Data Creation Pipeline for NLP Applications

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

Master of Science
in
Computer Science and Engineering
by Research

by

Pavan Baswani
2021701035
pavan.baswani@research.iiit.ac.in

International Institute of Information Technology, Hyderabad
(Deemed to be University)
Hyderabad - 500 032, INDIA
May 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled " **Data Creation Pipeline for NLP Applications**" by Pavan Baswani, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Dr. Manish Shrivastava

To Family and Friends

Acknowledgments

This thesis is a culmination of the efforts, guidance, and support of numerous people around me. I would like to express my deep gratitude to my thesis advisor, Dr. Manish Shrivastava, for his invaluable guidance, insightful feedback, and unwavering support throughout the process. His expertise and mentorship have been instrumental in shaping the direction and quality of this work.

I also want to extend my thanks to the faculty and staff at IIIT Hyderabad, who have created a dynamic academic environment and willingly shared their knowledge and expertise. Special thanks to Dr. Vasudev Varma, Dr. Anoop Namboodiri, Dr. Makarand Tapaswi, Dr. Charu Sharma, and Dr. Vikram Pudi for their enlightening lectures, thought-provoking assignments, and challenging quizzes. The concepts I learned from their courses have greatly enhanced my technical skills, and the projects I completed under their guidance have served as a strong foundation for my research work.

Furthermore, I am indebted to my family, my loved ones, my friends, and my lab mates for their unwavering support, encouragement, and for their belief in me. Without their love and motivation, this thesis would not have been possible.

Finally, I would like to extend special thanks to Lokesh Madasu, Gopichand Kanumolu, and Ashok Urlana for their invaluable help, not only with the lab work but in all aspects of this journey. I have thoroughly enjoyed their companionship and support.

Abstract

Keywords: Data Creation Pipeline, Indic News Scraper, Annotation tools.

In the rapidly evolving landscape of Natural Language Processing (NLP) applications, a critical need arises for a versatile data creation pipeline capable of addressing the diverse requirements of various tasks. This thesis introduces a Data Creation Pipeline that significantly enhances the efficiency of data creation for a spectrum of NLP applications, including Abstractive Summarization, Question Answering, Paraphrasing, Legal Named Entity Recognition, Headline Classification, Semantic Relatedness, and Machine Translation correction. This pipeline offers a unified and adaptable solution, streamlining the entire data creation pipeline.

The motivation for this pipeline stems from the availability and limitations of existing task-specific tools for open-source usage. While these tools excel in their designated areas, they lack the flexibility to accommodate a wide range of NLP applications. Our pipeline bridges this gap by offering a solution that ensures quality data collection.

Key contributions of this work include the development of a systematic and extensible data creation pipeline that begins with the scraping and extraction of pertinent information from news articles. This encompasses not only the article text but also metadata such as publish date, author, category, summary, highlights, headline, sub-headline, tags, images, external links, and miscellaneous details. A noteworthy feature is the pipeline's capability to derive pre-annotations from instruction-based models. This unique approach transforms the annotation task into a correction task, expediting the annotation process while contributing to the iterative improvement of instruction-based models.

The impact of this pipeline on modern data collection methods for NLP applications is profound. By offering a versatile tool that accommodates a myriad of tasks, it streamlines the entire data creation process. The iterative model training based on human instructions not only ensures the development of state-of-the-art models for specific tasks but also signifies a paradigm shift in the way instruction-based models are refined over time.

Also, language diversity is a critical aspect of NLP, and our pipeline acknowledges this by supporting a wide range of languages. This inclusivity ensures that the pipeline can be applied globally, fostering

linguistic diversity in NLP research and development. The uniqueness of the Data Creation pipeline lies in its adaptability to various NLP tasks, serving as a comprehensive solution for data creation. The iterative improvement guided by human instructions sets it apart from existing pipelines, offering a dynamic and efficient approach to developing high-performance NLP models.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the fast growing era of Natural Language Processing (NLP) domain, the landscape has been reshaped by remarkable advancements, fundamentally altering how machines comprehend and generate human language. A pivotal component within this transformative sphere is the Data Collection Pipeline, serving as the keystone in the intricate process of gathering, organizing, and enhancing data. This pipeline constitutes the systematic gathering, curation, and augmentation of data, providing the foundational support for the training and development of high-performing NLP models, including Abstractive Summarization, Question Answering, Paraphrasing, Legal Named Entity Recognition, Headline Classification, Semantic Text Relatedness, Machine Translation correction, and more.

