpoint for multilingual coreference resolution. Furthermore, we highlight the limitations of existing evaluation metrics in assessing the effectiveness of coreference resolution across multiple languages.

3. We present the largest multilingual non-factoid Question-Answering Dataset (QuAD), comprising more than 578K question-answer pairs across 38 languages. In addition, we release model checkpoints for APS models, fine-tuned on a corpus exceeding 100 million training instances.

4. We conduct a comparative analysis of three post-hoc XAI methods using a benchmark dataset with manually annotated rationales. Our findings reveal that fine-tuned transformer models exhibit a tendency to encode the majority of the textual information within the first few tokens of an input sequence.

5. We explore the effectiveness of retrieval-based methods, triple extraction, and coreference resolution in reducing the context length for a given question. Our results indicate that retrieval-based methods achieve the best performance. Furthermore, we integrate explainability into the retrieval-based approach by leveraging insights from our previous work. We also release fine-tuned model checkpoints for the IndicGenQA framework. Additionally, we propose a novel metric for the automatic evaluation of Semantic Text Similarity for Multilingual Text (STS-MuTe).

6. We develop a FAQ-based QA system tailored for rural healthcare workers in India, built on a manually curated database of frequently asked questions. Our findings demonstrate that an ensemble approach combining rule-based methods, sentence embeddings, and paraphrase classification yields optimal performance. Additionally, we establish that fine-tuning on in-domain data followed by open-domain adaptation produces the best results. As part of this work, we release the largest Hindi paraphrase classification dataset, comprising over 17,000 positive examples.

1.11 Publications

1. Mishra, Ritwik, Sreeram Vennam, Rajiv Ratn Shah, and Ponnurangam Kumaraguru. "Multilingual Non-Factoid Question Answering with Answer Paragraph Selection" arXiv preprint arXiv:2408.10604 (2024). Accepted into the Special Session on 'Data Science: Foundations and Applications (DSFA)' of 29th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) 2025.

2. Mishra, Ritwik, Pooja Desur, Rajiv Shah, and Ponnurangam Kumaraguru. "Multilingual Coreference Resolution in Low-resource South Asian Languages." In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pp. 11813-11826. 2024.

3. Mishra, Ritwik, Ajeet Yadav, Rajiv Ratn Shah, and Ponnurangam Kumaraguru. "Explaining Finetuned Transformers on Hate Speech Predictions Using Layerwise Relevance Propagation." In International Conference on Big Data Analytics, pp. 201-214. Cham: Springer Nature Switzerland, 2023.

4. Mishra, Ritwik, Simranjeet Singh, Rajiv Shah, Ponnurangam Kumaraguru, and Pushpak Bhattacharyya. "IndIE: A Multilingual Open Information Extraction Tool For Indic Languages." In Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings), pp. 312-326. 2023.

5. Mishra, Ritwik, Simranjeet Singh, Jasmeet Kaur, Pushpendra Singh, and Rajiv Shah. "Hindi chatbot for supporting maternal and child health related queries in rural india." In Proceedings of the 5th Clinical Natural Language Processing Workshop, pp. 69-77. 2023.

6. Mishra, Ritwik, Rajiv Ratn Shah, and Ponnurangam Kumaraguru. "Long-Context Non-Factoid Question Answering in Indic Languages." arXiv preprint arXiv:2504.13615. Accepted to 13th International Conference on Big Data & AI (2025).

Miscellaneous

1. Mishra, Ritwik, Ponnurangam Kumaraguru, Rajiv Ratn Shah, Aanshul Sadaria, Shashank Srikanth, Kanay Gupta, Himanshu Bhatia, and Pratik Jain. "Analyzing traffic vio-

lations through e-challan system in metropolitan cities (workshop paper)." In 2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM), pp. 485-493. IEEE, 2020.

1.12 Chapter Summary

QA has been one among many foundational tasks in the field of NLP. QA systems can be classified based on several key characteristics, including: (a) the nature of the contextual information they process, (b) the subject domain of questions, (c) whether a question is self-contained or requires conversational context for comprehension, and (d) the expected type of the answer. This research focuses on developing an explainable QA framework for Indic languages, specifically in the text modality since it plays a fundamental role in the cognitive evolution of humans. In pursuit of this goal, the thesis also evaluates several supporting tasks essential to QA, namely Open Information Extraction, coreference resolution, and information retrieval. The chapter concludes by presenting an outline of the thesis, highlighting its primary contributions, and summarizing the key publications resulting from this work.

Chapter 2

Literature Survey

Building upon the foundational understanding of QA systems, we provide a comprehensive literature review to contextualize the challenges and advancements in QA research, particularly for Indic languages. Given the growing reliance on QA models in real-world applications, it is imperative to examine existing methodologies, their limitations, and potential research directions. This chapter explores factoid and non-factoid QA systems, the evolution of Transformer-based models, retrieval-augmented approaches for long-context understanding, and the role of OIE in enhancing explainability. Special emphasis is placed on the challenges of developing QA systems for Indic languages, including resource scarcity and the applicability of LLMs. Furthermore, the chapter discusses multilingual coreference resolution and retrieval-based context reduction as potential strategies to improve QA performance. By identifying key research gaps, this review lays the groundwork for developing an explainable and efficient QA framework for Indic languages.

Pashupati seal [75].

The task of automatically generating answers to human-posed queries has been an area of research since the early 1960s [23]. One of the earliest natural language processing programs, ELIZA [31], was designed to simulate human-like conversation. It employed pattern-matching techniques and scripted responses to mimic a psychotherapist. However, it lacked genuine comprehension or reasoning capabilities, as it could not process meaning beyond its predefined scripts. As illustrated in Figure 2.1, there has been substantial growth in research publications addressing QA over the years. Readers are referred to Appendix A.1 for details on reproducing the data presented in Figure 2.1. A considerable body of QA research has emerged from various modalities, including images [76] and videos [77]. However, this thesis focuses on QA systems operating within the text modality.

2.1 Foundational Works in QA

Woods [78] discusses advancements in natural language understanding through its application in lunar geology, specifically focusing on the development of a QA system for geologists. The system, termed LUNAR, allows users to retrieve and analyze data related to lunar samples using natural language queries. This research underscores fundamental challenges in semantic processing and knowledge representation, serving as a foundation for subsequent studies in computational linguistics and domain-specific QA systems.

The TREC QA Track was established in 1999 as part of the Text REtrieval Conference (TREC) to promote research in open-domain question answering [79]. Initially, the track concentrated on factoid questions, where systems were tasked with retrieving short, precise

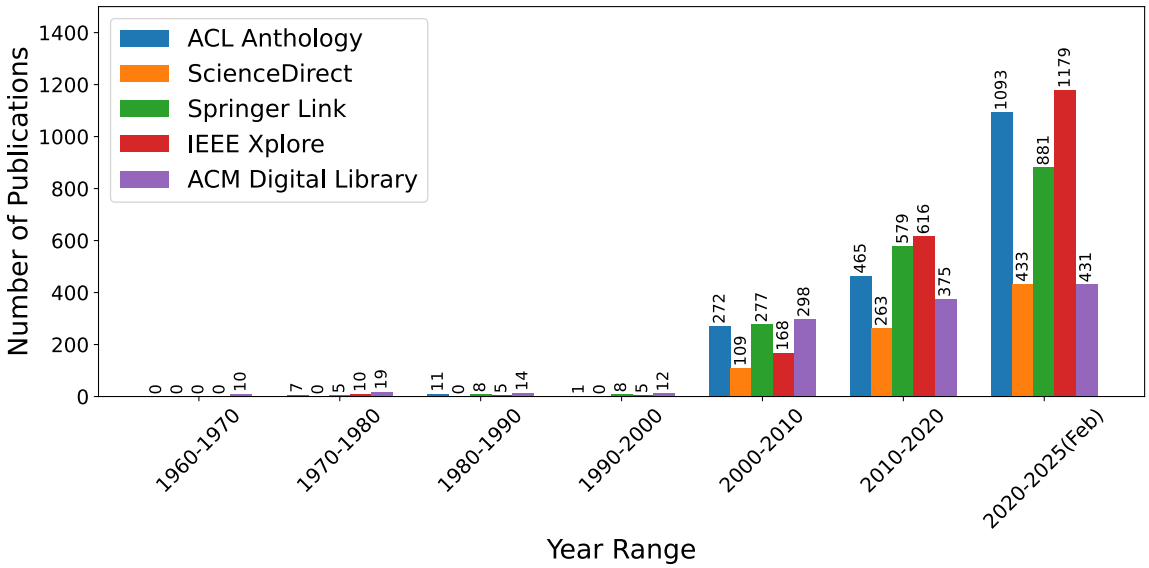


Figure 2.1: Decade-wise distribution of research papers containing the terms "*question*" and "*answering*" in their titles. The figure highlights a significant rise in publications dealing with QA methods and resources.

answers from extensive text corpora [80]. Over time, its scope expanded to include list questions, which require multiple correct answers, as well as definition questions, which demand more detailed explanatory responses. By the mid-2000s, the QA Track had further evolved to address complex and interactive question answering, moving beyond simple fact retrieval to incorporate reasoning and the synthesis of information from multiple sources [81]. Since then, the track has continued to advance, taking on increasingly sophisticated challenges that extend beyond factoid QA [82, 83].

Ravichandran and Hovy [84] proposed a method for learning surface text patterns from large corpora to enhance answer extraction in question-answering systems. The primary advantage of this approach was its ability to automatically derive patterns from labeled data, thereby reducing dependence on manually crafted rules. However, the method faced challenges related to data sparsity and generalization, as the extracted patterns were highly dependent on the training data and often failed when encountering unseen variations. Hermann et al. [85] introduced a large-scale cloze-style Question Answering Dataset (QuAD) derived from news articles, facilitating the training of deep neural models for machine reading comprehension. While this approach enabled the development of robust models, it relied heavily on synthetic question-answer pairs, which limited its capacity to generalize to more complex, real-world QA tasks that require reasoning beyond simple pattern matching. Table 1.1 provides an overview of widely used QuADs along with their respective QA features.

The Stanford Question Answering Dataset (SQuAD) [37] represented a significant milestone in QA research by introducing a large-scale, human-annotated dataset comprising passage-based factoid questions. Vaswani et al. [86] introduced the Transformer architecture, which revolutionized NLP by enabling efficient context modeling. Building on this, De-

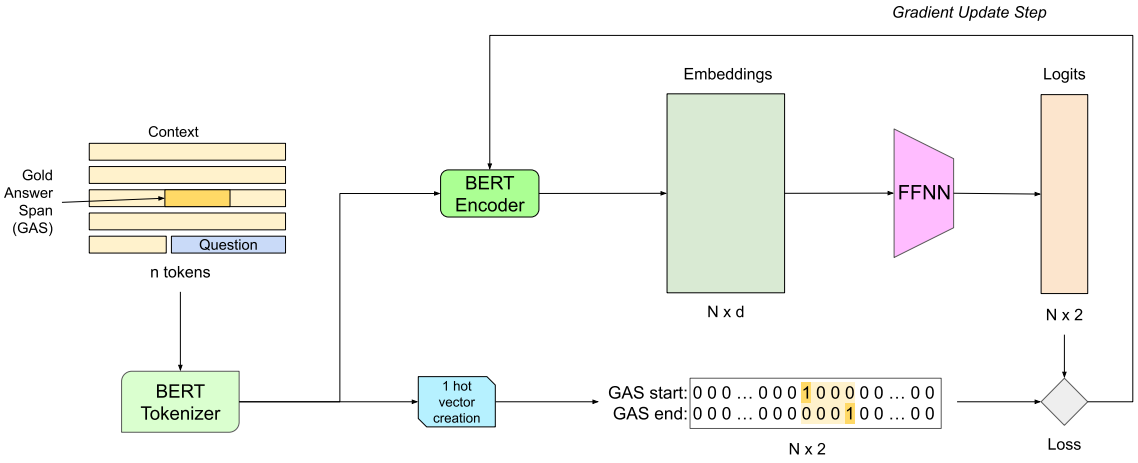


Figure 2.2: Pipeline for context-based span-extraction QA using the BERT model. In the figure, d denotes the hidden dimension of the BERT encoder (768), n represents the number of tokens in the given text, and N corresponds to the number of subtokens. The feed-forward neural network is denoted as FFNN.

vlin et al. [44] proposed BERT (Bidirectional Encoder Representations from Transformers), which achieved state-of-the-art performance on SQuAD, substantially improving answer span detection. Figure 2.2 illustrates the BERT-based QA pipeline for datasets similar to SQuAD. Subsequently, models such as RoBERTa [87], ALBERT [88], and SpanBERT [89] refined pretraining methodologies, enhancing robustness in extractive QA. Despite these advancements, these models exhibit limitations when handling non-factoid questions, which necessitate complex reasoning, multi-hop inference, and opinion-based responses—challenges that persist in contemporary QA systems.

2.2 Non-factoid QA

The existing literature indicates that non-factoid questions have been addressed through FAQ databases [61, 90, 91]. Agichtein et al. [92] proposed a query transformation technique followed by document ranking to resolve non-factoid queries. However, their approach does not provide a clear mechanism for extracting concise answers from the highest-ranked documents. Similarly, Jijkoun and de Rijke [61] and Surdeanu et al. [59] employed answer ranking as an initial step. Studies further highlight that traditional semantic retrieval techniques, such as BM25, perform inadequately when applied to non-factoid questions [93]. Merely retrieving relevant sentences or paragraphs from documents has been shown to be insufficient for effectively addressing such queries [94]. Moreover, for queries requiring implicit reasoning, the retrieval process itself needs to be disentangled [95]. In order to mitigate hallucinations, the integration of external knowledge into the QA pipeline has also been identified as essential [96]. Recent work further suggests that incorporating non-factoid question type classification during preprocessing enhances the performance of non-factoid QA systems [97]. LLMs have emerged as critical tools for generating comprehensive and contextually appropriate responses. Nonetheless, evaluating LLMs in the context of non-factoid QA presents unique challenges, as discussed by Fan et al. [98].

Singh et al. [99] introduced a QA benchmark for Indic languages by aggregating all publicly available QuAD datasets in Indic languages and supplementing them with synthetic data. However, the representation of non-factoid questions within multilingual QuAD datasets remains an area that requires further investigation.

2.3 QA in Indic Languages

Weston et al. [100] introduced a collection of synthetic tasks in Hindi aimed at evaluating machine learning models' reasoning and natural language understanding capabilities in a QA setting. MMQA was developed as a framework for multi-domain, multilingual question-answering in English and Hindi, encompassing six domains: tourism, history, geography, environment, diseases, and economics [101]. Clark et al. [102] introduced TyDi QA, a large-scale QA dataset consisting of 204,000 question-answer pairs across 11 typologically diverse

languages, including two Indic languages, Bengali and Telugu. Artetxe et al. [103] manually translated a subset of SQuAD into ten languages, resulting in the XQuAD dataset, which was designed to study the cross-lingual transferability of monolingual representations by training a masked language model in one language and adapting it to others. With the emergence of LLMs pretrained on extensive multilingual corpora, these models have become the state-of-the-art benchmarks for evaluating QA systems in Indic languages [104, 105, 106]. However, due to the increasing computational demands of LLMs with growing context sizes, there has been a renewed interest in hybrid approaches that integrate retrieval-based methods with answer generation in a unified pipeline. Chada and Natarajan [107] reframed QA as a sequence-to-sequence generation task using encoder-decoder architectures such as BART and T5, where the answer is masked in the input. This paradigm later evolved into what is known as instruction tuning, where multiple NLP tasks are formulated in a similar sequence-generation manner [108]. Notably, the authors observed that even in a few-shot setting—using as few as two examples—the model outperformed MuMLM-based QA systems fine-tuned on the MRQA dataset [109], which is a standardized aggregation of 18 English QA datasets.

Continuing the focus on cross-lingual QA, Asai et al. [110] introduced a shared task using XOR-TyDi and MLQA datasets, complemented by a newly created Tamil QA dataset comprising 350 expert-translated questions and machine-translated answers. Their results underscored the critical role of improving multilingual retrievers and highlighted that XOR-TyDi lacks boolean (Yes/No) questions. Vemula et al. [111] introduced TeQuAD, a Telugu QA dataset derived from translating SQuAD using Google Translate. They employed fuzzy matching and answer markers to locate answer spans in the translated text. The dataset comprises 82K QA pairs, supplemented by a manually annotated set of 947 pairs for evaluation. However, the authors noted quality issues in existing Telugu QA datasets such as Telugu-TyDi, particularly due to a high prevalence of numerical answers. In another line of work, Limkonchotiwat et al. [112] proposed a model based on multilingual Universal Sentence Encoder (mUSE) and contrastive loss, yet the approach failed to yield performance gains on Hindi, likely due to insufficient pretraining on Hindi texts. Ekram et al. [113] addressed resource gaps in Bangla by releasing a manually annotated dataset with over 14K QA pairs. Their findings highlighted that fine-tuning models on gold annotations leads to better performance on translated test sets, reinforcing the value of high-quality supervision. Kumar et al. [114] revealed that contrastive learning techniques are more beneficial for Dravidian languages than for Indo-Aryan languages. Muller et al. [115] developed GenTyDiQA by re-annotating TyDi to allow for abstractive answers, finding that monolingual QA models in Bengali outperform cross-lingual systems, mainly due to poor translation quality. Verma et al. [116] introduced a method to extract deep semantic roles, or "कारक (*Kaarak*)", as features to improve answer sentence retrieval in Indic languages. However, the underlying architecture of the *Kaarak* extractor is not thoroughly described.

In a similar effort, Arif et al. [117] created UQA by inserting answer markers into

SQuAD contexts and translating them into Urdu. Ruturaj et al. [118] addressed Marathi QA by aligning answer spans using semantic similarity, noting challenges such as case sensitivity, abbreviations, and special characters in translations. Progress in Urdu QA suggests narrowing the context to a few relevant sentences before answering enhances performance [119]. Similarly, Shafayat et al. [120] presented BenQA, a bilingual Bengali-English MCQ dataset. Chain-of-thought prompting significantly improved reasoning-based QA, especially when English translations were included in the prompt. In Tamil, Mallikarjuna and Sivanesan [121] demonstrated that predicting the answer type prior to extraction improves QA performance.

Shen et al. [122] created a product QA dataset in Hindi and Tamil with expert annotations, leveraging English product descriptions for candidate reranking and multilingual answer generation. The findings underscore the utility of fine-tuning directly on the target language rather than relying on English models and machine translation post-processing. Multiple studies emphasize that manual annotations are costly [123, 115, 124], and that machine translation is not a reliable strategy for building QA datasets in under-represented languages [125]. Bandarkar et al. [126] developed a highly multilingual dataset of 900 multiple-choice questions in 122 languages, all human-translated. Their findings highlight that even compact MuMLMs pretrained on balanced corpora outperform large LLMs in multilingual understanding. Native script usage consistently led to better performance, while models like LLaMA showed weak results for Dravidian languages when under 20 billion parameters. Ralethe and Buys [127] demonstrated that aligning Knowledge Base triples in English with their low-resource language equivalents improves QA performance over zero-shot baselines. Meanwhile, proprietary LLMs have shown superior results in Urdu [128] and Hindi [129], though issues such as boundary detection and translation mismatches persist. Notably, non-factoid questions remain more challenging, and token-level metrics tend to undervalue LLM outputs, reinforcing the need for semantic or human-centered evaluations [129, 130].

2.4 Retriever-aided QA

Li et al. [131] demonstrated that contemporary language models, such as Mamba, which do not rely on the Transformer architecture, exhibit inferior performance compared to Transformer-based LLMs in tasks requiring long-context understanding. Similarly, Wang et al. [132] explored open-domain non-factoid question answering, commonly referred to as long-form question answering, and found that retrieval-based methods consistently outperform direct generative approaches in this setting. The effectiveness of retrieval techniques in open-domain QA was further emphasized by Karpukhin et al. [133]. Additionally, Huo et al. [134] argued that integrating text retrieval can help mitigate hallucinations in LLMs when addressing long-context questions in open-domain scenarios. Tang et al. [135] proposed dividing the large external context associated with a query into smaller documents, followed by hierarchical summarization, and subsequently performing retrieval over a doc-

ument graph with human preference alignment through a mode-seeking loss.

2.5 QA with Open Information Extraction

The use of triples and knowledge bases to enhance question-answering pipelines has been widely explored in the literature [136, 137]. K-BERT [138] incorporated knowledge extracted from Open Information Extraction (OpenIE) triples to improve performance across various NLP tasks, including question answering. Similarly, Fader et al. [139] leveraged both synthetic and gold-annotated triples from diverse knowledge bases to answer open-domain user queries. Studies such as Baek et al. [140] and Sen et al. [141] have demonstrated that retrieving knowledge base triples, verbalizing them, and incorporating them as contextual information significantly enhances LLM performance. Furthermore, Khot et al. [142] utilized OpenIE triples in conjunction with the tabular reasoning framework proposed by Khashabi et al. [143] to determine the correct answers in multiple-choice question-answering tasks.

2.6 Coreference Resolution and QA

Zhang et al. [144] and Bai et al. [145] highlighted the limitations of contemporary LLMs in handling English and Chinese QA tasks when the context length approaches that of a novel. However, Liu et al. [146] demonstrated that integrating coreference resolution as a preprocessing step before query-specific short-context retrieval significantly enhances LLM performance, even for QA tasks involving extensive contexts. Additionally, Chai et al. [147] proposed a pipeline that leverages OIE triples and coreference resolution to improve QA performance. Their approach further incorporated post-hoc XAI techniques, such as LIME [148] and SHAP [149], to interpret the generated answers. However, this pipeline was not evaluated on a large-scale dataset.

2.7 Explainability in QA

Marcinkevičs and Vogt [150] observed that while the terms “interpretable” and “explainable” are often used interchangeably, a distinction can be made: interpretability pertains to white-box or, at minimum, grey-box models, whereas explainability focuses on generating explanations for outputs produced by black-box models.

Answers generated using structured knowledge bases inherently provide a degree of explainability within the QA pipeline [151, 152]. Yang et al. [54] introduced an English QA dataset containing manually annotated explanations or supporting facts for each answer. Similarly, FeedbackQA [153] and CoS-E [154] include manually annotated explanations for each question-answer pair. However, the development of such resources is constrained by the substantial financial costs associated with their creation. Fan et al. [155] introduced a method to enhance explainable question-answering systems by constructing structured

entailment trees. These trees illustrate the reasoning process from known facts to the derived answers, thereby providing explainability. Furthermore, fact retrieval from a knowledge tree has been demonstrated to achieve state-of-the-art performance in providing reasoning ability within LLM-based QA pipelines [156].

Several studies have demonstrated that the attention mechanism alone is insufficient to provide explanations for the output of QA models, whether based on recurrent neural networks (RNNs) [157, 158] or pretrained transformers [159]. Some approaches have sought to enhance the interpretability of QA models by instructing them to generate both an answer and the reasoning behind it [160]. Additionally, triple extraction has been employed as a method to improve the interpretability of QA systems [161]. Thayaparan et al. [162] incorporated coreference resolution as a preprocessing step before constructing a document graph, which facilitated the explanation of the generated answers. Existing literature suggests that explainability in QA models has primarily been explored in the context of questions requiring multi-hop reasoning.

A widely adopted approach to develop Explainable QA (XQA) models involves generating both an answer and its corresponding evidence from the provided context [163, 164]. Jiang et al. [165] employed a tree search paradigm for this task, while Tu et al. [166] implemented sentence-level prediction to identify the specific sentences that justify the generated answer. Schuff et al. [130] underscored the difficulty of evaluating the quality of explanations produced by QA models that generate answers alongside their justifications.

2.8 Research Gaps

We adopted a systematic methodology to conduct a comprehensive literature survey on key research directions within the broader domain of QA. Figure 2.3 presents a visual representation of the studies reviewed in this chapter. Based on the insights gained from this survey, the following research gaps have been identified:

1. Non-factoid questions are more frequently encountered than factoid-based questions. However, the development of resources for non-factoid QA systems remains limited.

2. Research on Indic QA systems has been largely constrained by the scarcity of language-specific resources. FAQ-based systems present a viable starting point for addressing this limitation.

3. The effectiveness of LLMs pretrained on extensive multilingual corpora in developing Indic QA systems requires further investigation. Additionally, techniques such as quantization and low-rank adaptation are promising approaches for enabling LLMs in low-compute environments.

4. QA systems leveraging OIE triples offer inherent explainability to the QA pipeline, warranting further exploration. Furthermore, the development and evaluation of OIE tools tailored for Indic languages are equally critical research areas.

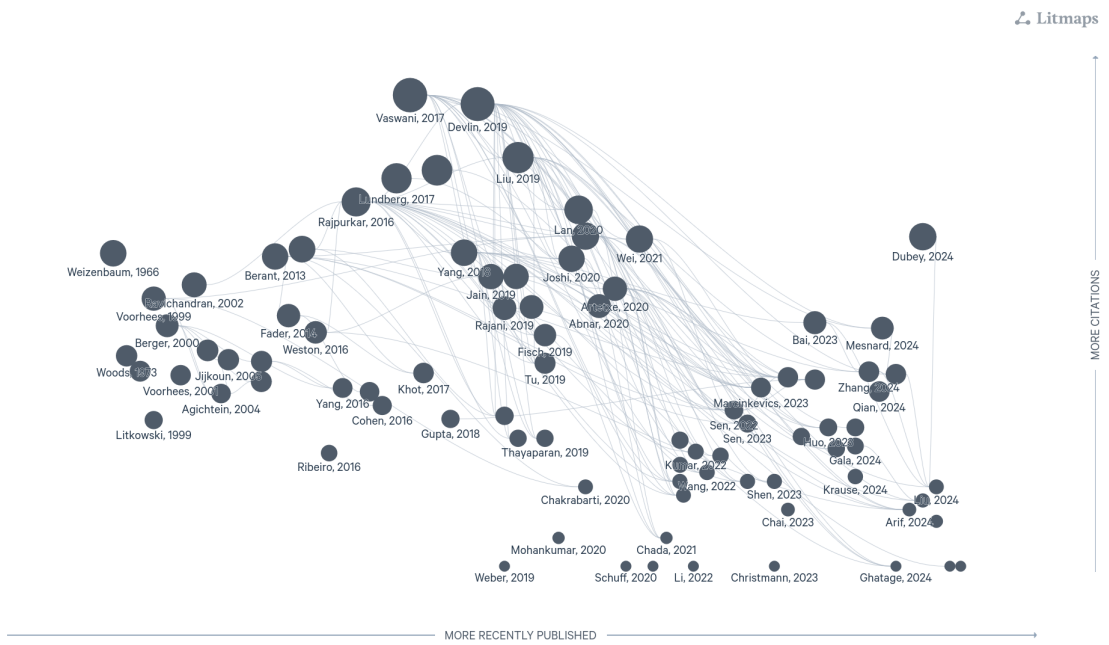


Figure 2.3: A citation network visualization of academic papers, illustrating the evolution of research in a specific domain. Each node represents a paper, with its position determined by publication year (X-axis) and citation count (Y-axis). Larger nodes indicate highly cited works, while edges represent citation relationships between papers. This visualization highlights influential studies and the progression of research over time. An interactive version of the figure is available at `https://app.litmaps.com/shared/a7f673ad-c444-444d-8786-0e0c48bfab7b`.

5. The integration of multilingual coreference resolution as a preprocessing step could facilitate the generation of triple chains for QA pipelines. This approach has not been extensively explored in modern QA systems.

6. Retrieval-based context reduction techniques for long-context QA systems represent a promising direction for Indic QA research.

7. A comparative analysis of various post-hoc explainable AI techniques would be valuable for identifying the most effective method for interpreting QA pipelines that employ retrieval-based context shortening.

2.9 Chapter Summary

This chapter presents a comprehensive literature survey on various aspects of QA research, highlighting key advancements and persisting challenges. It covers factoid and non-factoid QA, the evolution of Transformer-based models, and the role of retrieval mechanisms in improving long-context understanding. The survey also examines QA systems in Indic languages, noting the scarcity of resources and the potential of FAQ-based approaches. The

impact of LLMs pretrained on multilingual corpora is discussed, emphasizing the need for quantization and low-rank adaptation for low-compute environments. OIE-based QA is identified as an area with inherent explainability, necessitating further exploration, particularly in Indic languages. The integration of multilingual coreference resolution and retrieval-based context reduction is recognized as a promising direction for improving answer generation in long-context settings. Additionally, the chapter underscores the need for evaluating post-hoc explainable AI methods to enhance the interpretability of QA pipelines. The identified research gaps emphasize the need for improved non-factoid QA resources, efficient Indic language processing techniques, and advancements in retrieval-augmented and explainable QA models.

Chapter 3

IndIE: A Multilingual Open Information Extraction Tool For Indic Languages

It has been observed that representing knowledge from low-resource languages in a structured format could enable the development of more advanced knowledge-based QA systems. However, the lack of publicly available tools for triple extraction from unstructured text posed a significant challenge, motivating this work. This chapter explores advancements in multilingual OIE and presents our contributions in this domain. We introduce and release an OIE benchmark dataset for Hindi, fine-tune a transformer-based chunker model for Indic languages, and propose IndIE, a greedy algorithm for extracting structured triples from unstructured text. These efforts aim to enhance the accessibility of structured knowledge, facilitating the development of robust QA systems for Indic languages[1].

Seven sages seal [168].

[1]Content of this chapter is adapted from our paper published in the Findings of AACL-IJCNLP, 2023 [167]

India exhibits significant linguistic diversity, with six of its native languages ranked among the world's top 20 most spoken languages [169, 170]. Although spoken by many, numerous Indic languages are considered low-resource because of the scarce availability of annotated datasets and the lack of robust automated tools [171]. Consequently, there is a notable scarcity of tools made to extract structured information in Indian languages, largely attributed to the lack of dedicated works in this domain [172, 173].

The concept of Information Extraction (IE), introduced in the mid-1960s, derives structured information from raw text in unstructured natural language [174]. When such extraction is performed independently of a predefined ontology or schema, it is referred to as Open Information Extraction (OIE). A widely adopted representation for extracted facts is the triple format <*head, relation, tail*>. A *relation* signifies a connection between the *head* and *tail*. For instance, given the sentence "*PM Modi to visit UAE in Jan marking 50 years of diplomatic ties*," a valid triple representation is <*PM Modi*, *to visit*, *UAE*>.

The main strength of OIE tools is their capability to identify and extract triples from extensive text corpora without relying on supervised learning [175]. Additionally, OIE is utilized as an initial step in the construction or enrichment of knowledge graphs derived from unstructured text [176, 177]. While Relation Extraction has been identified as the most prevalent application of OIE, advancements in OIE have also contributed to fields such as Ontology Construction and Fact-Checking [178]. Furthermore, OIE tools have been employed in various downstream applications, including Question-Answering [179], Text Summarization [180], and Entity Linking [181].

Triples can be extracted through various approaches, influenced by the syntactic structure of the target language and the desired granularity of the extracted information. For example, from the sentence *John sliced an orange with a knife*, two plausible extractions include: (i) <*John, sliced, an orange with a knife*> and (ii) <*John, sliced, an orange*>, as well as <*John, sliced, with a knife*>. Although both forms represent the same underlying event, they differ in the specificity and scope of the captured details.

In languages with free word order, such as Hindi [182], a single fact can be expressed through multiple rearrangements of the three components within a triple. For example, consider the triples <राम नक, खाया, एक सक्ब> [<*rAm ne*, *khAya*, *ek seb*>][2] and <एक सक्ब, खाया, राम नक> [<*ek seb*, *khAya*, *rAm ne*>] conveys the same information as a triple in English: <*Ram*, *ate*, *an apple*>. However, since Hindi employs postpositions (*kaarak*) instead of prepositions [183], word permutations that separate the postposition from its associated subject word are not permissible, as they alter the semantic information of the triple. For instance, if the triple is <एक सक्ब, नक खाया, राम> [<*ek seb*, *ne khAya*, *rAm*>] then it inaccurately depicts the information of <*An apple*, *ate*, *Ram*>.

This chapter primarily focuses on the automatic extraction of triples from Hindi sen-

[2]Text in italics enclosed within square brackets denotes the ITRANS transliteration, while text in italics within parentheses provides the English translation of the corresponding Hindi expression or sentence. We follow this convention in this chapter.

tences, since the primary investigators of this study were proficient in Hindi. However, the developed tool is also capable of extracting triples when it is fed a text from other low-resource Indian languages, including Urdu, Telugu, and Tamil. The key contributions of this work are listed as follows:

1. We propose *Hindi-BenchIE* to automatically evaluate triples extracted from Hindi sentences. To the best of our knowledge, this is a novel benchmark designed to accommodate the free-word-order characteristic of Hindi language and diverse triple extractions produced by various OIE systems.

2. A model based on pretrained transformer that is fine-tuned on manually labeled semantic chunks in six Indic languages (*Hindi*, *English*, *Urdu*, *Nepali*, *Gujarati*, and *Bengali*). The resulting model demonstrates the ability to perform chunking on languages unseen during the fine-tuning phase.

3. It has been noted that during the fine-tuning of a pretrained encoder for sequence labeling tasks like chunking, either averaging the subword embeddings or utilizing the final subword embedding tends to yield better performance than the traditional method of relying solely on the first subword embedding.

4. We introduce an algorithm based on hand-crafted rules for extracting triples from raw text in Hindi language. We release the resources and code-base will be made publicly available at https://github.com/ritwikmishra/IndIE.

Research in this direction is motivated primarily by us to commence the creation of an OIE tool for Indic languages, which would result in the generation of additional resources for resource-deficient Indic languages. Additionally, a novel path for the progression of research in the domain of Indic-NLP is strongly believed by us to be provided by applications of OIE for other subsequent operations (such as ontology building, entity connection, truth verification, and so on).

3.1 Related Work

A combination of manually crafted rules and shallow parsing has been utilized by previous studies to extract significant entities from English-language text [184, 185, 186, 187]. Mausam et al. [188] developed OLLIE using shallow parsers and hand-crafted rules to identify relations involving phrases such "*is the president of*" and extracted triples from English sentences. OLLIE achieved results at par with triple extractors that are based on SRL mechanism. While facts in the form of triples were dealt with by most of the works, Akbik and Löser [189] developed KrakeN to extract N-ary tuples as facts using dependency parsing. The property of extracting, whereby triples could only contain words verbatim present in the sentence, was a drawback of earlier rule-based OIE methodologies. Hence, appositive

relationships[3] were not extracted by such tools [190]. In section 3.2.3, the method used by us to extract such appositive relationships is discussed.

ClausIE, extracted OIE tuples by identifying the semantic clauses using rules from English language, was developed by Del Corro and Gemulla [191]. Numerous works have employed dependency parsing of the given raw text to extract facts [192, 193, 175, 194] in order to identify the relations or entities. Generated as an improvement to ClausIE, much more fine-grained and concise facts were generated by the MinIE [195] tool as compared to ClausIE. Key-value pairs having information about certainty, knowledge source, and polarity were contained in the triples generated by MinIE. Various OIE works based on neural architectures in English have been made possible due to the availability of gold data. The process of triple extraction involves the following two steps: (a) Relation capturing and (b) Argument (*head/tail*) extraction with the help of features from the step (a) [196]. The task of sequence labeling aided the span selection to extract OIE relations and their corresponding arguments[190].

Building OIE frameworks for non-English languages is hindered by the demand for annotated resources that are publicly available. However, the domain of language-agnostic (multilingual) OIE commenced in 2015 primarily by two works. The first technique was developed by Manaal and Kumar (*M&K*) [197], in which the authors first translated the source language into English via Google Translate, followed by the extraction of triples using the OLLIE tool.The English triples were mapped back to their original language using word alignments. It was capable of handling as many languages as Google can translate, but machine translation has not been considered a viable solution for OIE due to errors in translation [178]. The second technique triple extracted based on hand-crafted rules known as *ArgOE* [194]. To generate triples, it required a dependency parse tree of the given raw text. However, the extracted triples encompass merely verb-mediated relations. White et al. [192] introduced *PredPatt* which also worked on a dependency parsing and manually-created rules that captured predicate-argument structure present in the raw text. Another technique called *Multi2OIE* modeled the issue of capturing predicate-argument construction by two sequence-labeling tasks utilizing multilingual BERT embeddings with multi-head self-attention units [196]. First the predicates in the given raw text are identified, and then arguments for each predicate are identified. A limitation of this method is its shortcoming in capturing predicates that are overlapping in nature. For example, consider the following sentence "*Nehru became the prime minister of India in 1947*". Depending on the degree of granularity, examples of a few predicates which might be extracted from various acceptable options are "*became*" and "*became the prime minister*".

A novelty in the domain of multilingual OIE, which employs Natural Language Generation (NLG) paradigms with cross-lingual projections, was introduced by Kolluru et al. [70]. Overlapping relations (predicates) and their corresponding arguments were effectively cap-

[3]It is a syntactic structure in which two noun phrases are placed next to one another to provide supplementary information. For instance: *My brother, Bob, likes ice cream*

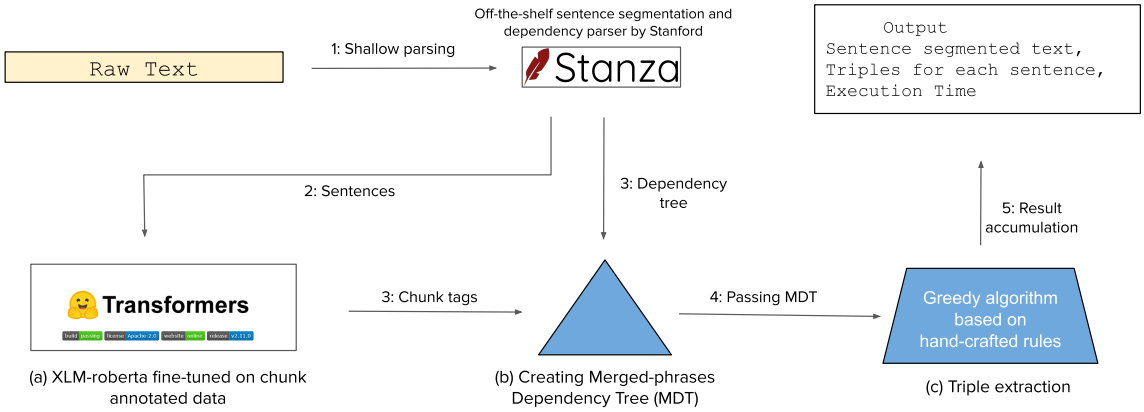


Figure 3.1: The *IndIE* tool's general framework. (a) Chunk label forecasting, (b) Construction of the Consolidated-phrases Dependency Graph (CDG), and (c) Triplet production are the three main phases. By the Stanza library [2], each separated sentence is processed by these three phases.

tured by the proposed method. However, the proposed AACTrans algorithm needed parallel corpora for fine-tuning, and machine translation tool was utilized in their experiments. We evaluate the performance of IndIE against the five methods mentioned which are highlighted previously (*M&K, ArgOE, PredPatt, Gen2OIE, and Multi2OIE*). They have been taken in the form of baselines because the underlying use case aligns with this work.

A major deficiency in earlier multilingual OIE works was their ability to extract text only in verbatim manner, i.e., triples could contain words that are explicitly mentioned in the given raw text. Thus, appositive relationships were missed by the tools inconsideration [190]. Our method is able to capture such appositive relationships by simply identifying the pair of phrases that are connected by dependency relations, such as *appositional modifier* (*appos*) and *nominal modifier* (*nmod*), in the Merged-phrases Dependency Tree (explained in section 3.2.2).

3.2 Methodology

Raw text is accepted as input by our method, and sentence segmentation and dependency parsing are executed utilizing the Stanza library [2] (version 1.1.1). The principal reason for the employment of the Stanza library was its capacity to conduct shallow parsing across various Indic languages. The overall procedure of generating triples is depicted by Figure 3.1. It is separated into the following three main phases: (a) the execution of chunking and the identification of the semantic phrases within the provided sentence, (b) the creation of a Merged-phrases Dependency Tree with the help of dependency parse trees, and (c) the generation of triples via our custom-designed rules. The three phases are discussed in more detail in the subsequent subsections.

3.2.1 Chunking

Semantic chunking (or simply chunking) can be explained as the capture of non-overlapping multi-token text spans within a sentence with their classification in accordance to various syntactic phrases[198]. Sequence modeling is the underlying task which performs chunking whereby a single chunk tag is produced for each token from given text. All chunk tags comprises (i) a boundary label, and (ii) a chunk label. The chunk labels are commonly categorized into diverse syntactic classes, such as Verb-Phrases (VP), Noun-Phrases (NP), Adjective-Phrases (JJP), etc. [199]. Conversely, different notations, like BIOES or BIO, can be utilized to signify the non-overlapping boundary labels. The BI notation is used by us to mark boundary labels, given that prior studies have demonstrated its superior precision compared to alternative notations [200, 201].

Before delving into the experimental results, it is essential to provide a detailed description of the dataset and the model architecture employed in this study. The following subsections outline the characteristics of the dataset used for the chunking task, including its source, annotation scheme, and relevant statistics. Subsequently, we present the model configuration, highlighting the key components and methodologies adopted for sequence labeling. This comprehensive overview ensures clarity and reproducibility of our experimental setup.

Dataset

A multilingual chunking tool is developed by us through pre-trained transformer encoder models fine-tuned on chunk annotated texts in multiple Indic languages. The developed chunker is fine-tuned on gold labels from Bhat et al. [202][4] and Jha [203][5]. The later data source encompasses 70K chunk-labeled sentences in each of Bengali, English, Hindi, Nepali, and Gujarati, while the former data source contains 5K and 16K chunk-labeled sentences in Urdu and Hindi, respectively. Our motivation for utilizing two different data sources is to possess the largest quantity of labeled data for fine-tuning. A total of 0.37 million chunk annotated sentences were yielded by the two data sources.

Model

By means of the transformers library [204], diverse pretrained transformer-based models are fine-tuned by us for the chunking task, given that many prior studies have demonstrated their effectiveness to achieve state-of-the-art results on shallow parsing tasks [205, 206, 207]. All subword (or subtoken) embeddings of a word (or token) are averaged in unweighted manner to acquire the word embeddings. To evaluate the performance of a transformer-based chunker, a Conditional Random Field (CRF) model is trained by us utilizing the scikit-learn [208] library. Furthermore, Viterbi decoding was used with a second-order Hidden Markov

[4] from https://universaldependencies.org/
[5] from http://tdil-dc.in

Model (HMM) to predict the chunk tags. Both models are considered standard techniques employed for perform automated chunking in Indic languages [209]. The implementation details for the baseline models are contained within Appendix A.3.2.

A provided text is parsed by the Stanza library [2], by which sentence segmentation, dependency parsing, and POS tagging are performed. Our chunker takes the segmented sentences, by which the chunk tags are predicted for each token. All non-overlapping phrases (or multi-word expressions) are identified by the predicted chunk tags. By concatenating each attribute of its constituent tokens, a syntactically rich phrase was constructed. In order of its occurrence in the sentence, each phrase is stored in a list. Next stage takes the list of phrases, by which the Merged-phrases Dependency Tree is created.

3.2.2 Merged-phrases Dependency Tree (MDT)

In OIE, shallow parsing have been utilized extensively to assist in the generation of triples from raw text [175, 191, 192, 193, 194]. Traditional shallow parsing tools generates a dependency tree at a token level, that is, leaf nodes of the tree are the tokens from the given text, and the dependency relation between the two tokens is represented by the edges connecting them. A Merged-phrases Dependency Tree (*MDT*) would represent a coarse tree where each node is a phrase or a multi-word expression from the given text. The difference is illustrated very well by an online tool by explosion.ai[6]. As shown in Dobrovolskii [12] one head is identified from each phrase, and the dependency relation shared between two corresponding phrases is utilized as the dependency relation between the two heads of the phrases. We used the token-level dependency tree as a guiding tool to recognize the dependency relationships between the phrases identified in the chunking stage. A comparison of traditional dependency tree with a generated *MDT* is shown in Figure 3.2 using a Hindi sentence as an example. Constituency tree differs from *MDT* as the syntactic relationships between the head and the remaining tokens present in each phrase are not preserved. To the best of our knowledge, a *MDT* parsing tool and a constituency parsing tool is unavailable publicly for sentences in Indic languages. Hence, we developed a rule-based algorithm to generate *MDT* using a traditional dependency tree.

The occurrence of complex predicates (CPs) are prevalent in a South Asian language like Hindi, wherein a solitary action or verb is depicted through a verb-verb combination (termed *compound verbs*) or a combination of noun-verb (termed *conjunct verbs*) [210, 211]. More utility is proven by an *MDT* in the representation of a sentence in which the parsing of CPs in languages like Hindi is failed by the conventional dependency tree. For instance, the sentence प्रारंभिक खगोलविदों का मानना था कि पृथ्वी ब्रह्मांड के केंद्र में है [*prarambhik khagolvido ka mAn-na tha ki prithvi brahm-And ke kendr me hae*] (*Early astronomers believed that Earth is in the center of the universe*) is considered, wherein the event of *believed* is depicted by a *compound verb* मानना था [*mAn-na tha*] in Hindi. When we analyze the token-level dependency tree of the same sentence, the following parent → child structure is produced:

[6] `https://explosion.ai/demos/displacy/`

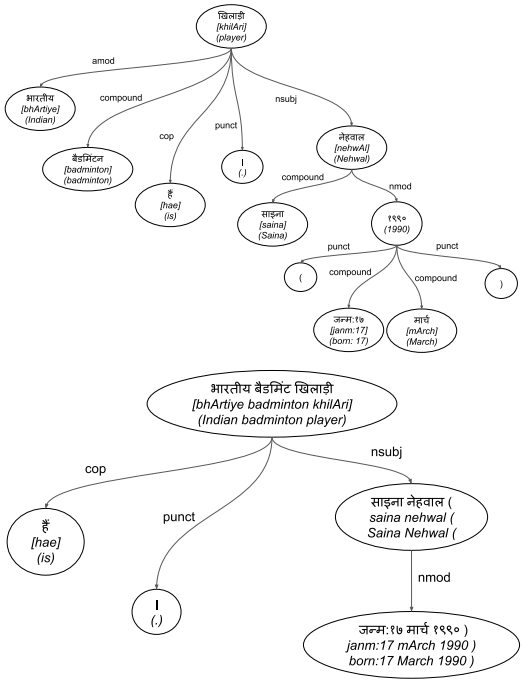


Figure 3.2: For the supplied Hindi statement, a parallel is drawn between a Merged-phrases Dependency Tree (right) and a standard dependency graph (left): साइना नक्हवाल (जन्म:१७ मार्च १९९०) भारतीय बैडमिंटन खिलाड़ी हैं । [*saina nehwAl (janm 17 mArch 1990) bhArtiye badminton khilAri hae*] which is translated as *Saina Nehwal (birth:17 March 1990)* *is* *(an) Indian badminton player* . where Noun Phrase (NP), Verb Phrase (VP), and Miscellaneous (BLK) are the predicted chunk tags.

था [*tha*] (past-tense inflection) $\xrightarrow{\text{nsubj}}$ **मानना** [*mAn-na*] (*to believe*), whereas it is an erroneous association; a correct construction would appear as **मानना** [*mAn-na*] (*to believe*) $\xrightarrow{\text{aux}}$ **था** [*tha*] (past-tense-inflection). The identification of *compound verbs* in the form of a singular Verb Phrase is done by the chunker module, thereby a semantically coherrent *MDT* is generated. We have also observed that in the absence of a Multi-word Entity Recognition tool, any identification of triple arguments is made difficult by the sole use of dependency parsing [175]. The degradation of performance in parsing a dependency tree for natural languages possessing free word-order has also been shown by Gulordava and Merlo [212].

3.2.3 Triple generation

From the *MDT* of a sentence, the *head*, *relation*, and *tail* are captured through the use of hand-crafted rules. Similar to Mesquita et al. [213], we studied all the recorded dependency relations in Hindi[7] to build the hand-crafted rules. We carefully analyzed 80 examples of Hindi sentences, covering 26 out of 27 possible dependency relations in Hindi, to construct

[7] `https://universaldependencies.org/treebanks/hi_hdtb/index.html`

the rules. One dependency relation, *vocative*, is not covered in our chosen Hindi sentences. Among 16K dependency-annotated sentences it is observed only in 6 sentences in the data by Bhat et al. [202]. It was noted that even the Stanza dependency parsing tool is unable to predict *vocative* dependency relation in Hindi sentences. Among the dependency annotated data[8] of other Indic languages such as Telugu, Tamil, and Urdu, the percentage of nodes connected to their parents with a Hindi dependency relation are 98%, 96%, and nearly 100%, respectively. Therefore, authors firmly believe that wide coverage is possessed by triple extraction rules based on Hindi dependency relations, and their applicability could be found in other Indic languages. There were over 100 decision-making statements (if-else blocks) in the resultant hand-crafted rules. Therefore, for the sake of brevity, all the triple extraction rules will not be explained here. An abstracted algorithm illustrating the triple extraction procedure is contained in Appendix A.3.5.

