Building Telugu Corpora for NLP Applications: Paraphrasing, Question Answering, and Spelling Correction

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

Master of Science
in
Computer Science and Engineering by Research

by

Mani Kanta Sai Nuthi
2019701019
mani.kanta@research.iiit.ac.in

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

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Building Telugu Corpora for NLP Applications: Paraphrasing, Question Answering, and Spelling Correction**" by **Mani Kanta Sai Nuthi**, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. Manish Shrivastava

To my family

Acknowledgments

As I submit my MS thesis, I would like to take this opportunity to acknowledge all the people who helped me in my journey at IIIT-Hyderabad.

Firstly, I would like to express my gratitude to my guide, Prof. Manish Shrivastava, without whom this work would not have been possible. I want to thank him for his invaluable guidance and support throughout my research work. His expertise, insightful suggestions, and constructive feedback have been instrumental in shaping my research projects. I am sincerely grateful to him for providing me with the resources, encouragement, and motivation to complete my research work. I would also like to extend my thanks to him for including me in Sevak and Remote internship programs, which helped me to acquire valuable practical knowledge and experience.

I would like to express my appreciation to Rakesh and Saideep, who have been an integral part of my life and my journey at the institute. I want to thank my fellow labmates Pavan, Lokesh, Gopi, Ashok, and Hema for their constant support and for providing a positive learning environment.

Last but not the least, I express my deepest gratitude toward my parents, Shobha Rani and Ramesh, for their unconditional love and support throughout my journey.

Abstract

Natural Language Processing (NLP) is a rapidly growing field focusing on the interaction between computers and human languages. It involves utilizing computational techniques to understand and generate natural language text. Several NLP tasks, such as question answering, text summarization, and machine translation, are widely researched for many languages, including Indian languages. Indian languages are resource-scarce and have distinctive characteristics posing different challenges for NLP. Recent advancements in NLP have helped in the development of models and techniques that are specific to Indian languages. However, for Telugu, a south Indian language, a lot of research is still needed to improve the performance of several NLP systems. Progress of such systems will benefit the huge Telugu-speaking community worldwide to communicate and access information in Telugu through various NLP applications. This motivates us to develop essential NLP resources and systems for Telugu.

Firstly, the thesis provides an overview of the fundamental concepts and techniques used in NLP. Then we approach three specific NLP tasks: paraphrasing, question answering, and spelling correction. For these tasks, we address the problem of resource scarcity and then present the techniques to create the data for such low-resource languages.

In this thesis, we have presented paraphrasing, question-answering, and spelling correction resources for the Telugu language. For paraphrasing, we presented two manually created and annotated corpora of size 1544 and 10000+ samples, respectively. We have also discussed the necessity for manual intervention while creating such resources. We have introduced a Telugu Question Answering Dataset - TeQuAD, with a size of 82k parallel triples. We also proposed the guidelines and methodologies that can be followed to create a Question Answering dataset for low-resource languages. We presented a Spell correction system for the Telugu language with the help of a synthetically created dataset.

Contents

List of Figures

List of Tables

Chapter *1*

Introduction

Every day humans use Natural language to communicate with each other. On the other hand, machines cannot understand the words or text in Natural language, as they interpret information in 1s and 0s. Molding machines to comprehend the Natural language will make day-to-day human life easier. Using the large amount of processed data produced by humans while communicating across the world in numerous languages, machines can be taught in a way to understand and derive meaning from human languages. The computer science field that explores the development of such applications is known as Natural Language Processing (NLP). NLP is a branch of Artificial Intelligence that has roots in the field of linguistics. It is an important, rapidly growing, and challenging research topic with many applications, such as machine translation, question answering, summarization, spelling correction, and paraphrasing. Building such NLP applications might need large volumes of processed data and massive computational resources, depending on the level of application. The data resources for several NLP applications are limited to a small set of high-resource languages like English and are still not a reality for many Indian languages. Hence, the advancement of NLP applications for such languages needs to be explored to benefit millions of users. In this thesis, we worked on developing resources for different NLP tasks in the Telugu language. We approached three tasks of NLP, namely Paraphrasing, Question Answering, and Spelling Correction.

Paraphrasing is rewriting a text to express the same meaning differently, using alternate words, phrases, or sentence structures. It helps in avoiding plagiarism, simplifying a complex piece of text, or restructuring the text for the intended audience or purpose. It is a crucial task in NLP that will help to improve natural language understanding and generation and can be embedded into high-level applications such as Question Answering, Summarization, and Machine Translation.

Question Answering is another important task in NLP, where the model outputs or generates the answer to the question posed by the user. Question Answering involves multiple components such as information retrieval, natural language understanding, and natural language genera-

tion. QA systems are utilized in different applications, such as chatbots, search engines, and information retrieval systems. It is evolving into a pivotal and challenging task in the context of the increasing amount of unstructured data and the need to access specific essential information.

Spelling correction is an NLP task that involves detecting and correcting spelling errors in texts. Spelling correction will help in improving the readability and understandability of text by correcting the errors such as typos, phonetic mistakes, and grammatical errors in texts. Like paraphrasing, spelling correction can be embedded to improve the performance of several downstream tasks like text summarization, machine translation, and question answering.

1.1 Need for Language-Specific Resources and Systems

India is a multilingual nation, and research in the NLP areas, especially focusing on Indian languages, is going on extensively. There is a huge necessity for language resources in India, as it will benefit the multilingual society through various applications. Language-specific resources are essential because different languages have different vocabularies and grammatical structures. These resources help boost the performance of different NLP tasks for the specific language.

Recent advancements in deep learning approaches helped in performing complex NLP tasks without relying on hand-crafted resources. The availability of large data will benefit the language models for learning the deep semantics of the languages. The data resources in the English language are abundantly accessible, therefore, English recorded significant progress for numerous NLP tasks. A system trained on large data of English might not perform that well when employed for the Telugu language. But when the system is exposed to Telugu-specific resources, such as Telugu-English parallel corpora, the system's performance can be significantly improved.

Our motivation is to develop similar resources for the Telugu language. We presented multiple corpora for the Telugu language corresponding to different NLP tasks. We also presented the approaches followed to create these datasets, which can be applied to develop data resources for other languages.

Telugu language

Telugu belongs to the Dravidian language family, a group of around 26 languages spoken mainly in southern and central India. Like many Indian languages like Hindi, though Telugu follows subject-object-verb word order, it can vary depending on the emphasis and context. Telugu is a morphologically rich language. It is written in a script called Telugu script, a

variation of the Brahmi script. Telugu is the third most spoken language in India, after Hindi and Bengali. It has more than 80 million native speakers.

Creating resources for such a largely spoken language will benefit the huge community to communicate and access information in Telugu. By utilizing these created resources; developing and providing tools in the language will help in preserving the Telugu linguistic heritage, and bringing more native speakers into the digital world.

1.2 Challenges Involved in Creating Telugu Corpora for NLP Applications

Creating a corpus for NLP applications in Telugu can involve several challenges. A few major challenges are discussed below:

- **Limited resources:**

  Even though Telugu is spoken by a large community around the world, limited resources are available for creating a Telugu data set. Telugu is one of the many low-resource languages, which have relatively fewer NLP data resources or annotated datasets.

- **Manpower:**

  The creation of Manual resources will act as the foundation step for providing high-quality resources for a specific language. Creating such resources requires manpower. A decent number of native language speakers have to be involved in the data creation activity, which can be costly. Moreover, human-generated data can be noisy and have to be supervised rigorously through extensive human effort.

- **Time-consuming:**

  Telugu is a morphologically rich language and consumes relatively more time in data resource creation. Additionally, irrespective of language, the data creation process itself will take time for training the participants, providing necessary feedback to participants, and filtering out the data.

- **Guidelines:**

  Proper guidelines are necessary to create a dataset of high quality. Guidelines help to ensure that the dataset is consistent, task-intended, and properly annotated. Guidelines

have to be clearly defined according to the task and data resources, with the help of linguistic experts.

- **Lack of standardization:**

  Similar to many languages, Telugu also has many dialects and variations. Such variations can lead to difficulty in standardizing the corpus that represents all the variations of the language.

- **Lack of pre-trained models:**

  As Telugu is a low-resource language, several state-of-the-art pre-trained models are not available for the Telugu language. Without the help of such pre-trained models, it is difficult to create synthetic datasets in Telugu for different NLP tasks.

Aside from these common challenges, there are task-specific challenges that will be discussed in the next chapters.

1.3 Reasons to choose these NLP problems

For Telugu, adequate research has been done on fundamental levels of NLP, such as Morphological, Lexical, and Syntactic levels. Different resources and tools were made available for Telugu in tasks such as POS Tagging, NER tagging, parsing, etc. However, there has been very little focus on developing tools and resources for higher-level NLP problems in Telugu. Research on high-level tasks leads to the development of models capable of analyzing the structure and meaning of text beyond a single sentence, making connections between words and sentences. Contributions to high-level tasks for a language will help in considering complex insights from the data, automating the task, and reaching much closer to the development of real-time tools/applications in the respective language.

Paraphrasing, question answering, and spelling correction are some of the crucial problems in the field of NLP, which requires a deeper understanding of language structure, meaning, and context to perform. Question answering and paraphrasing tasks involve texts with multiple sentences, and to operate on such data, a discourse level of understanding is essential. We also approached the spelling correction problem on a contextual level using pre-trained language models. We discuss the reasons to analyze and understand the input texts on a semantic level for fundamental spelling correction tasks in the later chapters.

1.4 Our contributions

- We have presented two manually created and annotated paraphrase corpora for Telugu, namely MCP and MAP respectively. Manually annotated paraphrase (MAP) corpus consists of 10k samples, while the Manually created paraphrase (MCP) corpus consists of 1544 paraphrases. We have also discussed the necessity for manual intervention while creating such resources.
- We have introduced a Telugu Question Answering Dataset - TeQuAD, with the size of 82k parallel triples created by translating triples from the SQuAD. We also proposed the guidelines and methodologies that can be used to create a Question Answering dataset for low-resource languages.
- We presented a Spell correction system for the Telugu language with the help of a synthetically created dataset.

We made the created datasets publicly available.

1.5 Organization of Thesis

This thesis is divided into six chapters. The current chapter gives an introduction to the different NLP tasks we approached, the motivation for choosing the problems, and the challenges involved in creating data resources for Telugu. Chapter 2 presents an overview of relevant work that was done in the past for paraphrasing, question-answering, and spelling correction systems. Chapter 3 discusses the Paraphrase corpora creation for Telugu. Chapter 4 discusses the creation of Question answering dataset for Telugu. Chapter 5 explores spelling correction task for Telugu. In chapter 6, we present the conclusion and future research works.

Chapter 2

Literature Review

We briefly discussed the introduction to paraphrasing, question answering, and spelling correction tasks in the previous chapter. In this chapter, we will discuss related work done for these NLP tasks (in section 2.1). We also present an overview of different word representation techniques (in section 2.2) and modeling approaches (in section 2.3) used in this thesis.

2.1 Related Work

2.1.1 Paraphrasing

Paraphrases are pairs of sentences constructed with different words or phrases restating the meaning of another piece of writing. In paraphrasing, an original or source text is considered, and a different text expressing the equivalent meaning is phrased. For example, "John was saddened by the sudden pandemic" could be paraphrased as "The unexpected pandemic depressed John". Although the textual constructions in these sentences are different, the meaning conveyed is the same. Different levels of paraphrasing can be done based on the type of variation introduced in the source sentence. For instance, in Lexical level paraphrasing, words in the source sentences are replaced with their synonyms to generate the paraphrased texts. So for a source sentence like "Recent progress in language modeling is incredible", paraphrases such as "Recent advancement in language modeling is incredible", and "Recent growth in language modeling is incredible" can be generated by replacing the word "progress" with its synonyms. With phrasal level paraphrasing, sentences like "Although government implements privatization programs, progress has been slower than expected" can be paraphrased to "Although government implements privatization programs, progress has not been as fast as expected". On the other hand, changing the structure of a sentence, such as "NLP learning students", resulting in "The students who learn NLP", is another way of paraphrasing. Such paraphrased examples obtained using individual techniques might not exhibit substantial textual differences. So, combining multiple paraphrasing techniques is followed to generate diverse paraphrased texts.