1.1 Significance of a Robust Data Creation Pipeline

In the evolving landscape of data-driven technologies, the significance of a robust pipeline cannot be overstated, as it profoundly influences the effectiveness and adaptability of models. The core of this importance lies in the pivotal role datasets play in shaping a model's capabilities. The quality and diversity of datasets used during training directly impact a model's ability to generalize and perform accurately across various domains. This significance is further highlighted by the diverse nature of tasks. A comprehensive pipeline not only streamlines the data creation process but also ensures the adaptability of models, allowing them to handle the intricacies inherent in diverse linguistic tasks. The pipeline stands as a linchpin, guiding the journey towards developing high-performing models with unparalleled efficacy and versatility.

1.2 Large-scale Data Creation Frameworks

Current widely-used frameworks, though proficient in specific tasks, often fall short in addressing the diverse landscape of NLP applications. Task-specific limitations prevail, hampering their adaptability to a wide range of applications. Furthermore, some tools encounter challenges in seamless integration with instruction-based models, impeding their potential for iterative improvement using human-annotated

data. Simultaneously, previous endeavors exploring analogous workflows might not have successfully provided a unified solution, leaving voids in the overall data creation pipeline.

1. **Label-Studio:** It stands out for its versatility in creating labeled data for machine learning models. It offers a user-friendly interface and supports various annotation types. However, its general-purpose design may result in limitations when handling the intricate requirements of diverse NLP applications. Challenges arise when integrating with instruction-based models, potentially requiring additional customization for seamless cooperation. Its adaptability to iterative improvements using human-annotated data could be further explored.

2. **Inception:** It is a task-centric framework, has demonstrated effectiveness for specific tasks. However, its task-specific nature (limited to tagging) might limit its adaptability across diverse NLP applications. The challenge lies in seamlessly incorporating instruction-based models for iterative improvements. Inception's design tailored for particular tasks might hinder its flexibility when confronted with a diverse set of NLP applications. Its adaptability to instruction-based models for iterative refinement could be a potential area for improvement.

3. **Shoonya:** Recognized for its capabilities in data annotation and creation, Shoonya facilitates various annotation types. However, its effectiveness in handling the nuanced requirements of instruction-based models for iterative refinement remains an area for potential enhancement. Shoonya, while proficient in annotation tasks, may face challenges in seamlessly integrating with instruction-based models. Its adaptability to iterative improvement using human-annotated data could be a focal point for refinement.

1.3 Motivation

The motivation driving this pipeline stems from the critical need to address challenges in NLP applications, particularly in the context of data creation and annotation. In the evolving landscape of language-related tasks, the demand for efficient, scalable, and language-agnostic pipeline is more pressing than ever.

The advent of sophisticated language models and the surge in data-centric NLP applications highlight the necessity for innovative solutions that bridge gaps in existing methodologies. Traditional data creation methods often face limitations in scalability, language diversity, and the ability to adapt to evolving linguistic patterns. As a result, there is a compelling motivation to explore and develop novel frameworks that not only overcome these challenges but also pave the way for enhanced accuracy and efficiency in language-related tasks.

Furthermore, the motivation extends to the practical application of cutting-edge technologies, such as

Large Language Models (LLMs), as pre-annotators or Evaluators/Raters. The transformative potential of these models in refining annotation and evaluation processes presents an exciting opportunity to elevate the quality of linguistic analysis in NLP applications.

In addition, the motivation for this pipeline is rooted in the democratization of tools and resources for the NLP community. By providing open-source annotation and evaluation tools, there is a concerted effort to foster collaboration, knowledge-sharing, and community-driven advancements. This motivation is driven by the belief that accessibility to robust tools accelerates progress, enabling researchers and practitioners to contribute effectively to the collective understanding and improvement of language processing.

1.4 Data Creation Pipeline

The proposed pipeline endeavors to contribute to the refinement and augmentation of language-based computational tasks through the development and integration of three essential components as follows:

1. Structure Aware Indic News Scraper
2. LLMs as pre-annotators or Evaluators/Raters
3. Annotation Tools for NLP Applications

This pipeline is not just about collecting data; it's about making life easier for the folks who annotate by giving them a head start with pre-annotations from existing instruction-based models. Imagine it as a smoother road for tasks like Abstractive Summarization, Question Answering, Paraphrasing, Legal Named Entity Recognition, Headline Classification, Semantic Text Relatedness, Machine Translation correction, and more. As we dig into the upcoming chapters, we'll discuss in detail about this pipeline, discovering its inner workings, methods, and the big impact it brings to the world of modern data collection for NLP applications.

1.4.1 Structure Aware Indic News Scraper

The **Indic News Scraper** represents a sophisticated and versatile tool meticulously crafted to address the intricate process of systematically extracting content from news articles. Its design is underpinned by two integral modules: the *URL Extractor* and the *Content Extractor*, each playing a crucial role in streamlining the data acquisition process.