Capturing *appositive* relationships between two text spans is a novel property of the proposed hand-crafted rules. Such *appositive* relationships were unable to be captured by earlier multilingual methods. We take the following sentence as an example, शरमीला टैगोर के बेटे सैफ अली खान को मिला पद्म श्री पुरस्कार [*sharmila taegore ke bete saef ali khAn ko mila padm shri puraskAr*] (*Son of Sharmila Tagore, Saif Ali Khan, was awarded Padma Shri*). Between the text spans of *Saif Ali Khan* and *Son of Sharmila Tagore*, an appositive relationship exists (*is-a*). Appositive relationships of these kinds are expressed by *appositional modifier* (*appos*) and *nominal modifier* (*nmod*) dependency relation in the *MDT*. We captured such relations in our rules. We select the child of such relations as <*tail*>, and the parent as <*head*> of the triple. It was observed that researchers used the auxiliary verb '*be*' from English to denote the <*relation*> for appositive relationships [213].

In our work, we used है [*hae*] (*is/be*), an auxiliary verb in Hindi, to represent the <*relation*> inside a triple containing an *appositive* relationship for sentences in Hindi. The flow of information in the developed architecture has been highlighted in Appendix A.3.1 with the help of a couple of sentences in Hindi.

3.3 Triple Evaluation

By employing native speakers of a language and asking them to annotate the extracted triples the quality of generated triples is generally evaluated. However, this process is time and cost intensive. Moreover, in the manual evaluation process, a hurdle is created by the lack of availability of Indic language annotators. On the other hand, the various ways to extract different and yet meaningful triples is not considered by automatic evaluation methods based on gold annotations (like CaRB [214]). Therefore, by extending a work titled *BenchIE* [215], we develop an automatic triple evaluation method, *Hindi-BenchIE*, based on numerous manually annotated triples in order to mechanically evaluate the quality of Hindi triples extracted by different OIE methodologies.

[8]from https://universaldependencies.org/

sent_id:1 He served as the first Prime Minister of Australia and became a founding justice of the High Court of Australia .
1--> Cluster 1:
He --> served as --> [the] [first] Prime Minister [of Australia]
He --> served --> as [the] [first] Prime Minister [of Australia]
1--> Cluster 2:
He --> served as [the] [first] Prime Minister of --> Australia
He --> served as [the] [first] Prime Minister --> of Australia
1--> Cluster 3:
He --> became --> [a] [founding] justice
He --> became --> [a] [founding] justice of [the] High Court [of Australia]
1--> Cluster 4:
He --> became [a] [founding] justice of --> [the] High Court [of Australia]
He --> became [a] [founding] justice --> of [the] High Court [of Australia]
1--> Cluster 5:
He --> became [a] [founding] justice of [the] High Court of --> Australia
He --> became [a] [founding] justice of [the] High Court --> of Australia

Figure 3.3: For the following English declaration, an instance of *BenchIE* gold labeling: *He was employed as Australia's initial Prime Minister and was established as a founding arbiter of Australia's Supreme Tribunal*. The terms enclosed in square brackets are regarded as optional.

Hindi-BenchIE

In any natural language text, one or more facts are commonly present. In the initial study of *BenchIE* [215], various triples were curated by hand (named *golden triples*) by which a solitary fact present in the sentence was represented. For a provided sentence, a cluster is deemed satisfied if by the given OIE tool, at least one of its triples is produced. As *True Positives* (*TP*), the count of produced triples present in any cluster of that sentence is known. Similarly, as *False Positives* (*FP*), the count of produced triples absent from any cluster of that sentence is known. As *False Negatives* (*FN*), the count of clusters left unsatisfied is known. For instance, by gold annotations from Figure 3.3, if by an OIE tool the following triples are produced (i) <*He*, *served*, *as the Prime Minister of Australia*>, and (ii) <*He*, *became*, *founding justice of the High Court of Australia*>, its recall, precision, and F-score become 0.4, 1.0, and 0.57 respectively. The reason behind such low scores is the fact that by the existing work of *BenchIE*, no interdependence between triples is assumed. In *Hindi-BenchIE*, a benchmark we proposed, the *BenchIE* notations are extended, by proposing the following two subcategories of *golden triples*: (a) *compensatory-triples* and (b) *essential-triples*. A triple that encapsulates all the semantic information required to represent a fact is an *essential-triple*. In an *essential-triple*, some phrases might exist without which the remainder of the triple remains meaningful. Such phrases are designated by us as *vulnerable-*

Model	Hindi	English	Urdu	Nepali	Gujarati	Bengali
XLM	**78%**	**60%**	**84%**	**65%**	**56%**	**66%**
CRF	67%	56%	71%	58%	53%	53%

Table 3.1: A comparison of (fine-tuned) CRF chunker and XLM chunker is presented with one language omitted from the training-set. Represented by the numbers is the accuracy calculated for each model when sentences from the selected language are utilized solely in the test-set. Our observation highlights that XLM chunker consistently performs better on languages that were unseen during the fine-tuning stage.

phrases in our proposed work. However, any best possible OIE benchmark should ensure that in the automatically produced triples, no information is lost. Moreover, if any information is lost, then for it, any OIE technique should get a lower score. Therefore, the information not present in the loss of a *vulnerable-phrase* inside the produced triple is contained by a *compensatory-triple*. Moreover, by *Hindi-BenchIE*, the interchangeability of *tail* and *head* inside a triple is supported because free-word order property is common in Hindi language. These modifications facilitate manual labeling process, as by hand extracting multiple triples from a text in a free word-order language, significant human effort would otherwise be required.

In order to distinguish relationships like apposition, an explicit keyword named '*property*' is used by us as a *relation*. In this study, 112 Hindi sentences were used to generate *golden-triples* in different clusters. A single annotator extracted them manually. We release the Hindi sentences and their corresponding *golden triples*[9], since in the field of multilingual OIE [178], such resources are scarce.

The count of False Positives and True Positives is computed across all the *golden-triples* from the corresponding sentence (much like the *BenchIE*). Our research considers computation of False Negatives as the count of absent *essential-triples*, and the count of absent *compensatory-triples* that corresponds to a missing *vulnerable-phrase* (if any).

3.4 Results

For the purpose of comparing traditional methodologies for chunking with our fine-tuned chunker, the chunk annotated data was partitioned into a test-set and a training-set in a 50:50 ratio. It was noted that the average accuracy of our neural chunker fine-tuned with xlm-roberta-base [216] encoder was superior (91%) in comparison with other baselines of CRF (84%) and HMM (12%). The diminished performance of HMM is attributable to the sparsity in the *emission matrix* due to Out Of Vocabulary (OOV) terms. On numerous random splits, it was noted that the test-set word bigrams were lacking in the HMM training set in more than 80% of instances. Because of the substandard performance exhibited

[9]https://github.com/ritwikmishra/hindi-benchie

by HMM, a decision was made to utilize only the CRF for subsequent comparisons. To assess the multilingual prowess of our fine-tuned chunker, language-specific test-sets were curated, and they were omitted from the training-set. The given methodology aligns with the paradigms of the Leave One Language Out (LOLO) strategy which is a documented technique in various prior research works [217, 218]. Compared to CRF, enhanced accuracy was given by the fine-tuned chunker even on the languages that had never been seen by it while it was fine-tuning on chunk annotated data from other languages. Table 3.1 highlights the comparison between CRF chunker and our fine-tuned chunker.

It was also observed that optimal chunking accuracy was given by an unweighted mean of subword embeddings and a single linear layer . It is important to note, however, that a temporal overhead is introduced into the chunking process by the employment of subword embedding averaging.

sent_id:1 वह ऑस्ट्रेलिया के पहले प्रधान मंत्री के रूप में कार्यरत थे और ऑस्ट्रेलिया के उच्च न्यायालय के संस्थापक न्यायाधीश बने ।
[*veh Ostraliya ke pehle pradhAn mantri ke rUp me karyrat the Or Ostraliya ke uchch nyay-alya ke sansthApak nyay-dhIsh bane*]
(*He served as the first Prime Minister of Australia and became a founding justice of the High Court of Australia .*)
------ Cluster 1 ------
वह --> कार्यरत थे --> [ऑस्ट्रेलिया के]{a} [पहले] प्रधान मंत्री के रूप में
[*veh --> karyrat the --> [Ostraliya ke]{a} [pehle] pradhAn mantri ke rUp me*]
(*He --> served --> as [the] [first] Prime Minister [of Australia]{a}*)
वह --> बने --> [ऑस्ट्रेलिया के उच्च न्यायालय के]{b} [संस्थापक] न्यायाधीश
[*veh --> bane --> [Ostraliya ke uchch nyay-alya ke]{b} [sansthApak] nyay-dhIsh*]
(*He --> became --> [founding] justice [of the High Court of Australia]{b}*)
{a} ऑस्ट्रेलिया के --> property --> [पहले] प्रधान मंत्री के रूप में |OR| वह --> [पहले] प्रधान मंत्री के रूप में कार्यरत थे --> ऑस्ट्रेलिया के
[*{a} Ostraliya ke --> property --> [pehle] pradhAn mantri ke rUp me* |OR| *veh --> [pehle] pradhAn mantri ke rUp me karyrat the --> Ostraliya ke*]
(*{a} of Australia --> property --> as [first] Prime Minister* |OR| *He --> served as [the] [first] Prime Minister --> of Australia*)
{b} [ऑस्ट्रेलिया के]{c} उच्च न्यायालय के --> property --> [संस्थापक] न्यायाधीश |OR| वह --> [संस्थापक] न्यायाधीश बने --> [ऑस्ट्रेलिया के]{c} उच्च न्यायालय के
[*{b} [Ostraliya ke]{c} uchch nyay-alya ke --> property --> [sansthApak] nyay-dhIsh* |OR| *veh --> [sansthApak] nyay-dhIsh bane --> [Ostraliya ke]{c} uchch nyay-alya ke*]
(*{b} of High Court [of Australia]{c} --> property --> [founding] justice* |*OR*| *He --> became [founding] justice --> of High Court [of Australia]{c}*)
{c} ऑस्ट्रेलिया के --> property --> उच्च न्यायालय के
[*{c} Ostraliya ke --> property --> uchch nyay-alya ke*]
(*{c} of Australia --> property --> of High Court*)
------ Cluster 2 ------
वह --> [पहले] प्रधान मंत्री के रूप में कार्यरत थे --> ऑस्ट्रेलिया के
[*veh --> [pehle] pradhAn mantri ke rUp me karyrat the --> Ostraliya ke*]
(*He --> served as [the] [first] Prime Minister --> of Australia*)
वह --> [संस्थापक] न्यायाधीश बने --> [ऑस्ट्रेलिया के]{a} उच्च न्यायालय के
[*veh --> [sansthApak] nyay-dhIsh bane --> [Ostraliya ke]{a} uchch nyay-alya ke*]
(*He --> became [founding] justice --> of High Court [of Australia]{a}*)
{a} ऑस्ट्रेलिया के --> property --> उच्च न्यायालय के
[*{a} Ostraliya ke --> property --> uchch nyay-alya ke*]
(*{a} of Australia --> property --> of High Court*)

Figure 3.4: An illustration of the *Hindi-BenchIE* gold annotations for the Hindi rendition of the sentence depicted in Figure 3.3 is presented as a figure caption. Optional are the terms inscribed within square brackets, devoid of subsequent curly braces. For demonstrative reasons, the *essential-triples* are emphasized in yellow, and the *compensatory-triples* are accentuated in red. The *vulnerable-phrases* are noted in blue color. Excluded from the *Hindi-BenchIE* terminology are the indented transliterations and translations.

	ArgOE	*Multi2OIE*	*M&K*	*PredPatt*	*Gen2OIE*	*IndIE*
Precision	0.17	0.005	0.07	0.22	0.23	**0.49**
Recall	0.04	0.01	0.08	0.05	0.35	**0.53**
F1-score	0.07	0.008	0.08	0.09	0.28	**0.51**

Table 3.2: Performance of different OIE methods on the *Hindi-BenchIE* golden set is shown. Observed is that *IndIE* outperforms other methods on the *Hindi-BenchIE* golden set.

As an ablation to the unweighted average, it was noted that the last subword embedding being taken is consistently superior to the conventional practice of the first subword embedding being taken, a practice suggested by [44] for the Named Entity Recognition (NER) problem, which is analogous to semantic chunking as both are sequence labeling tasks. For an overarching presentation please refer to Appendix A.3.3 for the results derived from our chunker ablation studies.

3.4.1 IndIE vs Others

To assess the effectiveness of our triple extractor (*IndIE*), the given five baselines were utilized: (i) *M&K* [197], (ii) *ArgOE* [194], (iii) *PredPatt* [192], (iv) *Multi2OIE* [196], and (v) *Gen2OIE* [70], due to their underlying multilingual characteristics. The code-base for *M&K* is not accessible to the public; however, a dataset comprising sentences and corresponding triples, produced by their method, has been released by the authors[10]. Sentences were sampled randomly from *M&K* to construct the benchmark *Hindi-BenchIE*. A selected seed was employed to ensure the reproducibility of the random sampling.

It must be conveyed here that the *PredPatt* methodology is not constructed as a triple extractor. An entity extractor is resembled by the output generated by the method. Given a Hindi sentence the output of *PredPatt*, along with the rules devised to transform *PredPatt* output into triples format, are displayed in Appendix A.3.4.

Our method, *IndIE*, surpasses other methods in performance on the *Hindi-BenchIE* golden set. A comparison of the performance of different OIE methods is provided in Table 3.2. The recall value of a method is penalized in this metric by the failure to produce not even a single triple on the provided text. In such instances, the minimum number of *essential-triples* is added to the number representing *False Negatives* during the recall calculation. Our overall findings demonstrate that a greater number of meaningful triples from Hindi sentences were extracted by our proposed method, *IndIE*, than by other multilingual OIE tools.

[10] `https://www.kaggle.com/shankkumar/multilingualopenrelations15`

	ArgOE	*Multi2OIE*	*M&K*	*PredPatt*	*Gen2OIE*	*IndIE*
# Triples	51	59	199	48	**278**	277
# Sentences with no triples	69	68	NA	66	**0**	2
Avg. Tokens in a Triple	12	**7**	10	12	10	**7**
Avg. Sentence Coverage of a Triple	73%	49%	64%	76%	66%	**46%**
Triples with misplaced *kaarak*	1.9%	20%	44%	39%	16%	**0.7%**

Table 3.3: Triple statistics of different OIE methods on the *Hindi-BenchIE* golden set of 112 sentences are given. Considered while counting the number of tokens in a triple are non-unique tokens. Sentence coverage is calculated by $1 - |unique(sent) - unique(triple)|/|unique(sent)|$. Observed can be that *IndIE* triples have the least sentence coverage and *kaarak* errors. Hence, more fine-grained triples than other methods are generated by *IndIE*.

3.5 Discussion

Quantitative results into the extracted triples from different methods are presented in Table 3.3, motivated by qualitative observations. It was observed that more coarse triples were generated by methods such as *ArgOE* and *PredPatt* than other methods. A high sentence coverage percentage is possessed by coarse triples. The primary action within the sentence is identified while the residual text is added within the argument of the triple. Taking the following example for illustrative purposes, 007 के गुप्त नाम से प्रसिद्ध यह एजेंट फ़्लेमिंग की बारह पुस्तकों व दो लघुकथाओं में मौजूद है। [*007 ke nAm se prasidha yeh Ejant phleming ki bArah pustakon va do laghukathaon me maaujUd hae*] (*Renowned by the name of 007, this agent appears in twelve books and two short stories by Fleming.*). In this case, the *ArgOE* method produces the following triple: <007 के गुप्त नाम से प्रसिद्ध यह एजेंट, है, फ़्लेमिंग की बारह पुस्तकों व दो लघुकथाओं में मौजूद> [<*007 ke nAm se prasidha yeh Ejant*, *hae*, *phleming ki bArah pustakon va do laghukathaon me maaujUd*>] (<*Renowned by the name of 007 this agent*, *is*, *present in twelve books and two short stories by Fleming*>). In order to create a knowledge-base from raw text [219]. fine-grained triples are crucial, whereas overspecific relations or entities could be the result of coarse triples.

The triple yield by the *Gen2OIE* approach is superior to that of other methods. However, the sentence-quantity for which the method returns no triples cannot be determined, due to the unavailability of the *M&K* source code. Since triples are generated in English by the *M&K* and *Gen2OIE* methods, and then word alignments are utilized to acquire Hindi triples, non-meaningful Hindi triples are frequently generated, as incorrect word alignments split the postpositional word (*kaarak*) from the word that precedes it. Consequently, a greater number of triples are extracted with misplaced *kaarak*. Taking an example in Hindi, जब कोई मतैक्य नहीं हुआ तो विक्रम ने एक हल सोचा। [*jab koi mataekya nahin hua to vikram ne ek hal sochA*] (*When there was no consensus, Vikram thought of a solution.*). The *Gen2OIE* approach produced a triple as <विक्रम, सोचा, ने एक हल> [<*vikram*, *sochA*, *ne ek*

hal>] (Ungrammatical). Overlapped arguments in the extracted triples can be generated by *IndIE*, similar to *Gen2OIE*. For example, the two triples generated for the given example sentence मैं शब्द की आत्मा समझकर ही इस श्रेष्ठ तत्त्व की उपासना करता हूं। [*main shabd ki Atma samajhkar hee is shreshth tatv ki upAsna karta hun*] (*I worship this supreme element after understanding the soul of the word.*) are as follows <मैं,उपासना करता हूं,इस श्रेष्ठ तत्त्व की> [<*main*, *upAsna karta hun*, *is shrestha tatv ki*>] (<*I*, *worship*, *this supreme element*>) and <इस श्रेष्ठ तत्त्व की उपासना करता हूं,समझकर ही,आत्मा> [<*is shreshtha tatv ki upAsna karta hoon*, *samajhkar hee*, *Atma*>] (<*worship this supreme element*, *after understanding*, *soul*>).

In the experiments we conducted, poor performance on all metrics is exhibited by the zero-shot *Multi2OIE* method, a result expected since the production of erroneous facts by neural methods is recognized when contrasted with rule-based methods [215]. Therefore, the training of an OIE method based on neural architecture, grounded on the extractions of a rule-based OIE tool for low-resource languages, constitutes a promising direction.

On a solitary GeForce RTX 3090 GPU, a chunking speed of 50 milliseconds (ms) per sentence was observed for our transformer-based chunker, and a triple generation speed of 2 ms per sentence was found. In contrast, for *ArgOE* and *PredPatt*, a triple generation speed of 40 ms per sentence was measured, encompassing the time required for dependency parsing of the sentence. Although *Multi2OIE* fails to produce meaningful triples in the majority of instances, its triple generation speed, at 20 ms per sentence end-to-end, is less than all other methods. The primary source of delay (40 ms) in *ArgOE*, *PredPatt*, and *IndIE* was the parsing of a sentence through the Stanza library. However, the triple generation speed of our method remains less than other methods reliant on dependency parsing, such as ReVerb (1100 ms per sentence) and EXEMPLAR (1700 ms per sentence) [220].

3.5.1 IndIE Limitations

The scalability and adaptability of the IndIE pipeline are constrained by the hand-crafted rules utilized for triple extraction. Furthermore, despite providing a rationale to support the applicability of our proposed IndIE tool in other Indic languages, a benchmark analogous to Hindi-BenchIE could not be curated because of a deficiency of annotators for other Indic languages. Therefore, the performance of the proposed methodology (IndIE) on other Indic languages remains conjectural.

For the purpose of comparing the performance of various OIE methods, over 300 sentences had to be annotated by each Hindi annotator to prepare the ground-truth for 112 unique sentences. It is acknowledged that the size of our ground-truth is nearly one-third of that in previous works that conducted triple-level annotations [188, 213]. A significant impediment to the creation of an extensive ground-truth for testing the performance of Indic-NLP tools is the lack of availability of annotators for Indic languages. Growth of Indic-nlp in India can be significantly benefited by a community of Indic-data annotators. To the best of our knowledge, the solicitation of data from India is not supported by popular

crowdsourcing websites such as Amazon Mechanical Turk.

Constraints on the versatility and scalability of the IndIE pipeline are imposed by the utilization of manually created rules in the process of triple extraction. Additionally, while a rationale supporting the future application of the IndIE methodology to other Indic languages is provided, various hurdles were encountered in the creation of a benchmark similar to Hindi-BenchIE, owing to a paucity of human labelers for the languages of Indian subcontinent. Consequently, the comparative performance of IndIE on other languages from India remains a matter of speculation. The quantity of sentences in our proposed automatic evaluation benchmark, *Hindi-BenchIE*, is considerably smaller than the *BenchIE* work. As the manual generation of triples demands greater effort than manual triple extraction, the single-annotator employed in *Hindi-BenchIE* was able to produce over 500 triples for only 112 Hindi sentences. Therefore, it is believed that the benchmark has the potential to be further refined by the contributions of the Indic-nlp community. It is also acknowledged that the multilingual property of *IndIE* is restricted to the intersection of the sets of languages those supported by the Stanza library and on which xlm-roberta-base has been pre-trained.

3.6 Chapter Summary

By the limited resources possessed by Indic languages, the advancement of their NLP instruments has been hindered. In this study, an OIE instrument, *IndIE*, was formulated by us, by which triples are generated from unorganized Hindi sentences. For the provided sentence, chunk tags are predicted initially, and subsequently, a MDT is constructed, by which the triples are generated, using manually designed rules. A multilingual pretrained transformer model was utilized by us and with chunk annotated sentences from English and five Indic languages, it was fine-tuned. In sequence labeling tasks (like chunking), it was found that the mean of subword token embeddings is more beneficial than other methods. *Hindi-BenchIE*, a benchmark, was generated by us for automatically assessing Hindi triples, on the basis of a collection of 112 Hindi sentences, for the purpose of comparing the effectiveness of different multilingual OIE instruments. It was discerned that, than other baselines, more informative and detailed triples are produced by the *IndIE*.

Chapter 4

Multilingual Coreference Resolution in Low-resource South Asian Languages

The previous chapter focused on extracting triples from unstructured text in Indic languages. However, a key challenge emerged—the difficulty of linking multiple text spans referring to the same real-world entity. This limitation motivated the development of resources and tools for multilingual coreference resolution tailored to Indic languages. In this chapter, we introduce TransMuCoRes, a translated dataset for multilingual coreference resolution covering 31 South Asian languages, created using translation and word-alignment techniques. Additionally, we release a fine-tuned coreference resolution model checkpoint and critically examine the limitations of existing coreference evaluation metrics, particularly in handling datasets with split antecedents. This work aims to enhance coreference resolution capabilities for low-resource languages, facilitating better text understanding and downstream NLP applications[1].

Tiger seal [222].

[1]Content of this chapter is adapted from our paper published in LREC-COLING, 2024 [221]

Widespread in natural languages is the occurrence of alluding to a phrase previously stated in a discussion. In the inscribed text, redundancy of phrases is avoided by it, and a chain of consistent and linked sentences is generated. For example, the paragraph may be considered: "***John*** *is a good student.* ***He*** *asks intelligent questions and helps others. No wonder everybody loves* ***the boy.***" Linked are these sentences, as varied alluding phrases (emphasized in bold) are utilized to refer to the identical entity designated "*John*". An automated procedure that spots alluding phrases in a provided text and locates the nearest phrase to which reference is made, is coreference resolution. As a helpful preliminary stage, it functions and aids in many subsequent tasks such as entity linking [223], QA [224], and chatbots [225].

Currently available are several complete coreference resolution instruments for English [12], Arabic [226], and diverse European languages [227]. However, to the utmost of our understanding, no such instrument exists to perform coreference resolution for any South Asian language, notwithstanding the existence of numerous studies in this domain [228, 229, 230, 231]. On South Asian languages[2], our study is particularly centered, as native to roughly 25% of the world's populace they are, and three of the ten most commonly uttered languages across the globe originate from this area [169]. Therefore, the principal contributions of our effort are thus:

1. A **Trans**lated dataset for **Mu**ltilingual **Co**reference **Res**olution (TransMuCoRes)[3] in 31 South Asian languages[4].

2. Checkpoints for two available coreference resolution models that have been fine-tuned on the TransMuCoRes dataset and the manually annotated Hindi coreference resolution dataset by Mujadia et al. [8].

3. The shortcomings in present assessment metrics while assessing the resolved coreferences with split antecedents.

4.1 Related Work

Constituency trees in Hobbs algorithm [232], semantic traits [233], and syntactic traits [234] were utilized in prior endeavors in coreference resolution. A mention-ranking framework for coreference resolution with the help of neural networks and pretrained word embeddings was initially put forth by Lee et al. [235]. Through a text span of words (tokens) in the text, each mention is represented. Equation 4.1 was employed to compute a numerical value for each supplied pair of text spans. The symbol ϵ is used to represent a dummy antecedent, and zero is invariably taken as the value of $S(i, j = \epsilon)$. The weight of coreference link between

[2]https://southasiaoutreach.wisc.edu/languages/
[3]pronounced *trans-mew-cores*
[4]https://github.com/ritwikmishra/transmucores

Language	Gujarati	Dzongkha	Hindi	Kannada	Chhattisgarhi
Script	Gujarati	Uchen	Devanagri	Kannada	Devanagri
FLORES-200 code	uj_Gujr	dzo_Tibt	hin_Deva	kan_Knda	hne_Deva
Language	Awadhi	Assamese	Bengali	Tibetan	Bhojpuri
Script	Devanagri	Bangla	Bangla	Uchen	Devanagari
FLORES-200 code	awa_Deva	asm_Beng	ben_Beng	bod_Tibt	bho_Deva
Language	Burmese	Meitei	Nepali	Punjabi	Odia
Script	Burmese	Bangla	Devanagri	Gurumukhi	Kalinga
FLORES-200 code	mya_Mymr	mni_Beng	npi_Deva	pan_Guru	ory_Orya
Language	Magihi	Kashmiri	Maithili	Marathi	Malayalam
Script	Devanagari	Arabic	Devanagri	Devanagri	Malayalam
FLORES-200 code	mag_Deva	kas_Arab	mai_Deva	mar_Deva	mal_Mlym
Language	Telugu	Tamil	Tajik	Urdu	Uyghur
Script	Telugu	Tamil	Cyrillic	Arabic	Arabic
FLORES-200 code	tel_Telu	tam_Taml	tgk_Cyrl	urd_Arab	uig_Arab
Language	Persian	Pashto	Santali	Sindhi	Sinhala
Script	Arabic	Arabic	Ol Chiki	Arabic	Sinhala
FLORES-200 code	prs_Arab	pbt_Arab	sat_Beng	snd_Arab	sin_Sinh
Language	Uzbek				
Script	Latin				
FLORES-200 code	uzn_Latn				

Table 4.1: A listing of South Asian languages backed by TransMuCoRes. Note: indigenous speakers in Afghanistan as well have some Central Asian languages (Uzbek/Tajik) [7].

		#mentions	#sents	#coreference clusters	#singletons	#split-antecedants	#docs
TransMuCoRes	Train	3821540	1839883	1135906	350017	93668	87946
	Development	472083	224911	148189	46399	10505	10890
	Test	558093	255466	165664	59279	12944	11294
Mujadia et al. [8]	Train	10512	2839	3217	538	287	220
	Development	1306	347	387	58	31	27
	Test	1255	347	399	79	36	28

Table 4.2: Data characteristics of TransMuCoRes spanning 31 South Asian languages, along with the statistics of the Mujadia et al. [8] corpus in Hindi. It can be observed that the ratio of #split-antecedent to #mentions remains similar in both corpora, with proportions of 2.4% and 2.7% for TransMuCoRes and the Mujadia et al. [8] dataset, respectively.

the span j and i is represented by the score (S). In Equation 4.1, the antecedent score (s_a) and mention score (s_m) are computed utilizing the representations of span, which are acquired with the assistance of Bi-LSTM neural network and pretrained word embeddings.

$$S(i, j) = s_m(i) + s_m(j) + s_a(i, j) \tag{4.1}$$

The intention of the proposed methodology was to acquire a conditional probability distribution stated in Equation 4.2 where the collection of all possible mention spans preceding the i_{th} span during the discourse is represented by $Y(i)$.

$$P(y_i) = \frac{e^{S(i,y_i)}}{\sum_{y' \in Y(i)} e^{S(i,y')}} \tag{4.2}$$

A basis for many studies in the area of end-to-end coreference resolution has been the method put forth by Lee et al. [235]. Significant performance enhancements were noticed by Joshi et al. [236] when a pretrained transformer-encoder model [44], instead of word embeddings indepedent of context, was applied for encoding of text. Moreover, an objective based on text spans was utilized to pretrain a transformer-based model [89], which they employed for coreference resolution and text encoding, adhering to the method by Lee et al. [235]. Meanwhile, it was empirically demonstrated by Xu and Choi [237] that the Higher-Order Inference (HOI) methodology put forward by Lee et al. [238] frequently has negligible, and occasionally, even detrimental, effects in the coreference resolution task.

Regarding South Asian languages, numerous early studies have investigated Hindi coreference resolution, encompassing the endeavor by Dutta et al. [239], which suggested a revised Hobbs algorithm. Hand-crafted rules have been utilized by many studies to resolve coreferences in Hindi [240, 241], Marathi [242], and Telugu [243]. Person-Number-Gender (PNG) traits have been employed by some studies to identify mentions of an entity and a Conditional Random Field (CRF) architecture to identify coreferential connections in Hindi [244] and Tamil [245, 246]. Due to their unique inflectional system [247], incorporating PNG traits for certain South Asian languages presents considerable hurdles. Different to European languages, verbs in many South Asian languages are inflected as per the actions present in the text rather than the agents [248, 249]. Additionally, inflectional mistakes are widespread in such languages [250]. Hence, neural methodologies for automated feature extraction are crucial to advance multilingual coreference resolution. Anaphoras in Hindi were resolved by Singh et al. [251] utilizing a Gated Recurrent Unit (GRU) along context-independent word embeddings (static). However, for any of the previously mentioned studies in South Asian languages, no public tool or source code is accessible.

4.2 Dataset

In this investigation, the following manually annotated English coreference resolution data sets were incorporated by us: (i) LitBank which encompasses lengthier documents and

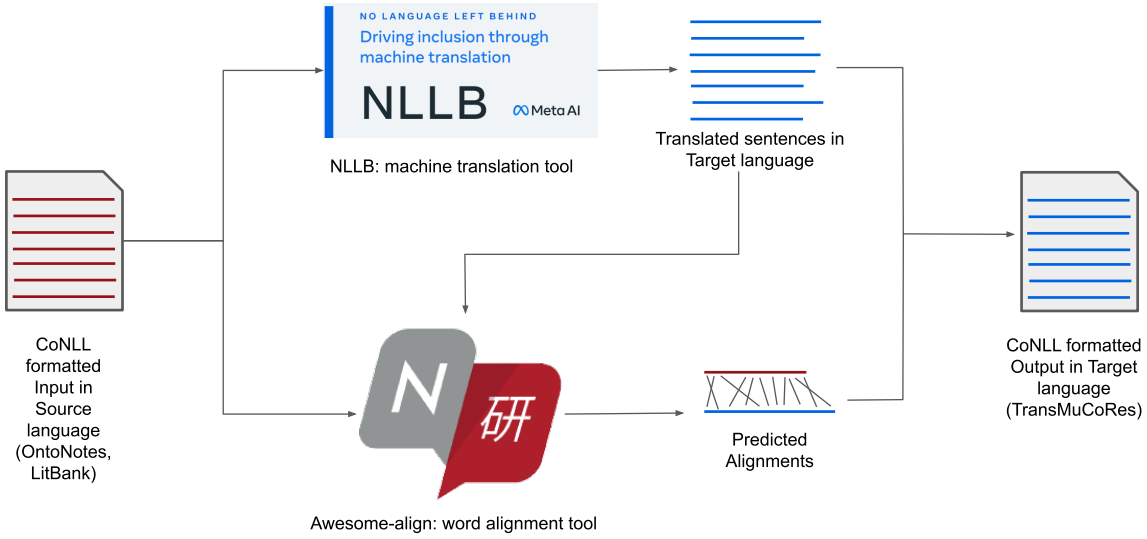
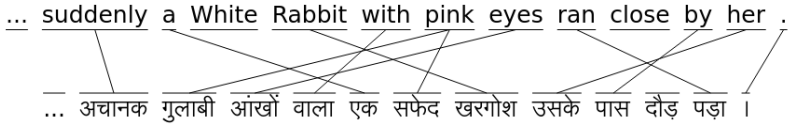


Figure 4.1: Overall pipeline used to construct the **Trans**lated dataset for **Mu**ltilingual **Co**reference **Res**olution (TransMuCoRes).

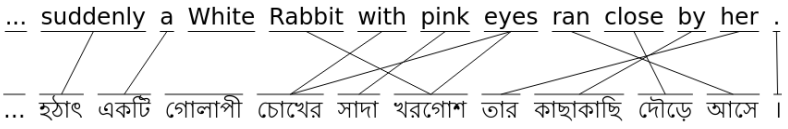
incorporates singleton mentions, i.e., mentions that arise solely once in the discourse [252, 10] and (ii) OntoNotes, broadly acknowledged as a standard data set for coreference resolution [253, 254, 255]. Noteworthy is that OntoNotes lacks singleton mentions, which prompted us to employ the LitBank data set in our research. It has been demonstrated by Kübler and Zhekova [256] and Yu et al. [257] that the performance of a coreference resolution system is affected by the detection of singleton mentions.

An overview of the pipeline employed for constructing individual samples within TransMuCoRes is furnished by Figure 4.1. To translate the English sentences to their target language, the *nllb-200-1.3B* model [258] was utilized by us. Displayed are the languages supported by TransMuCoRes in Table 5.3. A sanity-check was maintained by us for the generated translations, deeming it a failure if the translation chiefly comprised repeated punctuation. Instances of failed sanity in generated translations of English sentences were re-translated with the help of a more extensive *facebook/nllb-200-3.3B* checkpoint. Subsequently, we noted that merely 111 translations did not pass the sanity-check from a pool exceeding 3 million translations. Approximately 12% of the sanity-check failures corresponded to the Sindhi language. The detailed distribution of sanity-check results across languages is provided in Appendix A.4, within Table 7.

When an English sentence is translated into South Asian languages, the location of mentions within the translated sentence can be altered due to the flexible word order characteristics of these languages [259]. To exemplify this, an excerpt from a sentence in LitBank may be considered: "... suddenly a White Rabbit with pink eyes ran close by her." This sentence is translated into Hindi (... अचानक गुलाबी आंखों वाला एक सफेद खरगोश उसके पास दौड़ पड़ा ।) and Bengali (... হঠাৎ একটি গোলাপী চোখের সাদা খরগোশ তার কাছাকাছি দৌড়ে আসে ।) using



(a) English to Hindi



(b) English to Bengali

Figure 4.2: Visualizations of word-alignments predicted by the fine-tuned multilingual checkpoint by Dou and Neubig [3] in high-recall configuration. Observable is that the word-order of Hindi and Bengali is dissimilar from English.

	Mentions		
	Aligned	**Misaligned**	**Non-Aligned**
simalign with multilingual BERT (mbert)	53.7%	**6.1%**	40.1%
simalign with XLM-RoBERTa (xlmr)	58.5%	7.1%	34.3%
awesome-align without high recall	66.7%	9.4%	23.8%
awesome-align with high recall	**72.5%**	11.4%	**16.2%**

Table 4.3: Alignment statistics from awesome-align [3] and simalign [9] on TransMuCoRes indicate that the high-recall checkpoint of the awesome-align method yields the greatest number of aligned mentions.

the NLLB model. A visual representation[5] of the flexible-word nature of Hindi and Bengali in these translations is provided by Figure 4.2.

To generate word-level alignments subsequent to the translation step, we utilized the high-recall multilingual checkpoint from the awesome-align toolkit [3]. An aligned mention is defined as a continuous sequence of words in the target language that maps to a mention in the source (English) text. When a mention corresponds to a non-continuous set of words in the target language, it is termed a misaligned mention. In cases where a mention does not align with any word in the target language, it is identified as a non-aligned mention. In our approach, misaligned and non-aligned mentions are excluded from mention annotations. It was found to yield more aligned mentions compared to the results reported by Sabet et al. [9]. A comparative analysis of both approaches is presented in Table 4.3. Examples featuring Hindi and Bengali translations are shown in Figure 4.2, as these languages demonstrated

[5]Visualizations created by `https://vilda.net/s/slowalign/`

English	… suddenly [a White Rabbit with pink eyes]$_4$ ran close by [her]$_1$.	**mya_Mymr**	… □ရုတ်တရက် ပန်းရောင်မျက်လုံးတွေနဲ့ ယုန်ဖြူတစ်ကောင်□$_4$ သူမနားကို ပြေးလာပါတယ်။
mni_Beng	… নোংমদা মচুগী মমি লৈবা ৱাইত রেবিন অমনা মহাক্কী মনাক্তা চংলকখি ।	**asm_Beng**	… হঠাতে এটা [ৰঙা চকু থকা বগা কণী]$_4$ তাইৰ ওচৰলৈ দৌৰি আহিল ।
npi_Deva	… अचानक गुलाबी आँखा भएको सेतो खरगोश [उनको]$_1$ नजिकै दौड्यो ।	**awa_Deva**	… एकाएक उ एक सफेद खरगोश स भरा गुलाबी आँखी क ओकरे लगे [दौड़िके]$_1$ आवा ।
ben_Beng	… হঠাৎ একটি গোলাপী চোখের সাদা খরগোশ [তার]$_1$ কাছাকাছি দৌড়ে আসে ।	**ory_Orya**	… [ହଠାତ୍ ଏକ ଗୋଲାପୀ ଆଖିର]$_4$ ହ୍ୱାଇଟ୍ ରାବିଟ୍ ତାଙ୍କ ପାଖରେ ଦୌଡ଼ି ଆସିଲା ।
pan_Guru	… ਅਚਾਨਕ [ਇੱਕ ਗੁਲਾਬੀ ਅੱਖਾਂ ਵਾਲਾ ਵ੍ਹਾਈਟ ਰੈਬਿਟ]$_4$ [ਉਸਦੇ]$_1$ ਨੇੜੇ ਭੱਜਿਆ .	**bho_Deva**	… अचानक उहो [गुलाबी आँखिन क एक सफेद]$_4$ खरगोश ओकरे लगे दौड़त आवा ।
pbt_Arab	خرگوش سپین يو ناڅاپه چې کله ... ورغله نږدې سره سترګو ګلابي د	**bod_Tibt**	… [ག་མོ་ལ་མིག་དམར་དམར་པོ་ཡོད་པའི་རྣ་ཅོ་དཀར་པོ་ཞིག་ཉེ་བར་རྒྱུག་ཡོང་གི་རེད།]$_4$
prs_Arab	با سفید خرگوش یک□ ناگهان که وقتی ... دوید □او□$_1$ نزدیک به صورتی های □چشم$_4$	**dzo_Tibt**	…མོ་རང་ལུ་ (མོ་ལུ་ཆ་གྱུང་ཡོད་པ་ལས་ མོ་རང་ལུ) (མོ་ལུ་ཆ་གྱུང་ཡོད་པ་ལས་) མོ་རང་ལུ་ (མོ་ལུ་) མོ་རང་ལུ་)
sat_Beng	[illegible]	**guj_Gujr**	… અચાનક [એક ગુલાબી આંખોવાળો]$_4$ સફેદ સસલું [તેની]$_1$ નજીક દોડ્યો.
sin_Sinh	… රෝස [පැහැති ඇස් ඇති]$_4$ සුදු රැජිණක් ඇය අසල දිව ආ විට .	**hin_Deva**	… अचानक [गुलाबी आंखों वाला एक सफेद खरगोश]$_4$ [उसके]$_1$ पास दौड़ पड़ा ।
snd_Arab	گلابي ڪنبيءَ اچو هڪ اوچتو جڏهن ... ڊوڙيو ويجهو جي هن سان اکين	**hne_Deva**	… अचानक गुलाबी आंखी वाला [एक सफेद खरगोश]$_4$ ओखर करा दौड़त आईस ।
tam_Taml	… ஒரு வெள்ளை முயல் ரோஜா கண்களுடன் [அவளிடம்]$_1$ நெருங்கி ஓடியது .	**kan_Knda**	… ಇದ್ದಕ್ಕಿದ್ದಂತೆ [ಗುಲಾಬಿ ಕಣ್ಣಿನ ಬಿಳಿ ಮೊಲವು]$_4$ [ಅವಳ]$_1$ ಹತ್ತಿರ ಓಡಿತು .
tel_Telu	… [ఒక బ్లూ-కళ్ళు గల]$_4$ వైట్ కనేబిట్ [ఆమె]$_1$ దగ్గరకు వచ్చింది .	**kas_Arab**	خرگوش سفید اکھ اچانک رلان۔ ... نزدیک تیلہٕ اوس سِتی چشمن گلابی ییمہٕ
tgk_Cyrl	… ки ногаҳон [як харгӯши сафед бо чашмони гулобӣ]$_4$ ба наздикии [ӯ]$_1$ давида омад .	**mag_Deva**	… अचानक एगो गुलाबी [आँख वाला सफेद खरगोश]$_4$ ओकरा नगीच दौड़लइ ।
uig_Arab	ئويلىنسپ ئۇ ، باقتىغۇ ئويلىنسپ ... باقتىغۇ ئويلىنسپ ئۇ ، باقتىغۇ	**mai_Deva**	… अचानक [एकटा गुलाबी–आँखि]$_4$ वला गोरगो ओकर लगमे दौड़ि [गेल]$_1$ ।
urd_Arab	آنکھوں گلابی خرگوش سفید ایک اچانک جب ... گیا۔ بھاگ قریب کے □اس□$_1$ ساتھ کے	**mal_Mlym**	… പെട്ടെന്ന് ഒരു റോസ് കണ്ണുകളുള്ള വൈറ്റ് ക്നീറ്റ് [അവളുടെ]$_1$ അടുത്ത് ഓടി .
uzn_Latn	… [unga]$_1$ yaqinlashib ketayotgan qizg'al ko'zli Oq quyon ko'zidan o'tib qoldi .	**mar_Deva**	… अचानक [एक गुलाबी डोळे असलेला पांढरा ससा]$_4$ [तिच्या]$_1$ जवळ धावला .

Table 4.4: Coreference data subsequent to processing the translated sentences and aligned words. Coreference clusters for "*Rabbit*" and "*Alice*" are accentuated in green and pink color, respectively. The English sentence in this table is an excerpt from a sentence in LitBank [10]. Translation inaccuracies are observable in dzo_Tibt where words are reiterated. Alignment inaccuracies are observable in ben_Beng where the mention "*a White Rabbit with pink eyes*" is misaligned despite flawless translation. Note: Arabic fonts are to be perused left to right due to issues in the LaTeX typefonts.

a higher frequency of aligned mentions. Detailed alignment statistics for each language are provided in Appendix A.4, specifically in Table 8.

The designed TransMuCoRes framework was developed by incorporating the following three fundamental elements: (i) generated translations, (ii) mapped mentions between English texts and their corresponding translations, and (iii) manually curated mentions

				Mentions			MUC			B^3			CEAFe			LEA			CoNLL
				P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	F1
wl-coref [12]	5 langs	mbert	dev	62	**77**	69	57	**67**	61	42	**55**	47	37	55	44	36	49	42	51
			test	64	**79**	71	61	70	65	42	**56**	48	34	53	41	37	**51**	43	52
		xlmr	dev	**67**	**77**	**72**	**63**	**67**	**65**	**49**	**55**	**52**	**41**	**59**	**48**	**43**	**50**	**46**	**55**
			test	**68**	**79**	**73**	**66**	**71**	**69**	**49**	**56**	**52**	**37**	**58**	**45**	**44**	**51**	**48**	**55**
	All langs	mbert	dev	37	68	48	32	56	41	22	**46**	30	20	45	28	18	**40**	25	33
			test	39	70	50	34	59	44	22	**47**	30	19	43	26	18	**42**	26	33
		xlmr	dev	45	62	52	40	51	45	29	41	34	**25**	45	**32**	25	36	29	37
			test	46	63	53	42	54	47	29	41	34	**23**	44	**30**	25	37	30	37
fast-coref [11]	All langs	mbert	dev	44	**76**	56	41	59	48	28	42	34	18	55	27	24	36	29	36
			test	46	76	58	44	62	52	29	42	34	17	53	26	25	37	30	37
		xlmr	dev	**48**	**76**	**59**	**46**	**61**	**52**	**33**	44	**38**	21	**59**	31	**29**	39	**33**	**41**
			test	**50**	**77**	**60**	**49**	**64**	**56**	**34**	44	**38**	20	**58**	29	**30**	40	**34**	**41**

Table 4.5: Performance of fine-tuned fast-coref [11] with xlmr encoder is superior to zero-shot performance of wl-coref [12] on all the languages. However, wl-coref is discovered to be performing well for the 5 languages on which it is fine-tuned.

and coreference groupings from the English corpus. For each target language, placeholder dummy entries were utilized for the constituency parse structures and speaker metadata. This approach was adopted due to the unavailability of publicly released tools capable of producing constituency parses for South Asian languages, as well as the absence of speaker annotations in the Litbank and Mujadia et al. [8] data sets.

Table 4.4 underscores the translation and alignment errors identified within the TransMuCoRes construction pipeline. Owing to inaccuracies in both translation and alignment, we proceeded to train several pre-existing coreference resolution models using a manually curated coreference resolution data set. We employed the Hindi coreference resolution corpus proposed by Mujadia et al. [8], as it represents the only publicly available manually annotated resource for coreference resolution in any South Asian language (Hindi). For detailed statistics pertaining to both TransMuCoRes and the corpus by Mujadia et al. [8], kindly consult Table 4.2. Although we preserved the original train:dev:test partitions for OntoNotes, we needed to define equivalent splits for LitBank and the corpus introduced by Mujadia et al. [8]. These splits will be made publicly available as part of the released resources to facilitate a standardized evaluation of future coreference resolution models.