In everyday life, paraphrasing can be used in research papers, news articles, books, and different sorts of writing items. Paraphrasing is vital to simplify a piece of text which is difficult

to understand or to rewrite a piece of text to adjust to the flow of writing. It is helpful in avoiding plagiarism. In NLP, Paraphrasing can be broadly classified into two tasks: Paraphrase detection and Paraphrase generation. Paraphrase detection requires a deep semantic understanding of the natural language texts, hence improving the Language Understanding capability of NLU tasks. And, Paraphrase generation will assist the tasks such as Question Answering, Summarization, and Machine Translation in improving natural language generation.

Like many NLP tasks, paraphrasing research requires data resources for its advancement. Although there are few paraphrasing corpora available, they are limited to only high-resource languages like English.

Microsoft Research Paraphrase Corpus (MRPC) [16] is a popular paraphrase corpus available in the English language. MRPC was extracted by applying heuristic techniques on topic-clustered news data to detect candidate document pairs. Further, candidate sentence pairs were extracted from the candidate documents and filtered with an SVM classifier. Finally, the sentence pairs were manually annotated with the binary scores. '0' represents a non-paraphrase relation, while '1' represents a paraphrase relation.

Lan et al. [27] created Twitter URL Corpus (TUC) by using the data collected from Twitter. Candidate sentence pairs were extracted from tweets with similar URLs, followed by the manual annotation of the sentence pairs.

Besides English, few other languages have seen significant contributions in paraphrasing resources. Considering news articles and subtitles as the data sources, Kanerva et al. [22] and Demir et al. [13] introduced manually annotated paraphrase corpora for the Finnish (53k pairs) and Turkish (1270 pairs) languages, respectively.

Several other research works have created automatically extracted paraphrase datasets by using the language pivoting technique [4] on the multilingual parallel corpus. Tapaco [34], a multilingual paraphrase dataset available for 73 languages and Opusparcus released by Creutz [10] covering six European languages, are created by applying pivoting techniques on candidate paraphrases extracted from the Tatoeba (a large-scale multilingual crowdsourcing dataset) and OpenSubtitles2016 corpus (Tv and Movie subtitles collection) respectively.

For Indian languages, very few research works focus on resource creation for paraphrasing tasks. Singh et al. [36] created a paraphrasing dataset for four Indian languages: Hindi, Punjabi, Tamil, and Malayalam. The dataset is created by extracting the candidate paraphrase sentences from news headlines and articles discussing the same event on different news sites. Based on the semantic relation between the sentence pairs, around 30k pairs are manually annotated with three labels. IndicParaphrase, introduced in Indic NLG suite [25], is a synthetically created paraphrase dataset for multiple Indian languages, including Telugu. Sentence pairs for the paraphrase dataset are obtained from Samanantar Parallel corpus [31], and pivoting technique is applied to generate the paraphrase pairs. Assuming aligned paraphrase sentences are semantically similar, a variance filter is employed to filter out less diverse sen-

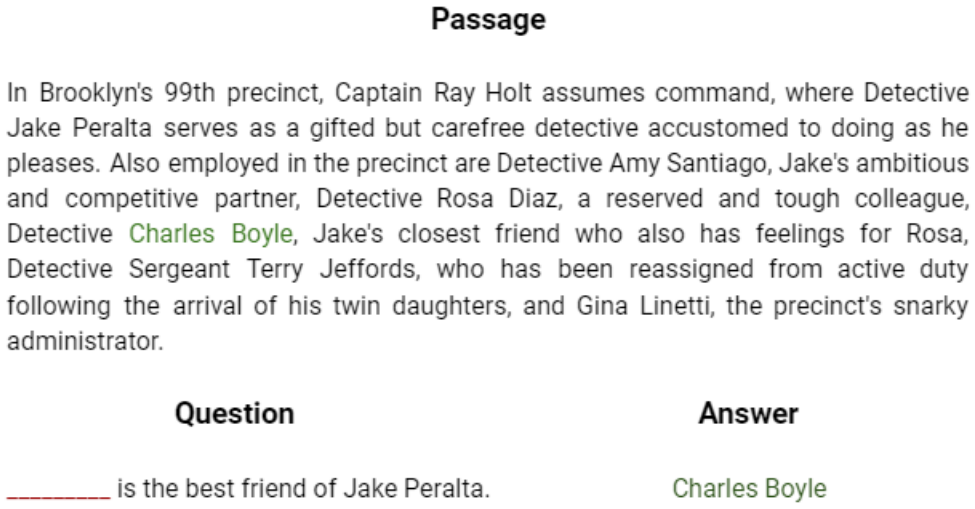

Figure 2.1: Example for Cloze-style Question Answering. For Span extractive QA instance, see 2.2.

tences. Aravinda Reddy et al. [2] manually annotated around 4000 Telugu sentence pairs for the paraphrase identification task, but not made the data publicly available.

2.1.2 Question Answering

With the advancement of deep learning and the availability of huge data resources, the reading comprehension ability of machines has also been improved. Machine Reading Comprehension is one of the prominent fields of NLP, where we test the ability of machines to understand natural language text. Machines are taught to understand the data provided and answer the posed questions by using the learned knowledge. The answers provided by QA systems can either be extracted from provided knowledge or can be generated. Locating the position of the answer from the provided text is known as Span Extractive Question Answering, while filling the blanks with generated answers is Cloze-style Question Answering. See 2.1 for examples of these QA systems. QA systems have significant applications in NLP. In our day-to-day life, QA systems can be seen in chatbots, speech assistants, and search engines. They provide relevant information efficiently and swiftly across several domains.

Among the available QA data resources, Stanford Question Answering Dataset (SQuAD) [30] is one of the popular Span-based QA datasets. The SQuAD was created through crowdsourcing and is available in the English language. High-quality Wikipedia articles belonging to different topics were collected, and the crowd workers created multiple questions on these articles. The SQuAD contains 100000 parallel triplets: sets of paragraphs, questions, and answers. In many instances, multiple question-answer pairs were generated for a single paragraph. The answer to

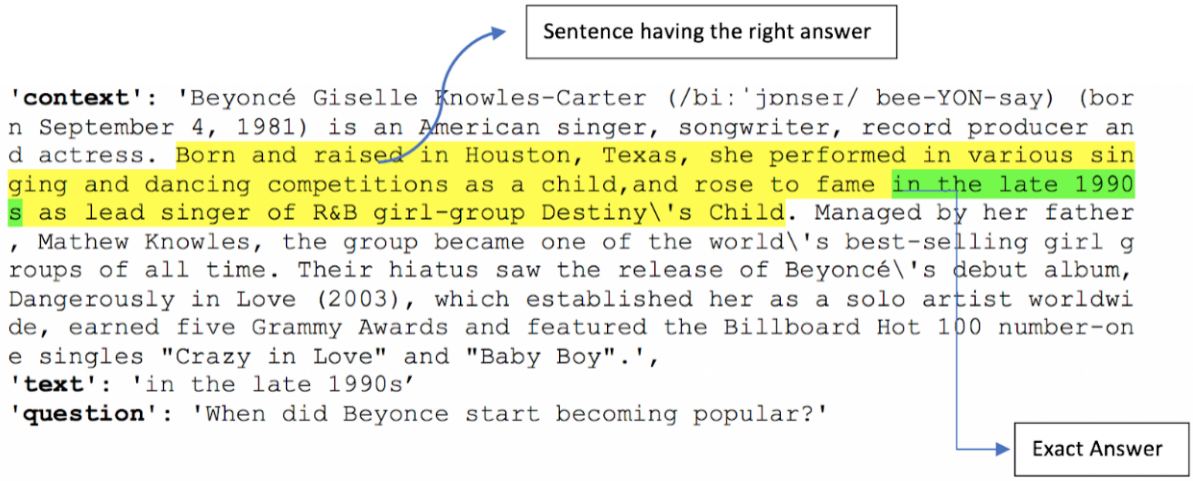

Figure 2.2: Example for Span Extractive Question Answering.

the question is present in the corresponding parallel paragraph itself, and the answer can be a single word or a lengthy phrase with multiple words. The answer is represented in terms of span indices, which denote the position of the answer in the paragraph. Indicating the answers with span indices will resolve the ambiguity of identifying the exact answers if multiple instances of answer phrases exist in the paragraphs. See 2.2 for a SQuAD instance example.

Besides SQuAD, the English language has also seen popular QA resources like NewsQA [38] and CNN/Dailymail [7] datasets. These datasets are huge and high-quality datasets and helped in achieving significant progress for this specific language in NLP. In order to improve the MRC task for other languages, accessibility of high-quality QA resources in respective languages is necessary. The primary reason for the unavailability of such resources in many languages is that creating a high-quality QA resource is time-consuming and costly.

Similar to SQuAD, a few research works have also contributed to creating QA data resources for low-resource languages by utilizing Wikipedia articles. This way, Lim et al. [28], Efimov et al. [17], Cui et al. [11], d'Hoffschmidt et al. [14] have introduced MRC datasets for Korean, French, Russian, and Chinese languages, respectively.

Clark et al. [9] presented a QA dataset covering 11 typologically diverse languages with 200k question-answer pairs.

Few others created the datasets in other languages by translating the SQuAD and then employing different methods to obtain the span indices of answers in target languages [6] [1].

Besides creating the data resources, a few works proposed methods to improve the performance of MRC models in low-resource settings. Hsu et al. [20] investigated zero-shot cross-lingual transfer learning on question-answering tasks. Through the experiments, they showed that translation of QA data resources from source languages to target languages, i.e., low-

resource languages is not essential in order to improve RC tasks in low-resource languages. Furthermore, they added that the translation might degrade the performance of the MRC model for low resource languages.

Bornea et al. [5] presented a data augmentation technique utilizing machine translation to boost multilingual transfer learning.

Liu et al. [29] and Cui et al. [12] addressed leveraging translated information from high-resource languages to improve the MRC performance for low-resource languages. Cui et al. [12] proposed different back translational approaches for cross-lingual experiments. They also discussed multiple methods to align the answer phrases in the target language. And then, pointing out the flaws of aligning approaches, they presented the 'Dual BERT' model. Dual BERT model exploits the semantic information from bilingual QA pairs and improves the performance of MRC for low-resource languages.

Such MRC models, which rely on knowledge from high-resource languages to perform well in low-resource languages, failed to extract exact answers from the paragraphs in low-resource languages. Yuan et al. [41] proposed phrase boundary supervision tasks to enhance the answer boundary detection capability in low-resource MRC models, which exploit cross-lingual transfer learning.

Reddy et al. [33] discussed the post-correction methods to improve the answer span extraction. New layers are added on top of pre-trained transformer-based language models to inspect and adjust the answer predictions.

2.1.3 Spell Correction

Spell correction is an essential task in NLP because it can help enhance the overall quality and readability of written text, which is necessary for many applications such as search engines, chatbots, and text-based communication. In addition, it can help resolve the ambiguity and confusion that can arise from spelling errors, which is particularly important in fields such as medicine and legal writing that require precision and accuracy.