The *URL Extractor* module functions as a meticulous navigator, systematically identifying and extracting URLs embedded within news web pages. This module serves as the initial step in the data collection workflow, facilitating the efficient retrieval of essential links that serve as gateways to the targeted content. On the other hand, the *Content Extractor* module operates as the core of the scraper, embodying a

structure-aware approach that eliminates the need for site-specific scraping code. This strategic design choice enables the tool to transcend the constraints of varying website structures, ensuring adaptability across various news sources.

Together, these modules contribute to the Indic News Scraper's effectiveness, enabling it to traverse the intricate web of news articles, extract relevant content systematically, and overcome the challenges posed by the dynamic nature of online news sources. In essence, the Indic News Scraper stands as a versatile, robust tool designed to meet the demands of modern data collection for NLP applications.

1.4.2 LLMs as pre-annotators or Evaluators/Raters

The strategic incorporation of **Large Language Models (LLMs)** as pre-annotators or evaluators/raters constitutes a pivotal component within our pipeline, targeting diverse Natural Language Processing (NLP) applications such as summarization, machine translation, and contract NER assessments.

Prompt engineering techniques form the crux of our approach, allowing us to harness the full potential of LLMs for extracting pre-annotations. By carefully crafting prompts tailored to specific NLP tasks, we guide the language models to generate annotations that align with the nuances and intricacies of the target application. This prompt engineering process ensures that the LLMs provide relevant and contextually rich pre-annotations, laying the groundwork for subsequent stages in the annotation or evaluation workflow. It is a versatile AI engineering technique, serves a dual purpose by fine-tuning large language models and guiding the refinement of inputs for generative AI services, resulting in the creation of text or images. In this overview, we explore several prompt engineering techniques:

1. **Zero-Shot Prompting:** This method enables models to respond effectively to prompts they haven't encountered during training. Leveraging general knowledge, zero-shot prompting enhances adaptability in tasks like language understanding and generation, proving valuable in diverse real-world applications.

2. **Few-Shot Prompting:** With this approach, models are trained to perform tasks or generate responses with very limited examples, typically fewer than five instances. Techniques like meta-learning and transfer learning enable effective generalization from minimal training data, crucial for applications requiring rapid adaptation to new tasks or domains.

3. **Chain of Thought (CoT):** CoT prompting involves structured, sequential prompts or questions to guide systematic thinking. Large Language Models (LLMs) exhibit enhanced capabilities in solving novel tasks by reasoning step-by-step [19].

4. **Fine-Grained Analysis:** This method involves detailed examination and analysis of data at a granular level. Employed for in-depth exploration and assessment, fine-grained prompting is used in research, data analysis, and various industries for extracting valuable insights.

5. **Translational Probability:** Assessing the likelihood that a given translation accurately represents the intended meaning of the source text, translational probability prompting is vital in evaluating the quality and fidelity of machine-generated translations.

6. **Majority Vote:** This decision-making approach aggregates the opinions or votes of multiple entities to make a final decision, leveraging collective wisdom to enhance decision-making accuracy or robustness.

7. **Self-Refinement:** A process of continuous improvement, self-refinement prompting involves providing prompts or questions that encourage models to reflect and self-assess, identifying areas for improvement to enhance performance.

These prompt approaches, integral in domains ranging from machine learning and artificial intelligence to cognitive psychology and decision-making processes, offer valuable insights. Understanding and applying these techniques contribute to more robust and informed solutions across a wide range of applications.

The utilization of LLMs as pre-annotators in the evaluation process involves deploying advanced language models, such as GPT-3 or similar counterparts, to assess the accuracy and quality of machine-generated text. A notable example is presented in the work of Yang Liu et.al [29], who introduced G-Eval, a summarization evaluation model constructed on the foundation of GPT-4. Impressively, G-Eval outperformed all preceding baseline models in summarization evaluation performance, as documented in their research findings. In the context of the recent WMT22 metrics shared task [11], the leading machine translation (MT) evaluation metric is identified as METRICX XXL, a robust multi-task metric fine-tuned on LLM model checkpoints. However, Kocmi et.al [21] demonstrates that, GEMBA, a GPT-based metric capable of operating with or without a reference translation, has exhibited superior performance compared to all metrics participating in the WMT22 shared task.

Further, the human corrected data can be utilized as instruction fine tuning of the LLMs on the specific task to improve the performance of pre-annotations or Evaluations. Incorporating human-annotated samples into the model training enhances the adaptability of the models, allowing them to learn from the nuanced annotations provided by human experts. This iterative loop facilitates a continuous improvement cycle, fine-tuning the models to exhibit increased accuracy, contextual understanding, and proficiency in generating pre-annotations.