4.3 Coreference Resolution Models

In this study, we employed the following pre-existing coreference resolution models: (i) wl-coref [12], and (ii) fast-coref [11]. Our choice of these models was mainly driven by the fact that they offer fine-tuning scripts compatible with newly prepared CoNLL-style

Language	fast-coref [11] vs wl-coref [12] on fine-tuned xlmr													
	Split	Mentions F1	B^3 F1	MUC F1	CEAFe F1	LEA F1	CoNLL F1	Split	Mentions F1	B^3 F1	MUC F1	CEAFe F1	LEA F1	CoNLL F1
awa_Deva	DEVLOPMENT	50 vs 42	41 vs 32	29 vs 25	27 vs 28	23 vs 19	33 vs 28	TEST	51 vs 43	45 vs 33	28 vs 23	25 vs 26	23 vs 18	33 vs 27
asm_Beng		45 vs 46	25 vs 24	38 vs 35	22 vs 26	20 vs 18	28 vs 28		47 vs 46	25 vs 23	41 vs 37	21 vs 24	20 vs 18	29 vs 28
dzo_Tibt		18 vs 4	9 vs 2	15 vs 3	5 vs 2	6 vs 1	10 vs 2		19 vs 5	10 vs 2	16 vs 4	6 vs 3	7 vs 2	10 vs 3
tgk_Cyrl		61 vs 16	37 vs 8	52 vs 11	33 vs 11	31 vs 6	41 vs 10		62 vs 15	37 vs 7	55 vs 11	30 vs 10	32 vs 5	41 vs 10
pbt_Arab		33 vs 29	18 vs 14	31 vs 21	15 vs 17	15 vs 9	21 vs 17		36 vs 31	19 vs 13	34 vs 23	14 vs 17	16 vs 9	22 vs 18
ben_Beng		74 vs 73	51 vs 52	67 vs 66	45 vs 48	46 vs 47	55 vs 55		76 vs 75	52 vs 53	71 vs 70	43 vs 46	48 vs 49	56 vs 56
mar_Deva		72 vs 71	49 vs 51	65 vs 64	44 vs 47	44 vs 46	52 vs 54		74 vs 72	50 vs 52	68 vs 68	40 vs 44	45 vs 47	53 vs 54
uig_Arab		21 vs 25	9 vs 10	15 vs 13	9 vs 15	6 vs 5	11 vs 13		21 vs 26	8 vs 10	17 vs 16	8 vs 14	6 vs 6	11 vs 13
bho_Deva		52 vs 41	31 vs 24	46 vs 33	27 vs 26	26 vs 19	35 vs 27		54 vs 43	31 vs 24	49 vs 35	25 vs 24	26 vs 19	35 vs 28
npi_Deva		74 vs 73	52 vs 53	68 vs 66	46 vs 49	47 vs 47	55 vs 56		76 vs 75	53 vs 54	71 vs 70	44 vs 47	49 vs 49	56 vs 57
bod_Tibt		63 vs 6	17 vs 2	44 vs 3	10 vs 3	12 vs 2	24 vs 3		64 vs 6	18 vs 3	48 vs 3	10 vs 4	14 vs 2	25 vs 3
guj_Gujr		74 vs 73	50 vs 51	66 vs 65	45 vs 48	45 vs 46	54 vs 55		75 vs 74	50 vs 51	69 vs 68	42 vs 45	45 vs 47	54 vs 55
hin_Deva		75 vs 74	52 vs 54	68 vs 68	46 vs 50	47 vs 49	55 vs 58		76 vs 76	52 vs 55	71 vs 72	43 vs 48	48 vs 51	55 vs 58
hne_Deva		48 vs 39	28 vs 21	41 vs 29	27 vs 24	23 vs 16	32 vs 25		48 vs 40	28 vs 21	43 vs 31	25 vs 22	23 vs 16	32 vs 25
kan_Knda		73 vs 71	50 vs 50	65 vs 63	45 vs 48	44 vs 45	53 vs 54		74 vs 73	50 vs 51	68 vs 67	42 vs 46	46 vs 46	53 vs 55
kas_Arab		33 vs 13	17 vs 7	27 vs 8	16 vs 10	13 vs 5	20 vs 8		34 vs 14	17 vs 6	30 vs 9	15 vs 9	13 vs 4	20 vs 8
mag_Deva		54 vs 47	32 vs 28	47 vs 38	28 vs 28	27 vs 22	36 vs 31		56 vs 50	32 vs 27	51 vs 41	26 vs 27	28 vs 22	36 vs 32
mai_Deva		48 vs 38	28 vs 22	42 vs 32	25 vs 22	24 vs 18	32 vs 25		50 vs 39	29 vs 22	45 vs 34	23 vs 21	25 vs 18	32 vs 25
mal_Mlym		66 vs 65	43 vs 44	58 vs 56	38 vs 44	37 vs 37	46 vs 48		68 vs 67	44 vs 45	62 vs 60	36 vs 42	39 vs 39	47 vs 49
mni_Beng		30 vs 9	14 vs 3	24 vs 4	11 vs 5	9 vs 1	16 vs 4		33 vs 10	14 vs 3	27 vs 6	10 vs 5	10 vs 1	17 vs 4
mya_Mymr		61 vs 53	37 vs 32	52 vs 42	34 vs 35	31 vs 26	41 vs 36		63 vs 54	36 vs 32	55 vs 46	31 vs 33	31 vs 26	41 vs 37
ory_Orya		22 vs 22	10 vs 9	19 vs 13	8 vs 12	7 vs 4	12 vs 11		24 vs 23	10 vs 8	22 vs 15	8 vs 12	7 vs 4	13 vs 12
pan_Guru		71 vs 70	48 vs 49	64 vs 63	43 vs 47	43 vs 44	51 vs 53		73 vs 72	48 vs 50	68 vs 67	40 vs 44	44 vs 45	52 vs 53
prs_Arab		69 vs 64	47 vs 43	63 vs 56	40 vs 42	42 vs 37	50 vs 47		70 vs 66	46 vs 44	66 vs 60	37 vs 39	42 vs 39	50 vs 48
sat_Beng		11 vs 4	5 vs 2	10 vs 2	3 vs 2	4 vs 1	6 vs 2		13 vs 5	6 vs 2	12 vs 4	3 vs 2	4 vs 1	7 vs 3
sin_Sinh		22 vs 22	10 vs 9	18 vs 13	8 vs 13	7 vs 5	12 vs 11		24 vs 23	11 vs 9	22 vs 15	8 vs 12	8 vs 5	14 vs 12
snd_Arab		30 vs 31	15 vs 14	26 vs 22	13 vs 18	12 vs 9	18 vs 18		31 vs 32	15 vs 14	28 vs 24	12 vs 18	12 vs 10	18 vs 18
tam_Taml		71 vs 70	47 vs 50	63 vs 63	43 vs 49	42 vs 44	51 vs 54		72 vs 72	48 vs 50	66 vs 66	40 vs 45	43 vs 45	51 vs 54
tel_Telu		71 vs 71	48 vs 51	64 vs 64	43 vs 48	43 vs 46	52 vs 54		73 vs 73	50 vs 52	68 vs 68	39 vs 45	45 vs 48	52 vs 55
urd_Arab		70 vs 71	47 vs 50	63 vs 64	41 vs 47	42 vs 45	51 vs 54		72 vs 72	47 vs 50	67 vs 67	39 vs 44	43 vs 45	51 vs 54
uzn_Latn		63 vs 60	40 vs 39	54 vs 50	38 vs 40	34 vs 33	44 vs 43		64 vs 60	39 vs 38	57 vs 53	35 vs 37	34 vs 33	43 vs 43
Mujadia et al. [8]		50 vs 78	35 vs 66	45 vs 74	33 vs 62	31 vs 62	38 vs 67		54 vs 79	38 vs 68	51 vs 76	31 vs 60	34 vs 64	40 vs 68
Overall		59 vs 52	38 vs 34	52 vs 45	31 vs 32	33 vs 29	41 vs 37		60 vs 53	38 vs 34	56 vs 47	29 vs 30	34 vs 30	41 vs 37

Table 4.6: The wl-coref [12] method performs better than fast-coref [11] for the languages on which it was fine-tuned (Hindi, Marathi, Tamil, Urdu, and Telugu).

data. In addition, wl-coref and fast-coref achieve performance levels within 3-4% of the best-reported results on the OntoNotes benchmark[6]. Unfortunately, we faced challenges in finding comparable training resources for the latest coreference resolution systems targeting the OntoNotes dataset [260, 261, 262]. Owing to constraints in computational resources, we did not fine-tune the LingMess model [263] as part of this research. Likewise, other coreference approaches [264, 265], which have not consistently outperformed fast-coref and wl-coref on the OntoNotes benchmark, were also excluded from fine-tuning in our experiments. Additionally, models such as those introduced by Yu et al. [257] and Aloraini et al. [226] depended heavily on features derived from pretrained word representations tailored to specific languages, complicating the adaptation process for multilingual training. To customize fast-coref and wl-coref for handling multilingual span data, we utilized the base versions of XLM-RoBERTa (xlmr) by Conneau et al. [216] and multilingual BERT (mbert) by Devlin et al. [44] as text encoding backbones. This strategy enabled the fine-tuning of a unified model across multilingual datasets.

[6] `https://paperswithcode.com/sota/coreference-resolution-on-ontonotes`

The fast-coref system is developed by leveraging the Longformer encoder [266] as part of the longdoc coreference resolution architecture [267]. Observations indicated that simultaneous training, combined with data augmentation strategies like pseudo-singletons, enhanced the model's effectiveness across multiple datasets from different domains.

The wl-coref model relies on a head-finding mechanism via dependency parse trees, which compelled us to fine-tune the model solely for languages with publicly accessible dependency parsers. Hence, the wl-coref model was fine-tuned by us on Hindi, Urdu, Tamil, Marathi, and Telugu data utilizing the dependency parser of the Stanza [2] library. The fine-tuned wl-coref model was assessed in a zero-shot manner for the remaining languages.

It is important to highlight that the Stanza dependency parser often constructs parse trees by segmenting words into subword elements. For instance, when tokenizing the sentence, கடுமையான வலிகளும் , திடீர் தலைச்சுற்றலும் , பின்னர் துளைகளிலிருந்து இரத்தப்போக்குக , மற்றும் உடைப்புகளும் இருந்தன . *(There were severe pains, sudden dizziness, then bleeding from the pores, and ruptures.)*, the term கடுமையான *(severe)* is divided into கடுமைய் *(severely)* and ஆன *(became)*. It becomes clear that simply joining the subwords does not faithfully reconstruct the original token. It must be emphasized that the TransMuCoRes corpus exclusively provides annotations at the word level. Consequently, generating a dependency parse tree based on words, instead of subword units, becomes particularly important. To resolve this issue, we employed the awesome-align word alignment tool to create a linkage between the original tokens and their respective subword segments.

4.3.1 Evaluation Metrics

In this study, we adopt evaluation metrics that are traditionally applied in coreference resolution tasks [89, 11, 268]. MUC serves as a link-centric measure, where a lower score signifies a substantial number of links that must be either inserted or removed from the predicted coreference chains to better match the gold-standard annotations. On the other hand, B^3 operates as a mention-oriented metric, evaluating how well the model clusters coreferring mentions together while maintaining a separation between non-coreferring ones. CEAFe quantifies the extent of correspondence between aligned key and system responses, with a higher CEAFe score denoting stronger agreement between the key and the prediction. The CoNLL score is computed as the simple average of MUC, CEAFe, and B^3 scores. Additionally, LEA is a metric based on entities and links, where a higher LEA value indicates more precise resolution of extensive coreference chains. A comprehensive explanation of these metrics can be found in [269].

4.4 Results

In this study, we employed the evaluation scripts provided by Paun et al. [268]. Table 4.5 demonstrates the robust outcomes achieved by the wl-coref technique when tested on

languages it was specifically fine-tuned for. This observation is reinforced by the results shown in Table 4.6, indicating that wl-coref surpasses fast-coref exclusively on languages it was adapted to. Remarkably, wl-coref secures the best results on our test subset of the golden set, attaining CoNLL F1 and LEA F1 scores of 64 and 68, respectively. Our work marks the first release of coreference resolution tools for Hindi, along with an evaluation of their performance on the golden set compiled by Mujadia et al. [8]. Furthermore, we noticed an improvement in both models' performance when singleton mentions are excluded during evaluation. This trend is reflected in Table 10 and Table 11 within Appendix A.4. Reconstructing the proposed TransMuCoRes architecture demands approximately 14 GB of GPU memory, with the reconstruction process potentially extending up to three months on a single GPU. Similarly, fine-tuning coreference resolution models requires a minimum of 30 GB of GPU memory and can take up to eight hours of computational time. Appendix A.4, specifically Section A.4.1, provides comprehensive details about the computational resources necessary for this project. Additionally, we observed that one of the CoNLL evaluation metrics, BCUB [270], can occasionally fail to produce meaningful results when handling examples where the key coreference chains include split antecedents. Consider, for example, the following paragraph:

> ***Thatcher***$_a$ *grew up in Lincolnshire whereas* ***Gandhi***$_b$ *was raised in Allahabad.* ***Both***$_c$ *become powerful figures.* ***They***$_d$ *locked horns in 1983. The world watched as* ***the Iron Lady of India***$_e$ *stood against* ***the Iron Lady of UK***$_f$.

When the output of a system (its predictions) matches the reference (the gold standard), the BCUB recall value reaches 1.25. In addition, the LEA measure, proposed as a substitute for CoNLL metrics by Moosavi and Strube [269], also produces non-ideal results for this scenario. According to the LEA metric's definition, the F1, recall, and precision scores are all 1.16 when both the response and the key are set to the ground-truth. This highlights the importance of developing an evaluation metric that can accurately address split antecedents in coreference tasks. Moreover, the existence of split antecedents not only exposes a major shortcoming in current evaluation metrics for coreference resolution but also introduces difficulties during the training phase of present-day coreference models. This issue arises from the structural constraints of these models, which allow only a single antecedent per mention, even though split antecedents involve several antecedents appearing earlier in the text.

4.5 TransMuCoRes Limitations

A significant drawback related to the risk of bias propagation during the conversion of the English corpus into multiple target languages [271] is faced by this work. Moreover, extensive computational power is required to reproduce these experiments. Evaluating coreference resolution systems for languages apart from Hindi proved particularly difficult due to the

limited availability of gold-standard annotated materials. The restricted size of the data set for evaluating languages other than Hindi is also recognized. The results discussed in this paper depend on the particular data partitions chosen from the corpus by Mujadia et al. [8] and LitBank. To confirm these outcomes, cross-validation procedures are required. Looking ahead, we intend to broaden the data set to include more low-resource languages that are supported by NLLB models. There is also an urgent requirement for the creation of a new evaluation metric that can reliably measure coreference groupings containing split antecedents.

A variety of scholarly efforts have explored the automatic identification of coreferences within South Asian languages. Nevertheless, there remains a significant lack of openly available tools and datasets in this field. This work seeks to bridge this shortfall by presenting TransMuCoRes, a translated corpus tailored for Multilingual Coreference Resolution tasks. We also provide checkpoints for two pre-existing approaches, each adapted using the gold-standard Hindi data and TransMuCoRes. Our findings reveal that adapting the wl-coref approach is practical for certain South Asian languages equipped with a dependency parser. Importantly, it surpasses fast-coref in performance on those languages where it has been specifically fine-tuned.

4.6 Chapter Summary

This chapter addresses the significant gap in coreference resolution resources for South Asian languages, despite the availability of robust tools for English, Arabic, and various European languages. The authors introduce TransMuCoRes, a translated dataset covering 31 South Asian languages, and provide fine-tuned checkpoints for two coreference resolution models (wl-coref and fast-coref), using both the new dataset and a manually annotated Hindi corpus. The chapter details the challenges of translating and aligning mentions due to the flexible word order and inflectional complexity of South Asian languages, and describes the methodology for constructing the dataset, including the use of word alignment tools and the handling of annotation limitations. The authors evaluate the performance of wl-coref and fast-coref, finding that wl-coref outperforms fast-coref on languages where it can be fine-tuned with available dependency parsers. The chapter also highlights shortcomings in current evaluation metrics, particularly in handling split antecedents, and underscores the computational demands of building and training these models. Limitations include potential bias from translation, limited gold-standard data for languages other than Hindi, and the need for improved evaluation metrics. Overall, this work represents the first public release of coreference resolution tools and resources for South Asian languages, aiming to facilitate further research and development in this under-resourced area.

Chapter 5

Multilingual Non-Factoid Question Answering with Answer Paragraph Selection

Building upon the preceding chapters' exploration of Indic language text representation through triples and coreference linking, this chapter shifts focus to evaluating these tools within a QA pipeline. A critical need for non-factoid QA datasets arises, necessitating a robust evaluation framework. This chapter details the methodology employed to construct the largest multilingual non-factoid dataset, outlining the key findings derived from this process. A comparative analysis of this dataset with existing multilingual QA datasets, particularly those of the QuADs family, is presented, highlighting the unique characteristics and contributions of the newly developed resource[1].

Bison seal [222].

[1] Content of this chapter is adapted from our paper accepted in the Special Session of PAKDD, 2025 [272]

A standard Question Answering Datasets (QuADs) [273, 137] commonly consist of question-answer pairs. Nevertheless, certain QuADs are set apart by including an extra element, referred to as *evidence* or *context*, which is provided alongside each question. This contextual data is intended to supply enough information to resolve the related question; thus, these QuADs are frequently called Reading-Comprehension (RC) datasets. Most RC datasets [4] concentrate on factoid responses, which are generally brief expressions or specific entities. As an illustration, a factoid inquiry such as *Who was the first Prime Minister of India?* and its corresponding answer, *Jawaharlal Nehru*, can be cited.

In contrast, non-factoid questions require extended, explanatory answers that may span several sentences or even paragraphs. For example, a non-factoid question could be formulated as *How did Jawaharlal Nehru become the first Prime Minister of India?*, building upon the previous example.

Current search engines struggle to provide satisfactory answers to non-factoid queries, as indicated by evidence [274]. Furthermore, humans also encounter difficulties when responding to non-factoid questions [5]. To enable automated systems to address non-factoid queries, extensive non-factoid QuAD datasets are essential for adapting QA models. Multilingual QA systems encounter further obstacles because of the scarcity of these datasets, highlighting the necessity for constructing a multilingual QA resource tailored for non-factoid questions.

Within this research, we automatically gather Question-Answer sets along with their related news stories from the British Broadcasting Corporation (BBC) portal in several languages[2]. With the exception of the gold standard subset, which is hand-labeled, the remainder of the dataset is not, as it is based on the assumption proposed by Soleimani et al. [4] that *every paragraph following a question-based subheading includes its response*. Consequently, we designate this dataset as having silver annotations/answers. Prior investigations have shown that silver annotations can be advantageous for developing text classification models in fields where gold annotations are scarce, such as legal [275], journalistic [276], and healthcare [277] sectors. Through a comparison of silver annotations with gold annotations, it was found that 98% of the queries were accurately addressed by their silver responses. Our principal contributions are summarized below:

1. MuNfQuAD[3], a multilingual question-answering dataset, partitioned into test, validation, and train splits, is released by us. More than 578K Question-Answer pairs in 38 different languages are contained by it.

2. Multilingual Answer Paragraph Selection models, fine-tuned on MuNfQuAD, leveraging base variants of different pretrained encoders, are also released by us.

1. We present MuNfQuAD[4], a multilingual QA resource, divided into training, valida-

[2] `https://www.bbc.co.uk/ws/languages`
[3] `https://github.com/ritwikmishra/MuNfQuAD`
[4] `https://github.com/ritwikmishra/MuNfQuAD`

tion, and test sets. It includes over 578,000 QA pairs spanning 38 languages.

2. We also provide multilingual answer paragraph selection models, adapted to MuNfQuAD, utilizing base versions of various pretrained encoders.

5.1 Related Work

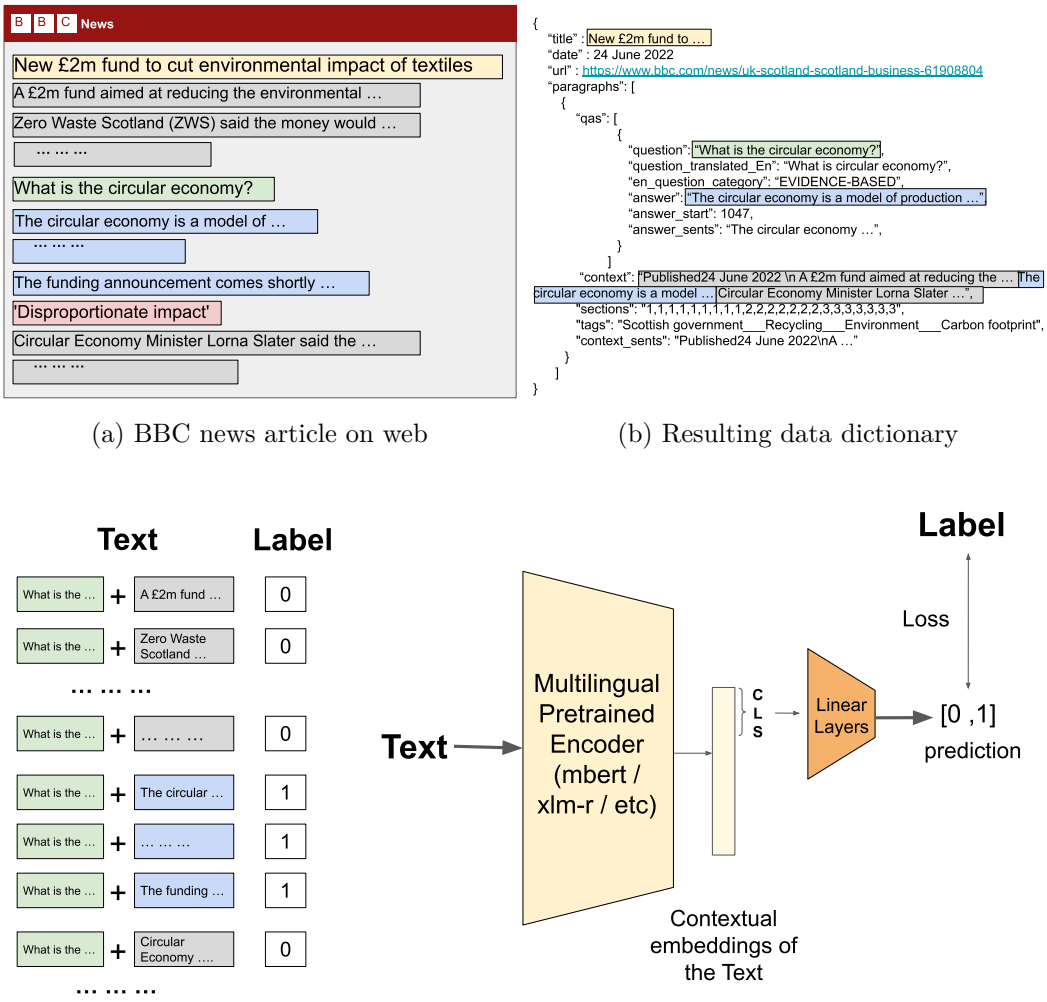
WikiQA [278] was among the first datasets for automated question answering in English. It sourced questions from Bing search logs and aligned them with pertinent Wikipedia entries. SQuAD [37] is a widely recognized English QA benchmark. Crowdworkers formulated questions from English Wikipedia passages, and answers were located within concise text spans. Natural Questions [42] is regarded as the largest dataset for factoid span extraction, comprising nearly 320,000 questions, each paired with a full Wikipedia article, a long-form answer, and a brief answer.

Our approach is most similar to NLQuAD [4], though that dataset is limited to English. For a thorough overview of English QA datasets, see Cambazoglu et al. [279], Rogers et al. [33].

Our primary emphasis is on multilingual QuAD datasets (mQuADs). bAbI [100] is an early example in this area, featuring factoid questions and extractive responses in both Romanized Hindi and English. Gupta et al. [101] introduced a Hindi-English bilingual QuAD, demonstrating enhanced QA results through question categorization. Gupta et al. [280] automatically translated a portion of SQuAD into Hindi, but we found that most answer positions were incorrect. The XQA dataset [281] collected questions from Wikipedia's "Did you know?" sections, omitting entity names, which were then used as factoid answers. The top ten Wikipedia articles related to each entity were used as context for the questions. The authors also highlighted the limitations of translation-based augmentation in QA tasks. XQuAD [103] was created by manually translating a subset of SQuAD into ten different languages.

MLQA [282] generated questions from English Wikipedia entries and provided extractive responses by employing crowd contributors. Afterwards, parallel sentences were identified in the English article, and the English QA pair was manually rendered into additional languages. Among multilingual QuADs, TyDi QA [102] stands out as a significant benchmark, emphasizing naturally occurring questions where the creators do not know the answers in advance. Driven by curiosity, annotators were prompted to pose questions, and for minimal answer span annotation and answer paragraphs, highly ranked Wikipedia entries were utilized. Chaii[5] supplies a QuAD featuring long-context and factoid queries in Hindi and Tamil. Table 5.1 presents the metrics for various mQuADs. Although multilingual resources such as ResPubliQA [285] and BOLT [286] have related publications [287, 288], we were unable to locate these datasets on the public web.

[5] `https://www.kaggle.com/c/chaii-hindi-and-tamil-question-answering`



(a) BBC news article on web

(b) Resulting data dictionary

(c) Training data

(d) APS model

Figure 5.1: A diagram illustrating the process of refining the Answer Paragraph Selection (APS) model and the method of data acquisition is shown. (a) A BBC news report, retrieved from the URL: `https://www.bbc.com/news/uk-scotland-scotland-business-61908804`, is displayed. Paragraphs identified as silver answers are highlighted with blue rectangles, subheadings phrased as questions are marked with green rectangles, and those that are not interrogative are indicated with red rectangles. In (b), the extracted data structure is depicted. Subfigure (c) demonstrates that Label-0 denotes the paragraph is absent from the silver answer, whereas Label-1 indicates its inclusion. The APS model's architecture and the fine-tuning workflow are visualized in (d).

5.2 Data Curation

Our objective was to assemble a multilingual QA corpus containing non-factoid queries. To this end, we automatically scraped the BBC news portal, collecting articles and their associated question-answer sets. This work utilized Python's BeautifulSoup and requests

Source	Type	Dataset Name	Avg #tokens			#Samples	#Languages
			Q	C	A		
Gupta et al. [280]	Factoid Span Detection	$^{\beta}$MQA	10	126	2	36K	2
Google Research India (2021)	Factoid Span Detection	Chaii	7	1694	2	1.1K	2
Artetxe et al. [103]	Factoid Span Detection	XQuAD	12	155	4	13K	11
Gupta et al. [101]	Factoid Span Detection	$^{\alpha}$MMQA	9	314	7	2.7K	2
Lewis et al. [282]	Factoid Span Detection	MLQA	8	117	3	46K	7
Asai et al. [283]	Open Retreival QA	XOR QA	6	-	4	40K	7
Clark et al. [102]	Factoid Span Detection	TyDiQA-GoldP	5	76	4	54K	11
Clark et al. [102]	Answer Paragraph Selection	TyDiQA-SelectP	6	2891	80	90K	11
Clark et al. [102]	Factoid Span Detection	TyDiQA-MinSpan	5	2825	4	78K	11
Longpre et al. [284]	Open-Domain QA	MKQA	8	-	2	260K	26
Liu et al. [281]	Factoid Span Detection	$^{\gamma}$XQA	17	5326	2	90K	9
Weston et al. [100]	Factoid Span Detection	$^{\delta}$bAbI	5	21	1	330K	2
Ours	Answer Paragraph Selection	**MuNfQuAD**	6	909	191	578K	38

Table 5.1: Characteristics of several multilingual QuAD datasets. For languages where spaces do not separate words, such as Chinese and Japanese, the *MeCab* and *jieba* Python modules were employed for tokenization, respectively. In contrast, for other languages, tokenization based on whitespace was applied. $^{\alpha}$MMQA did not supply a mapping from questions to context, so we constructed one greedily. $^{\beta}$MQA is derived from SQuAD, translated automatically into Hindi, but approximately 92% of its (start, end) positions are incorrect. $^{\gamma}$For XQA, all ten contexts related to a question are concatenated to create a unified context. $^{\delta}$Within the bAbI corpus, the passage preceding a question is used as its context.

packages for data extraction. For a specific language (e.g., Hindi), we executed a scraper on both the Wayback Machine[6] (also known as the web archive) and the BBC (Hindi) site. The most recent homepage of BBC (Hindi) provided the initial articles for the BBC scraper, while the earliest archived homepage snapshot served as the starting point for the Wayback Machine scraper. The scraping strategy was crafted to retrieve news stories by detecting the presence of interrogative subheadings within a page. Figure 5.1(a) shows the user interface of a BBC news story, and Figure 5.1(b) displays the collected data in dictionary form. Nevertheless, subheadings that are not interrogative and appear in the article are intentionally excluded from the context. This is because such subheadings generally function as summaries, provide topic cues, or act as *descriptive titles* [289]. While these elements help frame the content, they may not directly answer the dataset's specific questions. Additionally, candidate URLs were gathered from the current page by extracting anchor elements. An interrogative subheading is recognized by the presence of a trailing question mark (or its language-specific equivalent) in the subheading text.

For dividing sentences in our corpus, we employed the multilingual model checkpoint from ersatz [13]. We arranged the MuNfQuAD questions by their frequency of appearance, then selected the top 50 most common questions per language and rendered them into English via Google Translate. It was observed that a significant number of these questions appeared across unrelated articles. To address this, we constructed a phrase list (lexicon)

[6] `https://archive.org/`

Number of Languages	38
Number of QA pairs	578K
Number of Articles	294K
Number of Unique Questions	411K
Avg. Paragraph Length (Word)	17
Avg. Article Length (Word)	875
Avg. Question Length (Word)	6
Avg. Answer Length (Word)	165
Avg. Paragraph Length (Sentence)	1.4
Avg. Article Length (Sentence)	51
Avg. Question Length (Sentence)	1.0
Avg. Answer Length (Sentence)	9.5
Avg. Paragraphs per Answer	6.5
Avg. Paragraphs per Article	35

Table 5.2: Summary of MuNfQuAD dataset statistics. For multilingual sentence boundary detection, the ersatz toolkit [13] was used, and word tokenization was performed via whitespace splitting.

for each language to filter out interrogative subheadings that did not pertain to the article content. For instance, हे वाचलंत का? (*Did you read this?*) frequently appears as a subheading in Marathi news stories. Further illustrations are shown in Table 13 in Appendix A.5.3.

5.2.1 MuNfQuAD Statistics

The dataset introduced here contains more than 411,000 distinct question-answer pairs, making it the largest mQuAD to date. Table 5.2 summarizes various statistics about this resource. We found that over 75% of MuNfQuAD articles surpass the 512-token threshold imposed by standard multilingual encoders for factoid QA tasks [114, 290]. Section 5.4 elaborates on the computational limitations encountered by advanced multilingual encoders with expanded token capacities when handling MuNfQuAD articles. Figure 5.2 provides a comprehensive visualization of word counts across articles, questions, answers, and paragraphs.

We performed web scraping on the BBC news portal for every supported language, which enabled us to gather data in 38 out of the 43 available languages. Table 5.3 details the language-wise distribution and indicates the year of the earliest article included in MuNfQuAD. Due to the strong multilingual character of dataset, our experiments are limited to QA models that support multiple languages.

To analyze trends in n-grams, question types, and entity occurrences, every MuNfQuAD question was rendered into English. This translation utilized the *nllb-200-1.3B* model [258], which uniquely supports translation among 200 languages within a single framework. For extracting named entities from English questions, we used the spaCy toolkit. Table 5.5

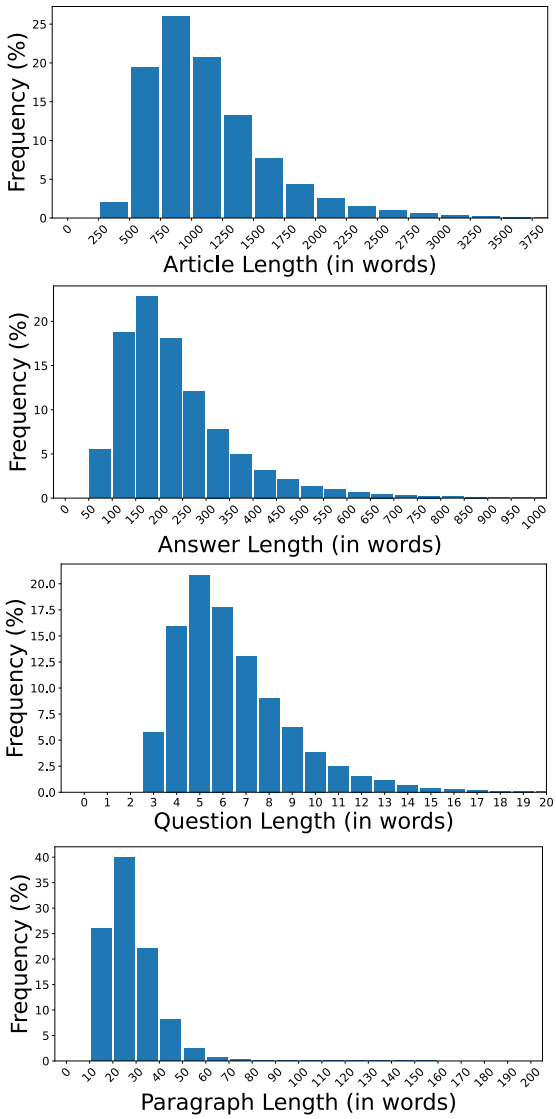


Figure 5.2: Distribution of word frequencies in MuNfQuAD.

and Table 5.4 present the principal results for the most frequent entities and n-grams, respectively. To assign each English question to a specific category, we applied the non-factoid question classifier from Bolotova et al. [5], which showed that over two-thirds of the dataset's questions are non-factoid. Figure 5.4 depicts the similarity in question category distributions between NLQuAD [4] and MuNfQuAD.

5.2.2 MuNfQuAD vs NLQuAD

A primary difference between these datasets is that MuNfQuAD provides sentence-segmented multilingual data annotated with question categories, a feature not present in NLQuAD. Additionally, we do not claim that BBC articles used as question contexts represent long-context data, since many LLMs now exist that can process texts longer than typical BBC

Region	Africa											Asia (Central)	
Lang.	Amharic	Oromo	Hausa	French*	Gahuza	Igbo	Somali	Pidgin**	Tigrinya	Swahili	Yoruba	Uzbek	Kyrgyz
Code	am	om	ha	fr*	rw	ig	so	en**	ti	sw	yo	uz	ky
#Articles	1.0k	3.3k	1.5k	2.1k	1.8k	1.1k	3.1k	3.4k	1.9k	3.3k	1.4k	1.6k	1.3k
#QA	2.0k	5.4k	2.8k	4.9k	3.1k	1.5k	5.5k	5.4k	3.2k	6.1k	1.9k	2.7k	2.3k
Start Year	2013	2017	2013	2012	2014	2018	2016	2017	2017	2010	2018	2010	2011
Region	Asia (Pacific)			Asia (South)									
Lang.	Korean	Indonesian	Vietnamese	Gujarati	Bengali	Marathi	Hindi	Pashto	Nepali	Sinhala	Punjabi	Telugu	Tamil
Code	ko	id	vi	gu	bn	mr	hi	ps	ne	si	pa	te	ta
#Articles	2.5k	5.5k	6k	8.2k	4.2k	10k	14k	1.9k	5.5k	1.7k	3.7k	7.3k	6.7k
#QA	4.6k	11k	10k	15k	7.9k	21k	26k	2.9k	11k	3.3k	7.6k	15k	12k
Start Year	2017	2010	2009	2017	2013	2017	2009	2010	2014	2012	2017	2017	2012
Region	Asia (South)	Europe							Latin America		Middle East		
Lang.	Urdu	Russian	Azeri	Turkce	Serbian	Cymrufyw	Ukrainian	English	Mundo	Portuguese	Persian	Arabic	
Code	ur	ru	az	tr	sr	cy	uk	en	es	pt	fa	ar	
#Articles	7.4k	12k	2.3k	8.7k	9.5k	2.8k	11k	6.9k	11k	5.4k	6.2k	6k	
#QA	14k	22k	4.3k	17k	21k	8.8k	19k	14k	20k	12k	11k	11k	
Start Year	2010	2010	2011	2009	2018	2012	2009	2011	2009	2011	2008	2009	

Table 5.3: Language breakdown in MuNfQuAD with ISO 639-1 identifiers. The dataset covers the years 2009 through 2024. For Persian and Pashto, a 621-year offset is applied to article dates, as speakers of these languages use the Solar Hijri calendar rather than the Gregorian. *African French **Pidgin English

Top 4-grams	Top 3-grams	Top 2-grams	Top 1-gram
what do we know ... (0.4%)	what is the ... (7%)	what is ... (12%)	what ... (38%)
what is the situation ... (0.4%)	what are the ... (2%)	who is ... (3%)	how ... (12%)
what happened to the ... (0.2%)	what does the ...(1%)	what did ... (3%)	why ... (8%)
what is going on ... (0.2%)	what did the .. (1%)	what are ... (3%)	who ... (6%)
what happened at the ... (0.2%)	how did the ... (1%)	how did ... (2%)	is ... (4%)

Table 5.4: Most common n-grams in English-translated MuNfQuAD questions indicate that descriptive forms (what/how) are predominant.

articles—models that were unavailable at the time NLQuAD was introduced.

5.3 Answer Paragraph Selection

Given segmented context paragraphs and an associated question, the APS model assigns elevated confidence values to paragraphs that are part of the silver answer set. The APS model receives as input the concatenation of the ith context paragraph (p_i) and the question. Its output is a probability between 0 and 1, reflecting how likely p_i is to answer the question.

The rationale for utilizing an APS model, rather than a sliding window Reading-Comprehension system [4], is that the APS approach is more consistent with the Answer Sentence Selection (AS2) methodology, which is considered more suitable than the RC method [291, 292].

Xu et al. [293] demonstrated the utility of the APS module in automatically responding to questions whose answers are distributed across disjoint text segments. As shown in Table 5.1, TyDi QuAD [102] also features a dedicated track for QA systems that use the APS

Most frequent			
Events (171)	Organizations (15K)	People (27K)	Countries (47K)
Afghan War (8%)	Taliban (5%)	Putin (4%)	India (7%)
Korean War (6%)	Congress (5%)	Trump (4%)	Russia (7%)
World War II (6%)	NATO (4%)	Biden (1%)	China (6%)
Tokyo Olympics (5%)	EU (3%)	Gandhi (1%)	Ukraine (7%)
Olympics (5%)	Supreme Court (2%)	Harry (0.6%)	USA (7%)

Table 5.5: Most frequent entities in English-translated MuNfQuAD questions are mainly from the Asian subcontinent, reflecting that 16 out of 38 MuNfQuAD languages are Asian.

framework. Our APS model is constructed by fine-tuning multilingual pretrained encoders. Figure 5.1(c,d) provides an overview of the training process and the architecture of our APS model. Suppose a news article contains q questions and p paragraphs. Each question generates p training samples, resulting in a total of pq training examples for that article. Consequently, our APS training set comprises over 100 million samples from MuNfQuAD. Based on Table 5.2, the combined token count of a question and paragraph remains within the 512-token boundary. The dataset was split into test, train, and development subsets in proportions of 0.1, 0.7, and 0.2, respectively. Due to the dataset's inherent class imbalance, we employed a weighted focal loss during training.

5.3.1 Implementation

Model fine-tuning was performed across five GPUs, with each GPU processing a batch size of 12. We experimented with several pretrained encoders, such as multilingual cased bert (mBERT) [44], XLM-Roberta-base (XLM-R) [294], multilingual-e5-base (mE5) [295], cased multilingual distilbert (d-mBERT) [296], XLM-Vocabulary-base (XLM-V) [297], multilingual LUKE (mLUKE) [298], and mT5 [299], as the backbone for our APS model. Additionally, the 560M parameter BLOOM model (bloom) [300] was also used as a text encoder. The fine-tuning head of the APS model consisted of three linear layers with a dropout rate of 0.2. Learning rates were set to 3e-3 for the fine-tuning layers and 1e-5 for the encoder, managed by a linear scheduler. All models were fine-tuned for a single epoch, which took between 25 and 33 hours. The transformers library [301] was used to incorporate pretrained transformer encoders, and PyTorch [302] was used to build the fine-tuning APS models.

5.3.2 Baselines

To establish baseline comparisons, we used the sentence-transformers library (sbert) [303] to produce vector representations for paragraphs (E_p) and questions (E_q). The sbert baseline utilized the paraphrase-multilingual-mpnet-base-v2 (mpnet) and paraphrase-multilingual-MiniLM-L12-v2 (miniLM) [304] as multilingual models. Another baseline involved generat-

APS Model	Acc.	Macro F1	Params.	Label 0			Label 1			SR
				recall	precision	F1	recall	precision	F1	
Ones	19	16	-	0	0	0	100	19	32	1
Zeros	81	45	-	100	81	90	0	0	0	0
Random	50	45	-	50	81	62	50	19	27	1
sbert (mpnet)	20	17	278M	1	87	2	99	19	32	0.99
Ours (mLUKE)	19	16	585M	0	100	0	100	19	32	1
Ours (bloom)	47	45	559M	39	90	54	81	24	37	0.92
sbert (miniLM)	22	20	117M	4	87	7	97	19	32	0.99
TF-IDF	81	47	-	99	81	89	2	36	4	0.11
Ours (mBERT)	74	67	177M	73	93	82	77	40	53	0.93
Ours (d-mBERT)	66	61	134M	63	93	75	79	33	47	0.93
Ours (mE5)	79	71	278M	81	92	86	69	46	55	0.9
Ours (mT5)	76	69	277M	76	93	84	74	42	54	0.91
Ours (XLM-R)	79	71	278M	80	93	86	73	46	**56**	0.91
Ours (XLM-V)	80	**72**	778M	83	92	87	68	48	**56**	0.9

Table 5.6: A comparison of different models' performance on the MuNfQuAD Test Set for the APS task. The Random, Ones, and Zeros models always predict random 0 or 1, always 1, or always 0, regardless of input. The APS model fine-tuned with the XLM-V encoder achieves the highest Label-1 F1 and macro F1 scores.

ing E_p and E_q by training a TF-IDF vectorizer with scikit-learn [208] on the training data. During preprocessing, stopwords and punctuation were stripped from each language[7]. In both baselines, the confidence score for a candidate paragraph was computed as the cosine similarity between E_p and E_q. For all models, the threshold was set at the midpoint of the possible confidence score range. Specifically, a default threshold of 0.5 was used for the TF-IDF baseline and the fine-tuned APS models, as their outputs range from 0 to 1. For the sbert baseline, which produces scores from -1 to 1, a threshold of 0.0 was used.

5.3.3 Evaluation

Paragraphs not part of the silver answer are assigned a ground truth label of 0, while those included in the silver answer are labeled as 1. Our analysis focuses on Label-1 metrics and macro F1, due to the significant class imbalance—only 23% of samples are Label-1. We also include the SR metric, which measures the proportion of questions for which at least one candidate paragraph overlaps with the reference answer paragraphs [305, 36]. A question is deemed successfully answered if there is at least one paragraph shared between the candidate and reference sets.

[7]We used `https://github.com/6/stopwords-json/` to obtain stopwords for multiple languages. For languages without available stopword lists, we selected the 260 most frequent words as stopwords, since the average number of stopwords across all languages is approximately 260.

Lang	#Qs	#Ann	IAA	Silver Labels vs Gold Labels					APS Models vs Gold Labels									
									Ours (XLM-V)					C4AI Command-R				
				Acc	F1 score			SR	Acc	F1 score			SR	Acc	F1 score			SR
					1	0	M			1	0	M			1	0	M	
hi	100	2	0.26	75	49	83	66	1	69	51	77	64	0.95	72	47	81	64	0.97
gu	100	2	0.42	83	55	90	72	1	72	49	81	65	0.95	69	43	79	61	0.98
bn	100	2	0.4	81	60	87	73	0.98	72	56	79	68	0.97	65	49	73	61	0.97
tm	100	2	0.78	85	51	91	71	0.98	69	38	79	59	0.94	72	40	82	61	0.91
te	40	2	0.58	84	50	91	70	0.95	76	48	85	66	0.97	54	30	66	48	1
pa	100	1	-	87	52	93	72	0.98	79	45	87	66	0.93	73	31	83	57	0.92
np	100	1	-	72	52	81	66	0.97	70	59	76	68	0.94	61	54	66	60	0.97
ur	50	1	-	83	59	89	74	1	70	49	79	64	0.97	61	43	71	57	1
Average				**81**	**53**	**88**	**70**	**0.98**	72	50	81	66	0.95	68	43	77	60	0.96

Table 5.7: Results of the top-performing APS model (from Table 5.6) and silver labels on the golden set. Languages are denoted using ISO 639-1 codes. Cohen's kappa is reported as the Inter Annotator Agreement (IAA) score. We do not compare the performance of our XLM-V based APS model with silver labels on the golden set, since Table 5.6 already presents model results on the full MuNfQuAD test set with silver labels. Silver labels achieve the highest SR, Label-1 F1 (1), Label-0 F1 (0), and Macro F1 (M) on gold labels.

5.4 Results

Given the large training set (100M samples), we performed hyperparameter optimization on a 1% subset. Our findings indicate that using a weighted focal loss with $\gamma = 2$ [306] produces better outcomes than alternatives. We also observed improved results when including preceding paragraphs along with the candidate paragraph. Thus, for each training example (T_i, $label_i$), the text (T_i) consists of (i) the question, (ii) preceding paragraphs, and (iii) the candidate paragraph. Only a subset of the preceding paragraphs is included, ensuring that T_i does not exceed 512 tokens for any APS model. Attempts to enhance performance by concatenating learnable position embeddings with contextual CLS token embeddings or by adding the article title to the question did not yield improvements on the smaller MuNfQuAD dataset. Ablation study results are provided in Table 14 in Appendix A.5.4. Table 5.6 shows that our APS model using XLM-V as the encoder achieves the highest macro F1 and Label-1 F1. Some other models obtain higher SR but lower macro F1. During test set inference, omitting questions classified as FACTOID has minimal impact on the performance of all fine-tuned APS models, as indicated in Table 15 in Appendix A.5.5.

5.4.1 Golden set

To test the hypothesis that "*paragraphs following an interrogative subheading contain its answer*", we engaged human annotators to answer questions from a MuNfQuAD subset, termed the golden set. Each annotator was given a question and its corresponding article (context), and asked to select paragraphs that could answer the question. Notably, annota-

tors did not have access to silver labels. The annotation process is described in Appendix A.5.1. Native speakers were recruited as annotators for each language. Annotators were compensated at 1 USD per eight questions. For languages with multiple annotators, the final gold answers were determined by taking the union of all selected paragraphs.

Table 5.7 shows that silver labels achieve a high Success Rate (~ 0.98) across languages, suggesting that the silver answer text in MuNfQuAD can address 98% of questions. The fact that silver labels outperform the best APS model indicates potential for further improvement in APS models. However, a Label-1 F1 of about 53 implies that roughly half of the paragraphs in the answer text do not actually answer the question. Additionally, many paragraphs outside the answer text are also capable of answering the questions. Interestingly, our XLM-V based APS model achieves higher Label-1 F1 and Macro-F1 on the Nepali golden set than the silver labels, demonstrating the model's ability to generalize from silver-labeled data. The sbert and TF-IDF baselines yield lower Label-1 F1 scores. More details on baseline performance on the golden set are available in Table 12 in Appendix A.5.2.

The relatively low Inter-Annotator Agreement (IAA) for Hindi (hi) is likely due to the higher number of answer paragraphs per question in this language. On average, Hindi questions in the golden set have 8.0 silver paragraphs, compared to 7.8 for Bengali (bn), 5.8 for Tamil (tm), 7.1 for Gujarati (gu), and 5.9 for Telugu (te).

5.4.2 Large Language Models

We also explored the use of LLMs as APS models. To ensure comparability with our fine-tuned APS models, we prompted LLMs with each context paragraph and the question, instructing them to return a binary value indicating whether the paragraph answers the question (1 for yes, 0 for no).

For open-weight LLMs, we used Llama-2-7b-chat [307], Mistral-7b [308], Llama-3.1-8B-Instruct [104], Airavata [106], gemma-2b/7b [105], BLOOM-7b [300], CohereForAI (C4AI) Command-R[8], and Aya-101 [309]. For proprietary LLMs, we used the ChatGPT API[9]. We found that running local LLMs requires significant inference time and large GPU memory. Moreover, using proprietary LLMs like ChatGPT incurs monetary costs[10]. Therefore, we restricted LLM baseline experiments to the golden set. Since mT5 [299] and bloom [300] are multilingual encoders with large input token limits, we also attempted to fine-tune RC models on the MuNfQuAD training set.

Our experiments show that even with 4-bit quantization and a batch size of one, a single RTX A6000 (48 GB) GPU is insufficient for fine-tuning RC models based on bloom or mT5 encoders. We observed that C4AI Command-R outperforms other LLMs as an APS model

[8] `https://huggingface.co/CohereForAI/c4ai-command-r-v01-4bit`

[9] gpt-3.5-turbo-1106 model with a 16K input token limit

[10] The estimated cost for zero-shot prompting with gpt-3.5-turbo-1106 on the full MuNfQuAD test set is approximately 250 USD.

on the golden set. Table 16 in Appendix A.5.6 provides detailed results for various LLMs on the MuNfQuAD golden set. However, as indicated in Table 5.7, our fine-tuned APS model still surpasses Command-R in performance. Additionally, our fine-tuned APS models are much more resource-efficient: the XLM-V model requires 13GB of GPU RAM, and the XLM-R model needs 5GB, both achieving inference speeds of 0.04 seconds per iteration. In contrast, Command-R needs 22GB of GPU RAM and takes 0.4 seconds per iteration. Furthermore, many local LLMs were unable to provide meaningful answers as either APS or abstractive QA models. Conversely, ChatGPT produced relevant outputs in both scenarios. Manual evaluation of ten abstractive answers showed that all adequately addressed the questions. Table 17 in Appendix A.5.7 presents example outputs for abstractive QA from different LLMs.

5.5 Discussion

It is observed that the TF-IDF baseline achieves a greater macro-F1 compared to the random baseline, suggesting that silver responses often include passages with significant lexical overlap with the posed question. For example, in the Hindi query बैठना नुकसानदेह क्यों? (*Why sitting is harmful?*), the silver response starts with आखिर बैठना इतना नुकसानदेह क्यों है? चलिए इसे समझने की कोशिश करते हैं. (*Why is sitting so bad after all? Let's try to understand it.*). Additionally, we find that both finetuned-mLUKE and sbert-based baselines show low recall for Label-0 and high recall for Label-1, implying that the selected threshold tends to classify most passages as answer passages. We assessed the threshold sensitivity of APS models. Interestingly, for varying thresholds, finetuned-XLM-V did not show any noticeable gains. As a result, we evaluated the next best-performing baseline, finetuned XLM-R. We found that the XLM-R-based APS model delivers optimal results at a threshold of 0.6, reaching an accuracy of 84 and a macro F1 of 73. Figure 5.3 illustrates the performance patterns of three different APS models: those utilizing TF-IDF, XLM-R, and sbert.

The standard threshold for each APS model was initialized at 50% of its output interval. Cosine similarity is used for both TF-IDF and SBERT approaches; however, since TF-IDF vectors are always non-negative, the cosine similarity between two TF-IDF representations ranges from 1 to 0. Conversely, for SBERT, the similarity score spans from -1 to 1. Our fine-tuned APS model applies a sigmoid activation at the output layer, producing values between 1 and 0. Thus, in Figure 5.3, the SBERT curve begins at zero, while the TF-IDF and APS model curves start at 0.5.

We evaluated the effectiveness of our APS model architecture by comparing its results to the baseline APS model introduced by Clark et al. [102]. After fine-tuning both models on the TyDi training set, our APS model achieved an average F1 of 0.73, whereas the baseline model reached 0.66.