For indian languages , spell correction is a challenging task in Natural Language Processing (NLP) because Indian languages have complex scripts and phonetic systems, with numerous characters and diacritics. Additionally, Indian languages also have a large number of words with similar spellings and pronunciations, which might result in ambiguity and confusion in text. For instance, in Telugu, the word "కల" (kala) that signifies "dream" can be mistaken for "కళ" (kaLa) meaning "art" causing mistakes in automated systems. To address these challenges, researchers and developers have developed various approaches for spell correction in Indian languages, including rule-based methods, statistical models, and machine learning techniques.

Dixit et al. [15] and Rao et al. [32] presented spell checkers for Marathi and Telugu languages, respectively, based on morphological analysis.

Dixit et al. [15] discussed the architecture and implementation of a rule-based spell-checker for Marathi, which is a major Indian Language. This is the first initiative to create a spell-checker for Marathi that utilizes morphological analysis. A spell checker based on rules of morphology and orthography was designed and produced promising results. However, it.is entirely dependent on the vocabulary list and set of rules.

Rao et al. [32] created a spell checker for Telugu, a challenging Indian language that has a complicated structure. This spell-checker uses morphological analysis and sandhi splitting rules.

For Indian languages [26], few works on spelling correction problems approached in two steps: error detection and error correction. The error detection process involves identifying any spelling errors in the provided input text, and the error correction process comes up with possible correct words. For spelling error detection, extensively researched algorithms are n-grams and dictionary lookup methods. And for spelling error correction, techniques like Edit-distance, similarity keys, rule-based techniques, n-gram-based techniques, probabilistic techniques, and neural networks are most commonly used.

Kaur and Singh [23] is a spell-checker specifically designed for the Hindi language that employs a hybrid approach. The dictionary lookup technique is used to detect errors in the input text by examining each word in the text against a Hindi dictionary that has been created. If the word is present in the dictionary, it is considered a correct word. If it is not found, it is deemed an error word and added to a list of error words. The error correction process consists of two stages: generating potential correct suggestions for the error word, and ranking these suggestions. Kaur and Singh [23] uses a weightage algorithm, the minimum edit distance method, and statistical machine translation techniques for error correction.

KS et al. [24] designed a spell checker for Malayalam using the two-stage approach. Similar to [23], error detection is done by using the dictionary lookup approach. The N-gram based technique corrects errors by determining the similarity between words and calculating a similarity coefficient.

More advanced approaches may use machine learning techniques like neural networks or deep learning to learn from large amounts of text data and provide more precise spell corrections.

Etoori et al. [18] built word-level spelling corrector for Hindi and Telugu languages. They proposed a sequence-to-sequence architecture with a Long Short Term Memory (LSTM) encoder and an LSTM decoder for spelling correction. For experimentation, they created synthetic data by incorporating character errors into the most frequently used words and movie names in Hindi and Telugu languages.

2.2 Word Embeddings

In general, the data in NLP tasks, i.e., the plain texts, are represented in terms of vectors. Such numerical representations of texts are essential, as machines are incapable of processing plain text for performing mathematical computations in NLP Applications. The vector representations for words are known as Word Embeddings. Word Embeddings are dense and low-dimensional vector representations of words. Instead of mapping some random vector to a word, several methods learn embeddings in a way that they preserve semantic and syntactical information of words in the text. So, the words with a similar meaning or the words that appear in a similar context will have a similar vector representation. Word representations can be broadly classified into Contextual and Non-Contextual word representations. Methods like Word2vec and FastText generate Non-Contextual representations, as they generate the same representation for a word irrespective of the context they appear. However, Contextual representation models like BERT and ALBERT might generate different vector representations for the same word seen in different contexts.

2.2.1 FastText

FastText is similar to Word2Vec, a very popular word embedding approach that uses information from neighboring words to learn the features of the current word in the text. Word2vec generates vector representations for every single word. But in the real world, it is nearly impossible to cover all the words in the vocabulary of train data. Such words are known as Out-of-Vocabulary words. And also there are rare words that are not frequent in the data. Word2Vec is incapable of learning representations for such OOV and rare words effectively. On the other hand, FastText learns representations for the words based on their n-grams. For instance, the tri-grams for the word 'fasttext' will be broken down into 'fas','ast','tte','tex', and 'ext'. Now the sum of all these n-grams is considered the word embedding for the word 'fasttext'. This way, FastText is able to obtain representations for the OOV and rare words by using their n-grams.

2.2.2 BERT

BERT stands for Bidirectional Encoder Representations from Transformers. BERT Model architecture is based on Transformers, a sequence-to-sequence model that learns contextual relations between words in the text. The transformer contains an encoder that reads the input and a decoder that generates the predicted output. As BERT's objective is to learn a language model, only the Transformer encoder is included. Unlike several models that read the text data from left to right or right to left, BERT reads the entire sequence of words in the text at once, hence called bi-directional. BERT is trained using two techniques, namely Masked Language

Model (MLM) and Next Sentence Prediction (NSP), to learn about the context of the word and the connection between the sentences, respectively.

2.2.3 ALBERT

The architecture of ALBERT is similar to BERT's architecture, and ALBERT also employs MLM tasks to learn embeddings from data. Nevertheless, ALBERT does not use NSP loss to learn sentence relations. Instead, it adopts Sentence Order Prediction (SOP) technique. NSP computes binary classification loss by predicting whether two sentences are consecutively related. The downside of this loss is that it checks for coherence along with the topic to recognize the immediate sentence, whereas SOP just focuses on sentence coherence. Also, in BERT, the input layers and hidden layers have the same embedding size. But these have been separated in ALBERT, reducing the size of the parameter by 80%. Parameter sharing between encoder layers is also introduced in the ALBERT model, reducing the size of parameters further when compared to BERT.

2.3 Modeling Approaches

2.3.1 Sequence-to-sequence

Sequence-to-sequence (seq2seq) models are popular deep learning models that have achieved great success in a variety of Natural Language Generation tasks such as Summarization, Machine Translation, and Speech Recognition. A Seq2seq model will process a sequence of units and generates or outputs some other sequence of units. The most common architecture of the seq2seq model comprises two neural network components: an encoder that takes the input and a decoder that outputs the predictions. The encoder reads the input sequence and transforms the information present in it into hidden state vector representation These representations are then used by the Decoder to generate the output sequences. The size of vector representations produced by the Encoder and Decoder might be different. Based on the application, Seq2seq models also incorporate different forms of attention integrated with the Encoder and Decoder. Attention, in general, helps the Encoder and Decoder to focus on a specific part of the sequence, reducing the burden of encoding or decoding the whole sequence. Encoders and Decoders are built with basic Neural Network blocks such as Recurrent Neural Networks (RNNs), Long Short Term Networks (LSTMs), and Transformers. Both Encoder and Decoder might have similar or different types of blocks.

2.3.2 BERT Adapters

Pretrained language models like BERT are finetuned further to perform well in the downstream NLP tasks. As an alternative to this finetuning process, the training of adapters is

proposed for the downstream tasks. Adapters are lightweight add-ons to the language models. They are a few additional modules introduced in between or on top of the language models, like BERT. While training with adapters, only the parameters of adapter layers are adjusted, while the weights of the language model are frozen. As adapters are layers with tiny sets of parameters, they can be trained fastly and reproduced easily. They are parameter efficient and recorded par performance when compared with BERT.

Chapter *3*

Paraphrase Corpora for Telugu Language

3.1 Introduction

Paraphrasing is restating the meaning of a text or passage using alternate words. A Paraphrase is a different textual construction that conveys the same information present in another text. The simplest way of creating a paraphrase is on the lexical level by replacing the words with their synonyms in the text. Besides the lexical level, paraphrasing can be observed at phrasal and sentential levels. Text pairs are determined as paraphrases based on the meaning or semantics of the text. So, all text pairs of the same length cannot be classified as paraphrases, and all the paraphrase pairs are not necessarily of the same length.

In NLP, many paraphrase datasets are introduced as paraphrasing resources, while few focus on either the Paraphrase detection task or the Paraphrase generation task. The task of detecting if a pair of sentences convey the same meaning is Paraphrase detection, where the task of generating new diverse sentences from provided natural language sentences without altering the semantic information is known as Paraphrase generation. Both of these tasks need a deep semantic understanding of the texts. These tasks are integrated into multiple NLP applications, such as data augmentation, question answering, summarization, and semantic parsing.

Progress in many NLP areas goes hand in hand with the availability of training and evaluation datasets. The lack of annotated resources hinders the advancement of different NLP tasks for several languages worldwide. Similarly, even though paraphrasing is an essential NLP task, accessibility to paraphrasing resources is limited to a small set of high-resource languages. Whilst some decent research work has been done in paraphrasing tasks for a few Indian languages, not many notable efforts are proposed for a Dravidian language like Telugu. No manually annotated or created datasets are publicly available for Telugu to date. Although manually created resources have appreciable advantages, creating a manually annotated or generated dataset of good quantity and quality is difficult, time-consuming, and requires manpower.

In this chapter, we discuss the creation of two types of paraphrase resources, namely the Manually Annotated Paraphrase (MAP) dataset and the Manually Created Paraphrase (MCP) dataset.

3.2 Manually Annotated Paraphrase Dataset

In this section, we will discuss the creation of the Manually Annotated Paraphrase dataset. We explain the sources considered for collecting the data, followed by the data creation process.

3.2.1 Data Source

Three different multilingual sentence-aligned parallel data sources, namely the Mann ki Baat corpus, PIB corpus, and Samanantar dataset, are considered for data collection.

3.2.1.1 Mann ki Baat corpus

Mann Ki Baat (MKB) is a collection of speeches expressed by the Prime Minister of India. MKB Speeches were addressed in the Hindi language and then translated & transcribed into 12 Indian languages (Hindi, Telugu, Kannada, Marathi, Gujarati, Urdu, Tamil, Malayalam, Oriya, Bengali, Manipuri, Assamese) and English. Translated texts are of good quality and are publicly available online on the MKB website. Siripragada et al. [37] crawled the multilingual article texts from the site and arranged document-level parallel alignments based on the posted date of the article. Sentence alignments were obtained using the BLEUAlign approach [35] and Transformer based NMT model.

3.2.1.2 PIB corpus

The Press Information Bureau (PIB) is the nodal agency of the Indian Government that releases information regarding government initiatives, achievements, and events to the media. Disseminated information is publicly available online on the PIB organized website[1]. A news article about an event is translated manually into multiple Indian languages, including Telugu, and made accessible on the website. Siripragada et al. [37] crawled such PIB news articles and extracted the document-level alignments using an approach similar to Uszkoreit et al. [39]. For sentence alignments, methods discussed in Mann Ki Baat Corpus are applied.

3.2.1.3 Samanantar

Samanantar is a parallel corpora collection available for Indic languages. It comprises sentence pairs between English and 11 Indic languages, including Telugu. Samanantar is built by collating multiple existing parallel corpora resources and then extending the data through

[1] https://www.pib.gov.in/

parallel sentences mined from the web. Among the resources collected for Samanantar, the PIB corpus is also included. So, we took care of the common samples that appeared from both sources.

3.2.2 Dataset Creation

As we discussed earlier, a paraphrase is a different textual construction carrying similar semantic information to the original text. So, first, we synthetically generated paraphrased texts for Telugu. And then, we manually annotate the paraphrases based on their semantic relation, and then we apply a filter to spot the less diverse samples. Therefore, the data creation process involves three phases: a. Pivoting approach, b. Manually annotation, c. Variation filter.

First, we collect the English-Telugu parallel sentences from MKB and PIB multilingual parallel corpora released by Siripragada et al. [37]. Then we apply the very common pivoting approach to these collected candidate sentence pairs to obtain paraphrases in the same language.