The resulting LLMs, infused with the knowledge gained through prompt engineering and iterative training, stand as significant tools for generating pre-annotations in diverse NLP applications. Their capacity to understand and mimic human-like language nuances empowers the pipeline to seamlessly integrate these pre-annotations into the subsequent annotation or evaluation processes, thereby enhancing the overall efficiency and efficacy of the NLP workflow.

1.4.3 Annotation Tools for NLP Applications

The development of Dedicated Interfaces stands as the third pivotal element within our pipeline, specifically crafted to cater to a spectrum of NLP applications. These purpose-built platforms serve as user-friendly hubs for diverse tasks. Each interface undergoes meticulous design to ensure seamless integration with NLP methodologies. The goal is to streamline and simplify the intricate processes associated with various language-related annotation tasks. These interfaces serve as specialized workspaces, catering to the unique requirements of each NLP application.

For tasks like summarization, the interface is structured to facilitate the quality check with the adapted intrinsic evaluation metrics. The contract NER tagging benefits from interfaces specifically calibrated to identify and classify entities within legal texts. Machine translation evaluation interfaces enable the systematic assessment of translated content, ensuring accuracy and linguistic fidelity.

The Semantic Text Relatedness interface provide a user-friendly environment for evaluating the likeness between texts, while Paraphrasing interface support the generation of alternative expressions while preserving the original meaning. Additionally, Question Answering interface is designed to handle queries and provide relevant responses in a coherent manner.

By meticulously tailoring each interface to the unique demands of its associated NLP task, we ensure accessibility and usability for both researchers and practitioners. The interfaces serve as intuitive tools that not only empower users to engage with complex language processing tasks effortlessly but also contribute to the overall enhancement of NLP applications in a user-centric manner.

1.5 Contributions

Our thesis delves into the exploration and analysis of a proposed data creation pipeline, presenting several key contributions:

1. **Development of Indic-News-Scraper:** We introduced a groundbreaking Indic-News-Scraper designed to carefully scrape news articles' contents. This tool contributes significantly to the field by providing raw data for pre-annotations. Notably, it operates iteratively, eliminating the need for site-specific scraping code. This scraper is structure-aware, enhancing its ability to extract information from diverse news sources effectively.

2. **Leveraging LLMs for pre-annotations or Evaluations:** A pivotal contribution lies in the exploration and utilization of Large Language Models as pre-annotators or Evaluators/Raters. By integrating these advanced models into the annotation and evaluation processes, we demonstrate their transformative impact on enhancing efficiency and accuracy in language-related tasks. This

contribution not only optimizes existing processes but also opens new avenues for leveraging LLMs in the broader NLP landscape.

3. **Open-Source Annotation/Evaluation Tools:** We provide a valuable contribution to the NLP community by offering open-source annotation and evaluation tools. These tools cater to large-scale annotation and evaluations, and their availability facilitates collaboration and knowledge-sharing within the community. Moreover, the tools are designed to harness the power of LLMs, streamlining the annotation and evaluation processes for a more effective and efficient workflow.

This integrated pipeline has been developed with the primary aim of overcoming limitations present in current frameworks, enhancing the ways in which data is generated, and providing practical tools for a range of NLP applications. By combining innovative approaches in data scraping, model leveraging, and interface development, this research aspires to significantly contribute to the ongoing discussions in NLP research and the practical development of applications.

1.6 Thesis Outline

Our thesis unfolds across six chapters, each delving into the intricate world of NLP applications. In **Chapter 1**, we introduce the challenges we're tackling. We discuss why these challenges matter in the broader context of NLP and give a quick overview of how we're planning to address them. Moving to **Chapter 2**, we shift our focus to the nuts and bolts of creating data for NLP. We explore existing methods, point out their limitations. This chapter provides a deep dive into exiting works, showcasing how they revolutionize the process of creating data for NLP applications.

In **Chapter 3**, we detail a handy tool designed to scrape content from news articles. This tool plays a crucial role in providing raw data for pre-annotations, making the data creation process smoother. We go into the nitty-gritty of the scraper, highlighting its capabilities, especially in dealing with languages like Indic. **Chapter 4** is a pivotal exploration where we integrate LLMs into annotation and evaluation processes. This chapter reveals how these models can significantly improve efficiency and accuracy in these tasks, with potential applications in various language-related processes.