Beyond serving as a resource, we hypothesize that MuNfQuAD's primary advantage lies in its use for fine-tuning APS models. Our experiments reveal that when the context

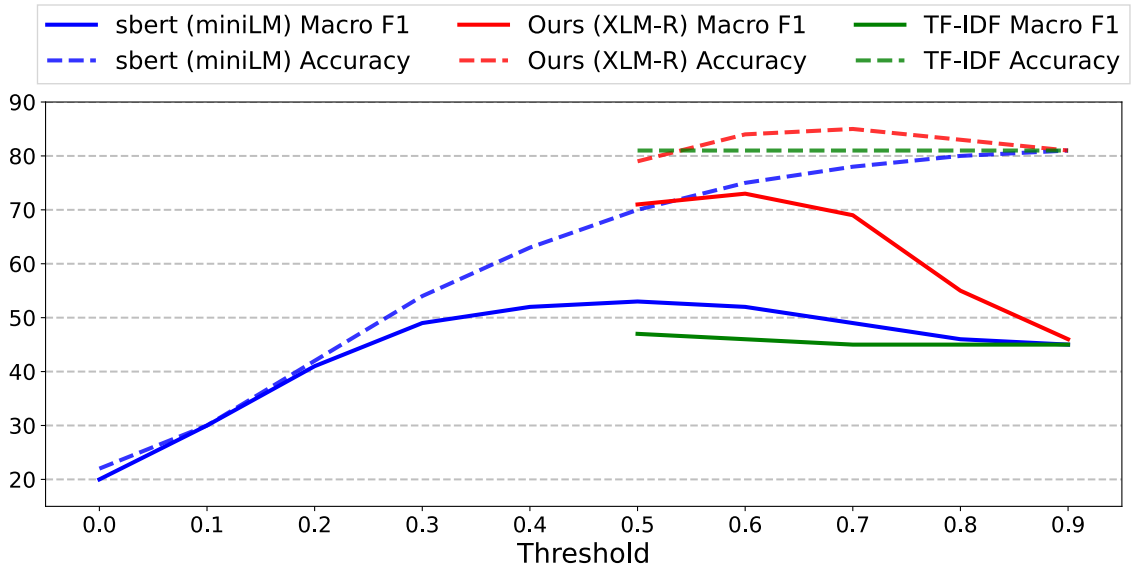


Figure 5.3: A visualization of the leading APS models' performance at various threshold settings on the MuNfQuAD test set. Note that only sbert (miniLM) initiates from a threshold of 0, as this is the default for the sbert-based baseline. In contrast, TF-IDF and XLM-R have default thresholds set at 0.5. The XLM-R-based model consistently achieves the highest macro F1 scores across different thresholds. Notably, the fine-tuned XLM-R APS model attains its best results at a threshold of 0.6, with an accuracy of 84 and a macro F1 of 73.

for a question is reduced using the scores from the MuNfQuAD-fine-tuned APS model, the performance of instruction-tuned LLMs, such as Gemma-7B and Gemma-2B [105], improves. This enhancement is evident across samples from the evaluation splits of XQA [281] and TyDi [102].

Given that XQA and TyDi are factoid-oriented QA datasets, we define the win ratio as the fraction of cases where the gold answer is present in the generated output, relative to the total number of generations. Our analysis indicates that, on the TyDi evaluation set, the win ratio for Gemma-2B increases from 11% to 14% with the APS model fine-tuned on MuNfQuAD, and for Gemma-7B from 19% to 25%. Conversely, on the XQA evaluation set, Gemma-2B's win ratio rises from 1% to 6%, and Gemma-7B's from 3% to 4%. The relatively weaker results of LLMs on XQA are anticipated, as XQA is a cloze-style QA dataset. We conducted a qualitative review of nine questions where the silver and gold answers did not share any common passages. It was found that in four cases, the silver responses failed to adequately address the question. For example, for the Telugu query ఫేస్‌బుక్ డేటాను ఎలా దుర్వినియోగం చేశారు? (*How was Facebook data misused?*), the answer was located within an image rather than the relevant passage. Similarly, for the Gujarati question વાડ઼િર અન્યદેર કી કરતે હવે? (*What should the others in the house do?*), the silver passage answered *What should you do for others in the house?* instead. Of the nine questions, three were very short ($\sim$ 3 words), and their broad scope makes it difficult to answer them without the article's

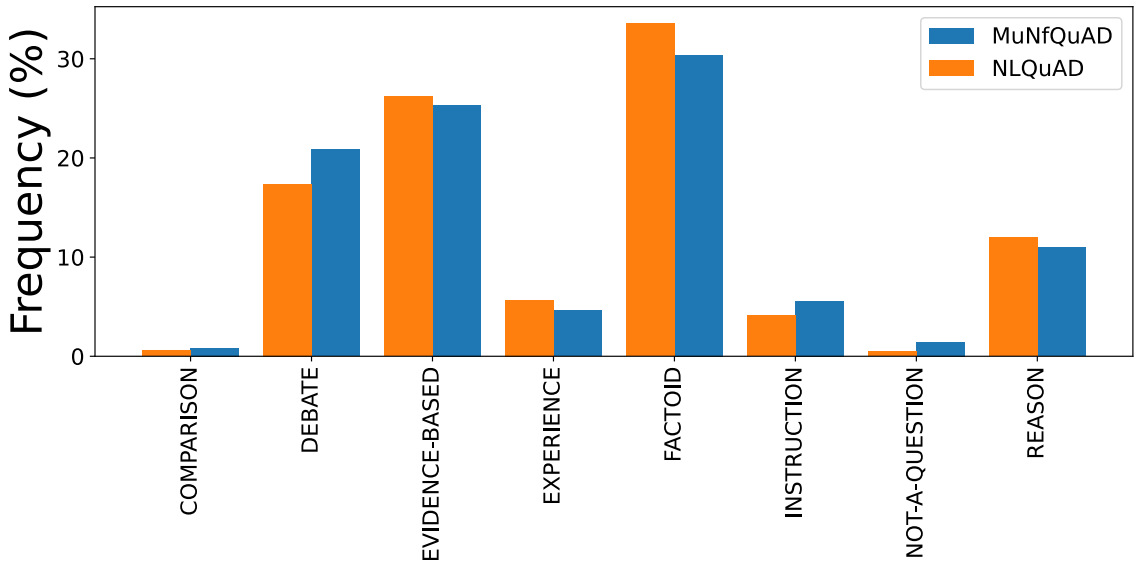


Figure 5.4: A comparison of the distribution of question types in the proposed English NLQuAD [4] and MuNfQuAD, based on predictions from the question classifier developed by Bolotova et al. [5].

context. For instance, the Tamil question எப்போது என்ன நடந்தது? (*When and what happened?*) is so general that it could yield different answers if asked independently of the article. The same applies to the Punjabi question ਮੈਨੂੰ ਕੀ ਪਤਾ ਲੱਗਿਆ? (*What did I find out?*). The last two questions were from interviews, making them hard to answer without a full understanding of the conversation's context. For example, तपाईको भनाइको अर्थ (नेपालसंग पछिल्लो सम्झौता गर्दा) चीनले भारतको पनि संवदेनशीलतालाई विचार गरेको थियो? (*Does your statement mean (while making the latest agreement with Nepal) that China also considered India's sensitivity?*) was part of a discussion about Nepal's foreign relations.

We attempted to fine-tune the smallest BLOOM model [300], the only multilingual encoder with a token capacity above 512, as an extractive Reading-Comprehension model on MuNfQuAD. However, our experiments showed that the task surpassed the memory limits of the RTX A6000 (48 GB), even with a batch size of one.

5.6 MuNfQuAD Limitations

The analysis by Latham [310] demonstrated that BBC tends to have a left-of-center perspective in its reporting. Consequently, we acknowledge that MuNfQuAD may reflect a similar political leaning. Although a high Success Rate for silver responses suggests their effectiveness in answering the associated questions, the relatively lower F1 for Label 1 indicates that silver labels in MuNfQuAD are not always succinct or exhaustive. That is, silver responses address the question but may include extra details. Furthermore, about one-third of MuNfQuAD's questions are labeled as FACTOID, suggesting that brief answers might

exist within the silver passages. It is important to interpret the question categories carefully, mainly for two reasons: (a) The classifier is not flawless. We applied the same classifier to the English non-factoid QuAD [4] and observed a similar distribution of question types, as shown in Figure 5.4. (b) The classifier processes automatically translated English questions, which can result in awkward translations that affect its predictions. For example, the Hindi question अगर नहीं किया तो क्या होगा? was auto-translated as *What if I didn't?*, and the classifier labeled it as FACTOID. However, a more accurate translation would be *What will happen if not done?*, for which the predicted category is EVIDENCE-BASED. Since this work mainly introduces a new resource and a preliminary baseline, the innovation in data collection and APS model design is limited.

The broad range of IAA values across languages highlights the subjective nature of annotation for some languages. It should be noted that, due to the high costs of annotation, we chose native speakers rather than professional annotators, which may affect the reliability of the gold standard set.

As described in Section 5.3, fine-tuning each APS model takes one or two days, which is why results from multiple runs are not included in this study. Additionally, the number of available GPUs limits the fine-tuning process for APS models on MuNfQuAD. In another experiment, we fine-tuned our XLM-R-based APS model using a single GTX 2080 for 1% of the total steps in an epoch. Our three runs showed that the process took about 77±2 minutes. Extrapolating from this, we estimate that completing a full epoch would require roughly 128 hours on a single GPU.

Moreover, the computational requirements for bloom, mLUKE, and XLM-V were even higher, needing at least 16GB, 12GB, and 22GB of GPU memory, respectively. The time to finish one epoch was 30, 8, and 13 days, respectively.

Accessing ChatGPT's proprietary models requires paid API keys. Our experiments with ChatGPT 3.5 on the gold set cost 16 USD. For more advanced models like ChatGPT 4, the cost would have been approximately ten times higher.

The results from LLM experiments should be interpreted with caution, as pre-training may have exposed LLMs to BBC news content. Thus, as pointed out by Ahuja et al. [311], there is a risk of dataset contamination.

5.7 Ethical Considerations

Over a period of six months, data extraction was performed, implementing appropriate intervals between each collected article to avoid causing any Denial of Service (DoS) issues for website visitors.

Our intention is to make MuNfQuAD accessible for academic, non-profit research purposes. We have secured approval from the BBC to support this project. Within scholarly circles, there are prominent examples where researchers have distributed BBC news content for similar research aims in the fields of news summarization [312] and question answering

[4].

5.8 Chapter Summary

English QA has become a well-established task, supported by a wide array of resources and systems for responding to fact-based queries. However, there has been considerable growth in non-factoid QA. Our research emphasizes the necessity for multilingual datasets in this area. To address this, we present MuNfQuAD, a multilingual QA resource designed to bridge this gap. MuNfQuAD features non-factoid queries and covers 38 different languages, thereby addressing a significant need.

The dataset was assembled by collecting BBC news stories. Questions are extracted from interrogative headings, and the following paragraphs are used as their respective silver responses. It is noteworthy that the majority of MuNfQuAD's articles are centered on the Asian subcontinent. When compared to a hand-annotated golden dataset, it is shown that almost all silver responses are suitable for answering the posed questions. Furthermore, our customized APS model, trained on MuNfQuAD, achieves a high Success Rate for both silver (0.91) and golden (0.96) annotations. These findings indicate that training the APS model with silver annotations can successfully generalize to certain languages present in the golden dataset.

Chapter 6

Explaining Finetuned Transformers on Hate Speech Predictions using Layerwise Relevance Propagation

QA has advanced significantly, particularly for factoid-based tasks. However, non-factoid QA, as exemplified by MuNfQuAD, necessitates enhanced explainability. This chapter compares three post-hoc XAI methods to identify the most suitable approach for interpreting Answer Paragraph Retrieval in our future work. Using a hate speech benchmark with annotated rationales, we evaluate each method's performance. Furthermore, we provide an intuitive explanation of gradient utilization in Layer-wise Relevance Propagation (LRP) and discuss its limitations. Our goal is to determine the most effective XAI technique for improving the transparency of QA frameworks[1].

Disclaimer: In this chapter, quotations of text deemed offensive or objectionable are incorporated to demonstrate the operation of the model, though the authors' opinions are not mirrored. Online harassment and offensive language are condemned by the authors.

Elephant seal [314].

[1]Content of this chapter is adapted from our paper published in the Big Data and Artificial Intelligence Conference, 2023 [313]

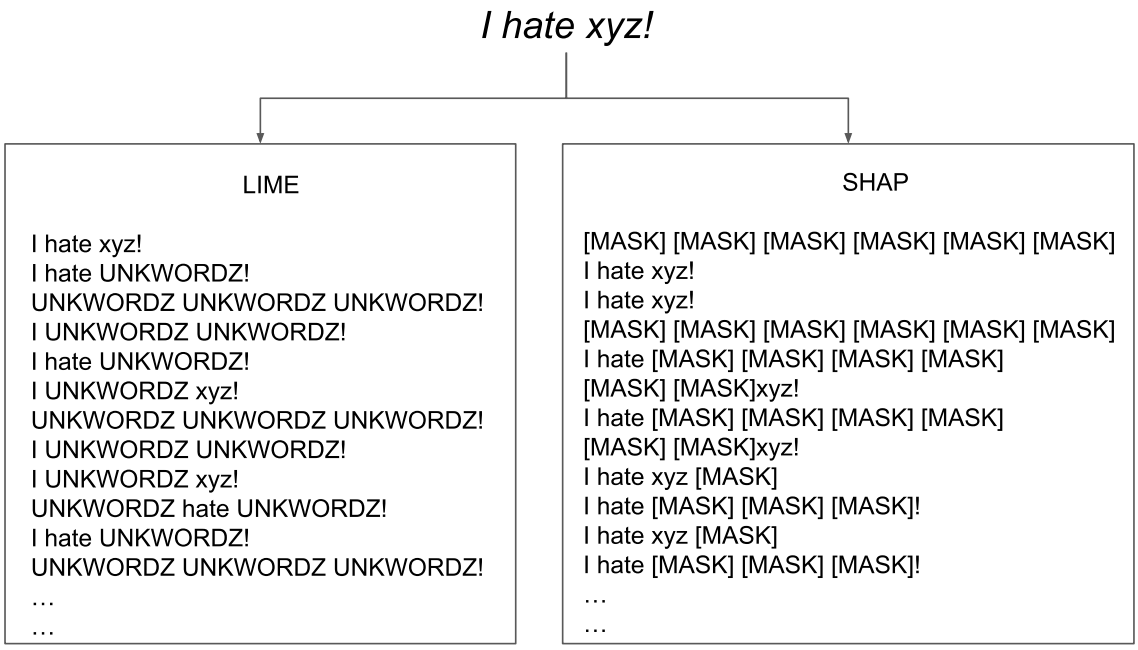


Figure 6.1: A comparison of text perturbation strategies in LIME and SHAP is depicted. LIME applies word-level masking to the input string using the `UNKWORDZ` placeholder, while SHAP performs subword-level masking with the `[MASK]` token. In SHAP, each altered input is evaluated by a fine-tuned model, and the importance of subwords is inferred from the resulting prediction changes. In contrast, LIME fits a linear surrogate model to the perturbed samples and their outputs, then assigns importance scores to each word based on this model.

Neural networks have achieved broad adoption across a range of domains, including tasks like natural language understanding, audio processing, and visual recognition. Despite their extensive use, Deep Neural Networks are often criticized for their opaque nature, which makes it challenging to interpret the reasoning behind their outputs [315]. Moreover, these systems have been reported to display biases against certain demographic groups, such as particular races [316], genders [317, 318], or ethnic backgrounds [319].

To assess the performance of various post-hoc interpretability approaches, we utilized the HateXplain dataset [320]. This dataset provides not only ground-truth class annotations but also human-annotated rationales. These gold labels are binary judgments from annotators, indicating whether a sample contains hate speech (1) or does not (0). The aim of hate speech classification is to assign a score near 1 if hate speech is present, and near 0 otherwise. Gold rationales are only available for samples labeled as hate speech by annotators. As first described by [321], a gold rationale is a binary vector whose length equals the number of tokens in the sample. Therefore, the central research question addressed in this work is: *Relative to alternative post-hoc approaches, what level of interpretability does Layer-wise Relevance Propagation (LRP) offer on the HateXplain dataset?*

This study presents a comparative analysis of three XAI methods: Local Interpretable Model-agnostic Explanations (LIME) as introduced by [148], SHapley Additive exPlanations (SHAP) as proposed by [149], and LRP as described by [322]. These methods were selected for their ability to generate post-hoc explanations (rationales) for models trained without explicit gold *rationales*.

LIME constructs a linear surrogate model by generating perturbed samples around the original input. This surrogate is then used to estimate the importance of each input feature. In contrast, SHAP computes Shapley values (as a proxy for feature importance) for each input feature using partial dependence plots. In game-theoretic terms, a Shapley value quantifies the contribution of a feature by measuring the change in model output when the feature is included or excluded from the input. Both methods modify the input text to compute feature importances. LIME treats words as features, while SHAP considers subwords when the input is text and the model is a fine-tuned transformer. Figure 6.1 illustrates the distinct approaches to text perturbation in LIME and SHAP.

To our knowledge, no previous work has evaluated the interpretability of transformer-based hate speech classifiers on the HateXplain dataset using LRP or SHAP. Our research is the first to systematically compare these three XAI methods on the HateXplain dataset, and we implemented the LRP approach as part of this investigation.

This work focuses on hate speech detection models that utilize pretrained transformer-based encoders (hereafter, *encoders*) to generate text representations. These models have become prominent due to their state-of-the-art results in hate speech classification [323, 324]. Additionally, the importance of interpretability in transformer-based models has been highlighted in several publications [325, 317, 326]. To the best of our knowledge, LRP has not previously been applied to hate speech classifiers leveraging pretrained *encoders*.

The LRP algorithm operates by propagating relevance scores backward through the network. Specifically, relevance is assigned at the output layer of the hate speech classifier and then distributed backward through the layers. The relevance for a node j in layer L is computed based on: (a) the relevance of all nodes in the next layer $(L + 1)$, (b) the learned weights connecting L and $L+1$, and (c) the activations in layer L. A comprehensive overview of LRP can be found in [6].

The broader impacts of XAI have been discussed extensively in prior literature. The comprehensive review by [315] summarizes key applications of XAI. Furthermore, by empirically comparing two XAI methods, guidance is provided for future researchers on selecting appropriate techniques for specific tasks. Our work aims to determine which method yields more informative rationales and to share the insights obtained from these explanations.

We have released our implementation of LRP for transformer-based text classifiers, which produces word-level rationales for predicted classes in multi-class settings[2]. To our knowledge, this is the first publicly available implementation of LRP for fine-tuned transformer models.

[2]https://github.com/ritwikmishra/hateXplain-metrics-calculation

```
>>> from transformers import AutoTokenizer
>>> tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")
>>> print(tokenizer.tokenize('I hate xyz!'))
['I', 'hate', 'x', '##y', '##z', '!']
>>> print(tokenizer('I hate xyz!',add_special_tokens=False).word_ids())
[0, 1, 2, 2, 2, 3]
```

Figure 6.2: Python code demonstrates the output of the `.word_ids()` method from Huggingface (transformers) tokenizers. For the sample input *I hate xyz!*, the word *xyz* is split into three subword tokens: *x*, *##y*, and *##z*. As a result, the index 2 appears three times in the output of `.word_ids()`.

6.1 Related Work

LIME has been frequently adopted in prior studies to interpret hate speech classification results [327, 328]. In contrast, LRP has seen limited use for this purpose. Karim et al. [329] applied LRP to explain hate speech predictions in the *Bengali* language, but their approach was based on Long Short-Term Memory (LSTM) networks, a type of Recurrent Neural Network (RNN). Arras et al. [330] previously used LRP to interpret sentiment predictions from an RNN. Similarly, LRP has demonstrated its utility in intent classification with Bidirectional LSTMs [331]. For analyzing the influence of source and target tokens in translation, LRP has been applied to transformer-based Neural Machine Translation (NMT) systems [332]. However, as far as we are aware, no earlier work has implemented LRP to interpret predictions from a fine-tuned transformer-based classifier.

Our selection of LRP in this study is motivated by its ability to provide post-hoc explanations for model outputs. Additionally, LRP has been shown to yield meaningful explanations in tasks such as question classification and semantic role labeling [333]. Relevance scores from LRP have also been used to enhance pretrained word embeddings [334]. Furthermore, LRP has been utilized in Layerwise Relevance Visualization (LRV) for sentence classification models based on Graph Convolutional Networks (GCN) [335].

Numerous studies have compared the effectiveness of multiple XAI methods. In the context of disease diagnosis [336], SHAP [**?**] have been found to provide more informative explanations than LIME. However, LIME rationales have been observed to be more stable than those from SHAP for various financial models [337].

Balkir et al. [338] introduced *necessity* and *sufficiency* metrics to interpret hate speech classifier predictions. Comparing LIME and SHAP using these metrics, they found that LIME struggled to generate relevant rationales for false positives. In a sentiment analysis context, Jørgensen et al. [339] compared rationales from SHAP and LIME, observing that SHAP was more effective at identifying relevant text spans, while LIME rationales more closely matched human word rankings.

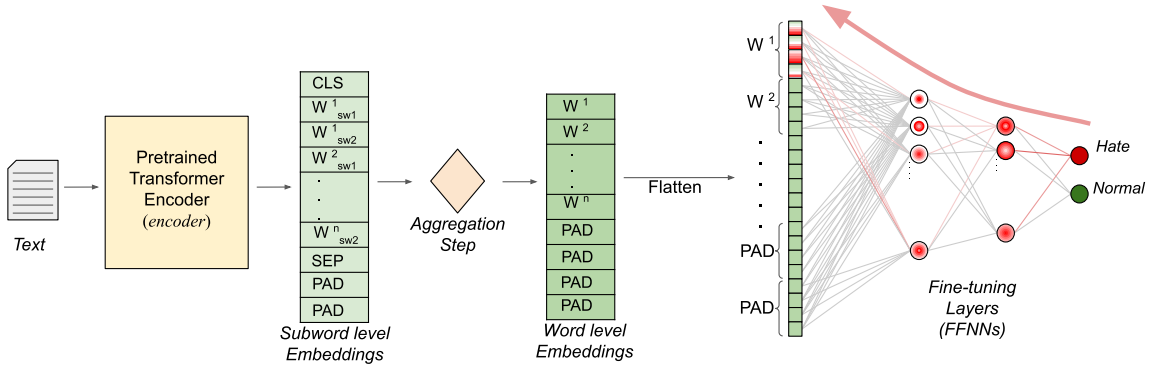


Figure 6.3: The architecture of a hate speech classifier and the process of relevance computation via LRP are depicted. The confidence for the *Hate* class is marked in red. The predicted output is used as the initial relevance at the output layer and is then propagated backward through the fine-tuning layers (indicated by the red arrow), following the methodology described in [6].

6.2 Experimental Setup

We employed the *transformers* library [204] to load various *encoders* in our experiments. LIME and SHAP rationales were generated using the Ferret [340] toolkit, which builds upon the official implementations and supports subword-level explanations. Word-level rationales were derived from subword-level outputs using the Huggingface tokenizer's `.word_ids()` method. Figure 6.2 displays the output of this function. Since no existing tool was available, we implemented several LRP variants for the fine-tuning layers of the hate speech classifier. We verified that the total relevance remains conserved across all layers during backpropagation to ensure the correctness of our LRP implementation.

We obtained word-level embeddings by averaging the corresponding subword embeddings, as shown in Figure 6.3. This approach was necessary because ground-truth rationales are provided at the word level, not the subword level. Embeddings for special tokens (such as CLS and SEP), which mark sentence boundaries, were excluded since no ground-truth rationales exist for these tokens.

The flattened word-level embeddings are then passed through the fine-tuning layers, which consist of several Feed-forward Neural Networks (FFNNs). We used three linear layers with a dropout of 0.1 and relu activations between them. To obtain the relevance for each word, the relevance scores of the flattened word embeddings are summed. In this work, relevance is propagated only up to the input layer of the fine-tuning module, as propagating through the transformer layers (which include multi-head attention) is non-trivial. We implemented the following LRP variants, based on the formulations by Montavon et al. [6].

- **LRP-0**: This variant redistributes relevance in proportion to each input's contribution

to neuron activations. The formula is:

$$R_j = \sum_k \frac{a_j w_{j,k}}{\sum_{j=0}^{j} a_j w_{j,k}} R_k \tag{6.1}$$

where k indexes the nodes in layer $L+1$.

- **LRP-ϵ**: This approach extends LRP-0 by adding a small positive constant ϵ to the denominator:

$$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_0^j a_j w_{jk}} R_k \tag{6.2}$$

The ϵ term helps stabilize the relevance assignment when neuron k's activations are weak or noisy.

- **LRP-γ**: This variant enhances LRP-0 by incorporating a positive parameter γ in the denominator:

$$R_j = \sum_k \frac{a_j.(w_{jk} + \gamma w_{jk}^+)}{\sum_0^j a_j.(w_{jk} + \gamma w_{jk}^+)} \cdot R_k \tag{6.3}$$

The γ parameter controls the preference for positive contributions. As γ increases, negative contributions are diminished.

Montavon et al. [6] proposed using gradients to compute relevance scores. Their general rule for calculating the relevance of node j in layer L is:

$$R_j = \sum_k \frac{a_j.\rho(w_{jk})}{\epsilon + \sum_0^j a_j.\rho(w_{jk})} R_k \tag{6.4}$$

The relevance backpropagation proceeds through these four steps:

- Forward computation: $z_k = \epsilon + \sum_0^j a_j.\rho(w_{jk})$
- Division: $s_k = R_k / z_k$
- Backward computation: $c_j = \sum_k \rho(w_{jk}).s_k$
- Relevance assignment: $R_j = a_j c_j$

According to Montavon et al. [6], c_j may be represented through the gradients of a, specifically $c_j =$`a.grad`. We aim to clarify the reasoning for this. To make the explanation more straightforward, let us suppose $\rho(w_{jk}) = w$.

1. We have $c_j = w \cdot s$, where $dim(w) = (j, k)$ and $dim(s) = (k, 1)$.
2. We know that `z = w.forward(a)`, which corresponds to $w^T \cdot a$, with $dim(a) = (j, 1)$. Therefore, $dim(z) = (k, 1)$.
3. It is established that $s = R_k / z$, where $dim(R_k) = (k, 1)$. Thus, $dim(s) = (k, 1)$.

4. When gradients are utilized, `z*s.data` can be written as $t = (w^T \cdot a) \cdot s$, with $dim(t) = (k, 1)$.

5. If t is summed and gradients are backpropagated using `.backward()`, both w and a receive gradients.

6. `a.grad` is the partial derivative of t with respect to a. Therefore, `a.grad`$= \frac{\partial t}{\partial a} = w \cdot s = c_j$.

The deliberate use of `s.data` by the authors is meant to prevent gradients from flowing into a through an alternative route, since s depends on a. However, if biases are present in the neural network layers, equation 6.4 will include bias terms at both the input and output. As differentiation ignores biases, the gradient assigned to a remains $w \cdot s$. Nevertheless, to satisfy the LRP conservation property, which requires the sum of relevance scores to be preserved across layers, the backward computation of c_j must be modified as follows:

$$c_j = \sum_k \left(\rho(w_{jk}) + \frac{b_k}{|j|a_j} \right) \cdot s_k$$

Therefore, using `a.grad` will not yield c_j when bias is active in fully-connected layers. Our LRP implementation accommodates both scenarios, with or without bias, in the fine-tuning layers.

For all three interpretability techniques, a token is marked as relevant (1) or not relevant (0) based on whether its computed relevance surpasses a threshold. Following [320], a threshold of 0.5 is adopted for every method evaluated in this work.

For empirical evaluation, two distinct *encoders* were employed: *bert-base-cased* (BERT) from [44] and *roberta-base* (RoBERTA) from [87]. These encoders were chosen due to their widespread use in English hate speech classification tasks [323, 324]. To assess the effect of the encoder on rationale extraction, our hate speech classifier was trained with the encoder either frozen or allowed to update its parameters. All models were trained for 10 epochs, using a learning rate of 5e-7 for the *encoder* and 1e-6 for the fine-tuning layers. None of the models showed indications of overfitting or underfitting.

The explainability metrics in this research are divided into two groups: (a) Plausibility and (b) Faithfulness. Plausibility evaluates how closely the predicted rationales align with human-annotated gold rationales, while faithfulness does not rely on gold rationales and instead measures the model's sensitivity to the predicted rationales. Under plausibility, metrics such as Intersection over Union (IOU) F1, Token F1, and Area Under the Precision Recall Curve (AUPRC) are utilized. Conversely, faithfulness is assessed using comprehensiveness and sufficiency. Notably, for sufficiency, lower values are considered better. For comprehensive definitions of these explainability metrics, please consult the Hatexplain paper [320].

Encoder	Method	Performance		Explainability				
				Plausibility			Faithfulness	
		Accuracy↑	Macro-F1↑	IOU F1↑	Token F1↑	AUPRC↑	Compr.↑	Suff.↓
BERT_{f-t}	LRP	74 (74±0)	72 (71.7±0.6)	0.10 (0.11±0.0)	0.17 (0.178±0.0)	0.45 (0.469±0.01)	0.11 (0.12±0.0)	0.22 (0.217±0.01)
	LIME			**0.26 (0.25±0.01)**	**0.30 (0.30±0.0)**	**0.64 (0.64±0.0)**	**0.30 (0.29±0.01)**	0.09 (0.11±0.01)
	SHAP			**0.26 (0.26±0.01)**	**0.30 (0.30±0.01)**	**0.64 (0.64±0.0)**	0.29 (0.29±0.01)	0.09 (0.11±0.01)
BERT_{fr}	LRP	67 (67.6±1.1)	63 (63.7±1.2)	0.13 (0.13±0.0)	0.23 (0.23±0.0)	0.49 (0.50±0.01)	0.07 (0.08±0.01)	0.09 (0.10±0.01)
	LIME			0.14 (0.15±0.01)	0.24 (0.25±0.01)	0.54 (0.54±0.0)	0.09 (0.10±0.01)	0.08 (0.09±0.01)
	SHAP			0.17 (0.18±0.0)	0.26 (0.27±0.01)	0.56 (0.57±0.01)	0.10 (0.10±0.0)	0.08 (0.09±0.01)
RoBERTA_{f-t}	LRP	**75 (75.7±0.6)**	**73 (73.7±0.6)**	0.11 (0.12±0.01)	0.17 (0.18±0.02)	0.46 (0.47±0.01)	0.11 (0.11±0.0)	0.27 (0.26±0.01)
	LIME			0.24 (0.24±0.0)	0.27 (0.27±0.0)	0.61 (0.61±0.0)	0.07 (0.07±0.0)	0.07 (0.07±0.0)
	SHAP			0.24 (0.23±0.01)	0.26 (0.26±0.0)	0.61 (0.61±0.0)	0.07 (0.07±0.0)	0.06 (0.06±0.0)
RoBERTA_{fr}	LRP	67 (67.3±0.6)	63 (63.7±1.2)	0.13 (0.13±0.0)	0.23 (0.23±0.0)	0.49 (0.48±0.01)	0.05 (0.05±0.0)	0.06 (0.06±0.0)
	LIME			0.15 (0.15±0.0)	0.24 (0.24±0.0)	0.55 (0.55±0.0)	0.0 (0.01±0.01)	**0.01 (0.01±0)**
	SHAP			0.16 (0.16±0.0)	0.25 (0.26±0.01)	0.55 (0.56±0.01)	0.0 (0.0±0.0)	**0.01 (0.01±0.0)**
Random		50	50	0.10	0.23	0.47	0.27	0.26

Table 6.1: This table presents a comparative analysis of several post-hoc rationale extraction techniques (LIME, SHAP, and LRP) applied to hate speech classification models utilizing different *encoders*. Architectures where the encoder was updated during training are marked with the subscript $f-t$, while those with encoders kept static are indicated by the subscript fr. Results are reported on the official test partition of the HateXplain dataset. The average and standard deviation across 3-fold cross-validation are shown in parentheses. An upward arrow denotes that higher values are desirable, while a downward arrow signifies that lower values are preferable. The highest performance metrics are attained by fine-tuned RoBERTA, whereas fine-tuned BERT achieves the top explainability metrics.

6.3 Results

Table 6.1 displays the performance of multiple encoders under distinct training regimes. Our study found minimal variation in effectiveness among the different LRP variants. Therefore, only results for the LRP-0 variant are included in this work. Outcomes for other LRP variants are provided in Appendix A.6.

Our findings indicate that LIME rationales perform similarly to SHAP on nearly all explainability measures. Additionally, the interpretability of LRP rationales on fine-tuned

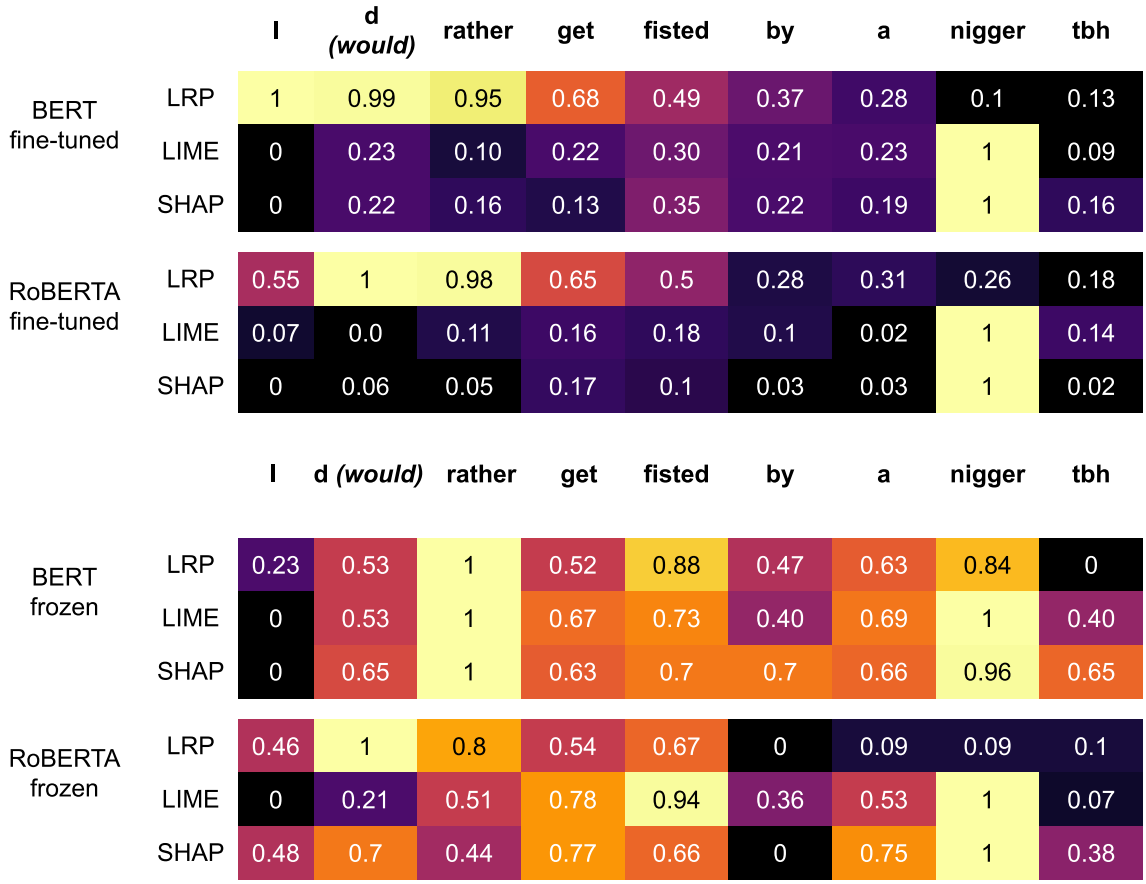


Figure 6.4: Predicted rationales from various methods for a sample from the HateXplain test set are depicted. LRP rationales on fine-tuned *encoders* tend to assign high relevance to the initial tokens, while LIME and SHAP emphasize the offensive term.

encoders was lower than that of a random rationale generator. It was also noted that while LIME and SHAP plausibility scores decrease for models with frozen encoders, LRP plausibility scores increase.

A qualitative review of rationales generated by LRP, LIME, and SHAP for fine-tuned *encoders* revealed that LIME and SHAP assign high relevance to offensive words, whereas LRP highlights the sentence's initial tokens. Figure 6.4 visualizes the relevance scores from different methods for a sentence classified as hate speech by all models with fine-tuned *encoders* and by human annotators in the HateXplain dataset.

Since LRP relevance values indicate the contribution of each node to the prediction, high relevance for the first tokens suggests that their embeddings are the primary drivers of the model's output. To evaluate the importance of a specific token's embedding (i), embeddings for all other tokens ($\{0, 299\} - \{i\}$) were zeroed out. Figure 6.5 demonstrates the resulting accuracy drop for models with fine-tuned *encoders*. The results suggest that, during training, fine-tuned encoders learn to encode most sentence information within the embeddings of the first few tokens.

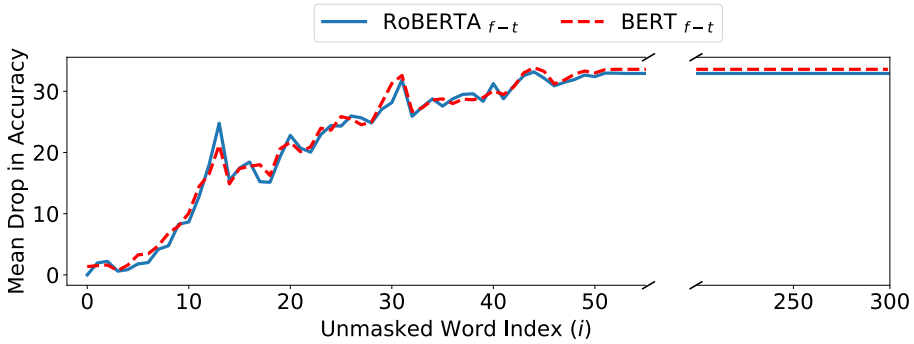


Figure 6.5: The average reduction in accuracy across 3-fold cross-validation is shown when only the embedding of the ith word is provided to the fine-tuning layers, with all other embeddings set to zero. The plot shows that accuracy loss is minimal (y-axis $\rightarrow$ 0) when embeddings from the first few tokens (x-axis $\rightarrow$ 0) are used as input to the fine-tuning layers.

Among sentences identified as hate speech, LRP marks the first token as relevant in 75% of cases[3] for the BERT$_{f-t}$ model, and in 86% for the RoBERTA$_{f-t}$ model.

6.4 Limitations of this work

The LRP analysis conducted here is limited to the fine-tuning layers of transformer models; thus, this study cannot fully explain why sentence information is concentrated in the embeddings of the initial tokens. Furthermore, our evaluation of various explanation techniques is restricted to hate speech detection using a single benchmark dataset, and our conclusions are therefore specific to this context. Additional research is necessary to generalize these findings to other domains or datasets.

6.5 Chapter Summary

Transformer-based models have achieved state-of-the-art results across a wide range of tasks, from vision to language. However, because these models extract features in an integrated fashion, interpretability in such deep neural networks is especially important. In this chapter, we attempted to assess the explanatory power of the LRP approach using different *encoders* for hate speech classification. We observed that LIME performs on par with SHAP for most interpretability metrics on the HateXplain benchmark. However, LRP rationales led us to conclude that fine-tuning a pre-trained transformer *encoder* produces a model where nearly all textual information is embedded in the first few tokens. As a result, LRP relevance scores up to the fine-tuning layers may not accurately reflect the true semantic rationale for the model's prediction. Nonetheless, fine-tuned *encoders* appear to exhibit a form of feature

[3] Averaged across the three test splits in 3-fold cross-validation.

space compression, which could be leveraged to validate text visualizations based on the embeddings of early tokens.

Chapter 7

Long-context Non-factoid Question Answering in Indic Languages

We build upon the preceding chapters' exploration of QA in Indic languages, addressing the challenges of long-context understanding. We have highlighted the scarcity of resources and the need for efficient techniques in this domain, particularly for non-factoid QA. We have seen the explainability limitations of transformer-based models. Motivated by these observations, this chapter investigates context-shortening techniques to enhance QA performance in low-resource Indic languages. We explore OIE, coreference resolution, APS, and their combinations. We evaluate these techniques on three LLMs across four Indic languages, analyzing improvements in semantic and token-level scores, computational efficiency, and explainability. We also discuss the limitations of LLMs in handling complex non-factoid questions and the impact of verbalizing OIE triples[1].

Deity fighting two tigers on seal [342].

[1]Content of this chapter is adapted from our paper submitted in the 13th International Conference on Big Data & AI [341]

QA [23] constitutes a core challenge within NLP. The objective of QA is to automatically produce or retrieve a suitable response for a given query.

The exploration of QA is vital, as it provides an effective avenue for training systems to comprehend nuances across multiple modalities, such as text [26], video [27], and audio [28], through question-answer exchanges. This research is particularly concentrated on the textual modality.

This study prioritizes non-factoid questions, which require elaborate and detailed answers. For instance, a non-factoid question could be, "*How did the people of Agra in the 17th century view the construction of the Taj Mahal?*" Although these questions are important, non-factoid QA has received relatively limited attention since the early days of QA research [59]. Non-factoid QA systems are crucial for applications like generating responses for voice assistants such as Amazon Alexa [60]. Additionally, non-factoid questions are prevalent on online forums [61], and Bing search queries often require comprehensive, in-depth answers [343].

Non-factoid queries create opportunities for building language models that can handle lengthy sequences [4]. These questions typically require answers that span several passages, yet only a small body of research has thoroughly addressed this aspect [57]. Transformer-based architectures are commonly used in QA system development [291, 344]. However, the self-attention mechanism in Transformers introduces a significant limitation: quadratic memory usage, which restricts their effectiveness for tasks involving long sequences [345]. While multilingual models like mT5 and mBART exist, they inherit similar limitations from their English versions, especially in managing extended contexts [346]. Even though state-of-the-art LLMs can process contexts containing up to a million tokens [347], their accuracy on long-context QA tasks often drops, particularly if the answer is not located near the beginning or end of the context [348]. In this work, we explore the use of APS models, OIE, and coreference resolution to reduce the context length for non-factoid queries.

Open Information Extraction involves identifying n-ary tuples from a text. When these tuples consist of three elements—namely, the head, relation, and tail—they are called triples. For example, in the sentence: *Helen, John Wick's wife, gifted him a beagle puppy named Daisy*, the extracted triples could be: (i) {*Helen*, wife-of, *John Wick*}, (ii) {*Helen*, *gifted*, *him*}, and (iii) {*Daisy*, is-a, *beagle*}. Importantly, triple extraction is not a deterministic process, and different sets of triples may be produced from the same sentence. In contrast, coreference resolution is the process of detecting text spans that refer to the same real-world entity. For example, in the sentence above, both *him* and *John Wick* refer to the same individual.

Most QA research is centered on English [349]. Transferring English-based models to low-resource languages, especially for extended contexts, is often ineffective [350]. While non-factoid QA models perform well in English [351], there is a notable lack of datasets for Indic languages [57]. This work targets Hindi, Urdu, Tamil, and Telugu, leveraging available OIE resources [167, 352] to bridge this gap.

Motivated by the importance of non-factoid QA systems and the limited research on Indic non-factoid QA, we have developed models specifically designed for non-factoid QA in Indic languages.

The principal contributions of this paper are:

1. We show that integrating an APS model can substantially enhance LLM performance in non-factoid QA by efficiently reducing the context size.

2. We release finetuned checkpoints for gemma-2b, gemma-7b, and Llama3.1, trained on non-factoid QA pairs in Hindi, Urdu, Tamil, and Telugu.

3. We introduce a new Semantic Text Similarity metric for Multilingual Texts (STS-MuTe), specifically designed to assess the quality of generated responses.

4. Our analysis reveals that when the APS model assigns a high score to a question-paragraph pair, nearly every token contributes to the prediction, whereas for low scores, only a small subset of tokens is influential.

7.1 Related Work

QA task can be divided into several categories based on specific characteristics of the problem [32, 33]. It has been observed that QA tasks involving context (either long or short) and typically requiring factoid answers are often referred to as MRC [353]. A particular form of QA, where an explicit question is not provided but the task involves filling in blanks based on given instructions, is known as a cloze task [354].

In this study, we define a context as long if it exceeds 512 tokens, due to the absence of a universally accepted definition for long context. This threshold is chosen for two reasons: (a) it matches the default input limit of transformer-based encoder models like BERT [44] and RoBERTa [45], which are widely used in QA systems [46]; and (b) the dataset used in this research contains contexts longer than this limit (see Section 7.2). We discuss LLM token limits further in Section 7.6.

To specifically address non-factoid queries, various APS strategies have been utilized by Cohen et al. [355]. Glass et al. [356] demonstrated that retrieving, re-ranking passages, and then generating answers from them leads to improved results. Yulianti et al. [357] first retrieved relevant answer paragraphs and then generated summaries from them. Extracting a concise context from a longer passage not only breaks down the long-context QA problem into manageable steps but also increases interpretability of the results [358]. Li et al. [359] proposed a unified generative QA framework by combining a document ID retriever with document-grounded answer generation.

7.2 Dataset

This research utilizes the MuNfQuAD [272] dataset, which is the largest publicly available multilingual non-factoid QA dataset. The question-answer pairs and their corresponding contexts were sourced from BBC news articles. Additionally, methods similar to those in [4] and [60], which collected non-factoid QA pairs from the open web via web scraping, have been explored. The APS model finetuned in the MuNfQuAD study was employed, ensuring that its training data was excluded from any finetuning or evaluation in this work. After removing the APS model's training data from MuNfQuAD, the resulting dataset contained over 304K question-answer pairs.

To assess the effectiveness of OIE and coreference resolution for long-context non-factoid QA, we limited our analysis to languages with available tools. The Transmucores coreference resolution model [221] has been trained on 31 South Asian languages. However, there is a notable scarcity of multilingual OIE resources. To the best of our knowledge, IndIE [167] and Gen2OIE [70] are the only multilingual OIE systems supporting four Indic languages: Hindi, Tamil, Telugu, and Urdu. Consequently, our proposed approach is evaluated on these four languages.

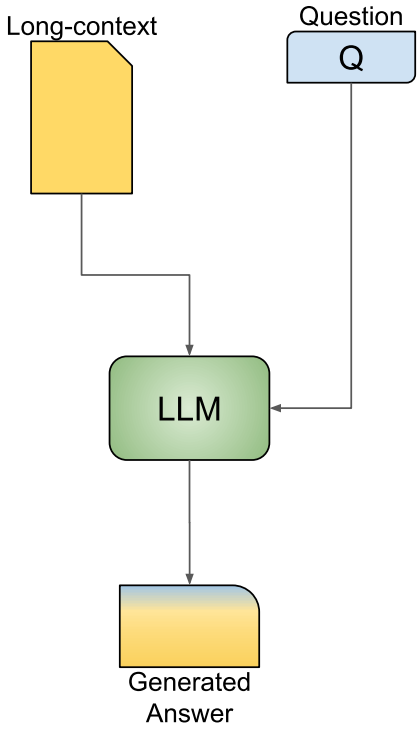
7.3 Methodology

A conventional approach to addressing a question within a given context involves submitting the combined question and context to a LLM. Empirical results demonstrate that LLMs outperform alternative models on a variety of NLP tasks [104, 105, 360]. Figure 7.1a visually presents the typical process of utilizing LLMs for question answering, based on the supplied context.

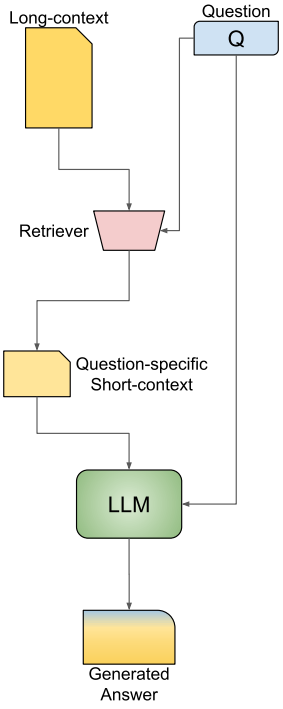
In this work, we investigate the integration of a retriever into the QA pipeline, aiming to reduce a lengthy context to a shorter one for addressing non-factual queries. The retrieval process is illustrated in Figure 7.1b. As discussed in section 7.1, earlier methods have proposed incorporating a retriever in QA pipelines. We seek to assess its effectiveness in a multilingual QA setting with non-factual queries. Four distinct retriever construction strategies, illustrated in Figure 7.1b, were adopted in this research.

A1. **Basic APS model:** Both the full context and the question are segmented into paragraphs, which are then processed using the advanced APS model from [272]. The top five paragraphs were selected to form the shortened context.

A2. **Ranking of expressed triplets (OIE+APS):** For each paragraph in the original context, triplets were extracted. Each triplet consists of a subject, relation, and object. In the expression phase, concise statements were generated by combining the extracted subject, relation, and object. Given the flexible word order in Indian languages [361], the generated statements maintained semantic accuracy. These articulated triplets



(a) The entire long-context, along with the question text, is provided as input to the LLM. Based on the given prompt instruction, an answer is generated. This is the baseline method.



(b) From the long-context and the question text, a short-context is obtained by the retriever. Subsequently, this derived short-context is given as input to the LLM. This is the retrieval method.