3.2.2.1 Pivoting approach

Assume an aligned multilingual sentence pair is semantically similar, upon translation, a sentence with different textual construction is generated while the semantic information is preserved, making it a paraphrase. The text in the source language will be translated into the target language to generate paraphrased texts in the target language. The source language will act as a pivoting language here.

From Telugu-English parallel sentence pairs, we consider the Telugu sentences as the original or actual texts. Then we apply the pivoting technique and translate English sentences into the Telugu language. Translated Telugu texts will have similar semantic information to their English texts. And English texts carry similar semantic information to the original Telugu texts. So, Translated Telugu texts and Original Telugu texts are semantically similar with textual variation, making them paraphrases of each other.

In this way, we used the sentence pairs from MKB and PIB multilingual parallel corpora and obtained synthetically generated paraphrases in the Telugu language. In the case of Samanantar corpora, instead of extracting sentence pairs and applying the pivoting approach, we considered the available Telugu candidate paraphrase pairs from the IndicParaphrase dataset [25] to reduce the effort.

Obtained candidate Telugu paraphrase pairs will be manually annotated based on their semantic relation, and then a variation filter will be applied to filter textually less diverse samples.

Data Source	No. of samples annotated	No. of valid paraphrases	Average no. of words per paraphrase
Mann ki Baat	4015	3506	14.17
PIB	1438	1096	17.89
Samanantar	4590	2657	8.27
Total	10043	7259	12.01

Table 3.1: Manually Annotated Paraphrase data Statistics

3.2.2.2 Manual annotation

The pair of Original and Translated Telugu texts were manually annotated with binary scores i.e., 1 and 0. '1' represents the semantic similarity relation between the sentences, while '0' represents a non-semantically similar relation. The dataset obtained after the annotation can be served as the Semantic Textual Similarity dataset for Telugu. Annotation of the corpus was done with the help of three annotators and obtained a Fleiss kappa score of 0.72, indicating a Substantial agreement. A total of 10043 sentence pairs were manually annotated.

3.2.2.3 Variation filter

A filter is applied to identify the samples with slighter or no textual variation. The variation score[2] is computed using the n-gram overlap between original and paraphrased texts. Sentence pairs with a score greater than 0.8 are marked as non-paraphrases for their textual similarity.

$a_n = \frac{o_n}{i_n}$

$b_n = \frac{o_n}{p_n}$

$$score = \frac{\sum_{n=1}^{4} \frac{2}{\frac{1}{a_n} + \frac{1}{b_n}}}{4}$$

,where o_n = n-gram overlap between input and paraphrase texts, i_n =Total n-grams in input text and p_n =Total n-grams in paraphrase text. This formula computes the average of 1-, 2-, 3-, and 4-gram overlaps.

Data statistics are shown in table 3.1.

3.2.3 Why Manual Annotation

In the above data creation process, we manually annotated the candidate Telugu paraphrase pairs obtained through the pivoting approach. The reasons for manually labeling the candidate paraphrase samples are discussed below.

- The candidate paraphrases are obtained through the pivoting approach using the parallel corpora. Hence, candidate paraphrase pairs are considered to be semantically similar

[2]Variation formula is used in [25]

based on the assumption that samples in parallel corpora are aligned flawlessly. On observation, several misaligned samples were found in the multilingual parallel datasets discussed above. These datasets were created using heuristic and deep learning approaches, therefore, a proportion of sentence pairs can be aligned incorrectly. On top of that, parallel sentences might contain different information. Sentences might contain additional information or suffer from information loss to some extent, as these sentences were extracted from different news articles or sources.

- Pivoting approach involves Machine translation, and translation challenges for Telugu, a morphologically rich language, will affect the generation of proper translations. So, the Telugu translation of a well-constructed and semantically similar sentence in English might suffer from attaining decent adequacy and readability.

3.3 Manually Created Paraphrase Dataset

In this section, we will discuss the creation of the Manually Created Paraphrase dataset for Telugu. We explain the data collection then data creation, followed by the evaluation of the data.

3.3.1 Data Collection

News articles are the common source of dataset creation in NLP for several languages. We crawled the Telugu news articles from different regional news websites. Then we applied the sentence tokenizer to the articles to extract texts of one, two, and three sentences. These texts are shared with creators to create paraphrases.

3.3.2 Data Creation

Data Creators were trained for multiple weeks to understand the paraphrasing data creation task. We guided the creators to create texts with not just lexical level or phrasal level variation but also sentential level by rewriting the text information in their own words. Different paraphrasing techniques, such as using synonyms, changing the word order, writing from a different point of view (active-passive or direct-indirect speech conversions), changing the grammatical structure, and changing the form of words are suggested with proper examples. Examples for paraphrasing techniques are mentioned in tables 3.2, 3.3, 3.4, 3.5, 3.6 and 3.7. Our aim is to create diverse paraphrased texts that are semantically similar to original texts. So, the creators are advised to develop paraphrases by combining the above-mentioned paraphrasing techniques to create maximum variation in the paraphrases while the readability and semantic information are not compromised.

Original text	Paraphrase text
Vidya, udyōgāllō reḍlaku tīrani naṣṭa jarugutundani vyākhyāniñcāru.	Vidya, udyōgāllō reḍlaku an'yāya jarugutundani annāru.

Table 3.2: Example for paraphrasing technique - Use synonyms.

Original text	Paraphrase text
45.5 Grāmula baṅgāra apaharanaku guraindi. Cōri cēsina vāri vivarālu iṅkā teliyalēdu.	45.5 Grāmula baṅgārānni gurtu teliyani vyaktulu apahariñcāru.

Table 3.3: Example for paraphrasing technique - Change the form of words.

Original text	Paraphrase text
Vēlādi mandi prajalu ī dvīpanlō nivāsistāru.	Ī dvīpa vēlādi mandi prajalaku āśraya istundi.

Table 3.4: Example for paraphrasing technique - Write from a different point of view : active-passive

Original text	Paraphrase text
Vaccē ennikallō vāri pārṭī gelupu khāyamani rēvant reḍḍi annāru.	"Vaccē ennikallō mā pārṭī gelupu khāya " ani rēvant reḍḍi annāru.

Table 3.5: Example for paraphrasing technique - Write from a different point of view : direct-indirect

Original text	Paraphrase text
Anēka prabhutvālu praivēṭīkaraṇa kāryakramālanu amalu cēyaḍāniki caryalu tīsukunnappaṭikī, purōgati ūhiñcina dāni kaṇṭē nem'madigā undi.	Anēka prabhutvālu praivēṭīkaraṇa kāryakramālanu amalu cēyaḍāniki caryalu tīsukunnappaṭikī, ī praṇāḷika āśiñcinanta vēgaṅgā lēdu.

Table 3.6: Example for paraphrasing technique - Change the grammatical structure.

Original text	Paraphrase text
Bhadrācala , ēṭūrunāgāra , adilābād aiṭīḍī'ēla paridhilō unna girijanulaku upādhi kalpiñcālanē uddēśantō minī prāsiṅg yūniṭlanu nelakolpālani kōrāru.	Ēṭūrunāgāra , adilābād, bhadrācala aiṭīḍī'ēla paridhilō unna akkaḍi girijanulaku upādhi kalpiñcālanē uddēśantō minī prāsiṅg yūniṭlanu nelakolpālani kōrāru.

Table 3.7: Example for paraphrasing technique - Change the word order.

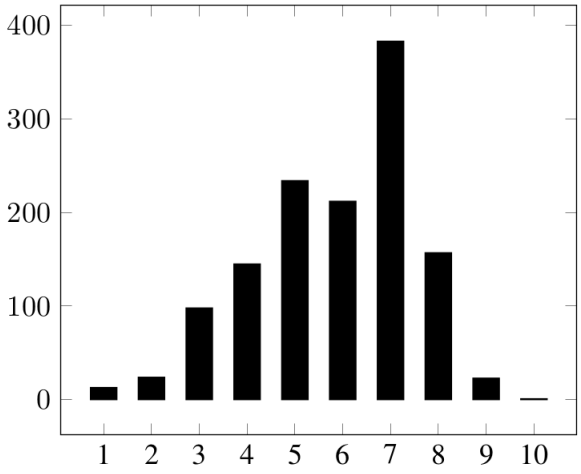

Figure 3.1: Distribution of Variance Scores in MCP

Data creators are provided with a tool that provides the interface to create paraphrases for original texts. The interface will not allow the creators to submit the paraphrased texts unless the variation in the texts is more than 30% compared to the original texts.

Creating paraphrases while focusing heavily on linguistic diversity might alter the semantic information. So, we also suggested them to not include out-of-context information, convey the wrong information, use incorrect tense/number/gender, lose the information provided in original texts, and introduce typos. During the pilot study, paraphrase samples created by annotators are collected, and relevant improvements are suggested. Through the data creation process, we collected a total of 1544 manually created paraphrases.

3.3.3 Data Evaluation

Unlike the simple binary evaluation in MAP, manually created paraphrases are evaluated with seven metrics, namely Readability, Adequacy, Variation, Grammatical errors, Out-of-the-context-info, Info loss, and Wrong info.

While the latter four metrics are evaluated with binary scores (0/1), the score ranges of Readability, Adequacy, and Variation are 0-5, 0-5, and 0-10, respectively.

All these metrics represent different linguistic features of the created sentence, but Adequacy and Variation evidently determine the paraphrase quality of the created sentence. While most of the sentences claim good Adequacy scores, Variation was observed to be normally distributed across the range (See figure 3.1). The reason is that creating a sentence with the lowest Variance score will result in a low-quality paraphrase, at the same time, possessing a high Variance observed to be affecting the Adequacy of the sentence.

Model	Precision	Recall	Accuracy
mBERT	0.901	0.899	0.835
Indic BERT	0.902	0.913	0.850
Adapter-BERT	0.852	0.966	0.838

Table 3.8: Experimental results of Paraphrase detection task on MAP dataset.

3.4 Experimental Results

3.4.1 Experimental Setups

We experimented with multiple setups to establish the baselines for the new dataset. We adopted Google's Tensorflow implementations of BERT, ALBERT, and Adapter-BERT for MRPC format paraphrase detection experiments. For BERT and Adapter-BERT, the encoded representations for Telugu were obtained from the pre-trained Multilingual-BERT (mBERT), meanwhile, for ALBERT setup, encoded representations for Telugu were obtained from Indic BERT, a pre-trained multilingual ALBERT trained in 12 Indic languages covering Telugu.

We finetuned and tested the above models with the dataset ratios: train (80%), dev (10%), and test (10%). We ran the experiments three times, randomizing the data and computing the average values of metrics. Accuracy, Precision, and Recall are used as evaluation metrics. Results for the classification tasks are shown in table 3.8. High Precision scores indicate that the quality of a synthetic dataset can be improved by utilizing this manual paraphrasing resource.

3.4.2 Improving the Quality of Synthetic data using Manually created resources

Compared to synthetic data creation, manually created resources assures quality but can be costly and lacks quantity. So, in order to improve the quality while preserving the quantity of the paraphrase dataset, we refined the synthetical dataset by using manually handcrafted resources.

We experiment with the paraphrase detection task by considering another synthetically generated Telugu paraphrasing dataset. We finetune the IndicBERT model with our manually annotated paraphrasing resources and test it on IndicParaphrase Telugu samples. As IndicParaphrase Telugu samples are not manually annotated, we randomly extracted 100 samples from the IndicParaphrase and manually annotated them for testing in the paraphrase detection task. Out of 100 samples, 58 are identified as paraphrases, while the remaining 42 are non-paraphrases. Due to this distribution, the necessity of manual annotation for such a synthetically created dataset, especially for Telugu, is of paramount importance. So, this effort also contributes to evaluating the quality of synthetically generated paraphrase datasets for low-resource languages.

For the Paraphrase detection task, where each pair is classified as either paraphrase or not, We finetuned the ALBERT model with the manually annotated paraphrase dataset and tested it with 100 manually annotated Indic Paraphrase samples. Results report 0.75 Accuracy, i.e, around 15 % improvement in the quality. This result further validates the application of this dataset in improving the quality of synthetically generated datasets. The performance of detection can be improved further by increasing the manual paraphrasing resources.

3.5 Conclusion

As discussed before, the lack of resources for any research topic has been an obstacle to the advancement of the respective topic. Similarly, research in Telugu paraphrasing has not recorded any notable progress until recent efforts on data resources were made. We took a step forward to create manual resources for Telugu paraphrasing. We discussed the methodologies for creating the paraphrase resources for low-resource languages. And then, we presented the manually created and annotated paraphrase corpora for Telugu. We have also discussed the necessity for manual intervention in the creation of such resources and demonstrated the same through observations from paraphrase detection experiments.

Chapter 4

TeQuAD: Telugu Question Answering Dataset

4.1 Introduction

State-of-the-art models and high-quality datasets have improved many Natural Language Processing fields, especially Machine Reading Comprehension tasks have advanced with the availability of resources like SQuAD (Stanford Question Answering Dataset). But, quality MRC datasets are still not a reality for several low-resource languages like Telugu. Our approach is to rely on existing quality datasets from high-resource languages to record decent MRC performance in low-resource languages. We adopted the SQuAD dataset and translated it to obtain QA resources in the target language i.e, Telugu. After translation, the position of the answers in the target paragraphs will change because of language divergences. So, for answer span extraction, applying simple matching methods like identifying translated answer phrases in the translated target paragraphs will be ineffective. In the QA data creation process that involves translation, the span extraction process is pivotal. We explored and discussed different heuristic matching techniques for the span extraction process. Applying such techniques, we created a parallel Telugu-English QA dataset consisting of 82k triples. The purpose of creating a parallel QA dataset is to exploit the advantage of Cross-lingual reasoning. Further, we propose a supervised approach for efficient span extraction in target languages. We also discuss the cases where the supervised approach performs more effectively than matching techniques because of its ability to understand semantic information using pre-trained language models.

In this chapter, we discuss the different methodologies followed to create a Question Answering dataset for low-resource languages. Then we present a Telugu Question Answering Dataset - TeQuAD with the size of 82k parallel triples created by translating triples from the SQuAD. Then, we discuss the performance of our models, which outperform baseline models on Monolingual and Cross-Lingual Machine Reading Comprehension setups.

4.2 Corpus Creation

4.2.1 Data Source

We considered SQuAD as the data source to create the QA dataset in the Telugu language. Because of its high quality and quantity, and adaptability to recent implementations of deep learning models, SQuAD has been chosen as the source to create QA datasets in different languages [6], [1], [3]. SQuAD was created through crowdsourcing. Crowd workers posed questions on paragraphs extracted from English Wikipedia articles. 10000+ triples were constructed, each triple consisting of a paragraph, question, and span indices of answer. Span indices of answer indicate the position of the answer phrase in the paragraph.

4.2.2 Data Creation

To create a data resource in an NLP task for any low-resource language, translating an available quality dataset is an effortless and cost-efficient approach. We translated the SQuAD dataset from English to Telugu using Google translator [1]. After translation, we obtained translated paragraphs and translated questions in the target language. We also extracted the English answer phrases from the English paragraphs using the span indices of the answers. Then translated these English answer phrases into Telugu to obtain the translated triples: translated paragraphs, translated questions, and translated answer phrases. To create a span extractive QA dataset similar to SQuAD, we also need span indices of the translated answers. Due to differences in language characteristics, the individual answer phrase and the answer phrase section present in the paragraph are translated differently. So, employing simple, straightforward matching techniques might not yield appropriate span indices in most cases. We discuss multiple approaches for span extraction of translated Telugu answer phrases. The intention of employing multiple techniques is to obtain as much synthetic data as possible for Telugu MRC.

4.2.2.1 Matching

Matching algorithms such as fuzzy search and cosine similarity are used. A threshold value of 0.7 has been assigned. A window of a certain size will move or slide across the translated paragraph words from beginning to end. At each iteration, the word or phrase present inside the window will be matched with translated Telugu answer phrase, and the matching score will be computed. If the matching score exceeds the threshold value, then the phrase inside the window will be considered the answer phrase, else ignored, and the window will move forward. In a few instances, multiple instances of answer phrases might exist in paragraphs. To handle such cases, we recorded the index of the English answer phrase among its repetitions present in the English paragraph. Then we picked the same instance of translated Telugu answer phrase

[1] https://translate.google.co.in/

English text	Translated Telugu text (ISO 15919)
China Unicom 's service in Wenchuan and four nearby counties was cut off , with more than 700 | **towers** | suspended.	Veñcuvān mariyu samīpanlōni nālugu kauṇṭīlalō cainā yunikām sēva nilipivēyabaḍindi, 700ki paigā | **ṭavarlu** | nilipivēyabaḍḍāyi.

Table 4.1: Marking answers using special symbol for answer span extraction.

among its repetitions in the translated Telugu paragraph. For example, let's say the answer phrase 'polar' is repeated four times in the English paragraph. And the third repeated instance is the actual answer phrase according to the question. Then for the corresponding target triple, we will also consider the third repeated instance of the translated Telugu answer phrase ('polar' in Telugu) from the Telugu paragraph as the actual Telugu answer phrase.

4.2.2.2 Explicit Position Indicator

If phrases with matching scores greater than the threshold value are not found, span extraction for such samples is skipped when matching techniques are applied. These skipped samples are considered to extract the answer span using the explicit position indicator technique. Using span indices of the answers, English answer phrases in the English paragraphs are identified and marked with a special symbol ('|'). After translating English paragraphs to Telugu, everything gets translated, but the symbol remains unchanged. Locating the symbol, we can identify the translated answer phrases in translated Telugu paragraphs. See table 4.1 for example.

Using above-discussed methods, we obtained 82,605 English-Telugu parallel triples, creating a Telugu Question Answering dataset - TeQuAD. See table 4.2 for a parallel English-Telugu instance in TeQuAD. And for evaluation and experiments, we have created two different test datasets. We later use these test datasets to analyze the performance of Telugu MRC models and present the results.

- **Translated & Corrected dataset**

  2000 English triples from the dev set of SQuAD1.1 are translated to Telugu and corrected manually. A set of guidelines [URL to github file] is prepared to correct the translated Telugu context, questions, and answers.

- **Wiki dataset**

  Similar to SQuAD, we created this data from Wikipedia articles. Randomly selected Wikipedia articles are split into paragraphs. From 125 Telugu Wikipedia paragraphs, 947 QA pairs are created manually by framing questions with answer types such as Person, Location, Date/Time, Quantities, Clauses, Verb phrases, Adjective phrases, and others.

	English	Telugu (ISO 15919)
Context	China Mobile had more than "**2,300**" base stations suspended due to power disruption or severe telecommunication "**traffic congestion**". Half of the wireless communications were lost in the Sichuan province . China Unicom 's service in Wenchuan and four nearby counties was cut off , with more than 700 towers suspended.	Vidyuttu antarāya lēdā tīvramaina ṭelikamyūnikēṣan "**ṭrāphik raddī**" kāraṇaṅgā cainā mobail "**2,300**" ki paigā bēs'sṭēṣanlanu nilipivēsindi. Sicuvān prānslō saga vairles kamyūnikēṣanlu pōyāyi. Veñcuvān mariyu samīpanlōni nālugu kauṇṭīlalō cainā yunikām sēva nilipivēyabaḍindi, 700ki paigā ṭavarlu nilipivēyabaḍḍāyi.
Question 1	Besides power disruption , what caused telecommunications to be suspended ?	Vidyuttu antarāyantō pāṭu, ṭelikamyūnikēṣanlanu nilipivēyaḍāniki kāraṇamēmiṭi?
Span	16 - 17	5 - 6
Answer	traffic congestion	ṭrāphik raddī
Question 2	How many base stations are suspended?	Enni bēs sṭēṣanlu saspeṇḍ cēyabaḍḍāyi?
Span	5 - 5	10 - 10
Answer	2,300	2,300

Table 4.2: Representation of QA pairs in parallel corpora

A minimum of five and maximum ten questions were created for each paragraph/context. To make it challenging, for fair evaluation, multiple types of queries were posed while creating the dataset.

4.2.2.3 Span Extractor

While creating the reading comprehension dataset for Telugu, TeQuAD, we used discussed techniques to retrieve span indices for the translated Telugu answers. These techniques might fail in cases where

- The translated answer might not exist in the translated paragraph. After the translation, there is a possibility that information about the answer phrase might have been lost or transformed into a different word form. See 4.1 for example.
- Multiple instances of the translated answer might be in the translated paragraph. See 4.2 for example
- A random phrase in the Telugu paragraph returns a greater matching score than the actual answer phrase when compared with translated Telugu answer phrase. See 4.3 for example

English Context:
. . . Trade liberalization may shift economic inequality from a global to a domestic scale . . .
English Question:
What scale does trade liberalization shift economic inequality from ?
English Answer:
Global
Translated Telugu Context:
. . . వాణిజ్య సరళీకరణ ఆర్థిక అసమానతలను ప్రపంచ స్థాయి నుంచి దేశీయ స్థాయికి మార్చవచ్చు . . .
(. . . Vāṇijya saraḷīkaraṇa ārthika asamānatalanu prapan̄ca sthāyi nun̄ci dēśīya sthāyiki mārcavaccu . . .)
Translated Telugu Question:
వాణిజ్య సరళీకరణ ఏ స్థాయి నుండి ఆర్థిక అసమానతను మారుస్తుంది ? (Vāṇijya saraḷīkaraṇa ē sthāyi nuṇḍi ārthika asamānatanu mārustundi)
Plausible Telugu Answer: ప్రపంచ స్థాయి (prapan̄ca sthāyi)
Translated Telugu Answer: గ్లోబల్ (Glōbal)

Figure 4.1: Example for the absence of translated Answer in the translated Context. Both *'prapan̄ca sthāyi'* and *'glōbal'* share the similar meaning.