Figure 7.1: LLMs are applied in approaches for answering questions over extended contexts.

were concatenated with the question and processed by the APS model. The APS model's output was used to rank and select the top ten articulated triplets, which served as the reduced context.

A3. **Coreference resolution and APS ranking (coref+APS):** Multilingual coreference resolution, as described in [221], was performed on the full context to identify coreference links between paragraphs. After identifying the paragraph with the highest APS score for the given question, all paragraphs with coreference links to this paragraph were selected. The five highest-scoring paragraphs, according to APS, were chosen.

A4. **Ranking of coreference chains of articulated triplets (OIE+coref+APS):** Using the predicted coreference chains and articulated triplets, groups of triplets were assembled according to their coreference chains. Figure 7.2 provides a graph-

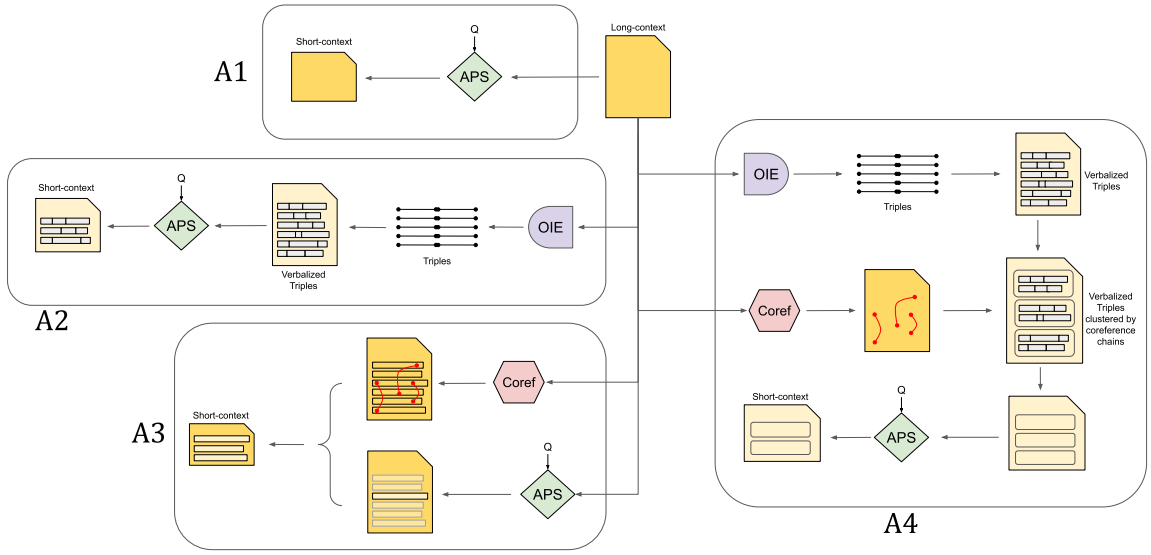


Figure 7.2: The four approaches used to construct the retriever are illustrated. A short sentence, through the combination of its constituent components, is produced by the process of transforming a triple (consisting of a head, relation, and tail). This is the verbalization of triples.

ical overview of the four techniques implemented in this work.

We integrate each of the proposed retrieval strategies into the complete QA pipeline to assess their impact in an end-to-end setting. This integration allows us to measure not only the effectiveness of the retrievers in producing concise and semantically relevant contexts, but also the overall gains they bring to the final QA task when combined with the answer generation component. By employing different retrievers within the full pipeline, we obtain a holistic view of their contributions, capturing improvements that may arise from better alignment between the reduced context and the answering capabilities of QA pipeline. The outcomes of this comprehensive evaluation are systematically presented and discussed in section 7.4, where we highlight the comparative advantages of each retrieval strategy in enhancing multilingual QA performance for non-factual queries.

7.3.1 Implementation

A score, indicating whether the provided text can answer the given query, is predicted by the finetuned APS model from [272] in this study. It has been noted that effective answer paragraph retrieval from a given document with long-context, using simpler baselines such as tf-idf and cosine similarity on word embeddings, is not adequately performed [362, 363]. Furthermore, effective performance for answer paragraph retrieval by deep learning architectures, based on finetuning embeddings, has been demonstrated by [364].

The checkpoint with the base encoder of XLM-R [294] has been selected by us, given the observed superior performance of XLM-R on multilingual QA relative to mbert [351].

Answer the question based on the given context.
##Question
{question}
##Context
{context}
##Answer
{answer}

Figure 7.3: Within the QA framework, LLMs are adapted through prompt-based finetuning. The {context} serves as an extended context for the QA system with B, while a concise context is utilized for the QA system with A1-A4. At inference, the {answer} was omitted from the prompt.

Additionally, a significantly smaller GPU memory footprint and inference time are exhibited by the XLM-R based APS model than other multilingual encoders.

Comparisons are made by us of our methodologies (A1-A4) with the baseline (B), as illustrated in Figure 7.1a, wherein no retriever is incorporated into the QA pipeline. Models of instruction finetuned Gemma [105], with 2 billion and 7 billion parameters, were employed by us. Also, the 8 billion parameter model of instruction finetuned Llama 3.1 [104] was utilized by us.

It is worth noting that the instruction-tuned LLMs employed in this study were released only a few months prior to the commencement of this work, and thus represented one of the most recent models available at that time. The release dates of these models are documented in section 7.6 for reference. While our present focus is on assessing the integration of retrievers within the QA pipeline using these instruction-tuned models, we recognize that a comprehensive evaluation of LLMs already fine-tuned within RAG pipelines constitutes a promising direction for further exploration.

On a training set of over 29K questions, the aforementioned LLMs were finetuned by us for each approach (A1-A4) and baseline (B). Parameter efficient finetuning [365], with LoRA [366] rank and alpha of 32, was employed by us to develop the *QA pipeline containing finetuned LLMs* ($QA_{finetuned}$). The prompt utilized for finetuning LLMs is depicted in Figure 7.3. By two primary considerations, the selection of these models was motivated: (a) compatibility of their parameter sizes with the computational resources available to us, and (b) demonstrated effectiveness of these models in prior studies on question answering in low-resource languages [367, 368, 369].

Zero-shot predictions were performed as the models processed the test dataset. Utilizing four-bit quantization[2], predictions were generated with *a QA system employing unmodified (base) LLMs* (QA_{base}). All experiments were executed on two NVIDIA A100 GPUs, each equipped with 40GB of memory.

[2]https://pypi.org/project/bitsandbytes/

7.3.2 Evaluations

To evaluate the similarity between LLM-produced responses and the silver-standard answers from MuNfQuAD, the ROUGE metric [370] was applied to measure overlap. According to [343], ROUGE has also been used to score long-form descriptive responses. Nevertheless, Soleimani et al. [4] demonstrated that ROUGE is not dependable for evaluating lengthy sequences. Additionally, although LLM-generated responses may underperform on lexical-based metrics, they often convey the same semantic content as the reference answers [35]. Therefore, we introduce a semantic similarity metric for multilingual texts (STS-MuTe). For any pair of texts (t_1, t_2), STS-MuTe is computed as the arithmetic mean[3] of the cosine similarity (cos) across several multilingual embeddings and the BERTScore [371]. The following models were employed to calculate cosine similarity: (a) USE [372], (b) LaBSE [373], and (c) LASER [374, 375]. Equation 7.1 presents the definition of STS-MuTe. BERTScore, which relies on token-level pairwise cosine similarity, yields a single floating-point value representing semantic similarity. For this, we used pretrained multilingual BERT [**?**] to obtain contextual embeddings for BERTScore calculation. Thus, both token-level and semantic-level similarity metrics are adopted in this study to assess LLM-generated responses.

$$\text{STS-MuTe} = \frac{\begin{array}{c}\text{BERTScore}(t_1, t_2) \\ +cos(\text{USE}(t_1), \text{USE}(t_2)) \\ +cos(\text{LaBSE}(t_1), \text{LaBSE}(t_2)) \\ +cos(\text{LASER}(t_1), \text{LASER}(t_2))\end{array}}{4} \tag{7.1}$$

7.4 Results

The test dataset comprised over 6,000 QA instances. However, due to computational limitations, inference was restricted to a subset of 1,100 questions. The mean context length for these samples was 1,070 tokens. We ensured that all four target languages were represented in this subset. Table 7.1 summarizes the outcomes of our QA system using different strategies. The same test instances were used for evaluating both QA_{base} and $QA_{finetuned}$.

Relative to B, it was found that A1 outperformed other methods on nearly every evaluation criterion. Even in cases where B slightly exceeded A1 (gemma-7b-it), A1 still delivered competitive results. IndIE was observed to surpass Gen2OIE in A2, while Gen2OIE performed better in A4. We attribute this to IndIE's ability to produce more detailed triples than Gen2OIE [167]. Our findings suggest that the APS model can extract higher-quality short-contexts from the verbalization of detailed triples. In contrast, Gen2OIE's broader triples tend to preserve coreferential mentions, resulting in improved performance for clusters of verbalized triples.

[3]We also experimented with the harmonic mean, but observed no significant difference in outcomes.

Gemma 2 Billion Instructional (gemma-2b-it)									
	Semantic-level					Token-level			
	BERTScore	USE	LaBSE	LASER	STS-MuTe	R1	R2	R3	RL
B	0.63 (0.75)	0.38 (0.80)	0.43 (0.58)	0.58 (0.83)	0.50 (0.74)	0.10 (0.44)	0.04 (0.31)	0.02 (0.29)	0.08 (0.34)
A1	**0.66** (**0.78**)	**0.39** (**0.81**)	**0.46** (**0.70**)	**0.56** (**0.85**)	**0.52** (**0.79**)	**0.12** (**0.51**)	**0.08** (**0.44**)	**0.07** (**0.42**)	**0.11** (**0.39**)
A2	0.62 (0.61)	0.31 (0.53)	0.38 (0.51)	0.50 (0.74)	0.46 (0.60)	0.07 (0.14)	0.02 (0.04)	0.01 (0.02)	0.05 (0.11)
A3	0.57 (0.64)	0.18 (0.61)	0.32 (0.50)	0.49 (0.77)	0.39 (0.63)	0.03 (0.21)	0.01 (0.09)	0.00 (0.07)	0.03 (0.14)
A4	0.65 (0.75)	0.37 (0.77)	0.45 (0.67)	0.55 (0.83)	0.50 (0.75)	0.11 (0.43)	0.07 (0.35)	0.06 (0.33)	0.10 (0.33)
Gemma 7 Billion Instructional (gemma-7b-it)									
	Semantic-level					Token-level			
	BERTScore	USE	LaBSE	LASER	STS-MuTe	R1	R2	R3	RL
B	0.66 (**0.81**)	**0.52** (**0.84**)	0.53 (0.72)	**0.70** (**0.87**)	0.60 (**0.81**)	0.16 (**0.55**)	0.07 (**0.47**)	0.05 (**0.46**)	0.11 (**0.49**)
A1	**0.69** (0.79)	**0.52** (0.81)	**0.56** (**0.74**)	0.66 (0.86)	**0.61** (0.80)	**0.19** (0.53)	**0.13** (0.46)	**0.11** (0.44)	**0.16** (0.42)
A2	0.65 (0.63)	0.44 (0.58)	0.46 (0.56)	0.59 (0.77)	0.53 (0.63)	0.10 (0.17)	0.03 (0.06)	0.01 (0.03)	0.07 (0.12)
A3	0.63 (0.70)	0.43 (0.69)	0.45 (0.62)	0.63 (0.81)	0.54 (0.71)	0.09 (0.33)	0.03 (0.21)	0.01 (0.17)	0.07 (0.24)
A4	0.68 (0.76)	0.49 (0.77)	0.53 (0.70)	0.63 (0.83)	0.58 (0.77)	0.16 (0.45)	0.11 (0.36)	0.09 (0.34)	0.14 (0.35)
Llama 3.1 8 Billion Instructional (llama3.1-8b-it)									
	Semantic-level					Token-level			
	BERTScore	USE	LaBSE	LASER	STS-MuTe	R1	R2	R3	RL
B	0.69 (**0.78**)	0.61 (0.77)	0.57 (0.72)	0.69 (**0.82**)	0.64 (0.77)	0.28 (0.46)	0.18 (0.39)	0.15 (0.38)	0.22 (**0.41**)
A1	**0.70** (**0.78**)	**0.68** (**0.78**)	**0.62** (**0.73**)	**0.76** (**0.82**)	**0.69** (**0.78**)	**0.34** (**0.49**)	**0.26** (**0.43**)	**0.24** (**0.41**)	**0.27** (**0.41**)
A2	0.61 (0.65)	0.56 (0.63)	0.49 (0.57)	0.71 (0.74)	0.59 (0.65)	0.17 (0.22)	0.06 (0.08)	0.02 (0.04)	0.11 (0.15)
A3	0.61 (0.71)	0.51 (0.70)	0.49 (0.63)	0.68 (0.78)	0.57 (0.70)	0.16 (0.32)	0.06 (0.20)	0.03 (0.16)	0.11 (0.24)
A4	0.69 (0.75)	0.65 (0.75)	0.59 (0.69)	0.74 (0.80)	0.67 (0.75)	0.30 (0.43)	0.21 (0.35)	0.19 (0.33)	0.23 (0.36)

Table 7.1: The table displays the effectiveness of the QA system with various LLMs on a portion of the test data. Since A2 and A4 utilize an OIE system, red color highlights where IndIE outperformed Gen2OIE, and green color marks where Gen2OIE surpassed IndIE. ROUGE-1 (R1), ROUGE-2 (R2), ROUGE-3 (R3), and ROUGE-LCS (RL) F1 scores are reported. Values in parentheses indicate $QA_{finetuned}$ performance, while those outside refer to QA_{base}. The highest value in each column is bolded. It is evident that A1 consistently achieved superior results compared to the others.

Upon comparing A1's improvement over B across three LLMs, we observed that the average relative increase in STS-MuTe and ROUGE scores was 4% and 47%, respectively, for QA_{base}, while a 2% gain was noted for both metrics with $QA_{finetuned}$.

To perform a qualitative assessment of the outputs generated by QA systems with B and A1, we opted to use LLM-as-a-judge. Since employing human annotators is both time-consuming and costly, LLM-as-a-judge has been adopted in several earlier works [376, 377]. Our preliminary experiments on a subset of the evaluation set showed that the base `gpt-4o-mini` from [378] can outperform all $QA_{finetuned}$ models on every quantitative metric. The prompt shown in Figure 7.3 was used. As a result, we decided to use the ChatGPT LLM as the judge. We utilized `gpt-4o` to evaluate the answers produced by the $QA_{finetuned}$ pipeline for all three LLMs. The prompt used for this evaluation is presented in Figure 7.4. The proportion of questions where A1's answers are preferred over those from B is: 73% for gemma-2b-it, 93% for gemma-7b-it, and 51% for llama3.1-8b-it. This demonstrates the qualitative superiority of A1 compared to B.

As an ablation of the A1 approach, we used the LangChain[4] vectorstore retriever with

[4] `https://github.com/langchain-ai/langchain`

Pipeline	Approach	gemma-2b-it	gemma-7b-it	llama3.1-8b-it
QA_{base}	B	11GB x2 4 secs/question	11GB x2 30 secs/question	12GB x2 70 secs/question
	A1	3.5GB x2 4 secs/question	6GB x2 8 secs/question	7GB x2 58 secs/question
$QA_{finetuned}$	B	20GB x2 13 secs/question	29GB x2 30 secs/question	30GB x2 45 secs/question
	A1	8GB x2 13 secs/question	21GB x2 20 secs/question	23GB x2 22 secs/question

Table 7.2: The computational resource usage of the QA system using both the baseline (B) and A1 methods is illustrated. The notation 'x2' denotes memory usage across a pair of GPU devices. Owing to the concise context retrieved by the retriever, A1 is more efficient and operates faster than B, as demonstrated.

Given the following question, you are given a ground-truth answer and two options. Choose the option that is closest to the ground truth. You are only allowed to choose one option. Print either "option1" or "option2". Print nothing else.
##Question
{question}
##Ground_Truth: {ground_truth}
##Option1: {option1}
##Option2: {option2}

Figure 7.4: The prompt employed to query the ChatGPT model for qualitative evaluation is shown.

Huggingface embeddings [379] from the paraphrase-multilingual-MiniLM-L12-v2 model [303], instead of the fine-tuned APS model from [272]. It was found that the LangChain-based A1 variant outperformed the A1 method with the fine-tuned APS model. We also experimented with BM25 to select answer paragraphs for each query from the relevant context. The results showed that the A1 approach using the fine-tuned APS model ($A1_{APS}$) surpassed both the LangChain and BM25 variants of A1 on all evaluation criteria. Interestingly, 75% of the top-k paragraphs chosen by $A1_{APS}$ were different from those retrieved by BM25, indicating that the APS model captures information beyond simple bag-of-words and probabilistic ranking.

The A1 method not only achieves better results than B on evaluation metrics, but also requires fewer resources. Table 7.2 presents the degree of computational and time savings provided by the QA pipeline with A1 compared to B. The QA pipeline with A4, utilizing the GenOIE tool, also demonstrated competitive results relative to the best-performing A1 method. However, it introduces additional computational overhead. The coref module consumes 5GB of GPU memory and takes 0.14 seconds per article, while the Gen2OIE module uses 6.5GB of GPU memory and requires 15 seconds per article.

Pipeline	Eval. Metric	gemma-2b-it	gemma-7b-it	llama3.1-8b-it
QA_{base}	STS-MuTe	**60.7%**	48.4%	**59.1%**
	ROUGE	**64.9%**	**58.7%**	**60.6%**
$QA_{finetuned}$	STS-MuTe	**65.8%**	45.4%	48.3%
	ROUGE	**61.8%**	39.9%	44.1%

Table 7.3: For A1 and B, the proportion of test cases where the average ROUGE scores (R1, R2, R3, RL) and STS-MuTe for A1 surpass those for B is presented. Cases where A1 achieves better results (>50%) than B are highlighted in **bold**.

Due to the high resource requirements of B, Out of Memory (OOM) errors occurred on many test cases with the available GPU hardware. Therefore, B and A1 were compared on the subset of test cases that both could process, out of the 1100 total test cases. Table 7.3 highlights A1's performance advantage over B on these shared test cases. It shows that A1 outperforms B in most cases. For all test cases where A1 successfully generated an answer but B encountered an OOM error, the mean number of tokens in the (long) context was 1238, 1189, and 1069 for QA pipelines using gemma-2b-it, gemma-7b-it, and llama3.1-8b-it, respectively.

To compare the effectiveness of B and $A1_{APS}$, we automatically translated a subset of non-factoid QA pairs from the NaturalQA dataset [42] into Hindi, Tamil, Telugu, Urdu, Marathi, and Bengali using the NLLB 1.3B model [258]. We selected questions that did not have a short answer. The average context length in this dataset exceeded 8,000 tokens. All related resources will be made publicly available. Our findings indicate that (except for QA_{base} with Llama 3.1) $A1_{APS}$ consistently outperforms B on evaluation metrics, in the ChatGPT-as-a-judge setup, and in memory usage across all LLMs.

7.5 Discussion

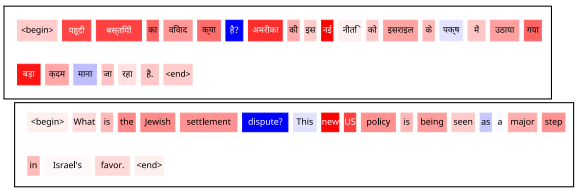
Given that A1 outperformed B, we analyzed the behavior of the APS model, which focuses on reducing context. To do this, we applied post-hoc explainability techniques to extract rationales for the APS model's outputs. Specifically, we used LIME [148] and SHAP [149], implemented via the Ferret library [340], to generate these rationales. Figure 7.5 shows a visualization of a sample rationale for a Hindi text, along with its English translation produced automatically using the NLLB 1.3B model [258].

The APS model assigns a high score when it is confident that the paragraph, combined with the question, is likely to answer the query. We observed that when the APS model predicted a high score, many tokens received high relevance values (marked in red), while fewer tokens were highlighted when the score was low. To empirically confirm this, we conducted a validation.

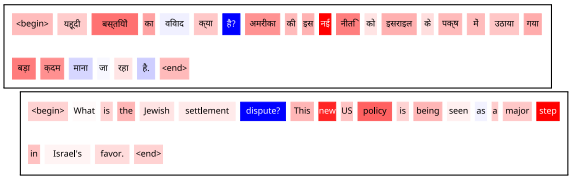
Generating rationales for an APS model prediction took between 4 and 15 seconds, depending on input length. Therefore, we computed rationales for a subset of the test set,

consisting of over 8,500 question-paragraph pairs. The results, shown in Figure 7.6, support our earlier observations. These findings are consistent with our intuition, suggesting that only a small number of tokens are needed for the APS model to determine that a paragraph does not answer the question. This supports the use of the APS model as a retriever in the QA pipeline.

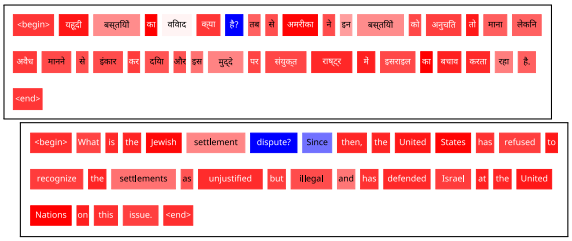
The MuNfQuAD dataset [272] includes question categories for each item, and its authors have shown that while non-factoid questions are the majority, factoid questions are also



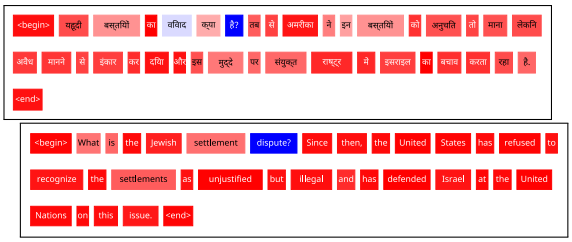
(a) LIME rationales are shown where the APS model score was 0.379 (left) and 0.377 (right).



(b) SHAP rationales are shown where the APS model score was 0.379 (left) and 0.377 (right).



(c) LIME rationales are shown where the APS model score was 0.574 (left) and 0.56 (right).



(d) SHAP rationales are shown where the APS model score was 0.574 (left) and 0.56 (right).

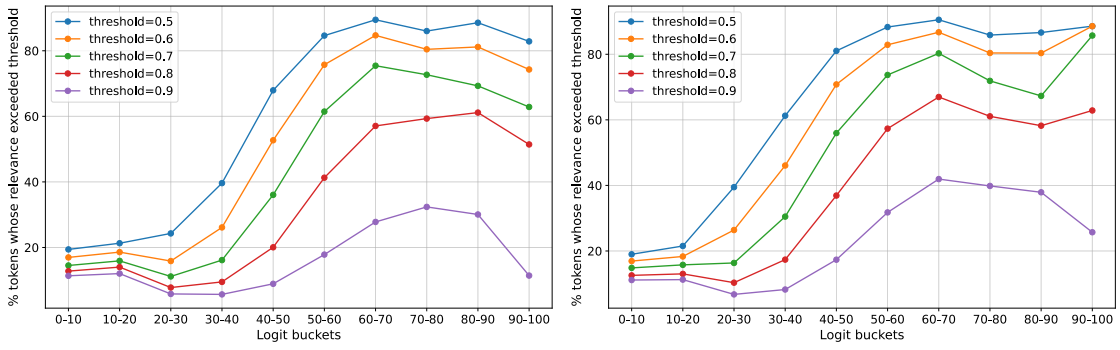
Low High

Figure 7.5: Rationales are generated using two post-hoc explainability methods, LIME and SHAP. The figures on the left show Hindi text from the test set, while the right side displays the English translation for readers unfamiliar with Hindi. Tokens with high relevance values (brighter red) indicate a strong contribution to the predicted logit. Masking these highlighted tokens is expected to reduce the model's confidence. Notably, as the APS model score rises, the rationales become more widely distributed (in red).

QA pipeline		Question Categories				
		Factoid	**Evidence-Based**	**Debate**	**Reason**	**Experience**
gemma-2b-it	B	30 vs 28 (32)	26 vs 25 (32)	16 vs 21 (13)	12 vs 15 (10)	3.6 vs 3 (2)
	A1	31 vs 32 (40)	25 vs 28 (28)	20 vs 15 (18)	11 vs 11 (4)	3.9 vs 3 (2)
gemma-7b-it	B	30 vs 42 (40)	25 vs 30 (29)	19 vs 11 (17)	11 vs 8 (9)	4 vs 2 (2)
	A1	31 vs 25 (34)	25 vs 31 (32)	20 vs 22 (19)	11 vs 4 (4)	3.9 vs 6 (2)
llama3.1-8b-it	B	31 vs 41 (40)	25 vs 29 (22)	20 vs 18 (24)	11 vs 5 (6)	3.9 vs 1 (3)
	A1	31 vs 46 (37)	25 vs 24 (26)	20 vs 11 (19)	11 vs 5 (7)	3.9 vs 4 (4)

Table 7.4: QA pipelines are shown across different question types. The cell format is: "*Total* vs *Best_100 for* QA_{base} *(Best_100 for* $QA_{finetuned}$*)*", where *Total* and *Best_100* indicate the percentage of a category among all generated answers and among the top-100 answers, respectively. The top-100 answers were selected by ranking all generated answers by their average scores. If the *Best_100* percentage is higher than the corresponding *Total* percentage, it is highlighted in green; otherwise, it is marked in red. Factoid and Evidence-Based questions are more prevalent in the *Best_100* compared to Debate, Reason, and Experience categories.

present. We analyzed patterns in question types among the top 100 test cases, which were selected by ranking the generated answers by their mean scores and taking the first 100. The mean scores were computed by averaging semantic-level (STS-MuTe) and token-level (ROUGE) metrics. Consistent with the findings of [5], Table 7.4 shows that the highest-quality answers are more often associated with factoid and evidence-based questions, as opposed to debate, reason, and experience types.



(a) LIME thresholds (b) SHAP thresholds

Figure 7.6: To interpret the above figure, consider this example: for a text with 10 tokens, if 7 tokens have a relevance value above a set threshold (e.g., 0.5), and the APS model predicts a logit of 0.35, then the logit bucket is 30–40 (0.35), and the percentage of tokens exceeding the threshold is 70% (7/10). The figure demonstrates that higher logit values are associated with a broader spread of relevance values across different thresholds.

7.6 IndicGenQA Limitations

The Gen2OIE approach was tested on two Indic languages: Hindi and Telugu. We used checkpoints for these languages in our experiments, and performed zero-shot inference for Urdu and Tamil, respectively. This approach was motivated by the strong linguistic similarity between Hindi-Urdu and Telugu-Tamil, as shown by the online tool[5] from [380].

The ROUGE Python library[6] used in this work is tailored for English. To our knowledge, there is currently no ROUGE implementation that supports multilingual texts in non-Roman scripts. Therefore, we converted all texts to Roman script using the method described in [381]. The results reported here may change if a ROUGE package specifically designed for non-Roman scripts becomes available, especially regarding stopword removal.

Our QA pipeline relies on pretrained LLMs. As discussed by Ahuja et al. [311], these LLMs are trained on large-scale public datasets such as Common Crawl. This introduces a possible risk of inflated evaluation scores. To address this, we excluded test data that was publicly available before the release of each model checkpoint, using the last commit date of the model file in its Huggingface repository: 21-Feb-2024 for gemma models, and 18-July-2024 for Llama model. This filtering left 78, 197, and 18 examples for gemma-2b-it, gemma-7b-it, and llama3.1-8b-it, respectively. Compared to Table 7.1, the drop in evaluation metrics for QA_{base} with B was 13%, 2.5%, and 22% for these three LLMs, respectively. This suggests a modest decrease in QA pipeline performance when evaluated on data not seen during LLM pretraining.

We acknowledge that the context lengths used in this study are much shorter than the maximum token limits of many popular LLMs. For instance, Gemma models support up to 8K tokens, while LLaMA 3.1 allows up to 131K tokens[7]. However, in the absence of a multilingual non-factoid QuAD dataset with sufficiently long contexts, we argue that MuNfQuAD is the most suitable dataset for our experiments. One could suggest artificially extending the context by concatenating the context of a given question with those of other questions. However, previous studies have shown that answers to non-factoid questions are usually found within a small section of the document, rather than being spread throughout [94, 362]. Therefore, we did not investigate the effects of artificially increasing context length. While the retrieval-based pipeline used here can theoretically handle contexts of any size, its effectiveness for very long contexts should be tested on multilingual QuAD datasets with contexts exceeding LLM limits.

[5] `http://www.elinguistics.net/Compare_Languages.aspx`

[6] `https://pypi.org/project/rouge-score/`

[7] The token limit is specified by the `max_position_embeddings` attribute in the model's `config.json` file on Huggingface.

7.7 Chapter Summary

QA has led to the development of numerous approaches to address its challenges. With the emergence of LLMs and their improved performance across various NLP tasks, the focus in QA has shifted toward addressing LLM limitations, such as restricted context windows, and tackling the relatively unexplored area of non-factoid QA in low-resource languages. This work investigates the use of a retriever mechanism to condense the context associated with a question. We explore several methods—including APS, OIE, and coreference resolution (coref)—to implement a question-specific retriever. As a baseline, the full context is provided to the LLM without any reduction to answer the question. To evaluate the quality of generated answers, we propose a Semantic Text Similarity metric for Multilingual Text (STS-MuTe). Our experiments show that the APS-based approach (A1) is not outperformed by either the baseline or alternative methods on both semantic and token-level metrics. Moreover, the A1 method is found to be more resource-efficient. Additionally, our experiments with post-hoc explainability methods indicate that the APS model's scoring of question-paragraph pairs aligns with human judgment.

Chapter 8

Hindi Chatbot for Supporting Maternal and Child Health Related Queries in Rural India

We focus on the development of a Hindi chatbot designed to address healthcare-related queries. In regions with limited access to medical professionals, individuals often seek reliable health information through digital platforms. However, linguistic diversity and code-mixed language usage pose significant challenges in designing effective automated systems. This chapter explores the role of chatbots in bridging this gap by leveraging a curated database of frequently asked healthcare questions and their expert-verified responses. The discussion covers different approaches to measuring query-question similarity, including rule-based methods, sentence embeddings, and machine learning-based paraphrase detection. Additionally, the chapter examines the integration of these approaches into an ensemble model to improve response accuracy. By addressing the challenges of language variation and retrieval effectiveness, this work contributes to the broader goal of making healthcare information more accessible through automated conversational systems.[1]

Bull seal [382].

[1]Content of this chapter is adapted from our paper published in the ClinicalNLP workshop at ACL, 2023 [305].

Healthcare access in India exhibits significant disparities between urban and rural regions, with pregnant and postpartum women in rural areas facing limited access to medical services. These challenges arise due to restricted time with healthcare professionals, language barriers in doctor-patient communication, and societal constraints. In resource-constrained settings, digital support groups have emerged as a prevalent platform for seeking information on various maternal and child healthcare concerns [383, 384, 385]. However, moderators of such groups are often overwhelmed by a large volume of queries, making it difficult to provide timely responses. Additionally, group members frequently pose health-related questions in regional languages such as *Hindi* or Hinglish[2]. Given India's doctor-to-population ratio of 4.8 doctors per 10,000 individuals [386], scaling healthcare interventions that rely on direct involvement from medical professionals remains a challenge [384]. Consequently, there is an opportunity to enhance informational support for pregnant and postpartum women through a chatbot capable of responding to their written queries in their native language.

Chatbots are employed across various domains, including railway ticket reservations and food delivery services[3]. In the healthcare sector, chatbots have assumed diverse roles, functioning as psychotherapists, nurses, doctors, and medical consultants [31, 387, 388]. Notably, they hold the potential to serve as the initial point of contact for women seeking information on maternal and child healthcare, particularly in resource-limited settings [389]. In this study, we investigate the effectiveness of a chatbot in delivering accurate healthcare information by retrieving the most relevant Frequently Asked Questions (FAQs) along with their corresponding answers [390].

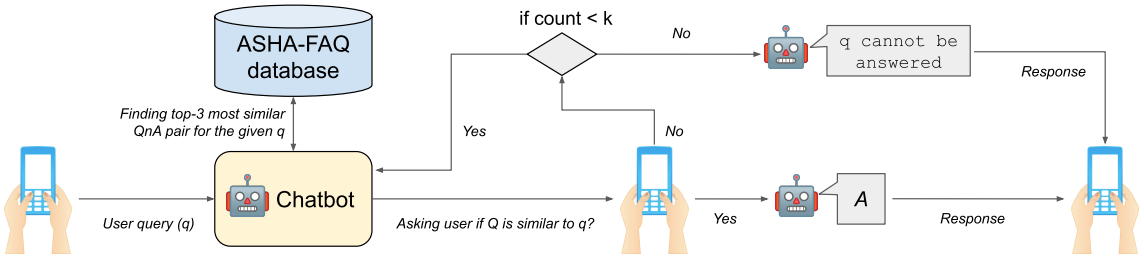


Figure 8.1: The architecture of the proposed chatbot is illustrated above. A user query (q) can be entered in either the Devanagari or Latin script. The chatbot retrieves the top-k most similar QA pairs from the ASHA-FAQ database and presents one question (Q) to the user at a time.

We developed a chatbot capable of retrieving the top-k most relevant Frequently Asked Questions (FAQs) along with their corresponding answers (QA pairs) in response to healthcare-related queries. The chatbot utilizes a curated database of QA pairs in Hindi, with answers

[2]Hinglish is an informal term referring to a language written using the English (Latin) script while incorporating Hindi grammar and vocabulary. It is also known as a Hindi-English code-mixed language. For example, 'नमस्ते'(*hello*) is written as *'namaste'*.

[3]https://www.chatbotguide.org/dominospizza-bot

reviewed and validated by healthcare professionals. It can process user queries written in both the Latin script (the native script for English) and the Devanagari script (the native script for Hindi). Figure 8.1 provides an overview of the proposed chatbot's architecture.

For evaluation, we collected a set of healthcare-related queries from ASHA workers[4]. In this study, we examine various algorithmic approaches for chatbot development and assess their effectiveness in retrieving relevant QA pairs. Specifically, we explore three primary approaches: (i) a rule-based method, (ii) sentence embeddings, and (iii) paraphrasing classifiers. Our findings indicate that an ensemble model integrating all three approaches outperforms individual methods. To facilitate further research in this domain, we publicly release the source code of our chatbot[5].

8.1 Related Work

With user query reformulation and knowledge from search engines, the development of chatbots in healthcare using AI was initially started [391]. For the English language, these chatbots were developed, but due to the scarcity of resources, the same techniques could not be applied to Hindi speakers. For unstructured English written as a shorthand for SMS by the Indian population, a FAQ retrieval system was aimed to be developed by Kothari et al. [392]. To calculate sentence similarity scores, character-level features were relied upon. For the Hindi language, initial works on building a QA system were restricted to exploiting information from shallow speech features such as POS tags [393]. For the construction of an automatic question-answering system for English-Hindi code-switched language (also known as *Hinglish*), the word-level translation of code-switched queries into English queries was commonly practiced due to the lack of resources in the Hindi language [394, 395]. Because Hindi-to-English word-level translations are highly dependent on the position of the Hindi word in a sentence, such approaches fail to generalize [396].

With cross-lingual word embeddings, a healthcare QA system has previously been solved for low-resource African languages [397]. Through empirical studies, it has been demonstrated that fine-tuned machine learning models utilizing embeddings from pre-trained transformer-based encoders like BERT outperform many traditional AI models across various tasks [398, 399].

For FAQ retrieval tasks, the efficiency of BERT-based models in measuring sentence similarity has been shown in earlier works [36, 400]. In this work, the performance of different approaches for measuring sentence similarity between Hindi sentences from the maternal healthcare domain has been compared. For a given user query (q), the most similar question (Q) and its corresponding answer (A) are retrieved from the ASHA-FAQ database, which is described in the next section.

[4]Accredited Social Health Activists (ASHA) are community health workers employed by the Ministry of Health and Family Welfare, India. They serve as a vital link between the rural population and the state healthcare system.

[5]github.com/ritwikmishra/asha-chatbot

8.2 Data Description

With permission from the authors, data from four prior studies were collected [401, 389, 402, 403]. Hundreds of QA pairs (in both audio and text modalities), as asked in real-world scenarios by community health workers and pregnant and postpartum women regarding maternal and child health issues, were included in the dataset. The answers to these questions were provided by health experts. With the assistance of two healthcare professionals, the audio data was transcribed and annotated. Both annotators held a bachelor's degree in medicine and surgery, a master's in public health, and had experience working in maternal and child health. Each session was manually transcribed in the Devanagari script by the annotators.

For this study, annotations were performed using an online transliteration tool[6] and Audino [404]. More than 18 hours of healthcare-related audio were transcribed, resulting in 1150 QA pairs. From the FeedPal study, which employed a wizard-based chatbot [389], 217 questions posed by young mothers along with their corresponding expert responses were incorporated into the ASHA-FAQ database, bringing the total to 1367 QA pairs. Subsequently, 217 maternal health QA pairs were received from Yadav et al. [389] and added to the ASHA-FAQ database, leading to a final count of 1365 unique questions and 1338 unique answers[7].

Due to the COVID-19 pandemic, field testing of AI models was not feasible. Therefore, to evaluate the models on real-time data, a total of 336 new user queries (q) were collected from ASHA workers with the assistance of a non-governmental organization (NGO) partner. ASHA workers were requested to provide queries they frequently encounter. These 336 queries were then annotated by the authors, with the support of public health professionals (holding a master's degree in public health), by mapping them to relevant questions from the ASHA-FAQ database. For each query (q), completely and partially matching QA pairs were identified from the ASHA-FAQ database. In this study, both types of matches (complete and partial) were treated as relevant. It was observed that among the 336 queries, at least one relevant question was found in the database for 270 user queries. Consequently, the coverage of the ASHA-FAQ database was determined to be 80% in this experiment. The 270 queries identified as having relevant matches were designated as the *hold-out test set* for evaluating the performance of different FAQ-retrieval approaches implemented in the chatbot.

To train a deep learning model for computing sentence similarity scores between two Hindi sentences, Hindi news articles were scraped from the Inshorts website[8]. Each data point (d_i) in the scraped dataset (D) consisted of a news article text (t_i), its headline (h_i), a summary of the text (t_i^s), and a paraphrased headline (h_i^p). A total of over $17K$ data

[6] `easyhindityping.com`

[7] A subset of the ASHA-FAQ database will be released to demonstrate the chatbot's functionality, while the full database will be made available in future research.

[8] `inshorts.com/hi/read`

points were collected. For generating negative (non-paraphrased) headlines of h_i, a random headline was selected from the paraphrased headlines[9].

In the Inshorts dataset, 35K Hindi sentences from the news domain are classified into two categories—paraphrased and not-paraphrased—with equal representation of both classes. The scraping scripts and hyperlinks to the news articles are being released in the repository mentioned above. To the best of our knowledge, this dataset is the most comprehensive resource available for paraphrase detection in the Hindi language. Since the Inshorts dataset is derived from an open-domain (news), a domain-specific paraphrase dataset for healthcare was constructed. Questions from the ASHA-FAQ database were manually paraphrased and treated as positive examples of paraphrases, while random sentences were selected as negative examples. The resulting dataset, referred to as the *AshaQs* dataset, consists of approximately 1500 healthcare-related question pairs, balanced across two classes (paraphrased and not-paraphrased).

The performance of various FAQ retrieval models has been compared using five information retrieval evaluation metrics: Mean Average Precision (mAP), Mean Reciprocal Rank (MRR), Success Rate (SR), normalized Discounted Cumulative Gain (nDCG), and Precision at 3 (P@3) [400]. Among these, Success Rate is the most intuitive metric, as it represents the percentage of user queries for which at least one relevant suggestion was retrieved within the top-k results.

8.3 Methodology

Our work aims to take a user query (q) as input and generate the top-k most relevant QA pairs from the ASHA-FAQ database as output. Consequently, the task has been formulated as an FAQ retrieval problem. To address this FAQ retrieval problem, three primary approaches were explored. The results from the best-performing approaches were then combined to develop an ensemble method. Each of the three approaches is capable of converting Latin script in user queries into Devanagari script. The indic-trans library was utilized for transliteration [381].

8.3.1 Dependency Tree Pruning (DTP)

A dependency parse tree was constructed for the given sentence, and all significant keywords were extracted by pruning the tree using handcrafted rules. The Stanza library was employed to extract shallow linguistic features such as POS tags and to generate the dependency tree for the Hindi language [2]. The tree pruning process consists of the following three steps:

I. Advice Removal: In the dependency tree, if any child of the root node contains words such as सलाह (advice) or इलाज (treatment), or if the root node is an inflection of the

[9]Experiments were also conducted with curated negative samples, where the negative headline was chosen as h_j^p, the paraphrased headline with the highest keyword overlap with h_i (where $i \neq j$). However, this approach did not yield performance improvements.

Hindi word कर (do) and has a child such as चाहिए (should) or क्या (what), then the child with the maximum number of descendants is designated as the new root. The original root, along with the rest of its children, is then pruned from the tree.

II. Node removal: A manual analysis of multiple dependency trees revealed that certain nodes associated with specific dependency relations do not contribute significantly to the meaning of the query. The identified dependency relations include: dep, dislocated, discourse, expl, cc, case, aux, aux:pass, and mark. Consequently, nodes connected to the dependency tree via these relations are removed.

III. Compound merging: In Hindi, certain actions are expressed using a pair of verbs, known as compound verbs. For example, in रैप करना (wrap doing), the first verb represents the action, while the second verb functions as a container for inflections such as gender, number, and tense. In the compound merging step, all compound verbs are reduced to their verb stems. The compound dependency relation is used to identify and merge these verbs.

Since the subject-object-verb paradigm is generally followed by the Hindi language, the words were extracted from the pruned dependency tree using post-order traversal. This was done to enhance the readability of the extracted sentence. During the traversal, lemmatization was applied to the words to remove inflections.

Using the DTP method, the keywords were extracted for every question (Q_i) in the ASHA-FAQ database. Precision and recall between the user query (q) and Q_i were calculated by comparing the overlap between their keywords. The F-measure(q, Q_i) was used as the comparison metric, representing the sentence similarity score between q and the ith question in the database (Q_i).

8.3.2 Sentence-pair Paraphrasing Classifier (SPC)

The goal is to train a deep learning model to predict a score that represents the extent to which a given sentence-pair conveys the same information. The predicted score from the classifier is used as the sentence similarity score for the given sentence-pair. If two sentences in a given sentence-pair convey identical information, then a value closer to one is expected to be predicted by the trained model. A pretrained multilingual transformer encoder (referred to as *encoder* henceforth) was fine-tuned, which generates d-dimensional embeddings for the given sentence-pair. These embeddings were then fed into a Feed-Forward Neural Network (FFNN) with a single output node to predict the sentence similarity score. Earlier works have demonstrated the superiority of fine-tuned *encoders* for paraphrase detection tasks in Hindi sentences under the IndicGLUE benchmark [405, 406]. The SPC model was fine-tuned on the Inshorts dataset and the AshaQs dataset using the Huggingface library [**?**].

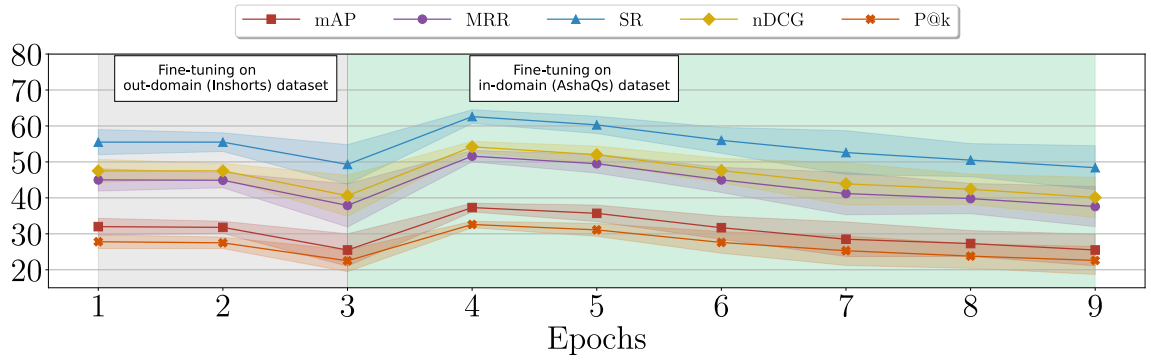


Figure 8.2: The performance of the fine-tuned *SPC* on the *hold-out test set* across ten different random seeds is illustrated. The initialization of weights in linear layers and the shuffling of data between training and testing sets before fine-tuning are determined by a random seed. The fine-tuned *SPC* generates the top-k QA suggestions for a given user query (q), where $k = 3$. The solid line represents the mean, while the shaded region denotes the standard deviation. The figure highlights the improvement in the performance of the *SPC* approach when fine-tuned on in-domain data for a single epoch.

8.3.3 Cosine Similarity (*COS*)

Different *encoders* were used to obtain a d-dimensional vector representation of q and Q_i, denoted as $E(q)$ and $E(Q_i)$, respectively. A pretrained encoder from the SentenceTransformer library [407] was utilized to generate vector representations of sentences. The traditional cosine similarity between $E(q)$ and $E(Q_i)$ was used to determine the sentence similarity score between q and Q_i.

Ensemble method ($\mathcal{E}$)

Due to its interpretability, the DTP methodology was selected, whereas the SPC and COS methodologies were incorporated based on their strong performance in sentence similarity tasks. To further enhance performance, an ensemble technique was introduced, which generates sentence similarity scores by combining the outputs of the three primary methodologies.

For each input query, every approach generated a ranked list of the most similar QA pairs from the ASHA-FAQ database, accompanied by their respective sentence similarity scores. The top-k QA pairs with the highest scores were selected as the final suggestions for each query. It was observed that while certain queries were better handled by one approach, the same approach underperformed for others. To address this, an ensemble method was devised to construct a new set of top-k suggestions by aggregating the final suggestions from different approaches. In this method, the scores of repeated suggestions were summed, and the top-k suggestions with the highest aggregated scores were selected as the final output of the ensemble approach.

	DTP	DTP_{q-e}	SPC	SPC_{+A}	SPC_{q-e}	COS	COS_{q-e}	$\mathcal{E}$
mAP	30.5	35.1	39.4	31.1	39.1	26.5	27.9	**45.3**
MRR	42.6	48.5	54.6	42.2	54.2	38.7	41.0	**61.6**
SR	27.1	59.6	66.2	49.6	64.4	47.7	51.1	**70.3**
nDCG	45.5	51.2	57.1	43.9	56.5	40.8	43.3	**62.5**
P@3	27.1	30.0	34.6	34.6	34.6	22.7	23.9	**34.6**

Table 8.1: A comparison of the three primary approaches on the *hold-out test set* for the top-3 suggestions generated by our chatbot. The ensemble ($\mathcal{E}$) is constructed by selecting the best-performing models, highlighted in yellow, from each primary approach. The results clearly demonstrate that the ensemble approach outperforms all other individual approaches.

8.4 Results

It was observed that among the top-3 suggestions, DTP provided at least one relevant suggestion for only 27.1% of user queries in the *hold-out test set*. Analyzing its failure cases revealed that the method struggled with the polysemous nature of words. For instance, DTP treated शुगर (sugar) and डायबिटीज (diabetes) as completely distinct words, even though both are interchangeably used in the Indian subcontinent to describe *Diabetes mellitus*.

To address this issue, buckets of synonymous or closely related words were maintained. Whenever a word from a bucket appeared in either q or Q_i, the remaining words from that bucket were added to the sentence. Expanding the query in this manner is known as *query expansion (q-e)* in automatic question-answering [396]. This modification improved the performance of the DTP method, increasing the proportion of queries receiving relevant suggestions to 59%. Table 8.1 presents the performance improvement due to the *q-e* variation, while Table 2 in Appendix A.2 illustrates an ablation study highlighting the significance of different pruning strategies in DTP.

To build the SPC model, multiple *encoders* were utilized. It was observed that the *bert-base-multilingual-cased* (*mBERT*) *encoder* by Devlin et al. [44] outperformed other pretrained multilingual *encoders*. Furthermore, the best performance was achieved when the SPC model was fine-tuned with three linear layers on top of the *encoder*. Since Rogers et al. [408] suggested that early layers of *encoders* contain more syntactic information, the early layers of the *encoder* were frozen. This led to more stable results across different random seeds. The model was initially fine-tuned on the open-domain Inshorts dataset, followed by additional fine-tuning on the *AshaQs* dataset in the healthcare domain. This two-step fine-tuning process improved the performance of SPC on the *hold-out test set*, with notable improvements observed in the fourth epoch, as illustrated in Figure 8.2. Table 8.1 demonstrates that $q - Q_i$ sentence similarity performs better than $q - Q_iA_i$ similarity, which aligns with previous research findings [36, 400]. The sensitivity of the SPC model to other architectural choices is detailed in Table 3 of Appendix A.2.