Translated Telugu Context:
. . . పూర్తిగా పెట్టుబడిదారీ ఉత్పత్తి పద్ధతిలో కార్మికుల వేతనాలు ఈ సంస్థలు లేదా యజమాని ద్వారా నియంత్రించబడవు, కానీ మార్కెట్ ద్వారా, వేతనాలు ఏ ఇతర మంచి కోసం ధరల మాదిరిగానే పనిచేస్తాయి. అందువలన, వేతనాలు నైపుణ్యం యొక్క మార్కెట్ ధర యొక్క విధిగా పరిగణించబడతాయి . . .
(. . . Pūrtigā peṭṭubaḍidārī utpatti pad'dhatilō kārmikula vētanālu ī sansthalu lēdā yajamāni dvārā niyantrin̄cabaḍavu, kānī mārket dvārā, vētanālu ē itara man̄ci kōsaṁ dharala mādirigānē panicēstāyi. Anduvalana, vētanālu naipuṇyaṁ yokka mārkeṭ dhara yokka vidhigā parigaṇin̄cabaḍatāyi . . .)
Translated Telugu Question:
పూర్తిగా పెట్టుబడిదారీ ఉత్పత్తి పద్ధతిలో వేతనాలను ఏది నియంత్రిస్తుంది ? (Pūrtigā peṭṭubaḍidārī utpatti vidhānanlō vētanālanu ēdi niyantristundi ?)
Translated Telugu Answer:
మార్కెట్ (mārkeṭ)

Figure 4.2: Example for multiple instances of Answer in the Context

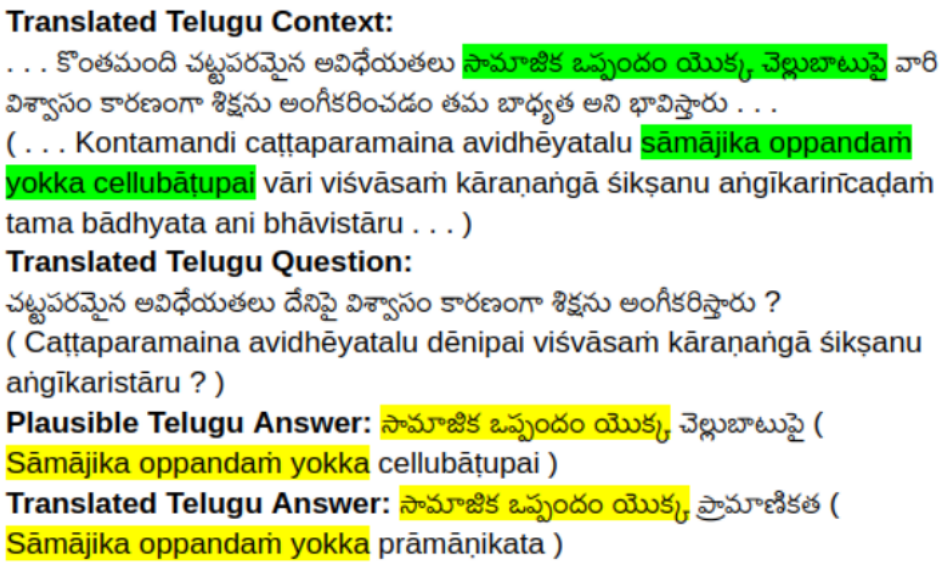

Figure 4.3: Example for partial matching answer scenario.

To handle the span extraction of answers in such cases, we introduced a supervised span extraction approach. We employed the Dual BERT method proposed in [12] for Chinese MRC. Along with parallel QA pairs, we additionally added answer phrases as inputs to the model, and the span indices of the answers are predicted as output. See 4.4. Dual BERT considers the contextual semantic information from both Telugu-English parallel triples and can identify the answer phrases even if they exist in different forms in the translated Telugu paragraphs. As this model relies on contextual information to identify the answer phrase, it is able to find the correct instance of the answer phrase, even if there are multiple instances of the answer phrase present in the Telugu paragraph. Along with the translated Telugu answer, information about English answers will help predict span indices of the exact Telugu answer phrases, avoiding the retrieval of partial answer phrases.

In order to evaluate this supervised span extraction method, we need data to train the model. We considered 82k parallel triples from TeQuAD for training and validation. Telugu answers were extracted using the span indices obtained through heuristic techniques. Although some of these answers were partial and fall into the cases discussed above, providing information about Telugu answers in combination with corresponding English answers will help the model to learn and predict answers accurately. For testing, we considered the Translated & Corrected test dataset, where we have the manually created span indices of answers.

The experimental setup for this supervised method is similar to the Cross-lingual MRC experimentation, which will be discussed in later sections. Results from experiments show the performance of 88% F1 Score and 73% EM Score. Besides apparent advantages, such supervised methods need sufficient resources to learn and perform well.

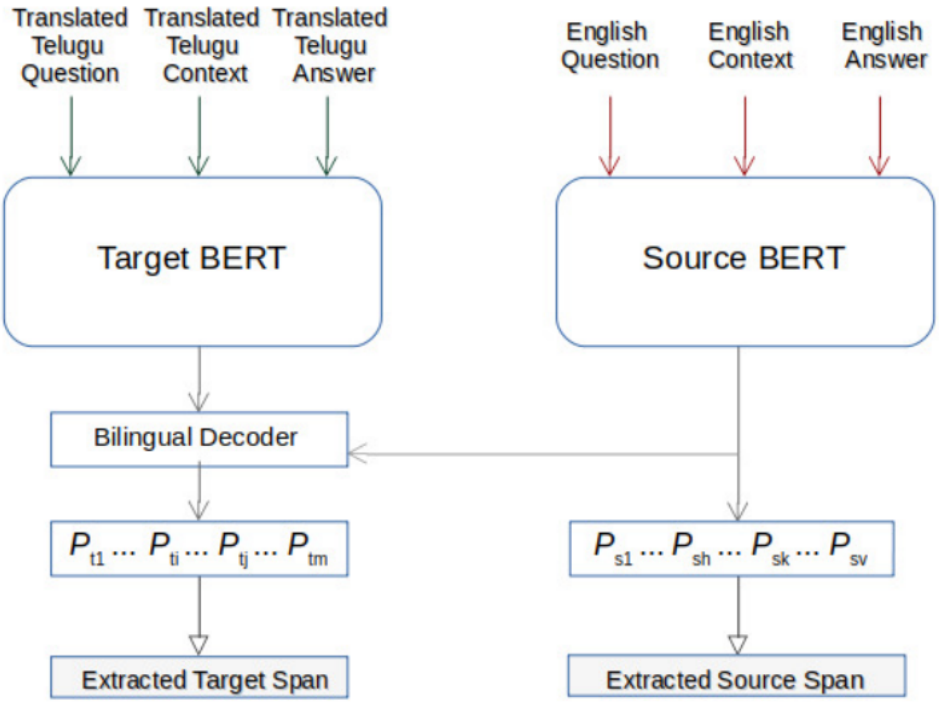

Figure 4.4: Architecture of Span Extractor

4.3 Experiments

We experimented with TeQuAD in monolingual and cross-lingual setups. The pre-trained Multilingual-BERT (mBERT) trained in 104 languages, including Telugu and English, is employed for obtaining encoded representations for both languages. We use NLTK tokenizer followed by the BERT Word Piece tokenizer to sub-tokenize the tokens in all the experiments. Experimented with a batch size of 64 and sequence length of 512. As in Google's Tensorflow implementation of BERT, ADAM with weight decay optimizer is considered with different learning rates for different experimental setups. Our models have been trained on Google Cloud TPU v2.

4.3.1 Monolingual setup

In the monolingual setup, 82k Telugu triples from TeQuAD are considered for fine-tuning the mBERT model for the MRC task. We used Google's Tensorflow implementation of BERT for running SQuAD tasks and trained it for 3 epochs with a learning rate of 1e-4.

Model	Test Dataset	Monolingual		Crosslingual	
		F1	EM	F1	EM
mBERT(Zero shot)	Translated & Corrected	28.4	0.0	27.1	0.01
	Wiki QA	27.1	0.0	27.6	0.0
	TyDi dev QA	21.0	0.0	21.3	0.0
mBERT(TeQuAD)	Translated & Corrected	69.4	43.7	69.4	43.5
	Wiki QA	**83.0**	**61.0**	**83.3**	**61.9**
	TyDi dev QA	61.0	41.6	69.1	43.3

Table 4.3: Experimental results of MRC on Test Datasets. Performance (in terms of %) F1 : F1 Score and EM: Exact Match Score

4.3.2 Cross-lingual setup

The dual BERT approach proposed in [11] is used for the CLMRC setup. In this approach, deep contextualized representations of the inputs from both languages are considered, and 'Bilingual Context' is computed, which will be used to exploit the semantic relations among the English and Telugu QA pairs. Parallel QA pairs of English and Telugu are passed as inputs to the model, and span indices of the Telugu answer phrases are predicted. 82k Parallel Telugu-English triples from TeQuAD are considered for fine-tuning the pre-trained mBERT model. We used the implementation in and trained it for 3 epochs with a learning rate of 2e-5. Both the Cross-Lingual and Monolingual fine-tuned models are evaluated on three test datasets. Along with Translated & Corrected (2000) and Wiki (947) test datasets, Telugu samples of the Gold Passage task (Span Extractive QA task) from the TyDiQA dev (667) dataset are considered for evaluation.

F1 score and EM score are used as evaluation metrics. Results of the evaluation for monolingual and cross-lingual setups are shown in Table 4.3.

4.4 Results and Observations

From the results attained, the key observation is that compared to the zero-shot mBERT model, the models finetuned on TeQuAD performed better for the Telugu MRC task. On average, a 40% increase in F1 and EM Scores was registered across all setups.

From the experiments, we observed that the performance of the finetuned model on the Wiki test dataset is much better than on other test datasets. Notice that the Wiki dataset is created from original Telugu Wikipedia articles, followed by the manual effort to produce QA pairs, hence has high quality than the Translated & Corrected dataset. On the other hand, the TiDyQA Telugu samples are of low quality and not preferable for evaluating Telugu MRC. Most of the QA pairs in TyDiQA revolve around the numbers such as zip codes, dates of birth/death, area of land, etc. MRC model exposed with such data resources will overfit to learn and answer

just these types of questions and lack the ability to comprehend other QA types. So, the model trained on the TyDiQA train dataset produced good results when evaluated on the TyDiQA dev dataset. However, compared to the TeQuAD model, the performance of the TyDiQA-trained model fell behind when evaluated on Translated Corrected and Wiki test datasets.

Why low EM scores

In the MRC experiments, though TeQuAD registered decent F1 scores, Exact Match scores are observed to be noticeably lower than F1 scores. Approximately 20% gap can be seen between these two metrics across all the setups. Partial answer predictions will affect the Exact Match score, and we tried to analyze the causes for such error predictions. The primary reason for the low EM scores is the multiple possible answers for a query. The existence of different answer phrases in the paragraph, all of which seem to be correct, will affect the ability of the MRC model to predict the exact answer.

Another obvious reason for faulty answer predictions is the low-to-moderate resources available for the language. Pre-trained models exposed to such fewer data resources might not be able to reason the context leading to false answer predictions. And even though such models leverage the information from high resource language(s), due to the linguistic divergences between the languages (here Telugu and English), answer boundary detection capability in the low resource language is poorer, failing to identify the complete answer phrase in the context.

In [41], they discussed the deficient answer boundary detection capability of MRC models for low-resource languages. Their work suggested improving the detection capability by training the MRC model on phrases in low-resource language mined from the internet. We experimented by mining approximately 32k Telugu phrases from Wikipedia and trained the model with the phrase masking prediction task. Results don't show any noticeable improvement in the EM scores.

On the other hand, several MRC works employ character-level span indices to point to the answer phrase specifically. This might lead to worse EM scores in Telugu, considering the rich morphology of the language. So, instead, we stuck to word-level span indices for the answer phrases.

Why Cross lingual experimentation?

As discussed, [12] proposed the Dual BERT approach to improve the MRC for low-resource languages by utilizing cross-lingual knowledge. With experiments, we observed that CLMRC setup helps in boosting the performance of the model when the size of the corpora is low (See 4.4). But with the creation of large synthetic data, the effect of the CLMRC setup is negligible. In table 4.3, results obtained by training the model on 82k data in the mono-lingual setup are identical to the results of the CLMRC setup. The creation of such resources helps the machine to learn from the target language itself instead of relying on High resource languages.