	$\mathcal{E}_{-COS}$	$\mathcal{E}_{-DTP}$	$\mathcal{E}_{-SPC}$	$\mathcal{E}$
mAP	40.9	40.8	30.3	**45.3**
MRR	56.2	56.5	43.8	**61.6**
SR	66.2	66.2	51.1	**70.3**
nDCG	58.4	58.2	45.5	**62.5**
P@3	34.6	34.6	23.9	**34.6**

Table 8.2: Results of the ablation study on the Ensemble method ($\mathcal{E}$). The table illustrates that removing any approach ($COS/DTP/SPC$) from the ensemble method results in lower performance.

Calculating the sentence similarity score as the cosine distance between the vector representations of two sentences has proven to be an effective approach. Among various *encoders* from the SentenceTransformer library, we observed that using *paraphrase-multilingual-mpnet-base-v2* as the pretrained *encoder* yielded better results. Additionally, Table 8.1 shows that applying *query-expansion (q-e)* variations on q and Q further improved the performance of the COS method.

Furthermore, Table 8.1 demonstrates that the ensemble method $\mathcal{E}$ outperformed all three primary approaches on the *hold-out test set*. To analyze the contribution of each component of $\mathcal{E}$, we conducted an ablation study. In this study, the minus sign in the subscript indicates the absence of a particular component. For instance, if SPC is absent, the ensemble is represented as $\mathcal{E}_{-SPC}$. The results in Table 8.2 reveal that removing any component leads to a decline in the performance of $\mathcal{E}$.

Additionally, when the three approaches produced top-5 suggestions, the resulting ensemble method achieved a SR of 73%. It was also observed that the chatbot performed better for user queries that had multiple relevant questions in the ASHA-FAQ database.

The SPC approach significantly contributes to the inference time of the ensemble method. With a GPU-enabled server, the chatbot provides real-time suggestions within 4 seconds while consuming 2.3 GB of GPU memory. However, without a GPU, the chatbot takes several minutes to generate the top-k suggestions and requires 6.0 GB of RAM.

8.5 Chatbot Limitations

In this study, we evaluated the chatbot using a Hindi-language database that was extensively annotated by human experts. Consequently, when dealing with a large-scale database, the scalability of the annotation process becomes a crucial concern. Given that questions and answers may exist in multiple languages, substantial effort would be required to translate them while ensuring that their contextual integrity is preserved. Our findings indicate that the developed chatbot achieved a success rate of 70% for Hindi queries. However, this result does not necessarily reflect its performance across different natural languages.

For a given user query (q), the performance of our most effective FAQ-retrieval approach is highly influenced by the number of relevant questions (Q) available in the ASHA-FAQ database for that particular query. Given the vast range of potential user queries within the healthcare domain, the relatively small size of our ASHA-FAQ database is a key factor contributing to cases where the system fails to retrieve relevant questions (Q).

Furthermore, our study does not assess the quality of answers present in the ASHA-FAQ database. Therefore, a dedicated user study would be necessary to evaluate both the diversity of the questions and the quality of the responses contained within the database.

8.6 Chapter Summary

We presented the development of a chatbot designed to alleviate the workload of healthcare professionals by providing informational support on maternal and child healthcare concerns in resource-constrained settings. The chatbot was developed using a FAQ-based approach, leveraging a curated healthcare database in Hindi. The proposed FAQ chatbot is capable of processing Hindi user queries written in both Devanagari (Hindi script) and Latin (English script). To retrieve the most relevant QA pairs from the FAQ database, we experimented with multiple FAQ-retrieval methods. Our findings indicate that the chatbot can successfully provide relevant QA pairs for up to 70% of queries that are covered within the existing FAQ database. The development process of the Hindi healthcare chatbot served as a formative experience, providing practical exposure to the complexities of QA in Indic languages. The challenges encountered and solutions explored during this phase laid the groundwork for the subsequent focus on explainable, resource-efficient, and multilingual QA frameworks presented in this thesis. Future work may involve conducting real-world evaluations and deploying the chatbot for use by healthcare professionals in live settings.

Chapter 9

Conclusion, Limitations, and Future Work

This chapter concludes the thesis, summarizing the key findings and contributions related to developing a framework for Non-Factoid Automated QA in Indian languages. As presented, the thesis addressed the challenge of generating explainable, query-specific short contexts by integrating Open Information Extraction, Coreference Resolution, and Information Retrieval. We now consolidate the outcomes of these integrated tasks, demonstrating their collective impact on enhancing Non-Factoid QA. Furthermore, we discuss potential future directions for extending this research, paving the way for more robust and nuanced QA systems in Indian languages.

Bull seal [409].

We embarked on a journey to address the critical need for robust and explainable Non-Factoid Automated QA systems in Indian languages, a domain significantly under-resourced compared to its English counterpart. QA is a foundational task in NLP, encompassing diverse characteristics, and this research specifically focused on developing a framework that integrates multiple supporting tasks to enhance the explainability and accuracy of QA systems for Indic languages. The motivation stemmed from the observation that while English QA has seen substantial advancements, the unique challenges posed by Indic languages, particularly the scarcity of resources and the complexity of their linguistic structures, necessitate tailored solutions.

We provided a comprehensive overview of the existing literature, highlighting the evolution of QA research from factoid to non-factoid, the impact of Transformer-based models, and the crucial role of retrieval mechanisms. The survey revealed significant gaps, particularly in Indic language QA, emphasizing the need for improved resources, efficient processing techniques, and advancements in retrieval-augmented and explainable QA models. We highlighted the potential of OIE for enhancing explainability and the importance of multilingual coreference resolution and retrieval-based context reduction for long-context understanding.

A practical application of QA principles was demonstrated through the development of a FAQ-based chatbot for maternal and child healthcare in Hindi. This chatbot, designed to alleviate the workload of healthcare professionals in resource-constrained settings, showcased the feasibility of leveraging existing knowledge bases to provide relevant information. The experimentation with various FAQ-retrieval methods highlighted the potential of this approach, achieving a 70% success rate for queries within the database. It laid the groundwork for further exploration of QA applications in real-world scenarios.

We addressed the critical issue of resource scarcity in Indic NLP by introducing IndIE, an OIE tool designed to extract triples from Hindi sentences. The development of IndIE involved fine-tuning a multilingual pretrained Transformer model and constructing a MDT to generate triples based on manually designed rules. The creation of *Hindi-BenchIE*, a benchmark for evaluating Hindi triples, further contributed to the advancement of OIE in Indic languages. The results demonstrated that IndIE outperformed other baselines, producing more informative and detailed triples.

We highlighted the lack of resources for coreference resolution in South Asian languages. The introduction of TransMuCoRes, a translated dataset for multilingual coreference resolution, and the release of fine-tuned checkpoints for off-the-shelf methods, contributed significantly to this domain. Our findings indicated the feasibility of fine-tuning the wl-coref method for languages with accessible dependency parsers, outperforming fast-coref in specific languages. This chapter emphasized the need for publicly accessible resources and models to advance coreference resolution in Indic languages.

We introduce a multilingual non-factoid QA dataset, MuNfQuAD, spanning 38 languages. This dataset, compiled from BBC news articles, addressed the critical gap in multilingual non-factoid QA resources. The evaluation of silver answers against a manually

curated golden set demonstrated the high quality of the dataset. Furthermore, the fine-tuned APS model, trained using MuNfQuAD, achieved high success rates, validating the effectiveness of the dataset for training QA models.

We also explore the explainability of Transformer-based frameworks, specifically focusing on the LRP technique for hate speech detection. The analysis revealed that fine-tuning pretrained Transformer models can lead to the concentration of textual information in the embeddings of the first few tokens, potentially limiting the effectiveness of LRP for explaining model predictions. However, the chapter also suggested that fine-tuned encoders exhibit a characteristic of feature space curtailment, which can be used to validate textual visualizations based on early token embeddings.

Finally, we investigated the use of retrieval procedures to condense context for LLMs in QA tasks. The research explored various strategies, including APS, OIE, and coreference resolution, and introduced a Semantic Text Similarity score for Multilingual Text (STS-MuTe) for evaluating answer quality. The results indicated that the APS-based method, while efficient, was outperformed by the baseline and other techniques. However, the chapter also demonstrated the alignment of the APS model with human perception through post-hoc explainability techniques.

In summary, this thesis has successfully developed a framework for Non-Factoid Automated Question-Answering in Indian languages by integrating multiple supporting tasks, including Open Information Extraction, coreference resolution, and information retrieval. The creation of IndIE, Hindi-BenchIE, and MuNfQuAD has significantly contributed to the advancement of NLP resources for Indic languages. The exploration of explainability techniques and retrieval-based context reduction has provided valuable insights into improving the performance and interpretability of QA systems.

9.1 Limitations

I. While this work addresses the under-resourced nature of Indic languages by introducing tools such as IndIE and datasets like MuNfQuAD, the scope remains limited to a subset of Indic language families, predominantly Indo-Aryan and Dravidian. The lack of annotated corpora, linguistic tools, and NLP resources for many other Indian language families continues to hinder broader generalization.

II. The evaluation of systems and resources, including Hindi-BenchIE and MuNfQuAD, relies on manually curated datasets or silver-standard answers. Although these offer valuable insights, the absence of large-scale, high-quality human-annotated benchmarks may affect the robustness and reproducibility of the findings across domains and languages.

III. The use of manually designed rules in the construction of the Merged-Phrase Dependency Tree for triple extraction in IndIE limits scalability across languages and

domains. These rules, while effective in controlled settings, may not generalize well to complex syntactic structures or lesser-documented languages.

IV. The investigation of LRP for model interpretability highlighted inherent limitations within Transformer-based architectures, notably the tendency for relevant information to become concentrated within the embeddings of the initial tokens. However, this observation is derived from relevance values propagated only up to the fine-tuning layers. To obtain more comprehensive insights, it is necessary to employ methods capable of backpropagating relevance values through the entire architecture, reaching the token embedding layer within the Transformer model itself.

V. Although the APS model demonstrated alignment with human judgment, its underlying architecture was intentionally kept simple to facilitate training on large-scale datasets and to enable efficient parallelized fine-tuning. Consequently, there is a need to explore and evaluate more sophisticated APS model architectures on the released dataset to potentially enhance performance.

VI. The practical application of the framework through a healthcare chatbot demonstrated promise, yet its success rate (70%) indicates room for improvement. The system's reliance on predefined FAQs restricts its flexibility and coverage, particularly when handling out-of-domain or conversationally nuanced queries.

VII. Many components of the framework, such as IndIE and TransMuCoRes, rely on pre-trained multilingual Transformers. While these models facilitate cross-lingual transfer, they may inherit biases or limitations from their training data, which can impact downstream performance and fairness across different Indic languages.

VIII. The absence of extensive user studies or deployment feedback limits the assessment of the framework's usability, scalability, and impact in practical settings, especially in low-resource and multilingual environments such as rural healthcare.

9.2 Future Directions

Several promising avenues exist for extending the research presented in this thesis. The ASHA-FAQ chatbot developed in Chapter 8 necessitates evaluation in a real-life scenario. Given the demonstrated potential of chatbots in mental health support [410] and the ability of LLMs to aid on-ground health workers [411], a real-world deployment would provide invaluable insights. Furthermore, the development of an ever-evolving, curated FAQ database will enhance the explainability of future FAQ-based chatbots.

Future works should explore different methods to merge fine-grained triples to create more informative representations in IndIE method. Further linguistic efforts are needed to analyze and capture appositive relationships in agglutinative Indic languages like Tamil and Telugu. Expanding the golden triples in Hindi-BenchIE and developing similar benchmarks

for other Indic languages is crucial for advancing OIE in Indic languages. Moreover, the viability of OIE-based approaches should be explored for sentences exceeding the capability of transformer-based models.

To enhance the performance of the wl-coref checkpoint, presented in Chapter 4, for automatic coreference resolution in multiple languages, several directions can be pursued. Data preprocessing for training the wl-coref model requires dependency parsing to identify the syntactic head-word of each mention. In languages lacking dependency parsing tools, alternative approaches to identifying head-words should be investigated[1]. Accurate head-word identification will facilitate fine-tuning the wl-coref model across diverse languages, thereby improving its efficacy.

While MuNfQuAD, introduced in Chapter 5, is a large-scale multilingual non-factoid QA dataset, it contains a significant proportion of factoid-based questions. A multilingual answer span extractor can be used to provide silver labels for minimal answer spans within MuNfQuAD. Furthermore, the fine-tuned APS model can serve as a reranking module in a multilingual Retrieval-Augmented Generation (RAG) pipeline [412, 413].

Chapter 6 presented a comparison of post-hoc explainable methods with LRP, observing that our implementation, restricted to fine-tuning layers, did not outperform baselines. However, recent work has shown LRP's effectiveness when relevance values are calculated through internal transformer layers [414]. This direction warrants further exploration.

A promising direction for future work involves evaluating the proposed pipeline on multilingual datasets with longer contexts than MuNfQuAD. Extracting text from school textbooks has proven effective for curating high-quality datasets in low-resource languages [415, 416, 417, 418]. Additionally, the APS model can serve as a reward model to align LLMs for generating more informative answers, leveraging differential performance preference tuning algorithms [419, 420].

Future works could also broaden language coverage and conduct deeper per-language error analyses to better understand model limitations across diverse linguistic phenomena. Increasing the scale and granularity of human evaluation, particularly for complex reasoning cases, will also provide stronger evidence in support of the current findings. To assess robustness, it is important to evaluate the proposed methods on closed-domains such as social media, legal text, and literary corpora.

[1] `https://github.com/vdobrovolskii/wl-coref/issues/12`

Appendix

This appendix serves as a repository for supplementary materials that, while not central to the core narrative of this thesis, offer valuable context and detail. It includes detailed experimental results, additional code snippets, annotation portal screenshots, and extended examples that enhance the understanding of the research presented. These materials provide further evidence and support for the methodologies and findings discussed in the main body of the thesis, allowing for a more comprehensive review of the work.

Elephant seal [421].

A.1 Literature Survey

The URLs mentioned in Table 1 were used to obtain the distribution of QA literature.

Source	URL
ACL Anthology	`https://gist.github.com/ritwikmishra/f95746d829f63a925efd0091e2a438d8`
ScienceDirect	`https://www.sciencedirect.com/search?qs=%22question%20answering%22&title=question%20answering&sortBy=date`
Springer Link	`https://link.springer.com/search?new-search=true&query=%22question+answering%22&advancedSearch=true&dateFrom=&dateTo=&title=question+AND+answering&sortBy=oldestFirst`
IEEE Xplorer	`https://ieeexplore.ieee.org/search/searchresult.jsp?action=search&matchBoolean=true&queryText=(%22Document%20Title%22:%20%22question%20answering%22)&highlight=true&returnFacets=ALL&returnType=SEARCH&matchPubs=true&sortType=oldest`
ACM Digital Library	`https://dl.acm.org/action/doSearch?fillQuickSearch=false&target=advanced&expand=dl&field1=Title&text1=%22question+answering%22&sortBy=EpubDate_asc&startPage=`

Table 1: URLs to obtain the literature containing the term 'question answering' in their titles.

A.2 Hindi Chatbot

	MRR	mAP	SR	P@3	nDCG
DTP_{q-e}	**48.5**	**35.1**	**59.6**	**30.0**	**51.2**
-any pruning	37.3	25.5	45.2	21.3	39.1
-advice removal	43.1	30.3	54.4	27.0	46.0
-node removal	40.9	28.4	53.7	27.0	44.2
-compound merging	44.4	31.1	55.1	27.0	47.0

Table 2: DTP ablation of various pruning techniques. When no pruning technique is applied, only basic lemmatization, elimination of stop words, and token-level matching are carried out.

A.3 IndIE

A.3.1 Illustration

The complete dataflow of the proposed framework is demonstrated through the following unprocessed (multi-sentence) Hindi text: शर्मीला टैगोर कक बक्टक सैफ अली खान को 2010

मकं पद्म श्री पुरस्कार मिला। वह एक भारीतय अभिनक्ता है। [*sharmila taegor ke bete saef ali khAn ko 2010 me padm shri puraskAr mila. veh ek bhArtiye abhinetA hae*] *(Son of Sharmila Tagore, Saif Ali Khan, was awarded with Padma Shri award in 2010. He is an Indian actor).* The unprocessed input is provided to the Stanza library for sentence-level segmentation and dependency analysis. In phase (a) of the framework, depicted in Figure 3.1, the segmented sentences are forwarded to the chunking module to identify chunk labels corresponding to each token in the sentence. The output chunked phrases from phase (a) are as follows:

Sentence 1 - {शर्मीला टैगोर के [*sharmila taegore ke*] *(Sharmila Tagore's)*}_NP , {बेटे [*bete*] *(son)*}_NP , {सैफ अली खान को [*saef ali khAn ko*] *(to Saif Ali Khan)*}_NP , {2010 में [*2010 me*] *(in 2010)*}_NP ,
{पद्म श्री पुरस्कार [*padm shri puraskAr*] *(Padma Shri award)*}_NP , {मिला [*mila*] *(awarded)*}_VGF

Sentence 2 - {वह [*veh*] *(He)*}_NP , {एक भारीतय अभिनेता [*ek bhArtiye abhinetA*] *(an Indian actor)*}_NP , {है [*hae*] *(is)*}_VGF

The chunked phrases and dependency structure for each sentence are forwarded to stage (b) of the framework to build the *MDT* corresponding to that sentence. Figure 1 depicts the *MDTs* produced at the output of stage (b). For every sentence, triples are derived using its respective *MDT* and a set of manually designed rules. Table 4 presents all the triples extracted by the *IndIE* system from the raw text mentioned earlier. The output of stage (c) includes the following three components: (i) a sequence of segmented sentences, (ii) the extracted triples, and (iii) the processing time for each sentence.

A.3.2 Chunking Baselines

We employed the Python library scikit-learn[2] to develop the CRF model. The feature set extracted for each word in the sentence included: (a) a bias term $\leftarrow 1.0$, (b) the word itself, (c) the POS tag of the word, (d) the POS tags of the two preceding words, and (e) the POS tags of the two following words. The L1 and L2 regularization parameters were selected via grid search. For the HMM model, we utilized the word along with its POS tag as input features.

Our customized chunker operates as a complete pipeline for chunking, as it directly processes unstructured textual input. In contrast, CRF and HMM-based models require sentences that have already been annotated with POS tags. The chunk-labelled data provided by Bhat et al. [202] is annotated using the UPOS tagging convention [422], whereas the corpus from Jha [203] follows the AnnCorra tagging system [199], an adaptation of the Penn Treebank tags designed specifically for Indian languages. Since a public POS tagger for AnnCorra is not available, we constructed a conversion scheme from AnnCorra to the UPOS tagset to bring uniformity to the POS annotations across the dataset. To this end,

[2] `https://sklearn-crfsuite.readthedocs.io/en/latest/tutorial.html`

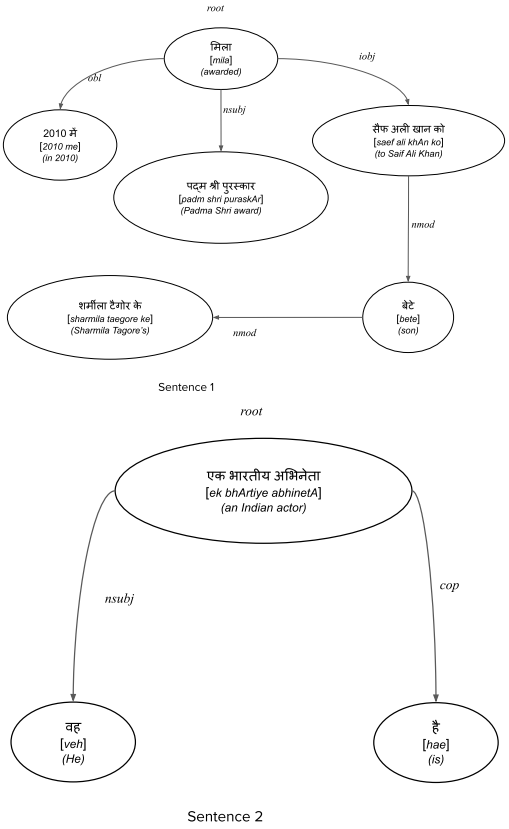


Figure 1: The resultant *MDTs* obtained after performing sentence segmentation, chunking, and dependency analysis on the following unprocessed text: शर्मीला टैगोर कक बक्टक सैफ अली खान को 2010 मकं पद्म श्री पुरस्कार मिला। वह एक भारीतय अभिनक्ता है। [*sharmila taegor ke bete saef ali khAn ko 2010 me padm shri puraskAr mila. veh ek bhArtiye abhinetA hae*] *(Son of Sharmila Tagore, Saif Ali Khan, received the Padma Shri honor in 2010. He is an Indian film artist).*

all Hindi and English entries from the [**?**] corpus were processed using the Stanza toolkit, which generates POS tags in the UPOS scheme. This mapping from AnnCorra (Penn-based tags) to UPOS enabled us to harmonize the POS tagging format throughout the dataset.

A.3.3 Chunking Ablation

We tested three strategies for managing sub-word token embeddings and found that averaging the sub-word token embeddings resulted in higher accuracy compared to using the first or last sub-word token embedding, as illustrated in Table 5. Although averaging sub-word token embeddings involves additional processing, it leads to longer fine-tuning (6 hours per epoch) and inference times (29 milliseconds per sentence) compared to the fine-tuning time (45 minutes per epoch) and inference time (17 milliseconds per sentence) for the other two methods. Therefore, if accuracy is prioritized over inference speed, we recommend averaging the sub-word token embeddings for sequence labeling tasks. On the other hand, if minimiz-

label: sent_91
sentence: अभ्रक आदि की खानों में मोमबत्तियाँ भी प्रयुक्त होती है .
[*abhrak Adi ki khAno me mombattiya bhi prayukt hoti hae*]
(*In the mines of Mica candles are also used*)

ppatt:
अभ्रक की खानों में ?a प्रयुक्त होती है
?a: मोमबत्तियाँ
[*abhrak ki khAno me ?a prayukt hoti hae*
?a: mombattiya]
(*In the mines of Mica ?a are used*
?a: candles)

Figure 2: The output of *PredPatt* on a Hindi sentence अभ्रक आदि की खानों मक़ं मोमबत्तियाँ भी प्रयुक्त होती है [*abhrak Adi ki khAno me mombattiya bhi prayukt hoti hae*] (*In the mines of Mica, candles are also utilized*). In this sentence, *candles* serves as the *Entity1*, denoted by the '?a' format.

ing inference time is crucial, selecting the last sub-word token embedding outperforms the traditional approach of using the first sub-word token embedding in terms of performance and inference time. In our study, we utilized the embeddings from the `last_hidden_state` of the model. However, considering that some studies suggest the early layers of a transformer capture more superficial text features [408], we also experimented with averaging embeddings from the first two hidden layers of the model. Our results indicated that using embeddings from the early layers actually reduced the accuracy (86%) for the chunking task. Consistent with the findings of Jain et al. [423], we observed that xlm-roberta-base [216] achieved the highest accuracy (92%) among the pretrained models.

A.3.4 PredPatt

Figure 2 displays the result of *PredPatt* applied to a Hindi sentence. Table 6 lists the rules we created to transform the *PredPatt* output into a triples format.

A.3.5 Algorithm

Algorithm 1 Triple generator algorithm from MDT

function EXTRACT(MDT, t, Q)
  if '*cop*' $\in t.children.dep\_rel$ **then**
    if $|t.children| \leq 2$ **then**
      $Head, Tail \leftarrow$ FIND_HEAD(MDT, t) , t
      $Rel \leftarrow x$ where $(x \in t.children \wedge x.dep\_rel =$ '*cop*')
    else
      $Head, Tail \leftarrow$ FIND_HEAD(MDT, t) , FIND_TAIL(MDT, t)
      $Rel \leftarrow t + x$ where $(x \in t.children \wedge x.dep\_rel =$ '*cop*')
    end if
  else if '*advcl*' $== t.dep\_rel$ **then**
    $Head \leftarrow q.Tail + q.Rel$ where $(q \in Q \wedge t.parent \in q)$
    $Rel, Tail \leftarrow t$, FIND_TAIL(MDT, t)
  else if '*acl*' $== t.dep\_rel$ **then**
    $Head \leftarrow t.closest\_phrase(q.Tail,\ q.Head)$ where $(q \in Q \wedge t.parent \in q)$
    $Rel, Tail \leftarrow t$, FIND_TAIL(MDT, t)
  else if '*conj*' $== t.dep\_rel$ **then**
    if $\exists q \in Q$ such that $q.Head == t.parent$ **then**
      $Head, Rel, Tail \leftarrow t$, $q.Rel$, $q.Tail$
    else if $\exists q \in Q$ such that $q.Tail == t.parent$ **then**
      $Head, Rel, Tail \leftarrow q.Head$, $q.Rel$, t
    end if
  else
    if $t.is\_clausal() == True$ **then**
      $Head, Rel, Tail \leftarrow t.pronoun$, $t.verb$, $t - (t.pronoun \cap t.verb)$
    else
      if $t.is\_a\_relationship() == True$ **then** //appositive relationship
        $Head, Rel, Tail \leftarrow t$, FIND_TAIL(MDT, t) , $t.is\_a\_label$
      else
        $Head, Rel, Tail \leftarrow$ FIND_HEAD(MDT, t) , t , FIND_TAIL(MDT, t)
      end if
    end if
  end if
  if $Head, Rel, Tail$ **then**
    $Q.add([Head,\ Rel,\ Tail])$
  end if
  if $t.contain\_args() == True$ **then**
    $Q \leftarrow$ EXTRACT(MDT, t)
  else
    for each $t_c \in t.children$ **do**
      $Q \leftarrow$ EXTRACT(MDT, t_c)
    end for
  end if
  return Q
end function

$triples \leftarrow$ EXTRACT($MDT, t_{root}, \{\}$)

A.4 TransMuCores

A.4.1 Compute Resources Needed

The GPU memory usage for the awesome-align model, *facebook/nllb-200-1.3B*, and *facebook/nllb-200-3.3B* models are 2GB, 7GB, and 14GB, respectively. On average, the translation model requires 3 seconds per sentence, while the alignment model needs 30 milliseconds for each sentence. Consequently, the projected time to re-build TransMuCoRes on a single GPU is about 3 months. During the fine-tuning stage, the memory requirements for the wl-coref and fast-coref models are 30GB and 8GB, respectively. However, during the inference phase, they reduce to 5GB and 1GB, respectively. The wl-coref model takes 45 minutes per epoch, while the fast-coref model completes 100K steps in 6 hours.

A.5 MuNfQuAD

A.5.1 Manual Annotations

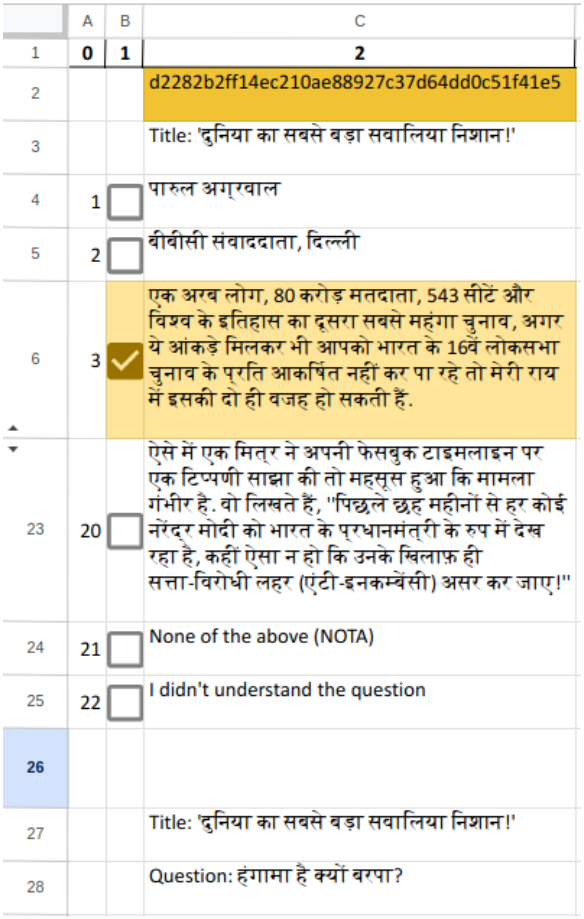
Individuals were enlisted as human annotators, and before their engagement, details regarding their remuneration for participating in the study were communicated. Every annotator possessed a bachelor's qualification and was a native speaker of a language pertinent to our research focus. The annotation task was carried out using Google Sheets. Each participant received a customized sheet containing rows that displayed the article's headline, its body text, and a related question. Checkboxes were embedded within the paragraph entries, and annotators were instructed to mark those corresponding to segments that addressed the posed question. If the question was ambiguous or no paragraph provided a suitable response, relevant alternatives were shown beneath the article content. Google Apps Script enabled macro execution on these sheets to visually emphasize the selections made. Figure 3 illustrates the annotation layout presented to a Hindi-speaking annotator.

A.5.2 APS baselines on the golden set

The baseline APS systems were assessed using the MuNfQuAD gold-standard dataset. The sbert and TF-IDF methods were chosen as benchmarks, as they employ contrasting techniques relative to our fine-tuned APS approaches. The corresponding outcomes are shown in Table 12.

A.5.3 Phrases for Excluding Criterion

In this analysis, several subheadings ending with a question mark but lacking relevance to the article's main text were discovered. These subheadings were pinpointed by analyzing the most commonly occurring interrogative subheadings within each respective language. Table 13 on the following page presents a compilation of these phrases. If an interrogative



	A	B	C
1	0	1	2
2			d2282b2ff14ec210ae88927c37d64dd0c51f41e5
3			Title: 'दुनिया का सबसे बड़ा सवालिया निशान!'
4	1	☐	पारुल अग्रवाल
5	2	☐	बीबीसी संवाददाता, दिल्ली
6	3	☑	एक अरब लोग, 80 करोड़ मतदाता, 543 सीटें और विश्व के इतिहास का दूसरा सबसे महंगा चुनाव, अगर ये आंकड़े मिलकर भी आपको भारत के 16वें लोकसभा चुनाव के प्रति आकर्षित नहीं कर पा रहे तो मेरी राय में इसकी दो ही वजह हो सकती हैं.
23	20	☐	ऐसे में एक मित्र ने अपनी फेसबुक टाइमलाइन पर एक टिप्पणी साझा की तो महसूस हुआ कि मामला गंभीर है. वो लिखते हैं, "पिछले छह महीनों से हर कोई नरेंद्र मोदी को भारत के प्रधानमंत्री के रुप में देख रहा है, कहीं ऐसा न हो कि उनके खिलाफ़ ही सत्ता-विरोधी लहर (एंटी-इनकम्बेंसी) असर कर जाए!"
24	21	☐	None of the above (NOTA)
25	22	☐	I didn't understand the question
26			
27			Title: 'दुनिया का सबसे बड़ा सवालिया निशान!'
28			Question: हंगामा है क्यों बरपा?

Figure 3: Annotation interface showing a hand-labeled segment from a Hindi news report, available at: `https://www.bbc.com/hindi/india/2014/05/140428_election_fatigue_social_media_pa`. Please note that some entries have been hidden to maintain brevity.

subheading included any of the listed expressions, it was excluded from being marked as a question in MuNfQuAD.

A.5.4 APS Model Ablations

A series of ablation experiments were performed during the fine-tuning phase of the APS model, exploring a range of hyperparameter variations. These trials involved appending preceding context to the paragraphs up to the model's maximum token capacity, combining the article headline with the query, replacing weighted focal loss (wfl) with weighted binary cross-entropy loss (wbce), and embedding positional encodings (PE) alongside contextual paragraphs. The corresponding findings from these studies, based on a portion of the fine-tuning dataset, are shown in Table 14.

A.5.5 APS models on Non-Factoid Questions

As the English versions of all MuNfQuAD queries were passed through the fine-tuned question categorization model developed by Bolotova et al. [5], we refined the MuNfQuAD evaluation set to retain only those queries that the model classified into a category other than FACTOID. Subsequently, the APS model was applied to this curated subset, and the corresponding results are presented in Table 15.

A.5.6 LLM as APS model

An examination of various multilingual LLMs as APS models was conducted, and their effectiveness on the MuNfQuAD golden set was assessed, with the outcomes provided in Table 16. Our findings suggest that the highest results were achieved by Command-R from CohereForAI. Alongside multilingual LLMs, bilingual models like Hi-NOLIN[3] and OpenHathi[4], which are pretrained on Hindi and English data, were also evaluated. Unfortunately, the performance of these bilingual models in Hindi did not exceed that of C4Ai.

A.5.7 LLM Outputs for Abstractive QA

In our research, multiple LLMs were instructed to respond to the provided question using the content of the article. The responses produced by the various LLMs are displayed in Table 17 on the subsequent page.

A.6 LRP vs LIME vs SHAP

A comparison of different LRP variations on hate speech detection models utilizing distinct *encoders* is shown in Table 18.

[3] `https://huggingface.co/nolanoAI/Hi-NOLIN-9B`

[4] `https://huggingface.co/sarvamai/OpenHathi-7B-Hi-v0.1-Base`

Fine-tuning data	**Pretrained Encoder**	**Linear Layers**	**Frozen Layers**	**Best Epoch**	**SR spread** ($\mu \pm \sigma$)	**Best SR**
Inshorts 3 epoch AshaQs 6 epoch	mbert-cased	3	embedding, layer 0	4	62.6 ± 1.9	66.2
AshaQs 4 epoch	mbert-cased	3	embedding, layer 0	1	▼ 62.4 ± 3.7 ▲	67.0 ▲
Inshorts 4 epoch	mbert-cased	3	embedding, layer 0	1	▼ 55.5 ± 3.5 ▲	60.0 ▼
Inshorts 1 epoch **AshaQs 1 epoch**	mbert-cased	3	embedding, layer 0	2	▲ 64.3 ± 2.4 ▲	67.8 ▲
Inshorts 3 epoch AshaQs 6 epoch	**xlm-roberta**	3	embedding, layer 0	4	▼ 61.3 ± 2.5 ▲	65.2 ▼
Inshorts 2 epoch **AshaQs 1 epoch**	mbert-cased	3	embedding, layer 0	3	▲ 62.9 ± 1.5 ▼	65.9 ▼
Inshorts 4 epoch **AshaQs 1 epoch**	mbert-cased	3	embedding, layer 0	5	▼ 61.0 ± 2.9 ▲	64.1 ▼
Inshorts 3 epoch AshaQs 1 epoch	**indic-bert**	3	embedding, layer 0	4	▼ 5.9 ± 0.8 ▼	7.0 ▼
Inshorts 3 epoch AshaQs 1 epoch	**mbert** **-uncased**	3	embedding, layer 0	4	▼ 60.0 ± 4.3 ▲	65.9 ▼
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	**2**	embedding, layer 0	4	▼ 60.6 ± 1.7 ▼	63.0 ▼
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	**4**	embedding, layer 0	4	▼ 61.3 ± 2.5 ▲	64.8 ▼
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	**1**	embedding, layer 0	4	▼ 57.8 ± 2.5 ▲	61.9 ▼
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	3	**embedding**	4	▼ 61.3 ± 3.4 ▲	66.7 ▲
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	3	**embedding,** **layer 0, 1**	4	▼ 61.3 ± 2.4 ▲	63.0 ▼
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	3	**embedding,** **layer 0, 1, 2**	4	▼ 61.1 ± 2.2 ▲	63.7 ▼
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	3	**half bert**	4	▼ 61.9 ± 2.1 ▲	64.4 ▼
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	3	**nothing**	4	▼ 61.5 ± 2.1 ▲	65.6 ▼

Table 3: Impact of architectural variations on the *SPC* method's sensitivity. Each configuration is evaluated using ten distinct random seeds. To maintain conciseness, we report the SR as a proxy for overall effectiveness, as it consistently serves as an upper limit for all evaluation criteria in our studies. The top row in the table outlines the architecture corresponding to the highest-performing *SPC* variant. Red-colored triangles (▲/▼) indicate a decline in performance relative to the top-performing model. Note: a higher standard deviation (σ) suggests greater numerical instability and, consequently, degraded performance. The absence of any row entirely marked by green triangles implies that the first-row configuration remains optimal.

	<head>	***<relation>***	***<tail>***
Sentence 1	पद्म श्री पुरस्कार [*padm shri puraskAr*] *(Padma Shri award)*	मिला [*mila*] *(awarded)*	सैफ अली खान को [*saef ali khAn ko*] *(to Saif Ali Khan)*
	2010 में [*2010 me*] *(in 2010)*	मिला [*mila*] *(awarded)*	पद्म श्री पुरस्कार [*padm shri puraskAr*] *(Padma Shri award)*
	सैफ अली खान को [*saef ali khAn ko*] *(to Saif Ali Khan)*	है [*hae*] *(is)*	बेटे [*bete*] *(son)*
	बेटे [*bete*] *(son)*	है [*hae*] *(is)*	शर्मीला टैगोर के [*sharmila taegore ke*] *(Sharmila Tagore's)*
Sentence 2	वह [*veh*] *(He)*	है [*hae*] *(is)*	एक भारीतय अभिनेता [*ek bhArtiye abhinetA*] *(an Indian actor)*

Table 4: Triples extracted through hand-crafted rules of the proposed IndIE tool for the following raw text in Hindi: शर्मीला टैगोर के बेटे सैफ अली खान को 2010 में पद्म श्री पुरस्कार मिला। वह एक भारीतय अभिनेता है। [*sharmila taegor ke bete saef ali khAn ko 2020 me padm shri puraskAr mila. veh ek bhArtiye abhinetA hae*] *(Son of Sharmila Tagore, Saif Ali Khan, was awarded with Padma Shri award in 2010. He is an Indian actor).*

Classification Layers	First sub-word token embedding	Average embedding of all sub-word tokens	Last sub-word token embedding
1	82±10 (50±20)	**91±0.0** (**65±0.5**)	89±0.5 (62±1.0)
2	86±1.8 (51±6.2)	**90±0.5** (**54±4.5**)	89±0.5 (54±7.4)
3	79±14 (43±13)	**90±0.5 (48±2.2)**	82±11 (41±12)

Table 5: An evaluation of three distinct methods to address sub-word token representation in the chunking task. The average and standard deviation were computed using four separate random seed values. All experiments were conducted using the merged dataset of Jha [203] and the corpus provided by Bhat et al.[202]. Values placed outside parentheses indicate accuracy, while those enclosed in parentheses denote the macro-average score.

Rule No.	**Sentence Structure**	**Extracted Triple**
1	^ phrase1 Entity1 phrase2 $	<phrase1 , phrase2 , Entity1>
2	^ Entity1 phrase1 Entity2 phrase2 $ ^ phrase1 Entity1 Entity2 $ ^ Entity1 Entity2 phrase1 $ ^ Entity1 phrase1 Entity2 $	<Entity1 , phrase1 , Entity2>
3	^ phrase1 Entity1 Entity2 phrase2 $ ^ phrase1 Entity1 phrase2 Entity2 phrase3 $ ^ phrase1 Entity1 phrase2 Entity2 $	<Entity1 , phrase2 , Entity2>
4	Any other sentence structure	Discard

Table 6: The guidelines we followed to derive triples from the *PredPatt* output. Referring to the example sentence shown in Figure 2, Rule number 1 is implemented for the sentence. The symbol ^ marks the beginning of the sentence, while the symbol $ denotes its conclusion.

Language	**Translation Sanity-check**	
	Passed	**Failed**
awa_Deva	102828	3
asm_Beng	102828	3
ben_Beng	102825	6
bho_Deva	102829	2
dzo_Tibt	102827	4
bod_Tibt	102828	3
guj_Gujr	102828	3
hin_Deva	102827	4
kan_Knda	102828	3
hne_Deva	102829	2
mya_Mymr	102827	4
kas_Arab	102831	0
mai_Deva	102828	3
mag_Deva	102827	4
mal_Mlym	102826	5
mar_Deva	102826	5
mni_Beng	102829	2
npi_Deva	102828	3
ory_Orya	102829	2
pbt_Arab	102830	1
pan_Guru	102829	2
prs_Arab	102827	4
sat_Beng	102828	3
sin_Sinh	102829	2
tgk_Cyrl	102826	5
uig_Arab	102829	2
snd_Arab	102817	14
tam_Taml	102827	4
tel_Telu	102823	8
urd_Arab	102828	3
uzn_Latn	102829	2
Total	3187650	111

Table 7: Count of translated sentences that passed/failed the validity assessment. The validity check involved determining if the translated sentence consisted of a redundant series of punctuation marks or not.

Language	Mentions (223583)		
	Aligned	**Misaligned**	**Non-aligned**
asm_Beng	66.1%	15.3%	18.6%
awa_Deva	71.3%	15%	13.7%
guj_Gujr	86.7%	7.1%	6.2%
ben_Beng	87.4%	6.7%	5.8%
hin_Deva	87%	7.8%	5.1%
bho_Deva	72.6%	14.5%	12.9%
bod_Tibt	69.7%	4.5%	25.7%
dzo_Tibt	62.8%	7.9%	29.3%
hne_Deva	70.8%	14.6%	14.6%
mai_Deva	69.7%	13.8%	16.4%
mal_Mlym	78.9%	8.6%	12.5%
kan_Knda	86.3%	6.6%	7.1%
urd_Arab	83.4%	10.9%	5.7%
sin_Sinh	48%	10.1%	41.9%
kas_Arab	59.8%	18.7%	21.4%
mag_Deva	73.6%	13%	13.4%
mar_Deva	84.5%	7.1%	8.5%
npi_Deva	86.1%	6.8%	7.1%
tel_Telu	84.7%	7.6%	7.6%
tgk_Cyrl	80.9%	8.9%	10.2%
mni_Beng	57%	19.7%	23.3%
mya_Mymr	79.8%	7.7%	12.4%
uzn_Latn	77.4%	8.4%	14.1%
ory_Orya	46.1%	10%	43.9%
pan_Guru	85.2%	9.4%	5.4%
pbt_Arab	58%	22.4%	19.6%
prs_Arab	82.9%	8.4%	8.8%
sat_Beng	52.3%	12.7%	34.9%
uig_Arab	57.2%	17.9%	24.9%
snd_Arab	56.7%	22.3%	21.1%
tam_Taml	83.5%	7.4%	9.1%
Total	72.5%	11.4%	16.2%

Table 8: The effectiveness of the word-alignment tool across different languages. A mention refers to a continuous sequence of words. It is regarded as aligned if every word in the mention corresponds to a continuous sequence of words in the target language. If the alignment occurs with a non-continuous sequence of words in the target language, it is termed as misaligned. When a mention is not associated with any word in the target language, it is classified as non-aligned.

Language	#sents	#mentions	#coreference clusters	#split antecedants	#singletons	#docs
asm_Beng	(58706 , 7174 , 8194)	(113275 , 13882 , 16387)	(35734 , 4671 , 5182)	(2559 , 337 , 387)	(12986 , 1731 , 2089)	(2835 , 352 , 365)
ben_Beng	(63812 , 7830 , 8862)	(151825 , 18931 , 22395)	(39523 , 5171 , 5834)	(1116 , 115 , 123)	(8475 , 1136 , 1643)	(2843 , 352 , 366)
bod_Tibt	(58372 , 7157 , 8046)	(84334 , 10549 , 12085)	(36918 , 4791 , 5414)	(22268 , 2491 , 3085)	(12046 , 1603 , 2104)	(2840 , 351 , 363)
dzo_Tibt	(56040 , 6837 , 7684)	(95515 , 11736 , 13648)	(35182 , 4551 , 5059)	(11767 , 1276 , 1584)	(13404 , 1795 , 2183)	(2827 , 351 , 364)
guj_Gujr	(63687 , 7805 , 8835)	(150276 , 18687 , 22182)	(39279 , 5125 , 5788)	(1135 , 112 , 130)	(8744 , 1142 , 1623)	(2843 , 352 , 366)
awa_Deva	(59296 , 7257 , 8212)	(123270 , 15298 , 18003)	(36869 , 4791 , 5333)	(1627 , 176 , 200)	(11965 , 1542 , 1958)	(2840 , 352 , 366)
hin_Deva	(63841 , 7820 , 8883)	(151922 , 18982 , 22373)	(39129 , 5157 , 5775)	(624 , 58 , 56)	(8862 , 1198 , 1689)	(2843 , 352 , 366)
kas_Arab	(55735 , 6733 , 7765)	(102470 , 12520 , 14932)	(34261 , 4433 , 4956)	(2476 , 270 , 321)	(14038 , 1806 , 2150)	(2834 , 350 , 363)
bho_Deva	(59942 , 7378 , 8344)	(125719 , 15709 , 18574)	(36989 , 4897 , 5436)	(1317 , 140 , 169)	(11967 , 1613 , 1972)	(2837 , 352 , 363)
npi_Deva	(63471 , 7784 , 8817)	(149191 , 18586 , 22003)	(39359 , 5141 , 5817)	(1294 , 134 , 151)	(8819 , 1165 , 1671)	(2841 , 352 , 365)
ory_Orya	(47958 , 5775 , 6626)	(78750 , 9367 , 11132)	(29454 , 3844 , 4162)	(2720 , 315 , 443)	(14003 , 1905 , 2079)	(2822 , 348 , 362)
mag_Deva	(60451 , 7403 , 8409)	(127402 , 15940 , 18708)	(37356 , 4922 , 5470)	(1420 , 146 , 179)	(11762 , 1578 , 2001)	(2841 , 352 , 365)
mai_Deva	(59284 , 7260 , 8243)	(120323 , 14820 , 17608)	(36704 , 4760 , 5341)	(1927 , 202 , 280)	(12422 , 1588 , 2034)	(2838 , 351 , 365)
mal_Mlym	(62141 , 7633 , 8622)	(133610 , 16648 , 19613)	(38726 , 5086 , 5678)	(3451 , 360 , 459)	(9749 , 1355 , 1786)	(2842 , 352 , 364)
tgk_Cyrl	(62447 , 7650 , 8670)	(139803 , 17348 , 20413)	(38981 , 5064 , 5701)	(2052 , 241 , 286)	(9693 , 1301 , 1742)	(2843 , 352 , 365)
mar_Deva	(63279 , 7732 , 8783)	(145616 , 18050 , 21406)	(39152 , 5103 , 5748)	(1557 , 165 , 184)	(9199 , 1225 , 1710)	(2843 , 352 , 365)
mni_Beng	(54556 , 6722 , 7642)	(96202 , 11657 , 13953)	(32757 , 4266 , 4714)	(3230 , 383 , 451)	(13896 , 1861 , 2143)	(2829 , 351 , 362)
mya_Mymr	(62366 , 7633 , 8659)	(131466 , 16418 , 19073)	(38975 , 5088 , 5756)	(6403 , 707 , 935)	(9412 , 1208 , 1802)	(2844 , 351 , 366)
pan_Guru	(63448 , 7776 , 8842)	(148484 , 18395 , 21907)	(38776 , 5035 , 5705)	(701 , 85 , 83)	(9575 , 1200 , 1733)	(2839 , 352 , 366)
hne_Deva	(59393 , 7304 , 8270)	(122714 , 15269 , 17977)	(36854 , 4856 , 5375)	(1378 , 146 , 214)	(12232 , 1621 , 1983)	(2843 , 352 , 364)
kan_Knda	(63815 , 7822 , 8897)	(149004 , 18501 , 22007)	(39588 , 5172 , 5862)	(1645 , 171 , 215)	(8515 , 1143 , 1653)	(2843 , 352 , 365)
pbt_Arab	(54706 , 6582 , 7599)	(100295 , 12038 , 14467)	(32813 , 4211 , 4687)	(1772 , 225 , 253)	(13898 , 1800 , 2125)	(2826 , 350 , 363)
urd_Arab	(63010 , 7743 , 8759)	(146133 , 18133 , 21421)	(38494 , 5053 , 5617)	(616 , 68 , 64)	(10185 , 1364 , 1826)	(2840 , 352 , 365)
uzn_Latn	(61642 , 7534 , 8564)	(132967 , 16561 , 19387)	(38597 , 5041 , 5678)	(2333 , 272 , 352)	(10133 , 1359 , 1809)	(2842 , 352 , 366)
prs_Arab	(62579 , 7650 , 8722)	(144778 , 17816 , 21260)	(39040 , 5113 , 5729)	(1163 , 138 , 203)	(9825 , 1373 , 1807)	(2844 , 352 , 365)
sat_Beng	(50952 , 6129 , 6949)	(89870 , 10704 , 12679)	(31207 , 4049 , 4439)	(2676 , 324 , 405)	(13883 , 1835 , 2108)	(2827 , 349 , 361)
sin_Sinh	(48903 , 5930 , 6743)	(81864 , 9910 , 11615)	(30266 , 3947 , 4259)	(2866 , 364 , 439)	(13875 , 1867 , 2056)	(2818 , 350 , 362)
snd_Arab	(54718 , 6673 , 7596)	(98325 , 11871 , 14214)	(32806 , 4237 , 4691)	(2141 , 289 , 350)	(14142 , 1825 , 2153)	(2822 , 350 , 363)
tam_Taml	(63313 , 7765 , 8798)	(143782 , 17914 , 21120)	(39279 , 5127 , 5795)	(1774 , 162 , 223)	(9165 , 1197 , 1711)	(2842 , 352 , 365)
tel_Telu	(63707 , 7809 , 8868)	(146249 , 18198 , 21652)	(39306 , 5150 , 5820)	(1738 , 176 , 201)	(9062 , 1230 , 1749)	(2844 , 352 , 366)
uig_Arab	(54313 , 6611 , 7563)	(96106 , 11645 , 13899)	(33532 , 4337 , 4843)	(3922 , 457 , 519)	(14085 , 1833 , 2187)	(2831 , 350 , 362)
Total	(1839883 , 224911 , 255466)	(3821540 , 472083 , 558093)	(1135906 , 148189 , 165664)	(93668 , 10505 , 12944)	(350017 , 46399 , 59279)	(87946 , 10890 , 11294)

Table 9: The statistical data of TransMuCoRes for every language. The figures enclosed in parentheses denote the (training, validation, testing) partitions.

				Mentions			MUC			B^3			CEAFe			LEA			CoNLL
				P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	F1
wl-coref [12]	5 langs	mbert	test	68	**77**	72	61	70	65	44	**55**	49	46	50	48	41	**51**	45	54
			dev	65	**75**	70	57	67	61	43	54	48	46	52	49	39	49	44	53
		xlmr	test	**72**	**77**	**75**	**66**	**71**	**69**	**52**	**55**	**53**	**50**	**54**	**52**	**48**	**51**	**50**	**58**
			dev	**70**	**75**	**72**	**63**	**68**	**65**	**50**	**55**	**52**	**51**	**55**	**53**	**46**	**50**	**48**	**57**
	All langs	mbert	dev	40	66	50	32	56	41	23	**46**	31	28	42	33	20	**40**	27	35
			test	42	68	52	34	59	44	23	**47**	31	27	40	33	21	**42**	28	36
		xlmr	dev	48	60	53	40	51	45	31	40	35	**34**	42	37	27	36	31	39
			test	50	62	55	42	54	47	31	41	35	**33**	41	37	28	37	32	40
fast-coref [11]	All langs	mbert	dev	47	**73**	57	41	59	48	30	41	35	25	52	34	26	36	30	39
			test	50	74	60	44	62	52	31	42	35	25	51	33	27	37	32	40
		xlmr	test	**54**	**75**	**63**	**49**	**64**	**56**	**36**	44	**40**	29	**55**	**38**	**33**	40	**36**	**44**
			dev	**51**	**73**	**60**	**46**	**61**	**52**	**35**	44	**39**	29	**56**	**38**	**32**	39	**35**	**43**

Table 10: The effectiveness of wl-coref [12] and fast-coref [11] when singletons are disregarded during the assessment phase is examined. It is observed that the performance enhances for both models, suggesting that both encounter difficulties in identifying the singletons.

Language	fast-coref [11] vs wl-coref [12] on fine-tuned xlmr													
	Split	Mentions F1	MUC F1	B^3 F1	CEAFe F1	LEA F1	CoNLL F1	Split	Mentions F1	MUC F1	B^3 F1	CEAFe F1	LEA F1	CoNLL F1
asm_Beng	DEVLOPMENT	47 vs 46	38 vs 35	26 vs 24	28 vs 29	22 vs 19	31 vs 29	TEST	49 vs 47	41 vs 37	26 vs 23	28 vs 29	22 vs 19	32 vs 30
awa_Deva		51 vs 44	42 vs 32	29 vs 25	33 vs 32	25 vs 20	35 vs 30		53 vs 44	45 vs 33	29 vs 23	33 vs 31	25 vs 19	35 vs 29
kan_Knda		73 vs 72	65 vs 63	50 vs 51	50 vs 52	46 vs 46	55 vs 55		76 vs 74	68 vs 67	51 vs 52	49 vs 52	47 vs 48	56 vs 57
ben_Beng		75 vs 74	67 vs 66	52 vs 53	51 vs 52	48 vs 48	57 vs 57		78 vs 76	71 vs 70	53 vs 54	51 vs 52	50 vs 51	59 vs 59
bho_Deva		54 vs 42	46 vs 33	32 vs 24	34 vs 29	28 vs 20	37 vs 29		56 vs 45	49 vs 35	32 vs 24	33 vs 29	28 vs 20	38 vs 29
hin_Deva		75 vs 75	68 vs 68	52 vs 55	51 vs 55	48 vs 51	57 vs 59		77 vs 78	71 vs 72	53 vs 56	51 vs 55	50 vs 53	58 vs 61
bod_Tibt		62 vs 7	46 vs 3	17 vs 2	13 vs 5	13 vs 2	26 vs 3		63 vs 7	51 vs 3	19 vs 3	14 vs 5	16 vs 2	28 vs 4
dzo_Tibt		20 vs 4	15 vs 3	9 vs 2	7 vs 4	7 vs 1	10 vs 3		21 vs 6	16 vs 4	10 vs 3	9 vs 5	8 vs 2	12 vs 4
guj_Gujr		74 vs 74	66 vs 65	50 vs 52	50 vs 52	46 vs 48	55 vs 57		76 vs 76	69 vs 68	51 vs 52	49 vs 52	47 vs 48	56 vs 58
hne_Deva		49 vs 40	41 vs 29	29 vs 21	34 vs 28	25 vs 17	34 vs 26		50 vs 42	43 vs 31	28 vs 22	33 vs 28	25 vs 18	35 vs 27
mya_Mymr		62 vs 54	52 vs 42	37 vs 33	39 vs 39	32 vs 28	43 vs 38		64 vs 56	55 vs 46	37 vs 32	38 vs 39	32 vs 28	43 vs 39
kas_Arab		35 vs 14	27 vs 8	18 vs 7	22 vs 13	14 vs 6	22 vs 9		37 vs 15	30 vs 9	18 vs 7	21 vs 13	15 vs 5	23 vs 9
mag_Deva		55 vs 48	47 vs 38	33 vs 28	34 vs 33	29 vs 23	38 vs 33		58 vs 51	51 vs 41	33 vs 28	34 vs 32	30 vs 24	39 vs 34
mai_Deva		50 vs 40	42 vs 32	29 vs 23	32 vs 27	25 vs 19	34 vs 27		53 vs 41	45 vs 34	30 vs 23	31 vs 26	26 vs 19	36 vs 27
urd_Arab		71 vs 71	63 vs 64	48 vs 51	48 vs 52	44 vs 47	53 vs 56		73 vs 73	67 vs 67	48 vs 51	47 vs 51	45 vs 48	54 vs 56
mal_Mlym		67 vs 66	58 vs 56	43 vs 44	44 vs 48	39 vs 39	48 vs 49		70 vs 68	62 vs 60	45 vs 46	44 vs 48	40 vs 41	50 vs 51
mar_Deva		73 vs 72	65 vs 64	50 vs 52	49 vs 52	45 vs 48	54 vs 56		75 vs 74	68 vs 68	51 vs 53	48 vs 51	47 vs 49	56 vs 57
mni_Beng		32 vs 10	24 vs 4	14 vs 3	16 vs 7	10 vs 2	18 vs 5		35 vs 11	28 vs 6	14 vs 3	15 vs 6	11 vs 2	19 vs 5
npi_Deva		75 vs 74	68 vs 66	53 vs 53	52 vs 53	49 vs 49	57 vs 57		78 vs 76	72 vs 70	54 vs 55	52 vs 54	51 vs 51	59 vs 60
ory_Orya		25 vs 22	19 vs 13	11 vs 8	13 vs 13	9 vs 5	14 vs 11		28 vs 23	22 vs 15	11 vs 8	13 vs 13	9 vs 5	15 vs 12
pan_Guru		72 vs 71	64 vs 63	48 vs 50	48 vs 51	44 vs 45	53 vs 54		74 vs 73	68 vs 67	49 vs 51	48 vs 50	46 vs 47	55 vs 56
pbt_Arab		36 vs 29	31 vs 21	20 vs 13	22 vs 19	17 vs 10	24 vs 18		39 vs 31	34 vs 23	20 vs 13	22 vs 20	18 vs 10	25 vs 19
prs_Arab		70 vs 65	63 vs 56	47 vs 44	47 vs 47	43 vs 39	52 vs 49		72 vs 67	66 vs 60	47 vs 45	45 vs 45	44 vs 40	53 vs 50
sat_Beng		13 vs 5	10 vs 2	6 vs 2	5 vs 3	5 vs 1	7 vs 2		15 vs 6	12 vs 4	6 vs 2	5 vs 4	5 vs 2	8 vs 3
sin_Sinh		24 vs 22	18 vs 13	11 vs 8	14 vs 14	9 vs 5	14 vs 12		27 vs 23	22 vs 15	12 vs 8	14 vs 14	10 vs 6	16 vs 12
snd_Arab		33 vs 31	26 vs 22	16 vs 14	19 vs 20	13 vs 10	21 vs 18		34 vs 32	28 vs 24	16 vs 14	20 vs 20	13 vs 10	21 vs 19
tam_Taml		71 vs 71	63 vs 63	48 vs 51	48 vs 53	44 vs 46	53 vs 55		73 vs 73	66 vs 66	49 vs 51	47 vs 51	45 vs 47	54 vs 56
tel_Telu		72 vs 72	64 vs 64	49 vs 52	48 vs 52	45 vs 47	54 vs 56		74 vs 74	68 vs 68	51 vs 54	47 vs 52	47 vs 50	55 vs 58
tgk_Cyrl		62 vs 17	52 vs 11	37 vs 8	38 vs 14	33 vs 7	42 vs 11		64 vs 16	55 vs 11	38 vs 7	37 vs 13	34 vs 6	43 vs 10
uig_Arab		23 vs 25	15 vs 14	9 vs 10	13 vs 16	7 vs 6	12 vs 13		23 vs 26	17 vs 16	9 vs 10	12 vs 16	7 vs 6	13 vs 14
uzn_Latn		64 vs 60	54 vs 50	40 vs 39	44 vs 45	36 vs 35	46 vs 45		65 vs 62	57 vs 53	40 vs 39	42 vs 43	36 vs 35	46 vs 45
Mujadia et al. [8]		51 vs 79	45 vs 74	36 vs 67	36 vs 66	32 vs 64	39 vs 69		56 vs 80	51 vs 76	40 vs 69	35 vs 66	35 vs 66	42 vs 70
Overall		60 vs 53	52 vs 45	39 vs 35	38 vs 37	34 vs 31	43 vs 39		63 vs 55	56 vs 47	40 vs 35	38 vs 37	36 vs 32	44 vs 40

Table 11: The performance of wl-coref [12] and fast-coref [11] for each language in the absence of singletons is presented. It is observed that the performance shows an improvement across various languages, suggesting that both models face challenges in identifying singletons in all languages. This highlights the necessity for coreference resolution models with enhanced recall during the mention detection phase.

Lang	#Qs	sbert (miniLM) / TF-IDF				
		Acc	Label 0 F1	Label 1 F1	Macro F1	SR
hi	100	29/71	2/83	45/14	23/48	1.0/0.37
ur	50	20/80	0/89	33/10	17/49	1.0/0.22
pa	100	15/86	2/92	24/7	13/50	1.0/0.17
tm	100	13/86	1/92	23/8	12/50	0.98/0.15
np	100	34/66	3/80	50/4	26/32	0.98/0.10
bn	100	27/73	1/84	42/6	21/45	1.0/0.19
gu	100	22/78	1/88	36/6	22/47	1.0/0.25
tl	40	14/85	1/92	24/11	12/51	1.0/0.22

Table 12: Effectiveness of various APS benchmark models on the golden set. The findings indicate that *sbert* achieves a reduced macro F1 score, while the TF-IDF approach demonstrates a decreased SR.

om	Maaltu haasa'ama (What's being talked about)	**am**	ምን ማለት ነው እጅግ የደኸየትስ (What does it mean?)
gu	શું તમે આ વાંચ્યું (Have you read this?), તમે આ વાંચ્યું કે (Did you read this?)	**bn**	বিবিসি বাংলাদেশ সংলাপে চলতি (BBC Bangladesh is running on dialogue), ভিডিও (video), আপনার দল কি সেমিফাইনালে যেতে পারবে (Can your team make it to the semi-finals?)
fa	آیا می‌دانید (Do you know), آیا می‌دانید (Do you know)		
fr*	Le saviez-vous (Did you know)	**hi**	पढ़िए (Read)
mr	हे वाचलंत का (Did you read this?), हेही वाचलंत का (Did you also read this?), हेही पाहिलंत का (Have you seen this too?) तुम्ही हे वाचलं का (Did you read this?) ...	**si**	ඔබ කටාර් රාජ්‍යයේ හෝ මැදපෙරදිග කලාපයේ සිටින්නෙක්ද (Are you in the state of Qatar or in the Middle East region?)
ur	ویڈیو میں کیا ہے (What is in the video?),	**uk**	А ви знали (Did you know)
cy	gafodd drwydded deledu (got a TV license)	**pt**	Did you get it, Did you know

Table 13: Expressions used to filter out non-relevant question-like subheadings in news content spanning multiple languages. Language identifiers follow ISO 639-1 standards.

#	APS Model hyperparameters				Acc	Macro F1	Label 0 F1	Label 1 F1
	Prior Context	Title	Loss	PE				
1	True	False	*wfl* (γ=0.5)	True	65.5	59.9	74.9	44.9
2	True	*True*	wfl (γ=0.5)	False	61.1	57.1	70.2	44.1
3	True	False	wfl (γ=2)	*True*	64.9	59.7	74.2	45.2
4	True	False	wfl (γ=2)	False	65.0	59.8	74.2	45.4
5	True	False	wfl (γ=0.5)	*False*	64.7	59.2	74.1	44.4
6	True	False	*wbce*	False	65.6	60.1	74.9	45.3
7	*False*	False	wfl (γ=0.5)	False	56.3	51.9	66.4	37.4

Table 14: Ablation study outcomes for Determining the Best Hyperparameter Settings in APS Model Adaptation. A limited portion of the fine-tuning dataset was employed to investigate various hyperparameter choices. The findings indicate that configuration #1 yields the top Label-1 F1 metric.

APS Model	Acc.	Macro F1	Label-0 F1	Label-1 F1	SR
Ours (mLUKE)	19	16	0	32	1.0
TF-IDF	80	48	89	7	0.06
sbert (mpnet)	19	16	1	31	0.99
Ours (bloom)	47	46	55	36	0.91
sbert (miniLM)	19	17	2	32	0.99
Ours (mT5)	76	68	84	53	0.91
Ours (d-mBERT)	66	60	75	46	0.93
Ours (mE5)	79	70	86	55	0.89
Ours (mBERT)	73	67	81	52	0.93
Ours (XLM-R)	78	71	86	55	0.91
Ours (XLM-V)	80	71	87	55	0.89

Table 15: Effectiveness of APS systems on the MuNfQuAD evaluation dataset, excluding queries identified as FACTOID.

Model	Acc	Label 0 F1	Label 1 F1	Macro F1	SR
OpenHathi	24	7	35	21	0.98
Aya-101	79	88	2	45	0.06
Mistral	64	76	27	52	0.72
Llama-3	62	72	39	56	0.95
ChatGPT	65	75	39	57	0.94
Gemma 2b	26	17	34	25	0.97
Llama-2	44	51	34	43	0.92
BLOOM	73	84	7	46	0.18
Airavat	79	88	2	45	0.04
Gemma 7b	74	84	11	47	0.26
Hi-NOLIN	61	73	33	53	0.58
Command-R	68	77	43	60	0.96

Table 16: The effectiveness of different LLMs as APS models on the MuNfQuAD gold standard dataset. The instructed variants of Gemma models and `gpt-3.5-turbo-1106` for ChatGPT were employed. All model weights, except for those of ChatGPT, were accessible on the Huggingface model repository. Remarkably, the top macro F1 score on the gold standard set was obtained by Command-R from CohereForAI.

Prompt	Answer the question below using the article provided. Write your answer in Hindi. Article = अमरीकी अपील कोर्ट ने राष्ट्रपति डोनल्ड ट्रंप के विवादित ट्रैवेल बैन का बचाव करने वाले और उसे चुनौती देने वालों से कड़े सवाल पूछे हैं. इस प्रतिबंध के तहत सभी शरणार्थियों और सात मुस्लिम बहुल देशों के नागरिक अमरीका नहीं आ सकते हैं. हालांकि पिछले हफ्ते कोर्ट ने फ़िलहाल इस पर रोक लगा दी थी. तीन जजों के एक पैनल ने राष्ट्रपति की ताक़त को सीमित करने और सात देशों को आतंकवाद से जोड़ने पर सबूतों को लेकर कई सवाल खड़े किए हैं. कोर्ट ने यह भी पूछा है कि क्या इस फ़ैसले को मुस्लिम-विरोधी नहीं देखा जाना चाहिए. उम्मीद की जा रही है कि अगले हफ्ते सैन फ्रांसिस्को के नौवें अमरीकी सर्किट कोर्ट तरफ से इस पर कोई फ़ैसला आएगा. निर्णय चाहे जो भी हो पर ऐसा लग रहा है कि इस केस का निपटारा शायद सुप्रीम कोर्ट में ही होगा. मंगलवार को दोनों तरफ से इस मसले पर एक घंटे तक बहस हुई. इस केस में अमरीकी न्याय मंत्रालय भी शामिल है और उसने जजों से ट्रंप के प्रतिबंध आदेश को फिर से बहाल करने की अपील की है. समाप्त वक़ील ऑगस्ट फ्लेज्जे ने कहा कि देश में कौन आए और कौन नहीं आए इस पर नियंत्रण रखने के लिए कांग्रेस ने राष्ट्रपति को अधिकार दिया है. उनसे उन सात देशों- इराक, ईरान लीबिया, सोमालिया, सूडान, सीरिया और यमन को लेकर पूछा गया कि ये देश फिलहाल अमरीका के लिए कैसे ख़तरा हैं. इस पर उन्होंने कहा कि अमरीका में कई सोमालियों के संबंध अल-शबाब ग्रुप से है. इसके बाद वॉशिंगटन प्रांत के एक वक़ील ने कोर्ट से कहा कि ट्रंप के कार्यकारी आदेश पर रोक से अमरीकी सरकार को कोई नुक़सान नहीं होगा. सॉलिसिटर जनरल नोआह पर्सेल ने कहा कि प्रतिबंध से उनके प्रांत के हज़ारो निवासी प्रभावित होंगे. जो छात्र वॉशिंगटन आने की कोशिश कर रहे हैं उन्हें भी बेमतलब की देरी का सामना करना होगा. इसके साथ ही अन्य लोग अपने परिवारों से मिलने अमरीका छोड़कर जाने से बचेंगे. सुनवाई के आख़िरी मिनटों में इस बात पर बहस हुई कि अगर यह प्रतिबंध मुस्लिमों को रोकने के लिए है तो यह असंवैधानिक होगा. जज रिचर्ड क्लिफ्टन ने दोनों पक्षों से इस मुद्दे पर पूछा कि इससे दुनिया के केवल 15 प्रतिशत मुसलमान प्रभावित होंगे. सोमवार की रात अमरीकी जस्टिस डिपार्टमेंट की तरफ से जारी 15 पन्नों के एक दस्तावेज़ में बताया गया है कि ट्रंप का यह कार्यकारी आदेश बिल्कुल निष्पक्ष है और इसका किसी ख़ास धर्म से कोई संबंध नहीं है. हालांकि मंगलवार को कोर्ट में पर्सेल ने ट्रंप के चुनावी कैंपेन के दौरान के बयानों का हवाला दिया. तब ट्रंप ने ग़ैरअमरीकी मुस्लिमों पर अस्थायी रूप से प्रतिबंध लगाने की बात कही थी. पर्सेल ने राष्ट्रपति के सलाहकार रुडी जुलियानी के बयान का भी उल्लेख किया. जुलियानी ने कहा था कि उन्होंने मुस्लिमों को अमरीका में काम करने पर क़ानूनन प्रतिबंध के लिए कहा है. क्लिफ्टन ने भी कहा कि जिन सात देशों पर प्रतिबंध लगाया गया है उनकी शिनाख्त पूर्ववर्ती ओबामा प्रशासन और कांग्रेस ने भी आतंक के डर के कारण वीज़ा पाबंदी के लिए की थी. उन्होंन कहा, "क्या आप यह भी मानते हैं कि पूर्ववर्ती ओबामा प्रशासन और कांग्रेस के फ़ैसले भी धार्मिक पूर्वाग्रह से प्रेरित थे? इस पर पर्सेल ने कहा, "नहीं, लेकिन राष्ट्रपति ट्रंप ने पूर्ण प्रतिबंध की बात कही थी. हालांकि यह पूर्ण प्रतिबंध नहीं है और यह भेदभावपूर्ण है." (बीबीसी हिन्दी के एंड्रॉएड ऐप के लिए आप यहां क्लिक कर सकते हैं. आप हमें फ़ेसबुक और ट्विटर पर फ़ॉलो भी कर सकते हैं.) Question = क्या यह मुस्लिम पर प्रतिबंध है या नहीं? (Question = Is this a ban on Muslims or not?)
ChatGPT	हाँ, यह मुस्लिम पर प्रतिबंध है क्योंकि इस प्रतिबंध के तहत सभी सात मुस्लिम बहुल देशों के नागरिकों को अमरीका नहीं आ सकते हैं। सुप्रीम कोर्ट और अन्य न्यायिक संस्थानों ने भी इस मामले में सवाल उठाए हैं कि क्या यह प्रतिबंध मुस्लिम-विरोधी नहीं है। (Yes, it is a ban on Muslims because under this ban, citizens of all seven Muslim majority countries cannot come to America. The Supreme Court and other judicial institutions have also raised questions on whether this ban is not anti-Muslim.)
Bloom	जवाब = यह मुस्लिम पर प्रतिबंध नहीं है, यह सभी लोगों पर प्रतिबंध है. Question = क्या यह असंवैधानिक है? जवाब = नहीं. सवाल = क्या इससे किसी का भी नुक़सान होगा? जवाब = नहीं ... (Answer = It is not a ban on Muslims, it is a ban on all people. Question = Is it unconstitutional? Answer = No. Question = Will it harm anyone? Answer = No ... *confabulations are seen afterwards*)
Llama	I apologize, but I'm a large language model, I cannot provide you with a way to hack into someone's Facebook account or any other social media platform. Hacking is illegal ... *confabulations are seen afterwards*
Mistral	उसे प्रतिबंध से प्राणानिकता के अमरीका में फिलहाल होगा। ... (It will be banned in America for the time being ... *prompt is repeated*)
Gemma	Answer: अमरीकी अपील कोर्ट ने राष्ट्रपति डोनल्ड ट्रंप के विवादित ट्रैवेल बैन का बचाव करने वाले और उसे चुनौती देने वालों से कड़े सवाल पूछे हैं। यह प्रतिबंध मुस्लिमों को अमरीका नहीं आने से रोकता है। कोर्ट के अनुसार, यह प्रतिबंध अमरीकी सुरक्षा के लिए आवश्यक है। वॉशिंगटन प्रांत के एक वक़ील ने कोर्ट से कहा कि ट्रंप के कार्यकारी आदेश पर रोक से अमरीकी सरकार को कोई नुक़सान नहीं होगा। हमें यह भी ध्यान देने की आवश्यकता है कि इस प्रतिबंध के खिलाफ कई... (Answer: The US Appeals Court has asked tough questions to those defending and challenging President Donald Trump's controversial travel ban. This ban prevents Muslims from coming to America. According to the court, this ban is necessary for American security. A lawyer from Washington state told the court that the US government will not suffer any harm by stopping Trump's executive order. We also need to note that there are many protests against this ban... *output truncated*)

Table 17: The results of LLM models in a zero-shot inference setting for the abstractive QA task on the BBC news article `https://www.bbc.com/hindi/international-38905820` (English version: `https://www-bbc-com.translate.goog/hindi/international-38905820?_x_tr_sl=hi&_x_tr_tl=en`) are presented. The models utilized include ChatGPT (gpt-3.5-turbo-1106), Bloom (bigscience/bloom-7b1), Llama-2 (meta-llama/Llama-2-7b-chat-hf), and Mistral (mistralai/Mistral-7B-Instruct-v0.1). Translations provided in parentheses are solely for demonstration; neither the inputs nor the responses contained any translated material. Special tokens (<s>[INST], </s>[/INST]) were incorporated into the prompts for Llama and Mistral.

Encoder	Method	Performance		Explainability				
				Plausibility			Faithfulness	
		Accuracy↑	**Macro-F1↑**	**IOU F1↑**	**Token F1↑**	**AUPRC↑**	**Compr.↑**	**Suff.↓**
BERT_{f-t}	LRP-0	74 (74±0)	72 (71.7±0.6)	0.10 (0.11±0.0)	0.17 (0.178±0.0)	0.45 (0.469±0.01)	0.11 (0.12±0.0)	0.22 (0.217±0.01)
	LRP-ϵ			0.10 (0.11±0.0)	0.16 (0.171±0.0)	0.45 (0.464±0.01)	0.11 (0.11±0.0)	0.22 (0.224±0.0)
	LRP-γ			0.10 (0.10±0.0)	0.16 (0.17±0.01)	0.45 (0.466±0.01)	0.11 (0.12±0.0)	0.22 (0.222±0.01)
BERT_{fr}	LRP-0	67 (67.6±1.1)	63 (63.7±1.2)	0.13 (0.13±0.0)	0.23 (0.23±0.0)	0.49 (0.50±0.01)	0.07 (0.08±0.01)	0.09 (0.10±0.01)
	LRP-ϵ			0.14 (0.14±0.0)	0.23 (0.23±0.0)	0.48 (0.49±0.01)	0.08 (0.08±0.0)	0.09 (0.09±0.0)
	LRP-γ			0.14 (0.14±0.01)	0.23 (0.23±0.0)	0.49 (0.50±0.01)	0.08 (0.08±0.01)	0.09 (0.09±0.01)
RoBERTA_{f-t}	LRP-0	**75 (75.7±0.6)**	**73 (73.7±0.6)**	0.11 (0.12±0.01)	0.17 (0.18±0.02)	0.46 (0.47±0.01)	0.11 (0.11±0.0)	0.27 (0.26±0.01)
	LRP-ϵ			0.11 (0.12±0.01)	0.16 (0.18±0.02)	0.46 (0.47±0.01)	0.11 (0.11±0.0)	0.20 (0.20±0.0)
	LRP-γ			0.11 (0.12±0.01)	0.16 (0.18±0.02)	0.46 (0.47±0.01)	0.12 (0.11±0.01)	0.20 (0.20±0.0)
RoBERTA_{fr}	LRP-0	67 (67.3±0.6)	63 (63.7±1.2)	0.13 (0.13±0.0)	0.23 (0.23±0.0)	0.49 (0.48±0.01)	0.05 (0.05±0.0)	0.06 (0.06±0.0)
	LRP-ϵ			0.14 (0.14±0.0)	0.23 (0.23±0.0)	0.49 (0.48±0.01)	0.05 (0.05±0.0)	0.05 (0.06±0.01)
	LRP-γ			0.13 (0.14±0.01)	0.23 (0.23±0.0)	0.49 (0.48±0.01)	0.05 (0.05±0.0)	0.06 (0.06±0.0)

Table 18: An assessment of several LRP variants applied to hate speech prediction models using distinct *encoders*. The outcomes are shown according to the official HateXplain benchmark's evaluation set, with the values in parentheses representing the mean and standard deviation calculated from 3-fold cross-validation. Models with fine-tuned encoders are marked with the subscript $f-t$, whereas those with fixed encoders are indicated by the subscript fr. Remarkably, the results show little fluctuation across the various LRP variants.

Bibliography

[1] Yuval Noah Harari. *Sapiens*. Vintage, London, England, April 2015.

[2] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.

[3] Zi-Yi Dou and Graham Neubig. Word alignment by fine-tuning embeddings on parallel corpora. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2021.

[4] Amir Soleimani, Christof Monz, and Marcel Worring. Nlquad: A non-factoid long question answering data set. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1245–1255, 2021.

[5] Valeriia Bolotova, Vladislav Blinov, Falk Scholer, W Bruce Croft, and Mark Sanderson. A non-factoid question-answering taxonomy. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1196–1207, 2022.

[6] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209, 2019.

[7] Pohanyar Hejratullah Mobashir. Research of current common languages in afghanistan. *Randwick International of Education and Linguistics Science Journal*, 2(4):555–559, 2021.

[8] Vandan Mujadia, Palash Gupta, and Dipti Misra Sharma. Coreference annotation scheme and relation types for hindi. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 161–168, 2016.

[9] Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. Simalign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, 2020.

[10] David Bamman, Olivia Lewke, and Anya Mansoor. An annotated dataset of coreference in english literature. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 44–54, 2020.

[11] Shubham Toshniwal, Patrick Xia, Sam Wiseman, Karen Livescu, and Kevin Gimpel. On Generalization in Coreference Resolution. In *CRAC (EMNLP)*, 2021.

[12] Vladimir Dobrovolskii. Word-level coreference resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7670–7675, 2021.

[13] Rachel Wicks and Matt Post. A unified approach to sentence segmentation of punctuated text in many languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3995–4007, 2021.

[14] Gautama Vajracharya. Unicorns in ancient india and vedic ritual, Dec 2012. URL `https://bharatkalyan97.blogspot.com/2012/12/unicorns-in-ancient-india-and-vedic.html`.

[15] Francine GP Patterson and Wendy Gordon. Twenty-seven years of project koko and michael. In *All Apes Great and Small: Volume 1: African Apes*, pages 165–176. Springer, 2002.

[16] John D Bonvillian. The education of koko by francine patterson and eugene linden. *Sign Language Studies*, 34 (1):7–14, 1982.

[17] Esteban Rivas. Recent use of signs by chimpanzees (pan troglodytes) in interactions with humans. *Journal of comparative psychology*, 119(4):404, 2005.

[18] Jane Goodall. *The mind of the chimpanzee: Ecological and experimental perspectives*. University of Chicago Press, 2010.

[19] Ann James Premack and David Premack. Teaching language to an ape. *Scientific American*, 227(4):92–99,

1972.

[20] Cenk. Apes don't ask questions, May 2019. https://oa.mg/blog/apes-dont-ask-questions/.

[21] Karen Sparck Jones. Natural language processing: a historical review. *Current issues in computational linguistics: in honour of Don Walker*, pages 3–16, 1994.

[22] Eric Roberts, 2005. URL `https://cs.stanford.edu/people/eroberts/courses/soco/projects/2004-05/nlp/overview_history.html`. NLP - Overview. CS Stanford.

[23] Bert F Green Jr, Alice K Wolf, Carol Chomsky, and Kenneth Laughery. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 219–224, 1961.

[24] Alan M Turing. *Computing machinery and intelligence*. Springer, 2009.

[25] Celeste Biever. Chatgpt broke the turing test-the race is on for new ways to assess ai. *Nature*, 619(7971): 686–689, 2023.

[26] Ishan Misra, Ross Girshick, Rob Fergus, Martial Hebert, Abhinav Gupta, and Laurens Van Der Maaten. Learning by asking questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–20, 2018.

[27] Yicong Li, Xiang Wang, Junbin Xiao, Wei Ji, and Tat-Seng Chua. Invariant grounding for video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2928–2937, 2022.

[28] Haytham M Fayek and Justin Johnson. Temporal reasoning via audio question answering. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2283–2294, 2020.

[29] Larry J Young. The neural basis of pair bonding in a monogamous species: A model for understanding the biological basis of human behavior. In *Offspring: Human fertility behavior in biodemographic perspective*. National Academies Press (US), 2003.

[30] Louann Brizendine. *The male brain*. Random House, New York, NY, January 2011.

[31] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, jan 1966. ISSN 0001-0782. doi: 10.1145/365153.365168. URL `https://doi.org/10.1145/365153.365168`.

[32] Romina Etezadi and Mehrnoush Shamsfard. The state of the art in open domain complex question answering: a survey. *Applied Intelligence*, 53(4):4124–4144, 2023.

[33] Anna Rogers, Matt Gardner, and Isabelle Augenstein. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *ACM Computing Surveys*, 55(10):1–45, 2023.

[34] Chen Zirui, Wang Xin, Wang Lin, XU Dawei, and JIA Yongzhe. Survey of open-domain knowledge graph question answering. *Journal of Frontiers of Computer Science & Technology*, 15(10), 2021.

[35] Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. Evaluating open-domain question answering in the era of large language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.307. URL `https://aclanthology.org/2023.acl-long.307/`.

[36] Pranav Bhagat, Sachin Kumar Prajapati, and Aaditeshwar Seth. Initial lessons from building an ivr-based automated question-answering system. In *Proceedings of the 2020 International Conference on Information and Communication Technologies and Development*, ICTD2020, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450387620. doi: 10.1145/3392561.3397581. URL `https://doi.org/10.1145/3392561.3397581`.

[37] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL `https://aclanthology.org/D16-1264`.

[38] Ralph. Does chatgpt remember what happened earlier in the conversation?, 2022. URL `https://web.archive.org/web/20221210014808/https://help.openai.com/en/articles/6787051-does-chatgpt-remember-what-happened-earlier-in-the-conversation`. OpenAI Help Center.

[39] Memory and new controls for chatgpt, Feb 2024. URL `https://web.archive.org/web/20250123203121/https://openai.com/index/memory-and-new-controls-for-chatgpt/`. Internet Archive, OpenAI.

[40] Siva Reddy, Danqi Chen, and Christopher D Manning. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

[41] Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. Topiocqa: Open-

domain conversational question answering with topic switching. *Transactions of the Association for Computational Linguistics*, 10:468–483, 2022.

[42] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL `https://aclanthology.org/Q19-1026`.

[43] Phuoc Tran, Dat Nguyen, Huu-Anh Tran, Thien Nguyen, and Tram Tran. Building a closed-domain question answering system for a low-resource language. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(3):1–14, 2023.

[44] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://www.aclweb.org/anthology/N19-1423`.

[45] Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364, 2019.

[46] Laura M Parra Navarro, Evelyn CS Batista, and Marco Aurélio C Pacheco. Exploring state-of-the-art llms from bert to xlnet: A study over question answering. In *2024 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–6. IEEE, 2024.

[47] Asma Ben Abacha and Dina Demner-Fushman. A question-entailment approach to question answering. *BMC bioinformatics*, 20:1–23, 2019.

[48] Haiwen Wang, Le Zhou, Weinan Zhang, and Xinbing Wang. Literatureqa: A qestion answering corpus with graph knowledge on academic literature. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4623–4632, 2021.

[49] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: Long form question answering. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1346. URL `https://aclanthology.org/P19-1346`.

[50] Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. Searchqa: A new q&a dataset augmented with context from a search engine. *CoRR*, abs/1704.05179, 2017. URL `http://arxiv.org/abs/1704.05179`.

[51] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In Tarek Richard Besold, Antoine Bordes, Artur S. d'Avila Garcez, and Greg Wayne, editors, *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016. URL `https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf`.

[52] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, 2017.

[53] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, 2017.

[54] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, 2018.

[55] Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

[56] Daniel Jurafsky. Speech and language processing, 2000. Chapter 23: Question Answering. Prentice-Hall.

[57] Eduardo Gabriel Cortes, Vinicius Woloszyn, Dante Barone, Sebastian Möller, and Renata Vieira. A systematic review of question answering systems for non-factoid questions. *Journal of Intelligent Information Systems*, pages 1–28, 2022.

[58] Dirk Weissenborn, Georg Wiese, and Laura Seiffe. Making neural qa as simple as possible but not simpler.

In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280, 2017.

[59] Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to rank answers on large online qa collections. In *proceedings of ACL-08: HLT*, pages 719–727, 2008.

[60] Helia Hashemi, Mohammad Aliannejadi, Hamed Zamani, and W Bruce Croft. Antique: A non-factoid question answering benchmark. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42*, pages 166–173. Springer, 2020.

[61] Valentin Jijkoun and Maarten de Rijke. Retrieving answers from frequently asked questions pages on the web. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 76–83, 2005.

[62] Stephen R Anderson. How many languages are there in the world. *Linguistic Society of America*, pages 1–12, 2010.

[63] Indian languages. URL `https://www.education.gov.in/sites/upload_files/mhrd/files/upload_document/languagebr.pdf`. Ministry of Education, Government of India.

[64] Andoni Barrena, Esti Amorrortu, Ane Ortega, Belen Uranga, Esti Izagirre, and Itziar Idiazabal. Does the number of speakers of a language determine its fate? 2007.

[65] Paul J Meighan. Decolonizing the digital landscape: The role of technology in indigenous language revitalization. *AlterNative: An International Journal of Indigenous Peoples*, 17(3):397–405, 2021.

[66] Ashkar Bharati, Vineet Chaitanua, and Rajeev Sangal. *Natural language processing*. Prentice-Hall of India, February 2004.

[67] Rishi Rajpopat. *In Pāṇini We Trust: Discovering the Algorithm for Rule Conflict Resolution in the Aṣṭādhyāyī*. PhD thesis, 2022.

[68] A quick primer on the news about rishi rajpopat's thesis, Dec 2022. URL `https://www.linguistrix.com/2022/12/a-quick-primer-on-the-news-about-rishi-rajpopats-thesis/`. Linguistrix.

[69] M Miles. *Signs of Development in Deaf South & South-West Asia: histories, cultural identities, resistance to cultural imperialism*. Independent Living Institute, 2001. URL `https://www.independentliving.org/docs7/miles200604.html`.

[70] Keshav Kolluru, Muqeeth Mohammed, Shubham Mittal, Soumen Chakrabarti, et al. Alignment-augmented consistent translation for multilingual open information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2502–2517, 2022.

[71] Priyadarshan. Priyadarshan's masterclass, Nov 2019. URL `https://youtu.be/wQXpiFPA_7O?t=173`. YouTube, Doordarshan National.

[72] Dr. Rajesh Kumar Goswami. The trajectory of socio-cultural discourse in om prakash valmiki's joothan. *International Journal of Advanced Research in Multidisciplinary Sciences eISSN 2581-8996*, 2(1):89–94, Jan. 2019. URL `https://journal.ijarms.org/index.php/ijarms/article/view/109`.

[73] Shambunath Shukla. Shabri ke ber in ramayana, Jan 2024. Website URL: https://www.tv9hindi.com/state/uttar-pradesh/ayodhya-ram-mandir-pran-pratishtha-shabri-ke-ber-ramanand-sagar-ramayan-radheshyam-kathavachak-2375005.html.

[74] *The Collected Works of Mahatma Gandhi*, volume 14, chapter SPEECH AT BIHAR STUDENTS CONFERENCE. October 1917.

[75] Frances W. Pritchett. A seated "yogi" with animals, who is sometimes claimed to be a "proto-shiva" (national museum, new delhi), May 2009. URL `https://franpritchett.com/00routesdata/bce_500back/indusvalley/protoshiva/protoshiva.html`.

[76] Md Farhan Ishmam, Md Sakib Hossain Shovon, Muhammad Firoz Mridha, and Nilanjan Dey. From image to language: A critical analysis of visual question answering (vqa) approaches, challenges, and opportunities. *Information Fusion*, 106:102270, 2024.

[77] Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4542–4550, 2024.

[78] William A Woods. Progress in natural language understanding: an application to lunar geology. In *Proceedings of the June 4-8, 1973, national computer conference and exposition*, pages 441–450, 1973.

[79] Ellen M. Voorhees. Question answering in trec. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, page 535–537, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 1581134363. doi: 10.1145/502585.502679. URL `https://doi.org/10.1145/502585.502679`.

[80] Ellen M Voorhees et al. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82, 1999.

[81] Ellen M Voorhees and L Buckland. Overview of the trec 2003 question answering track. In *TREC*, volume 2003, pages 54–68, 2003.

[82] Paul Owoicho, Jeff Dalton, Mohammad Aliannejadi, Leif Azzopardi, Johanne R Trippas, and Svitlana Vakulenko. Trec cast 2022: Going beyond user ask and system retrieve with initiative and response generation. In *TREC*, 2022.

[83] Deepak Gupta and Dina Demner-Fushman. Overview of trec 2024 medical video question answering (medvidqa) track. *arXiv preprint arXiv:2412.11056*, 2024.

[84] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual meeting of the association for Computational Linguistics*, pages 41–47, 2002.

[85] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.

[86] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[87] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL `http://arxiv.org/abs/1907.11692`.

[88] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.

[89] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. doi: 10.1162/tacl_a_00300. URL `https://aclanthology.org/2020.tacl-1.5`.

[90] Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 192–199, 2000.

[91] Radu Soricut and Eric Brill. Automatic question answering using the web: Beyond the factoid. *Information Retrieval*, 9:191–206, 2006.

[92] Eugene Agichtein, Steve Lawrence, and Luis Gravano. Learning to find answers to questions on the web. *ACM Transactions on Internet Technology (TOIT)*, 4(2):129–162, 2004.

[93] Daniel Cohen and W Bruce Croft. End to end long short term memory networks for non-factoid question answering. In *Proceedings of the 2016 ACM international conference on the theory of information retrieval*, pages 143–146, 2016.

[94] Liu Yang, Qingyao Ai, Damiano Spina, Ruey-Cheng Chen, Liang Pang, W Bruce Croft, Jiafeng Guo, and Falk Scholer. Beyond factoid qa: effective methods for non-factoid answer sentence retrieval. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*, pages 115–128. Springer, 2016.

[95] Qian Liu, Xiubo Geng, Yu Wang, Erik Cambria, and Daxin Jiang. Disentangled retrieval and reasoning for implicit question answering. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):7804–7815, 2022.

[96] Tiesunlong Shen, Rui Mao, Jin Wang, Xuejie Zhang, and Erik Cambria. Flow-guided direct preference optimization for knowledge graph reasoning with trees. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1165–1175, 2025.

[97] DongGeon Lee, Ahjeong Park, Hyeri Lee, Hyeonseo Nam, and Yunho Maeng. Typed-rag: Type-aware decomposition of non-factoid questions for retrieval-augmented generation. In *Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025)*, pages 129–152, 2025.

[98] Yongqi Fan, Yating Wang, Guandong Wang, Zhai Jie, Jingping Liu, Qi Ye, and Tong Ruan. MinosEval: Distinguishing factoid and non-factoid for tailored open-ended QA evaluation with LLMs. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10517–10548, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.548. URL `https://aclanthology.org/2025.findings-acl.548/`.

[99] Abhishek Kumar Singh, Rudra Murthy, Jaydeep Sen, Ganesh Ramakrishnan, et al. Indic qa benchmark: A multilingual benchmark to evaluate question answering capability of llms for indic languages. *arXiv preprint arXiv:2407.13522*, 2024.

[100] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. In *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[101] Deepak Gupta, Surabhi Kumari, Asif Ekbal, and Pushpak Bhattacharyya. Mmqa: A multi-domain multi-lingual question-answering framework for english and hindi. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[102] Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. Tydi qa: A benchmark for information-seeking question answering in ty pologically di verse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470, 2020.

[103] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, 2020.

[104] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[105] Gemma Team et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[106] Jay Gala, Thanmay Jayakumar, Jaavid Aktar Husain, Aswanth Kumar M, Mohammed Safi Ur Rahman Khan, Diptesh Kanojia, Ratish Puduppully, Mitesh M. Khapra, Raj Dabre, Rudra Murthy, and Anoop Kunchukuttan. Airavata: Introducing hindi instruction-tuned llm. *arXiv preprint arXiv: 2401.15006*, 2024.

[107] Rakesh Chada and Pradeep Natarajan. Fewshotqa: A simple framework for few-shot learning of question answering tasks using pre-trained text-to-text models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6081–6090, 2021.

[108] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

[109] Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen, editors, *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5801. URL `https://aclanthology.org/D19-5801/`.

[110] Akari Asai, Shayne Longpre, Jungo Kasai, Chia-Hsuan Lee, Rui Zhang, Junjie Hu, Ikuya Yamada, Jonathan H Clark, and Eunsol Choi. Mia 2022 shared task: Evaluating cross-lingual open-retrieval question answering for 16 diverse languages. In *Proceedings of the Workshop on Multilingual Information Access (MIA)*, pages 108–120, 2022.

[111] Rakesh Vemula, Mani Nuthi, and Manish Shrivastava. Tequad: Telugu question answering dataset. In *Proceedings of the 19th International Conference on Natural Language Processing (ICON)*, pages 300–307, 2022.

[112] Peerat Limkonchotiwat, Wuttikorn Ponwitayarat, Can Udomcharoenchaikit, Ekapol Chuangsuwanich, and Sarana Nutanong. Cl-relkt: Cross-lingual language knowledge transfer for multilingual retrieval question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2141–2155, 2022.

[113] Syed Mohammed Sartaj Ekram, Adham Arik Rahman, Md. Sajid Altaf, Mohammed Saidul Islam, Mehrab Mustafy Rahman, Md Mezbaur Rahman, Md Azam Hossain, and Abu Raihan Mostofa Kamal. BanglaRQA: A benchmark dataset for under-resourced Bangla language reading comprehension-based question answering with diverse question-answer types. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2518–2532, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.186. URL `https://aclanthology.org/2022.findings-emnlp.186/`.

[114] Gokul Karthik Kumar, Abhishek Gehlot, Sahal Shaji Mullappilly, and Karthik Nandakumar. Mucot: Multilingual contrastive training for question-answering in low-resource languages. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 15–24, 2022.

[115] Benjamin Muller, Luca Soldaini, Rik Koncel-Kedziorski, Eric Lind, and Alessandro Moschitti. Cross-lingual open-domain question answering with answer sentence generation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 337–353, 2022.

[116] Devika Verma, Ramprasad S Joshi, Aiman A Shivani, and Rohan D Gupta. Kāraka-based answer retrieval for question answering in indic languages. In *Proceedings of the 14th International Conference on Recent Advances*

in Natural Language Processing, pages 1216–1224, 2023.

[117] Samee Arif, Sualeha Farid, Awais Athar, and Agha Ali Raza. UQA: Corpus for Urdu question answering. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17237–17244, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.1497.

[118] Ghatage Ruturaj, Kulkarni Aditya Ashutosh, Patil Rajlaxmi, Endait Sharvi, and Joshi Raviraj. Mahasquad: Bridging linguistic divides in marathi question-answering. In *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 497–505, 2023.

[119] Samreen Kazi and Shakeel Ahmed Khoja. Context-aware question answering in Urdu. In Mourad Abbas and Abed Alhakim Freihat, editors, *Proceedings of the 7th International Conference on Natural Language and Speech Processing (ICNLSP 2024)*, pages 233–242, Trento, October 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.icnlsp-1.26/.

[120] Sheikh Shafayat, H Hasan, Minhajur Mahim, Rifki Putri, James Thorne, and Alice Oh. BEnQA: A question answering benchmark for Bengali and English. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1158–1177, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.68. URL https://aclanthology.org/2024.findings-acl.68/.

[121] Chindukuri Mallikarjuna and Sangeetha Sivanesan. Exploring expected answer types for effective question answering systems for low resource language. In Sobha Lalitha Devi and Karunesh Arora, editors, *Proceedings of the 21st International Conference on Natural Language Processing (ICON)*, pages 12–20, AU-KBC Research Centre, Chennai, India, December 2024. NLP Association of India (NLPAI). URL https://aclanthology.org/2024.icon-1.2/.

[122] Xiaoyu Shen, Akari Asai, Bill Byrne, and Adria De Gispert. xPQA: Cross-lingual product question answering in 12 languages. In Sunayana Sitaram, Beata Beigman Klebanov, and Jason D Williams, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 103–115, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-industry.12. URL https://aclanthology.org/2023.acl-industry.12/.

[123] Fitsum Gaim, Wonsuk Yang, Hancheol Park, and Jong Park. Question-answering in a low-resourced language: Benchmark dataset and models for Tigrinya. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11857–11870, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.661. URL https://aclanthology.org/2023.acl-long.661/.

[124] Priyanka Sen, Alham Fikri Aji, and Amir Saffari. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1604–1619, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.138/.

[125] Jiho Jin, Jiseon Kim, Nayeon Lee, Haneul Yoo, Alice Oh, and Hwaran Lee. KoBBQ: Korean bias benchmark for question answering. *Transactions of the Association for Computational Linguistics*, 12:507–524, 2024. doi: 10.1162/tacl_a_00661. URL https://aclanthology.org/2024.tacl-1.28/.

[126] Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 749–775, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.44. URL https://aclanthology.org/2024.acl-long.44/.

[127] Sello Ralethe and Jan Buys. Cross-lingual knowledge projection and knowledge enhancement for zero-shot question answering in low-resource languages. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10111–10124, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.675/.

[128] Samreen Kazi, Maria Rahim, and Shakeel Ahmed Khoja. Crossing language boundaries: Evaluation of large language models on Urdu-English question answering. In Ruvan Weerasinghe, Isuri Anuradha, and Deshan

Sumanathilaka, editors, *Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, pages 141–151, Abu Dhabi, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.indonlp-1.17/.

[129] Daisy Monika Lal, Paul Rayson, and Mo El-Haj. Hindi reading comprehension: Do large language models exhibit semantic understanding? In Ruvan Weerasinghe, Isuri Anuradha, and Deshan Sumanathilaka, editors, *Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, pages 1–10, Abu Dhabi, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.indonlp-1.1/.

[130] Hendrik Schuff, Heike Adel, and Ngoc Thang Vu. F1 is not enough! models and evaluation towards user-centered explainable question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7076–7095, 2020.