Experimental Setup	**Translated & Corrected**		**Wiki QA**	
	F1	EM	F1	EM
Mono-lingual	65.9	39.1	79.3	50.5
Cross-lingual	67.4	39.7	82.2	54.0

Table 4.4: Results of the Experimental setups trained on less corpora : 34k QA pairs. Performance (in terms of %) F1: F1 Score and EM: Exact Match Score

Test Dataset	**TyDiQA**		**TeQuAD**	
	F1	EM	F1	EM
Translated-&-Corrected	57.7	29.5	69.4	43.7
Wiki QA	77.3	48.4	83.0	61.0

Table 4.5: Comparison b/w TeQuAD and TyDi QA for Telugu MRC. Performance (in terms of %) F1: F1 Score and EM: Exact Match Score

Comparison with TyDiQA

Clark et al. [9] demonstrated the performance of the Gold-Passage MRC task in Telugu, which is similar to the SQuAD style Question Answering task. They trained the model on approximately 49k multilingual QA pairs and evaluated it on the Telugu test dataset. For comparison, we also considered 49k multilingual QA pairs from the TyDiQA dataset and finetuned the mBERT model for the MRC task. Then we evaluated this model on discussed Telugu test datasets - The wiki test dataset and Translated & Corrected test dataset. See 4.5 for a comparison of Telugu MRC performance between TyDiQA and TeQuAD models. The MRC model finetuned on TeQuAD was observed to be outperforming the TiDyQA model in the Telugu MRC task.

4.5 Conclusion

This work has advanced the creation of a Question Answering (QA) dataset for Indian languages by utilizing translation in the creation of a Machine Reading Comprehension (MRC) corpus. Adequate resources are crucial for achieving good results in NLP tasks for low-resource languages. However, creating these resources is challenging. To overcome this, resources from high-resource languages such as English may be utilized to generate ample data for various NLP tasks in low-resource languages like Telugu. We present methods for generating and refining datasets, which aim to enhance both the quantity and quality of the data. Employing these methods, we created a Telugu-English Question Answering dataset. We also evaluated the performance of the Telugu MRC model on different Telugu QA resources and presented the observations.

Chapter *5*

Spell Correction

5.1 Introduction

Spell Correction has been receiving attention in research for its NLP applications. Spell Correction is the task of correcting spelling errors in sentences. Spelling errors are mistakes in the text written incorrectly corresponding to a natural language. English and other high-resource languages have seen significant progress in Spelling correction problems. The necessity of spelling correction and the availability of resources has brought interest in this research topic for a few resource-scarce languages too. But only a few works discuss the Spelling Correction tasks for Indic languages.

For a morphologically rich language like Telugu, spelling errors are recurrent and effectual in the outcome of a word/sentence. Hence, Spelling Correction for Telugu is of paramount importance. Recent advancements and the availability of language models paved the path for research in languages with no or small set of linguistic resources available. Utilizing such language models, we approached the spelling correction task for Telugu.

As discussed earlier, the progress of an NLP task for a language goes hand in hand with the data resources available for the respective language. Due to the unavailability of such resources in Spelling correction for Telugu, we created a synthetic dataset using rule-based methods. The dataset is created in such a way that most of the introduced spelling errors resemble real-world spelling mistakes, while few of them are just randomly generated. Discussed approaches can be employed for other languages to create necessary spelling correction resources. Unlike existing word-based spell correctors for Telugu, we worked on context-based spell correctors. We will discuss the advantages of taking context into account when correcting spelling mistakes by using a few cases as examples. We also employed character-level features for this task and discussed the reasons for doing so in the next sections. Multiple Baseline methods were opted to experiment on the Spelling Correction task. Comparisons between the models with word level, sub-token level, and character level features were made to verify the effect of character-level features. Performance results and observations are presented in the next sections.

No. of sentences	2.5 M
Avg. no. of words per sentence	16
Avg. no. of chars per sentence	120
Min no. of words per sentence	3
Max no. of words per sentence	100

Table 5.1: Statistics of synthetically created spelling error data.

5.2 Data Creation

5.2.1 Data Source

In this section, we will discuss the detailed process followed in the creation of the Spelling Correction dataset for Telugu. As mentioned earlier, data from the news domains are of considerable size and publicly available, hence are the common source of dataset creation in NLP for several languages. We collected Telugu news articles from different online websites and preprocessed the raw data. We employed our own customized sentence tokenizer to extract sentences from news articles. In total, 2517608 Telugu sentences are collected for Spelling Correction data creation.

5.2.2 Protocol for data creation

In order to create the Spelling Correction dataset, we applied four rule-based techniques: Insertion, Deletion, Substitution, and Transposition. These techniques will introduce noise into the Telugu sentences, creating different spelling errors. Spelling errors are created in such a way that they resemble real-world errors or common errors. And also, to incorporate a few unusual cases, random errors are also introduced in minor proportions.

In the Insertion and deletion technique, we will insert/delete a diacritic or a consonant, or both in the sentence. Different cases of insertion and deletion are addressed in table 5.2 and table 5.4, respectively, with examples. Most of the substitution errors are introduced based on two types of common errors: phonetic and typo mistakes. And other few, around 10%, are randomly replaced to introduce errors. Different cases of substitution are addressed in table 5.3 with examples.

In Transposition, we will create noise by shuffling a consonant/diacritic with a consonant/diacritic in the sentence. See the below example of a transposition error in the sentence.

తామర -> తారమ (taamara) –> (taarama)

తారమ పువ్వు నీటిలో ఉంటుంది. (taarama puvvu neetiloe untundi)

Above mentioned techniques are applied combinedly into the sentences to introduce spelling errors. Errors are introduced into sentences in different frequencies.

- **Plain data -** Original data. No errors are introduced.

Insertion of a diacritic	పనస --> పనాస , పానస (panasa) –> (panaasa) , (paanasa)
	అరటి --> అరెటి (arati) –> (areti)
Insertion of a consonant	అలవాటు --> అలపవాటు (alavaatu) –> (alapavaatu)
Insertion of '్' (virama: mute diacritic)	కాలము --> కాల్ము (కాల + '్' + ము) (kaalamu) –> (kaalmu)
Insertion of [consonant + '్' (virama: mute diacritic)]	అరవింద --> అప్రవింద (అ + ప + ్ + రవింద) (aravinda) –> (apravinda)
Insertion of consonant between a consonant and diacritic	ఇంకొకరు --> ఇంకనొకరు (added న) (inkokaru) –> (inkanokaru)
Insertion of duplicate consonant	సరదా --> సరరదా (saradaa) –> (sararadaa)

Table 5.2: Different cases of insertion are addressed with examples.

Typo mistake	substitution of a consonant	కాపాడాడు -->> కాపాదాడు (kaapaaDaaDu) –> (kaapaadaaDu) ప --> ఫ (pa) –> (pha)
	substitution of a sign	కొండ --> కోండ (konda) –> (koenda)
Phonetic mistake	substitution of a consonant	ధమరుకం --> డమరుకం (Dhamarukam) –> (Damarukam)

Table 5.3: Different cases of substitution are addressed with examples.

consonant deletion	విజయదశమి --> విజదశమి (vijayadaSami) (vijadaSami)
sign deletion	\ కిరాణ --> కరాణ (kiraaNa) (karaaNa)
	\ రావణుడు --> రవణుడు (raavaNuDu) (ravaNuDu)
duplicate deletion	చెప్పు -->ు చెపు (cheppu) (chepu)

Table 5.4: Different cases of deletion are addressed with examples.

Erroneous sentence:

తారమ పువ్వులు సువాసన కలెరి అందంగా ఉంటాయి కాబట్టి వాటినిని పూలలో ఉపయోగిస్తారు.

Original sentence:

తామర పువ్వులు సువాసన కలిగి అందంగా ఉంటాయి కాబట్టి వాటిని పూజలలో ఉపయోగిస్తారు.

Misspelled words indices:
0 (transposition), 3 (substitution), 7 (insertion), 8 (deletion)

Figure 5.1: Example for triple error sentence.

- **Single error data -** One error per sentence.
- **Double error data -** Two errors per sentence.
- **Triple error data -** Three errors per sentence.
- **Quadruple error data -** Four errors per sentence.
- **Five error data -** Five errors per sentence.

The types of spelling errors introduced into the sentences are randomly chosen. See figure 5.1 for a triple error sentence example.

5.3 Experimental Setups

In this section, we present experimental setups with multiple baseline models for the Spelling correction task. The synthetic dataset is divided into train, dev, and test partitions randomly with dataset ratios of 80%,10% and 10% respectively.

5.3.0.1 RNN

A general RNN encoder-decoder architecture is employed for NLP generation tasks like spelling correction, summarization, and paraphrase generation. We have implemented this model using sequence-to-sequence Recurrent Neural Networks (RNNs) [8] with an attention mechanism. Word embeddings obtained from Telugu fastText pretrained model are passed as input features to the encoder-decoder model. We trained the model with a learning rate of 0.001, 512 as the hidden size of LSTM cells, batch size of 32, and maximum sequence length of 128 for 5 epochs.

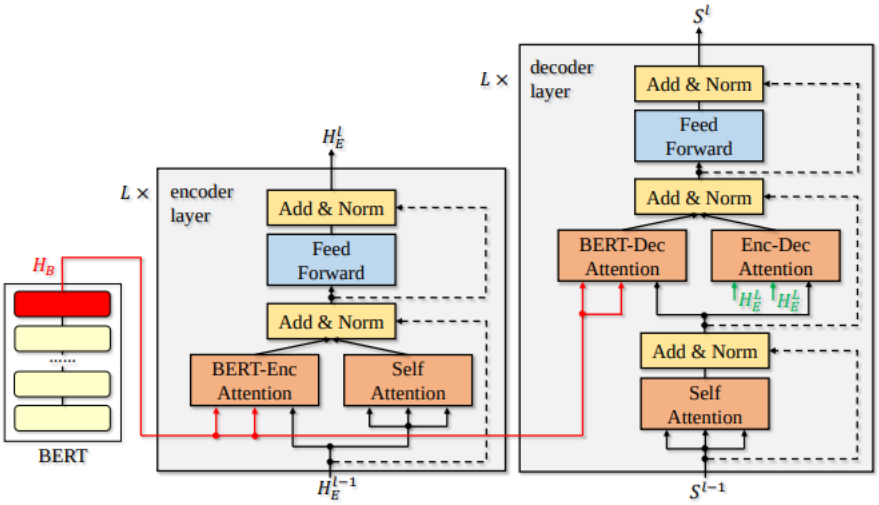

Figure 5.2: Overview of BERT-fuse model.

5.3.0.2 BERT-Fuse

Zhu et al. [42] proposed a technique where the pre-trained BERT model is fused with transformer encoder-decoder architecture. As shown in 5.2, input sentences are passed into the pre-trained BERT model and representations from the last layer of the pre-trained model are considered as input features for the transformer model. These representations will interact with the encoder and decoder by using an attention mechanism. Kaneko et al. [21] discussed the potential of this technique on English Grammar Error Correction tasks, while Wang et al. [40] further explored this technique on Chinese Grammar correction tasks. We have used mBERT pre-trained model to obtain features for Telugu sentences and then fed it to the six-layer transformer model with embedding size 512 and FFN layer dimension 1024. We utilized the pytorch implementation available and trained the model with a learning rate of 0.0005 and batch size of 8 for 45 epochs.

5.3.0.3 Convolutional Sequence-to-Sequence

Gehring et al. [19] proposed architecture for Convolutional sequence-to-sequence modeling for NLG tasks. In order to compute the intermediate encoder and decoder states, Convolutional Neural Networks are used instead of Recurrent Neural Networks. Unlike above-mentioned models, in this setup, we deal with character level features. We preprocess the data and build the vocabulary of Telugu characters. As we have moderate Telugu resources to train a transformer model, we experimented with a low-resource configuration setup comprised of four encoder layers and three decoder layers, both with embedding size 256. We utilized the

Model	Accuracy (EM score)
RNN	0.67
BERT Fuse	0.71
Convolutional Seq2Seq	0.85

Table 5.5: Experimental results of Spell correction task.

fairseq-pytorch implementation and trained the model with a learning rate of 0.25 and batch size of 64 for 50 epochs.