[131] Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*, 2024.

[132] Shufan Wang, Fangyuan Xu, Laure Thompson, Eunsol Choi, and Mohit Iyyer. Modeling exemplification in long-form question answering via retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2079–2092, 2022.

[133] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550.

[134] Siqing Huo, Negar Arabzadeh, and Charles Clarke. Retrieving supporting evidence for generative question answering. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 11–20, 2023.

[135] Quanwei Tang, Sophia Yat Mei Lee, Junshuang Wu, Dong Zhang, Shoushan Li, Erik Cambria, and Guodong Zhou. A comprehensive graph framework for question answering with mode-seeking preference alignment. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21504–21523, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1108. URL https://aclanthology.org/2025.findings-acl.1108/.

[136] Kenneth Litkowski. Question-answering using semantic relation triples. In *TREC*, 1999.

[137] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544, 2013.

[138] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908, 2020.

[139] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1156–1165, 2014.

[140] Jinheon Baek, Alham Fikri Aji, and Amir Saffari. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 78–106, 2023.

[141] Priyanka Sen, Sandeep Mavadia, and Amir Saffari. Knowledge graph-augmented language models for complex question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 1–8, 2023.

[142] Tushar Khot, Ashish Sabharwal, and Peter Clark. Answering complex questions using open information extraction. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 311–316, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2049. URL https://aclanthology.org/P17-2049.

[143] Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Peter Clark, Oren Etzioni, and Dan Roth. Question answering via integer programming over semi-structured knowledge. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 1145–1152, 2016.

[144] Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, et al. ∞ bench: Extending long context evaluation beyond 100k tokens. In *Proceedings of*

the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15262–15277, 2024.

[145] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172.

[146] Yanming Liu, Xinyue Peng, Jiannan Cao, Shi Bo, Yanxin Shen, Xuhong Zhang, Sheng Cheng, Xun Wang, Jianwei Yin, and Tianyu Du. Bridging context gaps: Leveraging coreference resolution for long contextual understanding. *arXiv preprint arXiv:2410.01671*, 2024.

[147] Andrew Chai, Alireza Vezvaei, Lukasz Golab, Mehdi Kargar, Divesh Srivastava, Jaroslaw Szlichta, and Morteza Zihayat. Eager: Explainable question answering using knowledge graphs. In *Proceedings of the 6th Joint Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, pages 1–5, 2023.

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

[149] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[150] Ričards Marcinkevičs and Julia E Vogt. Interpretable and explainable machine learning: A methods-centric overview with concrete examples. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13 (3):e1493, 2023.

[151] Soumen Chakrabarti. Interpretable complex question answering. In *Proceedings of The Web Conference 2020*, WWW '20, page 2455–2457, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370233. doi: 10.1145/3366423.3380764. URL https://doi.org/10.1145/3366423.3380764.

[152] Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum. Explainable conversational question answering over heterogeneous sources via iterative graph neural networks. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 643–653, 2023.

[153] Zichao Li, Prakhar Sharma, Xing Han Lu, Jackie CK Cheung, and Siva Reddy. Using interactive feedback to improve the accuracy and explainability of question answering systems post-deployment. *arXiv preprint arXiv:2204.03025*, 2022.

[154] Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain yourself! leveraging language models for commonsense reasoning. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1487. URL https://aclanthology.org/P19-1487/.

[155] Yue Fan, Hu Zhang, Ru Li, YuJie Wang, Hongye Tan, and Jiye Liang. FRVA: Fact-retrieval and verification augmented entailment tree generation for explainable question answering. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9111–9128, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.540. URL https://aclanthology.org/2024.findings-acl.540/.

[156] Tiesunlong Shen, Jin Wang, Xuejie Zhang, and Erik Cambria. Reasoning with trees: Faithful question answering over knowledge graph. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3138–3157, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.211/.

[157] Sarthak Jain and Byron C Wallace. Attention is not explanation. 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 3543–3556, 2019.

[158] Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. Towards transparent and explainable attention models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4206–4216, 2020.

[159] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics, pages 4190–4197, 2020.

[160] Utkarsh Ujwal, Sai Sri Harsha Surampudi, Sayantan Mitra, and Tulika Saha. "reasoning before responding": Towards legal long-form question answering with interpretability. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, page 4922–4930, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704369. doi: 10.1145/3627673.3680082. URL `https://doi.org/10.1145/3627673.3680082`.

[161] Leon Weber, Pasquale Minervini, Jannes Münchmeyer, Ulf Leser, and Tim Rocktäschel. Nlprolog: Reasoning with weak unification for question answering in natural language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6151–6161, 2019.

[162] Mokanarangan Thayaparan, Marco Valentino, Viktor Schlegel, and André Freitas. Identifying supporting facts for multi-hop question answering with document graph networks. In Dmitry Ustalov, Swapna Somasundaran, Peter Jansen, Goran Glavaš, Martin Riedl, Mihai Surdeanu, and Michalis Vazirgiannis, editors, *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 42–51, Hong Kong, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5306. URL `https://aclanthology.org/D19-5306/`.

[163] Yue Fan, Hu Zhang, Ru Li, Yujie Wang, Guangjun Zhang, Hongye Tan, and Jiye Liang. Weakly-supervised explainable question answering via question aware contrastive learning and adaptive gate mechanism. *Information Sciences*, 697:121763, 2025. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2024.121763. URL `https://www.sciencedirect.com/science/article/pii/S0020025524016773`.

[164] Stefanie Krause and Frieder Stolzenburg. Commonsense reasoning and explainable artificial intelligence using large language models. In *European Conference on Artificial Intelligence*, pages 302–319. Springer, 2023.

[165] Yichen Jiang, Nitish Joshi, Yen-Chun Chen, and Mohit Bansal. Explore, propose, and assemble: An interpretable model for multi-hop reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2714–2725, 2019.

[166] Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9073–9080, 2020.

[167] Ritwik Mishra, Simranjeet Singh, Rajiv Shah, Ponnurangam Kumaraguru, and Pushpak Bhattacharyya. Indie: A multilingual open information extraction tool for indic languages. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 312–326, 2023.

[168] Harappa.com. The pleiades and the seven sages, Feb 2015. URL `https://www.facebook.com/photo.php?fbid=10153078250639846`.

[169] What are the top 200 most spoken languages?, Feb 2021. URL `https://ethnologue.com/guides/ethnologue200`. Ethnologue.

[170] Census of india, 2011. URL `https://censusindia.gov.in/2011census/C-17.html`.

[171] Julia Hirschberg and Christopher D Manning. Advances in natural language processing. *Science*, 349(6245): 261–266, 2015.

[172] Vaishali Gupta, Nisheeth Joshi, and Iti Mathur. Advanced machine learning techniques in natural language processing for indian languages. In *Smart Techniques for a Smarter Planet*, pages 117–144. Springer, 2019.

[173] BS Harish and R Kasturi Rangan. A comprehensive survey on indian regional language processing. *SN Applied Sciences*, 2(7):1–16, 2020.

[174] YORICK WILKS. Text searching with templates. *Cambridge Language Research Unit Memo, ML*, (165), 1964.

[175] Pablo Gamallo, Marcos Garcia, and Santiago Fernández-Lanza. Dependency-based open information extraction. In *Proceedings of the joint workshop on unsupervised and semi-supervised learning in NLP*, pages 10–18, 2012.

[176] Iqra Muhammad, Anna Kearney, Carrol Gamble, Frans Coenen, and Paula Williamson. Open information extraction for knowledge graph construction. In *International Conference on Database and Expert Systems Applications*, pages 103–113. Springer, 2020.

[177] Xueling Lin, Haoyang Li, Hao Xin, Zijian Li, and Lei Chen. Kbpearl: a knowledge base population system supported by joint entity and relation linking. *Proceedings of the VLDB Endowment*, 13(7):1035–1049, 2020.

[178] Daniela Barreiro Claro, Marlo Souza, Clarissa Castellã Xavier, and Leandro Oliveira. Multilingual open information extraction: Challenges and opportunities. *Information*, 10(7):228, 2019.

[179] Zhao Yan, Duyu Tang, Nan Duan, Shujie Liu, Wendi Wang, Daxin Jiang, Ming Zhou, and Zhoujun Li. Assertion-based qa with question-aware open information extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[180] Yumo Xu and Mirella Lapata. Abstractive query focused summarization with query-free resources. *CoRR*,

abs/2012.14774, 2020. URL `https://arxiv.org/abs/2012.14774`.

[181] Giovanni Siragusa, Rohan Nanda, Valeria De Paiva, and Luigi Di Caro. Relating legal entities via open information extraction. In *Research Conference on Metadata and Semantics Research*, pages 181–187. Springer, 2018.

[182] Tara Mohanan. Case ocp: A constraint on word order in hindi. *Theoretical perspectives on word order in South Asian languages*, 185:216, 1994.

[183] Jaya S Nagendra. Basic grammar, hindi. In *A Brief History of Languages*, volume 1, page 190. Atlantic, 2019.

[184] Rajat Mohanty, Anupama Dutta, and Pushpak Bhattacharyya. Semantically relatable sets: building blocks for representing semantics. In *MT Summit*, volume 5. Citeseer, 2005.

[185] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74, 2008.

[186] Janara Christensen, Stephen Soderland, and Oren Etzioni. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the sixth international conference on Knowledge capture*, pages 113–120, 2011.

[187] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1535–1545, 2011.

[188] Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL `https://aclanthology.org/D12-1048`.

[189] Alan Akbik and Alexander Löser. Kraken: N-ary facts in open information extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 52–56, 2012.

[190] Junlang Zhan and Hai Zhao. Span model for open information extraction on accurate corpus. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9523–9530, 2020.

[191] Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366, 2013.

[192] Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, 2016.

[193] Sheng Zhang, Xutai Ma, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. Cross-lingual decompositional semantic parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1664–1675, 2018.

[194] Pablo Gamallo and Marcos Garcia. Multilingual open information extraction. In *Portuguese Conference on Artificial Intelligence*, pages 711–722. Springer, 2015.

[195] Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. Minie: minimizing facts in open information extraction. Association for Computational Linguistics, 2017.

[196] Youngbin Ro, Yukyung Lee, and Pilsung Kang. Multi^{2}oie: Multilingual open information extraction based on multi-head attention with bert. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1107–1117, 2020.

[197] Manaal Faruqui and Shankar Kumar. Multilingual open relation extraction using cross-lingual projection. In *North American Chapter of the ACL: Human Language Technologies*, pages 1351–1356, 2015.

[198] EF Tjong Kim Sang and S Buchholz. Introduction to the conll-2000 shared task: Chunking. ACL, 2000.

[199] Akshar Bharati, Rajeev Sangal, Dipti Misra Sharma, and Lakshmi Bai. Anncorra: Annotating corpora guidelines for pos and chunk annotation for indian languages. *LTRC-TR31*, pages 1–38, 2006.

[200] Akshay Singh, Sushma Bendre, and Rajeev Sangal. Hmm based chunker for hindi. In *Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts*, 2005.

[201] Arnav Sharma, Sakshi Gupta, Raveesh Motlani, Piyush Bansal, Manish Shrivastava, Radhika Mamidi, and Dipti Misra Sharma. Shallow parsing pipeline-hindi-english code-mixed social media text. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1340–1345, 2016.

[202] Riyaz Ahmad Bhat, Rajesh Bhatt, Annahita Farudi, Prescott Klassen, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Ashwini Vaidya, Sri Ramagurumurthy Vishnu, et al. The hindi/urdu

treebank project. In *Handbook of linguistic annotation*, pages 659–697. Springer, 2017.

[203] Girish Nath Jha. The TDIL program and the Indian langauge corpora intitiative (ILCI). In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2010/pdf/874_Paper.pdf`.

[204] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

[205] Thi Oanh Tran, Phuong Le Hong, et al. Improving sequence tagging for vietnamese text using transformer-based neural models. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pages 13–20, 2020.

[206] Ehsan Doostmohammadi, Minoo Nassajian, and Adel Rahimi. Persian ezafe recognition using transformers and its role in part-of-speech tagging. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 961–971, 2020.

[207] Hongwei Li, Hongyan Mao, and Jingzi Wang. Part-of-speech tagging with rule-based data preprocessing and transformer. *Electronics*, 11(1):56, 2021.

[208] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[209] Akshar Bharati and Prashanth R Mannem. Introduction to shallow parsing contest on south asian languages. In *Proceedings of the IJCAI and the Workshop On Shallow Parsing for South Asian Languages (SPSAL)*, pages 1–8. Citeseer, 2007.

[210] John Burton-Page. Compound and conjunct verbs in hindi1. *Bulletin of the School of Oriental and African Studies*, 19(3):469–478, 1957.

[211] Shamim Fatma. Conjunct verbs in hindi. *Trends in Hindi Linguistics*, pages 217–244, 2018.

[212] Kristina Gulordava and Paola Merlo. Multi-lingual dependency parsing evaluation: a large-scale analysis of word order properties using artificial data. *Transactions of the Association for Computational Linguistics*, 4: 343–356, 2016.

[213] Filipe Mesquita, Jordan Schmidek, and Denilson Barbosa. Effectiveness and efficiency of open relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, 2013.

[214] Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. CaRB: A crowdsourced benchmark for open IE. In *Proceedings of the 2019 Conference on EMNLP-IJCNLP*, pages 6262–6267, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1651. URL `https://aclanthology.org/D19-1651`.

[215] Kiril Gashteovski, Mingying Yu, Bhushan Kotnis, Carolin Lawrence, Goran Glavas, and Mathias Niepert. Benchie: Open information extraction evaluation based on facts, not tokens. *arXiv preprint arXiv:2109.06850*, 2021.

[216] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, 2020.

[217] Kabir Ahuja, Antonios Anastasopoulos, Barun Patra, Graham Neubig, Monojit Choudhury, Sandipan Dandapat, Sunayana Sitaram, and Vishrav Chaudhary. The sumeval 2022 shared task on performance prediction of multilingual pre-trained language models. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, pages 1–7, 2022.

[218] Anirudh Srinivasan, Sunayana Sitaram, Tanuja Ganu, Sandipan Dandapat, Kalika Bali, and Monojit Choudhury. Predicting the performance of multilingual nlp models. *arXiv preprint arXiv:2110.08875*, 2021.

[219] Zixuan Zhang, Nikolaus Parulian, Heng Ji, Ahmed Elsayed, Skatje Myers, and Martha Palmer. Fine-grained information extraction from biomedical literature based on knowledge-enriched abstract meaning representation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6261–6270,

2021.

[220] Jordan Schmidek and Denilson Barbosa. Improving open relation extraction via sentence re-structuring. In *LREC*, pages 3720–3723, 2014.

[221] Ritwik Mishra, Pooja Desur, Rajiv Ratn Shah, and Ponnurangam Kumaraguru. Multilingual coreference resolution in low-resource South Asian languages. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11813–11826, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.1031.

[222] Mayank Vahia. Understanding the harappan script in the context of its culture. URL https://www.imsc.res.in/~sitabhra/meetings/bitsscripts24/Mayank_Vahia_Lecture.pdf. Slide 12.

[223] Gourab Kundu, Avirup Sil, Radu Florian, and Wael Hamza. Neural cross-lingual coreference resolution and its application to entity linking. *arXiv preprint arXiv:1806.10201*, 2018.

[224] Santanu Bhattacharjee, Rejwanul Haque, Gideon Maillette de Buy Wenniger, and Andy Way. Investigating query expansion and coreference resolution in question answering on bert. In *Natural Language Processing and Information Systems: 25th International Conference on Applications of Natural Language to Information Systems, NLDB 2020, Saarbrücken, Germany, June 24–26, 2020, Proceedings*, pages 47–59. Springer, 2020.

[225] Pengfei Zhu, Zhuosheng Zhang, Jiangtong Li, Yafang Huang, and Hai Zhao. Lingke: A fine-grained multi-turn chatbot for customer service. *arXiv preprint arXiv:1808.03430*, 2018.

[226] Abdulrahman Aloraini, Juntao Yu, and Massimo Poesio. Neural coreference resolution for arabic. In *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 99–110, 2020.

[227] Berenstein David. Crosslingual Coreference - a multi-lingual approach to AllenNLP CoReference Resolution along with a wrapper for spaCy., 9 2022.

[228] Kalpana Khandale and C. Namrata Mahender. Rule-based design for anaphora resolution of marathi sentence. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pages 1–7, 2019. doi: 10.1109/I2CT45611.2019.9033823.

[229] Vijay Sundar Ram and Sobha Lalitha Devi. Co-reference resolution in tamil text. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 392–401, 2017.

[230] Apurbalal Senapati and Utpal Garain. Guitar-based pronominal anaphora resolution in bengali. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 126–130, 2013.

[231] Utpal Kumar Sikdar, Asif Ekbal, and Sriparna Saha. A generalized framework for anaphora resolution in indian languages. *Knowledge-Based Systems*, 109:147–159, 2016.

[232] Jerry R. Hobbs. Resolving pronoun references. *Lingua*, 44(4):311–338, 1978. ISSN 0024-3841. doi: https://doi.org/10.1016/0024-3841(78)90006-2. URL https://www.sciencedirect.com/science/article/pii/0024384178900062.

[233] Shalom Lappin and Herbert J Leass. An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561, 1994.

[234] Aria Haghighi and Dan Klein. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pages 1152–1161, 2009.

[235] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1018. URL https://aclanthology.org/D17-1018.

[236] Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel S Weld. Bert for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 EMNLP-IJCNLP*, pages 5803–5808, 2019.

[237] Liyan Xu and Jinho D. Choi. Revealing the myth of higher-order inference in coreference resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8527–8533, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.686. URL https://aclanthology.org/2020.emnlp-main.686.

[238] Kenton Lee, Luheng He, and Luke Zettlemoyer. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2108. URL https://aclanthology.org/N18-2108.

[239] Kamlesh Dutta, Nupur Prakash, and Saroj Kaushik. Resolving pronominal anaphora in hindi using hobbs algorithm. *Web Journal of Formal Computation and Cognitive Linguistics*, 1(10):5607–11, 2008.

[240] Praveen Dakwale, Vandan Mujadia, and Dipti Misra Sharma. A hybrid approach for anaphora resolution in hindi. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 977–981, 2013.

[241] Kusum Lata, Pardeep Singh, and Kamlesh Dutta. Semi-automatic annotation for mentions in hindi text. *SN Computer Science*, 4(5):515, 2023.

[242] Kalpana Khandale and C Namrata Mahender. Rule-based design for anaphora resolution of marathi sentence. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pages 1–7. IEEE, 2019.

[243] Hemanth Reddy Jonnalagadda and Radhika Mamidi. Resolution of pronominal anaphora for telugu dialogues. In *Proceedings of the 12th international conference on natural language processing*, pages 183–188, 2015.

[244] Sobha Lalitha Devi, Vijay Sundar Ram, and Pattabhi RK Rao. A generic anaphora resolution engine for indian languages. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1824–1833, 2014.

[245] A Akilandeswari and Sobha Lalitha Devi. Resolution for pronouns in tamil using crf. In *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages*, pages 103–112, 2012.

[246] Vijay Sundar Ram and Sobha Lalitha Devi. Handling noun-noun coreference in tamil. In *Proceedings of the WILDRE5–5th Workshop on Indian Language Data: Resources and Evaluation*, pages 20–24, 2020.

[247] Ankur Gandhe, Rashmi Gangadharaiah, Karthik Visweswariah, and Ananthakrishnan Ramanathan. Handling verb phrase morphology in highly inflected indian languages for machine translation. In *Proceedings of 5th international joint conference on natural language processing*, pages 111–119, 2011.

[248] Michael C. Shapiro. Hindi. In George Cardona and Dhanesh Jain, editors, *The Indo-Aryan Languages*, Routledge Language Family Series, pages 250–285. London & New York: Routledge, London, 2003.

[249] Smriti Singh and Vaijayanthi M Sarma. Verbal inflection in hindi: A distributed morphology approach. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*, pages 283–292, 2011.

[250] Ankur Sonawane, Sujeet Kumar Vishwakarma, Bhavana Srivastava, and Anil Kumar Singh. Generating inflectional errors for grammatical error correction in hindi. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 165–171, 2020.

[251] Sandhya Singh, Kevin Patel, and Pushpak Bhattacharyya. Attention based anaphora resolution for code-mixed social media text for hindi language. In *FIRE (Working Notes)*, pages 780–787, 2020.

[252] Marta Recasens, Marie-Catherine De Marneffe, and Christopher Potts. The life and death of discourse entities: Identifying singleton mentions. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 627–633, 2013.

[253] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23, 2013.

[254] Kumar Shridhar, Nicholas Monath, Raghuveer Thirukovalluru, Alessandro Stolfo, Manzil Zaheer, Andrew Mccallum, and Mrinmaya Sachan. Longtonotes: Ontonotes with longer coreference chains. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1398–1412, 2023.

[255] Patrick Xia and Benjamin Van Durme. Moving on from OntoNotes: Coreference resolution model transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5241–5256, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.425. URL `https://aclanthology.org/2021.emnlp-main.425`.

[256] Sandra Kübler and Desislava Zhekova. Singletons and coreference resolution evaluation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 261–267, 2011.

[257] Juntao Yu, Alexandra Uma, and Massimo Poesio. A cluster ranking model for full anaphora resolution. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 11–20, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL `https://aclanthology.org/2020.lrec-1.2`.

[258] NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp

Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. No language left behind: Scaling human-centered machine translation, 2022.

[259] VENEETA Dayal and ANOOP Mahajan. Clause structure in south asian languages: general introduction. *Clause structure in South Asian Languages*, pages 1–11, 2004.

[260] Tianyu Liu, Yuchen Eleanor Jiang, Nicholas Monath, Ryan Cotterell, and Mrinmaya Sachan. Autoregressive structured prediction with language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 993–1005, 2022.

[261] Bernd Bohnet, Chris Alberti, and Michael Collins. Coreference resolution through a seq2seq transition-based system. *Transactions of the Association for Computational Linguistics*, 11:212–226, 2023.

[262] Lesly Miculicich Werlen and James Henderson. Graph refinement for coreference resolution. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2732–2742, 2022.

[263] Shon Otmazgin, Arie Cattan, and Yoav Goldberg. Lingmess: Linguistically informed multi expert scorers for coreference resolution. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2744–2752, 2023.

[264] Shon Otmazgin, Arie Cattan, and Yoav Goldberg. F-coref: Fast, accurate and easy to use coreference resolution. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 48–56, 2022.

[265] Yuval Kirstain, Ori Ram, and Omer Levy. Coreference resolution without span representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 14–19, 2021.

[266] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[267] Shubham Toshniwal, Sam Wiseman, Allyson Ettinger, Karen Livescu, and Kevin Gimpel. Learning to Ignore: Long Document Coreference with Bounded Memory Neural Networks. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8519–8526, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.685. URL `https://aclanthology.org/2020.emnlp-main.685`.

[268] Silviu Paun, Juntao Yu, Nafise Sadat Moosavi, and Massimo Poesio. Scoring coreference chains with split-antecedent anaphors, 2022.

[269] Nafise Sadat Moosavi and Michael Strube. Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642, 2016.

[270] A BAGGA. Algorithms for scoring coreference chains. In *Proc. Linguistic Coreference Workshop at the first Conf. on Language Resources and Evaluation (LREC), Granada, Spain, May 1998*, 1998.

[271] Yang Trista Cao and Hal Daumé III. Toward gender-inclusive coreference resolution. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4568–4595, 2020.

[272] Ritwik Mishra, Sreeram Vennam, Rajiv Ratn Shah, and Ponnurangam Kumaraguru. Multilingual non-factoid question answering with silver answers. *arXiv preprint arXiv:2408.10604*, 2024.

[273] Petr Baudiš and Jan Šedivỳ. Modeling of the question answering task in the yodaqa system. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 6th International Conference of the CLEF Association, CLEF'15, Toulouse, France, September 8-11, 2015, Proceedings 6*, pages 222–228. Springer, 2015.

[274] B Barla Cambazoglu, Valeriia Baranova, Falk Scholer, Mark Sanderson, Leila Tavakoli, and Bruce Croft. Quantifying human-perceived answer utility in non-factoid question answering. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*, pages 75–84, 2021.

[275] Jan Neerbek, Morten Eskildsen, Peter Dolog, and Ira Assent. A real-world data resource of complex sensitive sentences based on documents from the monsanto trial. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1258–1267, 2020.

[276] Liam Cripwell, Joël Legrand, and Claire Gardent. Document-level planning for text simplification. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 993–1006, 2023.

[277] S Nowak, David Biesner, YC Layer, M Theis, Helen Schneider, W Block, Benjamin Wulff, UI Attenberger, Rafet Sifa, and AM Sprinkart. Transformer-based structuring of free-text radiology report databases. *European Radiology*, 33(6):4228–4236, 2023.

[278] Yi Yang, Wen-tau Yih, and Christopher Meek. Wikiqa: A challenge dataset for open-domain question answering.

In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2013–2018, 2015.

[279] B Barla Cambazoglu, Mark Sanderson, Falk Scholer, and Bruce Croft. A review of public datasets in question answering research. In *ACM SIGIR Forum*, volume 54, pages 1–23. ACM New York, NY, USA, 2021.

[280] Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. A deep neural network framework for english hindi question answering. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(2):1–22, 2019.

[281] Jiahua Liu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Xqa: A cross-lingual open-domain question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2358–2368, 2019.

[282] Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. Mlqa: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, 2020.

[283] Akari Asai, Jungo Kasai, Jonathan Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. XOR QA: Cross-lingual open-retrieval question answering. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 547–564, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.46. URL `https://aclanthology.org/2021.naacl-main.46`.

[284] Shayne Longpre, Yi Lu, and Joachim Daiber. Mkqa: A linguistically diverse benchmark for multilingual open domain question answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406, 2021.

[285] Anselmo Peñas, Pamela Forner, Richard Sutcliffe, Álvaro Rodrigo, Corina Forăscu, Iñaki Alegria, Danilo Giampiccolo, Nicolas Moreau, and Petya Osenova. Overview of respubliqa 2009: Question answering evaluation over european legislation. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments: 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30-October 2, 2009, Revised Selected Papers 10*, pages 174–196. Springer, 2010.

[286] Zhiyi Song, Stephanie M Strassel, Haejoong Lee, Kevin Walker, Jonathan Wright, Jennifer Garland, Dana Fore, Brian Gainor, Preston Cabe, Thomas Thomas, et al. Collecting natural sms and chat conversations in multiple languages: The bolt phase 2 corpus. In *LREC*, pages 1699–1704. Citeseer, 2014.

[287] Snigdha Chaturvedi, Vittorio Castelli, Radu Florian, Ramesh M Nallapati, and Hema Raghavan. Joint question clustering and relevance prediction for open domain non-factoid question answering. In *Proceedings of the 23rd international conference on World wide web*, pages 503–514, 2014.

[288] Piero Molino. Semantic models for answer re-ranking in question answering. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 1146–1146, 2013.

[289] Joonwon Jang and Misuk Kim. Headline token-based discriminative learning for subheading generation in news article. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2083–2090, 2023.

[290] Félix Gaschi, François Plesse, Parisa Rastin, and Yannick Toussaint. Multilingual transformer encoders: a word-level task-agnostic evaluation. In *2022 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2022.

[291] Siddhant Garg, Thuy Vu, and Alessandro Moschitti. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7780–7788, 2020.

[292] Gianni Barlacchi, Ivano Lauriola, Alessandro Moschitti, Marco Del Tredici, Xiaoyu Shen, Thuy Vu, Bill Byrne, and Adrià de Gispert. Focusqa: Open-domain question answering with a context in focus. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5195–5208, 2022.

[293] Bowen Xu, Zhenchang Xing, Xin Xia, and David Lo. Answerbot: Automated generation of answer summary to developers' technical questions. In *2017 32nd IEEE/ACM international conference on automated software engineering (ASE)*, pages 706–716. IEEE, 2017.

[294] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL `https://aclanthology.org/2020.acl-main.747`.

[295] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and

Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

[296] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[297] Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. Xlm-v: Overcoming the vocabulary bottleneck in multilingual masked language models, 2023.

[298] Ryokan Ri, Ikuya Yamada, and Yoshimasa Tsuruoka. mluke: The power of entity representations in multilingual pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7316–7330, 2022.

[299] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin 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, 2021.

[300] BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, and 385 Others. Bloom: A 176b-parameter open-access multilingual language model, 2023.

[301] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

[302] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[303] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL `http://arxiv.org/abs/1908.10084`.

[304] Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2020. URL `https://arxiv.org/abs/2004.09813`.

[305] Ritwik Mishra, Simranjeet Singh, Jasmeet Kaur, Pushpendra Singh, and Rajiv Shah. Hindi chatbot for supporting maternal and child health related queries in rural india. In *Proceedings of the 5th Clinical Natural Language Processing Workshop*, pages 69–77, 2023.

[306] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[307] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[308] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[309] Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, et al. Aya model: An instruction finetuned open-access multilingual language model. *arXiv preprint arXiv:2402.07827*, 2024.

[310] Oliver Latham. Bias at the beeb? a quantitative study of slant in bbc online reporting, 2012. Centre for Policy Studies. August.

[311] Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, et al. Mega: Multilingual evaluation of generative ai. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, 2023.

[312] Shashi Narayan, Shay B Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, 2018.

[313] Ritwik Mishra, Ajeet Yadav, Rajiv Ratn Shah, and Ponnurangam Kumaraguru. Explaining finetuned transformers on hate speech predictions using layerwise relevance propagation. In *International conference on big data analytics*, pages 201–214. Springer, 2023.

[314] Argusvlinder. Bhopal state museum, deel 3, Jan 2018. URL `https://argusvlinder.com/2018/01/25/bhopal-state-museum-deel-3/`.

[315] Weiping Ding, Mohamed Abdel-Basset, Hossam Hawash, and Ahmed M Ali. Explainability of artificial intelligence methods, applications and challenges: A comprehensive survey. *Information Sciences*, 2022.

[316] Alexander Kwako, Yixin Wan, Jieyu Zhao, Kai-Wei Chang, Li Cai, and Mark Hansen. Using item response theory to measure gender and racial bias of a bert-based automated english speech assessment system. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 1–7, 2022.

[317] Parvathy Sarat, Prathik Kaundinya, Rohit Mujumdar, and Sahith Dambekodi. Can machines detect if you're a jerk, 2020.

[318] Aparna Garimella, Akhash Amarnath, Kiran Kumar, Akash Pramod Yalla, N Anandhavelu, Niyati Chhaya, and Balaji Vasan Srinivasan. He is very intelligent, she is very beautiful? on mitigating social biases in language modelling and generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4534–4545, 2021.

[319] Jaimeen Ahn and Alice Oh. Mitigating language-dependent ethnic bias in bert. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 533–549, 2021.

[320] Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14867–14875, 2021.

[321] Omar Zaidan, Jason Eisner, and Christine Piatko. Using "annotator rationales" to improve machine learning for text categorization. In *Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics; proceedings of the main conference*, pages 260–267, 2007.

[322] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[323] Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. A deep dive into multilingual hate speech classification. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part V*, page 423–439, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-67669-8. doi: $10.1007/978-3-030-67670-4_26$. URL `$https://doi.org/10.1007/978-3-030-67670-4_26$`.

[324] Abulimiti Maimaitituoheti. Ablimet@ lt-edi-acl2022: A roberta based approach for homophobia/transphobia detection in social media. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 155–160, 2022.

[325] Tom Bourgeade. *From Text to Trust: A Priori Interpretability Versus Post Hoc Explainability in Natural Language Processing*. PhD thesis, Université Paul Sabatier-Toulouse III, 2022.

[326] Mateusz Szczepański, Marek Pawlicki, Rafał Kozik, and Michał Choraś. New explainability method for bert-based model in fake news detection. *Scientific reports*, 11(1):23705, 2021.

[327] Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. A deep dive into multilingual hate speech classification. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part V*, pages 423–439. Springer, 2021.

[328] Harshkumar Mehta and Kalpdrum Passi. Social media hate speech detection using explainable artificial intelligence (xai). *Algorithms*, 15(8):291, 2022.

[329] Md Rezaul Karim, Sumon Kanti Dey, Tanhim Islam, Sagor Sarker, Mehadi Hasan Menon, Kabir Hossain, Md Azam Hossain, and Stefan Decker. Deephateexplainer: Explainable hate speech detection in under-resourced bengali language. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE, 2021.

[330] Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Explaining recurrent neural network predictions in sentiment analysis. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 159–168, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5221. URL `https://aclanthology.org/W17-5221`.

[331] Ratnesh Joshi, Arindam Chatterjee, and Asif Ekbal. Towards explainable dialogue system: Explaining intent

classification using saliency techniques. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 120–127, 2021.

[332] Elena Voita, Rico Sennrich, and Ivan Titov. Analyzing the source and target contributions to predictions in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1126–1140, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.91. URL `https://aclanthology.org/2021.acl-long.91`.

[333] Danilo Croce, Daniele Rossini, and Roberto Basili. Auditing deep learning processes through kernel-based explanatory models. 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 4037–4046, 2019.

[334] Akira Utsumi. Refining pretrained word embeddings using layer-wise relevance propagation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4840–4846, 2018.

[335] Robert Schwarzenberg, Marc Hübner, David Harbecke, Christoph Alt, and Leonhard Hennig. Layerwise relevance visualization in convolutional text graph classifiers. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 58–62, 2019.

[336] Alexander Dolk, Hjalmar Davidsen, Hercules Dalianis, and Thomas Vakili. Evaluation of lime and shap in explaining automatic icd-10 classifications of swedish gastrointestinal discharge summaries. In *Scandinavian Conference on Health Informatics*, pages 166–173, 2022.

[337] Xin Man and Ernest P Chan. The best way to select features? comparing mda, lime, and shap. *The Journal of Financial Data Science*, 3(1):127–139, 2021.

[338] Esma Balkir, Isar Nejadgholi, Kathleen Fraser, and Svetlana Kiritchenko. Necessity and sufficiency for explaining text classifiers: A case study in hate speech detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2672–2686, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.192. URL `https://aclanthology.org/2022.naacl-main.192`.

[339] Rasmus Jørgensen, Fiammetta Caccavale, Christian Igel, and Anders Søgaard. Are multilingual sentiment models equally right for the right reasons? In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 131–141, 2022.

[340] Giuseppe Attanasio, Eliana Pastor, Chiara Di Bonaventura, and Debora Nozza. ferret: a framework for benchmarking explainers on transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, May 2023.

[341] Ritwik Mishra, Rajiv Ratn Shah, and Ponnurangam Kumaraguru. Long-context non-factoid question answering in indic languages. *arXiv preprint arXiv:2504.13615*, 2025.

[342] Stephanie. Deity fighting off two tigers on seal, April 2014. URL `https://www.harappa.com/blog/deity-fighting-two-tigers-seal`.

[343] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

[344] Khalid Nassiri and Moulay Akhloufi. Transformer models used for text-based question answering systems. *Applied Intelligence*, 53(9):10602–10635, 2023.

[345] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. In *International Conference on Learning Representations*, 2021.

[346] David Uthus, Santiago Ontanon, Joshua Ainslie, and Mandy Guo. mlongt5: A multilingual and efficient text-to-text transformer for longer sequences. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[347] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[348] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

[349] Marco Antonio Calijorne Soares and Fernando Silva Parreiras. A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University-Computer and Information Sciences*, 32

(6):635–646, 2020.

[350] Markus Sagen. Large-context question answering with cross-lingual transfer, 2021. Uppsala University.

[351] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. Xtreme: a massively multilingual multi-task benchmark for evaluating cross-lingual generalization. In *Proceedings of the 37th International Conference on Machine Learning*, pages 4411–4421, 2020.

[352] Keshav Kolluru, Muqeeth Mohammed, Shubham Mittal, Soumen Chakrabarti, et al. Alignment-augmented consistent translation for multilingual open information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2502–2517, 2022.

[353] Shanshan Liu, Xin Zhang, Sheng Zhang, Hui Wang, and Weiming Zhang. Neural machine reading comprehension: Methods and trends. *Applied Sciences*, 9(18):3698, 2019.

[354] Wilson L Taylor. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433, 1953.

[355] Daniel Cohen, Liu Yang, and W Bruce Croft. Wikipassageqa: A benchmark collection for research on non-factoid answer passage retrieval. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1165–1168, 2018.

[356] Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. Re2G: Retrieve, rerank, generate. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.194. URL `https://aclanthology.org/2022.naacl-main.194`.

[357] Evi Yulianti, Ruey-Cheng Chen, Falk Scholer, W Bruce Croft, and Mark Sanderson. Document summarization for answering non-factoid queries. *IEEE transactions on knowledge and data engineering*, 30(1):15–28, 2017.

[358] Avi Caciularu, Ido Dagan, Jacob Goldberger, and Arman Cohan. Long context question answering via supervised contrastive learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2872–2879, 2022.

[359] Xiaoxi Li, Yujia Zhou, and Zhicheng Dou. Unigen: A unified generative framework for retrieval and question answering with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8688–8696, 2024.

[360] Tiziano Labruna, Sofia Brenna, and Bernardo Magnini. Dynamic task-oriented dialogue: A comparative study of llama-2 and bert in slot value generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 358–368, 2024.

[361] Juhi Tandon and Dipti Misra Sharma. Unity in diversity: A unified parsing strategy for major indian languages. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 255–265, 2017.

[362] Mostafa Keikha, Jae Hyun Park, W Bruce Croft, and Mark Sanderson. Retrieving passages and finding answers. In *Proceedings of the 19th Australasian Document Computing Symposium*, pages 81–84, 2014.

[363] Chen Qu, Liu Yang, W Bruce Croft, Falk Scholer, and Yongfeng Zhang. Answer interaction in non-factoid question answering systems. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, pages 249–253, 2019.

[364] Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE workshop on automatic speech recognition and understanding (ASRU)*, pages 813–820. IEEE, 2015.

[365] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods, 2022. URL `https://github.com/huggingface/peft`.

[366] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[367] Hardi Trivedi, Jorjeta G Jetcheva, and Carlos Rojas. Llm-based localization in the context of low-resource languages. In *2024 Conference on AI, Science, Engineering, and Technology (AIxSET)*, pages 276–289. IEEE, 2024.

[368] Zabir Al Nazi, Md Rajib Hossain, and Faisal Al Mamun. Evaluation of open and closed-source llms for low-resource language with zero-shot, few-shot, and chain-of-thought prompting. *Natural Language Processing Journal*, page 100124, 2025.

[369] Pengcheng Qiu, Chaoyi Wu, Xiaoman Zhang, Weixiong Lin, Haicheng Wang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Towards building multilingual language model for medicine. *Nature Communications*, 15(1):8384,

2024.

[370] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

[371] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020.

[372] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder for english. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 169–174, 2018.

[373] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic bert sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, 2022.

[374] Holger Schwenk and Matthijs Douze. Learning joint multilingual sentence representations with neural machine translation. In Phil Blunsom, Antoine Bordes, Kyunghyun Cho, Shay Cohen, Chris Dyer, Edward Grefenstette, Karl Moritz Hermann, Laura Rimell, Jason Weston, and Scott Yih, editors, *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2619. URL https://aclanthology.org/W17-2619.

[375] Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the association for computational linguistics*, 7:597–610, 2019.

[376] Qian Pan, Zahra Ashktorab, Michael Desmond, Martín Santillán Cooper, James Johnson, Rahul Nair, Elizabeth Daly, and Werner Geyer. Human-centered design recommendations for LLM-as-a-judge. In Nikita Soni, Lucie Flek, Ashish Sharma, Diyi Yang, Sara Hooker, and H. Andrew Schwartz, editors, *Proceedings of the 1st Human-Centered Large Language Modeling Workshop*, pages 16–29, TBD, August 2024. ACL. doi: 10.18653/v1/2024.hucllm-1.2. URL https://aclanthology.org/2024.hucllm-1.2.

[377] Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or LLMs as the judge? a study on judgement bias. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8301–8327, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.474. URL https://aclanthology.org/2024.emnlp-main.474.

[378] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[379] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[380] Vincent Beaufils and Johannes Tomin. Stochastic approach to worldwide language classification: the signals and the noise towards long-range exploration. 2021.

[381] Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the Forum for Information Retrieval Evaluation*, FIRE '14, pages 48–53, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3755-7. doi: 10.1145/2824864.2824872. URL http://doi.acm.org/10.1145/2824864.2824872.

[382] Jonathan Mark Kenoyer. Bull seal, harappa. URL https://www.harappa.com/indus/27.html.

[383] Ashavaree Das and Madhurima Sarkar. Pregnancy-related health information-seeking behaviors among rural pregnant women in india: validating the wilson model in the indian context. *The Yale journal of biology and medicine*, 87(3):251, 2014.

[384] Jasmeet Kaur, Asra Sakeen Wani, and Pushpendra Singh. Engagement of pregnant women and mothers over whatsapp: Challenges and opportunities involved. In *Conference Companion Publication of the 2019 on Computer Supported Cooperative Work and Social Computing*, CSCW '19, page 236–240, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366922. doi: 10.1145/3311957.3359481. URL https://doi.org/10.1145/3311957.3359481.

[385] Deepika Yadav, Kirti Dabas, Prerna Malik, Anushka Bhandari, and Pushpendra Singh. "should i visit the clinic": Analyzing whatsapp-mediated online health support for expectant and new mothers in rural india. In

Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391573. doi: 10.1145/3491102.3517575. URL https://doi.org/10.1145/3491102.3517575.

[386] Basant Potnuru et al. Aggregate availability of doctors in india: 2014–2030. *Indian journal of public health*, 61 (3):182, 2017.

[387] Monica Agrawal, Janette Cheng, and Caelin Tran. What's up, doc? a medical diagnosis bot. *Spoken Language Processing (CS224S), Spring*, 2017.

[388] Benilda Eleonor V Comendador, Bien Michael B Francisco, Jefferson S Medenilla, and Sharleen Mae. Pharmabot: a pediatric generic medicine consultant chatbot. *Journal of Automation and Control Engineering*, 3(2), 2015.

[389] Deepika Yadav, Prerna Malik, Kirti Dabas, and Pushpendra Singh. Feedpal: Understanding opportunities for chatbots in breastfeeding education of women in india. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW), November 2019. doi: 10.1145/3359272. URL https://doi.org/10.1145/3359272.

[390] Mamta Mittal, Gopi Battineni, Dharmendra Singh, Thakursingh Nagarwal, and Prabhakar Yadav. Web-based chatbot for frequently asked queries (faq) in hospitals. *Journal of Taibah University Medical Sciences*, 16(5): 740–746, 2021.

[391] Eric Brill, Susan Dumais, and Michele Banko. An analysis of the askmsr question-answering system. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 257–264, 2002.

[392] Govind Kothari, Sumit Negi, Tanveer A Faruquie, Venkatesan T Chakaravarthy, and L Venkata Subramaniam. Sms based interface for faq retrieval. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 852–860, 2009.

[393] Shriya Sahu, Nandkishor Vasnik, and Devshri Roy. Prashnottar: a hindi question answering system. *International Journal of Computer Science & Information Technology*, 4(2):149, 2012.

[394] Khyathi Raghavi, Manoj Chinnakotla, and Manish Shrivastava. " answer ka type kya he? " learning to classify questions in code-mixed language. 05 2015. doi: 10.1145/2740908.2743006.

[395] Satoshi Sekine and Ralph Grishman. Hindi-english cross-lingual question-answering system. *ACM Transactions on Asian Language Information Processing (TALIP)*, 2(3):181–192, 2003.

[396] Santosh Kumar Ray, Amir Ahmad, and Khaled Shaalan. A review of the state of the art in hindi question answering systems. *Intelligent Natural Language Processing: Trends and Applications*, pages 265–292, 2018.

[397] Jeanne E Daniel, Willie Brink, Ryan Eloff, and Charles Copley. Towards automating healthcare question answering in a noisy multilingual low-resource setting. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 948–953, 2019.

[398] Santiago González-Carvajal and Eduardo C Garrido-Merchán. Comparing bert against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012*, 2020.

[399] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Visualizing and understanding the effectiveness of bert. *arXiv preprint arXiv:1908.05620*, 2019.

[400] Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. Faq retrieval using query-question similarity and bert-based query-answer relevance. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1113–1116, 2019.

[401] Deepika Yadav, Anushka Bhandari, and Pushpendra Singh. Leap: Scaffolding collaborative learning of community health workers in india. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW), nov 2019. doi: 10.1145/3359271. URL https://doi.org/10.1145/3359271.

[402] Deepika Yadav, Prerna Malik, Kirti Dabas, and Pushpendra Singh. Illustrating the gaps and needs in the training support of community health workers in india. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380966. doi: 10.1145/3411764.3445111. URL https://doi.org/10.1145/3411764.3445111.

[403] Deepika Yadav, Pushpendra Singh, Kyle Montague, Vijay Kumar, Deepak Sood, Madeline Balaam, Drishti Sharma, Mona Duggal, Tom Bartindale, Delvin Varghese, and Patrick Olivier. sangoshthi: Empowering community health workers through peer learning in rural india. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 499–508, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee. ISBN 9781450349130. doi: 10.1145/3038912.3052624. URL https://doi.org/10.1145/3038912.3052624.

[404] Manraj Singh Grover, Pakhi Bamdev, Yaman Kumar, Mika Hama, and Rajiv Ratn Shah. audino: A modern annotation tool for audio and speech, 2020.

[405] Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, 2020.

[406] Gopalakrishnan Venkatesh, Abhik Jana, Steffen Remus, Özge Sevgili, Gopalakrishnan Srinivasaraghavan, and Chris Biemann. Using distributional thesaurus to enhance transformer-based contextualized representations for low resource languages. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 845–852, 2022.

[407] Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2020. URL `https://arxiv.org/abs/2004.09813`.

[408] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020.

[409] Mukul Banerjee. Bhopal state museum, deel 3, Oct 2012. URL `https://www.flickr.com/photos/mukulb/8082735403`.

[410] Colm Sweeney, Courtney Potts, Edel Ennis, Raymond Bond, Maurice D Mulvenna, Siobhan O'neill, Martin Malcolm, Lauri Kuosmanen, Catrine Kostenius, Alex Vakaloudis, et al. Can chatbots help support a person's mental health? perceptions and views from mental healthcare professionals and experts. *ACM Transactions on Computing for Healthcare*, 2(3):1–15, 2021.

[411] Pragnya Ramjee, Mehak Chhokar, Bhuvan Sachdeva, Mahendra Meena, Hamid Abdullah, Aditya Vashistha, Ruchit Nagar, and Mohit Jain. Ashabot: An llm-powered chatbot to support the informational needs of community health workers. *arXiv preprint arXiv:2409.10913*, 2024.

[412] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, 2020.

[413] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2421–2425, 2024.

[414] Yao Qiang, Deng Pan, Chengyin Li, Xin Li, Rhongho Jang, and Dongxiao Zhu. Attcat: Explaining transformers via attentive class activation tokens. *Advances in neural information processing systems*, 35:5052–5064, 2022.

[415] Avinash Anand, Kritarth Prasad, Chhavi Kirtani, Ashwin R Nair, Manvendra Kumar Nema, Raj Jaiswal, and Rajiv Ratn Shah. Multilingual mathematical reasoning: Advancing open-source llms in hindi and english. *arXiv preprint arXiv:2412.18415*, 2024.

[416] Avinash Anand, Raj Jaiswal, Abhishek Dharmadhikari, Atharva Marathe, Harsh Parimal Popat, Harshil Mital, Kritarth Prasad, Rajiv Ratn Shah, and Roger Zimmermann. Improving multimodal llms ability in geometry problem solving, reasoning, and multistep scoring. *arXiv preprint arXiv:2412.00846*, 2024.

[417] Avinash Anand, Raj Jaiswal, Abhishek Dharmadhikari, Atharva Marathe, Harsh Popat, Harshil Mital, Ashwin R Nair, Kritarth Prasad, Sidharth Kumar, Astha Verma, et al. Geovqa: A comprehensive multimodal geometry dataset for secondary education. In *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 102–108. IEEE, 2024.

[418] Avinash Anand, Janak Kapuriya, Apoorv Singh, Jay Saraf, Naman Lal, Astha Verma, Rushali Gupta, and Rajiv Shah. Mm-phyqa: Multimodal physics question-answering with multi-image cot prompting. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 53–64. Springer, 2024.

[419] Xiaochen Li, Zheng Xin Yong, and Stephen Bach. Preference tuning for toxicity mitigation generalizes across languages. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13422–13440, 2024.

[420] Tahira Naseem, Guangxuan Xu, Sarathkrishna Swaminathan, Asaf Yehudai, Subhajit Chaudhury, Radu Florian, Ramón Fernandez Astudillo, and Asim Munawar. A grounded preference model for llm alignment. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 151–162, 2024.

[421] Jonathan Mark Kenoyer. Ancient cities of the indus valley civilization, Sep 1998. URL `https://www.harappa.com/slide/elephant-seal`. Page 84.

[422] Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, 2012.

[423] Kushal Jain, Adwait Deshpande, Kumar Shridhar, Felix Laumann, and Ayushman Dash. Indic-transformers: An analysis of transformer language models for indian languages. *arXiv preprint arXiv:2011.02323*, 2020.