5.4 Results and Observations

Exact Match Score is the evaluation metric for the experiments. We compared the predicted sentence text with the actual sentence text and marked accuracy as '1' only if both are equal, else '0'. The reason to opt for EM Score is that along with correcting the erroneous characters, model should also generate the correct characters as they are without any modifications.

Table 5.5 shows the results from the discussed experiments. Convolutional architecture outperformed the other baseline models for Spelling correction task. Besides possessing the architectural advantage, Convolutional model does not utilize any pretrained features and is trained from scratch with noisy spelling data. On the other hand, RNN is utilizing features obtained from a pretrained model which is not exposed to such noisy data, hence incapable to perform well in spell correction tasks. Similarly BERT-Fuse uses features from mBERT, a pretrained model which also not exposed to noisy data relatively.

Besides, input sentences were word tokenized and features for tokens were extracted and passed to the RNN model. For BERT-Fuse model, features for subtokens of input sentences were used. Features of these level observed to be less effective for the spelling correction task. So, using character level features also improved the performance of Convolutional Sequence model in Spelling Correction task.

Furthermore, we analysed the performance of Convolutional Sequence model on different erroneous sentence cases separately. EM Scores when tested on error less data and single error data turned out to be great, while on double and triple error data, they are decent. And EM Scores dips down further when tested on sentences with more than three errors. This is because it is difficult to predict the words when some good extent of information is missing/modified in the input sentence.

In a few cases, multiple errors were introduced into the same words in the sentences by adding or modifying, or deleting most of the information in those words. But even in such cases, models were able to detect and generate the actual words based on the surrounding context provided in the sentences.

5.5 Conclusion

In this chapter, we discussed the Spelling Correction task for the Telugu language. We made progress towards developing effortless tools for correcting spelling in Telugu. Using rule based-techniques, we introduced character-level noise into Telugu data and created a synthetic spell correction dataset. Then we experimented with different NLG models and analyzed their performance on the Spelling Correction task. We verified that the presence of context improves the spelling correction task significantly.

Chapter *6*

Conclusion

6.1 Summary

In this thesis, we have addressed the challenges involved in creating resources and systems for low-resource languages such as Telugu. The main contributions of the thesis are focused on resource creation and system development for paraphrasing, question answering, and spell correction tasks.

For paraphrasing, the processes for creating paraphrase resources for low-resource languages were showcased. Subsequently, we presented the manually created and annotated paraphrase corpora for Telugu, namely Manually created paraphrase(MCP) and Manually annotated paraphrase (MAP) datasets, respectively. MAP corpus consists of 10k+ samples, while the MCP corpus consists of 1544 paraphrases The importance of manual involvement in the production of such resources was highlighted and demonstrated through results from paraphrase detection experiments.

In Question answering (QA), many works focus on using English resources to create resources for low-resource languages by translating them. The main challenge with this approach is to ensure the accuracy of the translated answers and their span positions. We presented a few span correction/extraction techniques to improve both the quality and quantity of these translated QA datasets, specifically for SQuAD-style datasets. We translated English triples from SQuAD to Telugu and then applied the span correction techniques. In this manner, we developed a Telugu Question Answering Dataset - TeQuAD, consisting of 82,000 parallel Telugu-English triples. We then evaluated the performance of the Telugu MRC model on various Telugu QA resources and presented our findings.

We used rule-based techniques to add character-level errors to Telugu sentence data and created a synthetic dataset for spell correction. We experimented with different natural language generation models for their effectiveness in correcting spelling errors and found that context has a significant impact on the accuracy of the correction task.

6.2 Future Work

In the future,

- we would like to increase the manually created paraphrase resources for Telugu and integrate these resources into various downstream NLP tasks, improving these tasks' performance for Telugu.
- we aim to improve the Machine Reading Comprehension (MRC) task for the Telugu language by offering a set of pre-trained models that are trained on publicly available resources in Telugu, as well as creating additional QA data resources for the Telugu language.
- similar to spell correction task, grammar correction task can also be exploited and improved for Telugu with synthetically created grammatical error resources and recent state of the art language models.

Related Publications

Manikanta Sai Nuthi, Rakesh Vemula, and Manish Shrivastava. "**TeQuAD: Telugu Question Answering Dataset**". Proceedings of the nineteenth International Conference on Natural Language Processing (ICON-2022)

Bibliography

[1] N. Abadani, J. Mozafari, A. Fatemi, M. A. Nematbakhsh, and A. Kazemi. Parsquad: Machine translated squad dataset for persian question answering. In *2021 7th International Conference on Web Research (ICWR)*, pages 163–168. IEEE, 2021.

[2] D. Aravinda Reddy, M. Anand Kumar, and K. Soman. Lstm based paraphrase identification using combined word embedding features. In *Soft computing and signal processing*, pages 385–394. Springer, 2019.

[3] M. Artetxe, S. Ruder, and D. Yogatama. On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv:1910.11856*, 2019.

[4] C. Bannard and C. Callison-Burch. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 597–604, 2005.

[5] M. Bornea, L. Pan, S. Rosenthal, R. Florian, and A. Sil. Multilingual transfer learning for qa using translation as data augmentation. *arXiv preprint arXiv:2012.05958*, 2020.

[6] C. P. Carrino, M. R. Costa-jussà, and J. A. Fonollosa. Automatic spanish translation of the squad dataset for multilingual question answering. *arXiv preprint arXiv:1912.05200*, 2019.

[7] D. Chen, J. Bolton, and C. D. Manning. A thorough examination of the cnn/daily mail reading comprehension task. *arXiv preprint arXiv:1606.02858*, 2016.

[8] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[9] J. H. Clark, E. Choi, M. Collins, D. Garrette, T. Kwiatkowski, V. Nikolaev, and J. Palomaki. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8: 454–470, 2020.

[10] M. Creutz. Open subtitles paraphrase corpus for six languages. *arXiv preprint arXiv:1809.06142*, 2018.

[11] Y. Cui, T. Liu, W. Che, L. Xiao, Z. Chen, W. Ma, S. Wang, and G. Hu. A span-extraction dataset for chinese machine reading comprehension. *arXiv preprint arXiv:1810.07366*, 2018.

[12] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu. Cross-lingual machine reading comprehension. *arXiv preprint arXiv:1909.00361*, 2019.

[13] S. Demir, I. D. El-Kahlout, E. Unal, and H. Kaya. Turkish paraphrase corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 4087–4091, 2012.

[14] M. d'Hoffschmidt, W. Belblidia, T. Brendlé, Q. Heinrich, and M. Vidal. Fquad: French question answering dataset. *arXiv preprint arXiv:2002.06071*, 2020.

[15] V. Dixit, S. Dethe, and R. K. Joshi. Design and implementation of a morphology-based spellchecker for marathi, and indian language. *ARCHIVES OF CONTROL SCIENCE*, 15 (3):301, 2005.

[16] B. Dolan and C. Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[17] P. Efimov, A. Chertok, L. Boytsov, and P. Braslavski. Sberquad–russian reading comprehension dataset: Description and analysis. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 3–15. Springer, 2020.

[18] P. Etoori, M. Chinnakotla, and R. Mamidi. Automatic spelling correction for resource-scarce languages using deep learning. In *Proceedings of ACL 2018, Student Research Workshop*, pages 146–152, 2018.

[19] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR, 2017.

[20] T.-Y. Hsu, C.-L. Liu, and H.-y. Lee. Zero-shot reading comprehension by cross-lingual transfer learning with multi-lingual language representation model. *arXiv preprint arXiv:1909.09587*, 2019.

[21] M. Kaneko, M. Mita, S. Kiyono, J. Suzuki, and K. Inui. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. *arXiv preprint arXiv:2005.00987*, 2020.

[22] J. Kanerva, F. Ginter, L.-H. Chang, I. Rastas, V. Skantsi, J. Kilpeläinen, H.-M. Kupari, J. Saarni, M. Sevón, and O. Tarkka. Finnish paraphrase corpus. *arXiv preprint arXiv:2103.13103*, 2021.

[23] B. Kaur and H. Singh. Design and implementation of hinspell—hindi spell checker using hybrid approach. *International Journal of scientific research and management*, 3(2): 2058–2062, 2015.

[24] P. KS, N. Subash, et al. Automatic error detection and correction in malayalam. *IJSTE-International Journal of Science Technology & Engineering*, 3(02), 2016.

[25] A. Kumar, H. Shrotriya, P. Sahu, R. Dabre, R. Puduppully, A. Kunchukuttan, A. Mishra, M. M. Khapra, and P. Kumar. Indicnlg suite: Multilingual datasets for diverse nlg tasks in indic languages. *arXiv preprint arXiv:2203.05437*, 2022.

[26] R. Kumar, M. Bala, and K. Sourabh. A study of spell checking techniques for indian languages. *JK Research Journal in Mathematics and Computer Sciences*, 1(1):105–113, 2018.

[27] W. Lan, S. Qiu, H. He, and W. Xu. A continuously growing dataset of sentential paraphrases. *arXiv preprint arXiv:1708.00391*, 2017.

[28] S. Lim, M. Kim, and J. Lee. Korquad1. 0: Korean qa dataset for machine reading comprehension. *arXiv preprint arXiv:1909.07005*, 2019.

[29] J. Liu, L. Shou, J. Pei, M. Gong, M. Yang, and D. Jiang. Cross-lingual machine reading comprehension with language branch knowledge distillation. *arXiv preprint arXiv:2010.14271*, 2020.

[30] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[31] G. Ramesh, S. Doddapaneni, A. Bheemaraj, M. Jobanputra, R. AK, A. Sharma, S. Sahoo, H. Diddee, D. Kakwani, N. Kumar, et al. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *Transactions of the Association for Computational Linguistics*, 10:145–162, 2022.

[32] U. M. Rao, A. P. Kulkarni, C. Mala, and K. Parameshwari. Telugu spell-checker. In *International Telugu Internet Conference Proceedings*, 2011.

[33] R. G. Reddy, M. A. Sultan, E. S. Kayi, R. Zhang, V. Castelli, and A. Sil. Answer span correction in machine reading comprehension. *arXiv preprint arXiv:2011.03435*, 2020.

[34] Y. Scherrer. Tapaco: A corpus of sentential paraphrases for 73 languages. In *Proceedings of The 12th Language Resources and Evaluation Conference*. European Language Resources Association (ELRA), 2020.

[35] R. Sennrich and M. Volk. Mt-based sentence alignment for ocr-generated parallel texts. 2010.

[36] S. Singh, P. Ramanan, V. Sinthiya, K. Soman, et al. Creating paraphrase identification corpus for indian languages: Opensource data set for paraphrase creation. In *Handbook of Research on Emerging Trends and Applications of Machine Learning*, pages 157–170. IGI Global, 2020.

[37] S. Siripragada, J. Philip, V. P. Namboodiri, and C. Jawahar. A multilingual parallel corpora collection effort for indian languages. *arXiv preprint arXiv:2007.07691*, 2020.

[38] A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, and K. Suleman. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*, 2016.

[39] J. Uszkoreit, J. Ponte, A. Popat, and M. Dubiner. Large scale parallel document mining for machine translation. 2010.

[40] H. Wang, M. Kurosawa, S. Katsumata, and M. Komachi. Chinese grammatical correction using bert-based pre-trained model. *arXiv preprint arXiv:2011.02093*, 2020.

[41] F. Yuan, L. Shou, X. Bai, M. Gong, Y. Liang, N. Duan, Y. Fu, and D. Jiang. Enhancing answer boundary detection for multilingual machine reading comprehension. *arXiv preprint arXiv:2004.14069*, 2020.

[42] J. Zhu, Y. Xia, L. Wu, D. He, T. Qin, W. Zhou, H. Li, and T.-Y. Liu. Incorporating bert into neural machine translation. *arXiv preprint arXiv:2002.06823*, 2020.