Chapter 5 takes a practical turn, exploring various tools we've developed for specific NLP tasks. Each tool is discussed in detail, showcasing their functionalities and how they contribute to enhancing NLP applications. In **Chapter 6**, we conclude the outcomes of this thesis work. We reflect on what we've learned, summarize our contributions, and look ahead to potential improvements and future directions. This concluding chapter serves as a guide for future researchers in the ever-evolving landscape of NLP applications.

Chapter 2

Literature Survey

2.1 Structure Aware Indic News Scraper

The extraction of content from news websites stands as a pivotal task in today's information-driven world, given the sheer volume of digital content available [33]. This necessitates the development of effective techniques to scrape, process, and leverage news articles efficiently. The task, however, is riddled with challenges, primarily stemming from the diverse structures adopted by different websites. Each website employs a unique structure, demanding a tailored approach for effective data extraction. This complexity is further exacerbated for individuals lacking expertise in scraping techniques. The ever-evolving landscape of website designs and technologies adds another layer of complication, requiring constant adaptation of scraping methods to accurately extract news content.

Content extraction plays a pivotal role in applications that rely on obtaining accurate and pertinent information. One such application is summarization, where the ability to extract key information from news articles facilitates the creation of concise and informative summaries. Additionally, tasks like headline generation or classification heavily rely on accurately extracting titles and categorizing content. The ability to filter scraped data based on published dates, categories, and tags provides a mechanism to customize information according to specific application requirements. For example, certain applications might not necessitate crime news or beauty tips articles, underscoring the importance of the filtering capability. Furthermore, the proposed approach addresses common scraping issues, such as content concatenation and inaccurate representation, ensuring the elimination of duplicate content and enhancing the precision of article text representation.

Understanding the structure of web pages, encompassing the diverse formats and layouts adopted by different websites, is paramount for enabling the accurate extraction of key elements from news articles. Several libraries have been developed to extract news content from web pages. However, these libraries are predominantly tailored to English or other specific languages and may not deliver satisfactory performance for Indian languages. Existing libraries, including NewsOne [46], NewsPlease [12], News-

Paper[1], NewsFetch[2], NewsCatcher[3], PyGoogleNews[4], FeedParser[5], Goose3API[6], and Mozilla Readability[7], have demonstrated varying degrees of success. Nevertheless, they lack comprehensive support for Indian languages and fall short in addressing the unique challenges posed by Indian news websites. This inadequacy motivates the exploration of a novel approach to news content extraction, specifically designed to overcome language limitations and cater to the intricacies of Indian news websites.

Numerous rule-based approaches have been suggested to develop news scrapers and tackle the issues linked to content extraction as follows:

2.1.1 Template Dependent

Several techniques have been proposed in the literature for designing news wrappers and addressing the challenges associated with layout-dependent extraction methods. One approach focuses on detecting the webpage layout to generate appropriate wrappers. This involves the automatic framing of the main content of the webpage using computer programs known as wrappers [37]. Another popular technique involves utilizing the Document Object Model (DOM) of the news webpage for layout detection. Comparing DOM using tree matching techniques can help identify noise contents that may affect extraction accuracy [62].

Alternative methods include transforming the extraction problem into the Largest Continuous Subsequence Sum extraction problem, which enables the identification of omitted contents on webpages [56]. Some approaches involve detecting blocking tags that divide the webpage into functional areas, facilitating content extraction [68]. Tag paths have also been utilized to locate the main content, providing valuable guidance in the extraction process([53], [52], [54]). Linguistic and structural features are extracted from merged textual blocks, and classifier models are applied to enhance extraction accuracy [66]. Additionally, some studies focus on grouping similar pages based on structural similarities and then finding general structure representations using methods such as the Tree Edit Distance (TED) Method [42].

[1]https://github.com/codelucas/newspaper
[2]https://github.com/santhoshse7en/news-fetch
[3]https://github.com/kotartemiy/newscatcher
[4]https://github.com/kotartemiy/pygooglenews
[5]https://github.com/kurtmckee/feedparser
[6]https://github.com/goose3/goose3
[7]https://github.com/mozilla/readability

2.1.2 Template Independent

Layout-dependent methods are criticized for their inability to handle new and unseen webpages, requiring individual website analysis and customized wrappers ([50], [13], [43]). Moreover, the constant structural changes in webpages make the task of updating wrappers for thousands of news websites impractical, often leading to semi-automatic solutions rather than fully automatic ones.

On the other hand, layout-independent methods offer alternative approaches for news content extraction. One efficient method involves converting HTML webpages into paragraphed strings and detecting the main content based on word count [64]. Another approach utilizes heuristics and static analysis of common tags used in news websites to identify different parts of the news article, such as the title, body, and dates [10]. These studies highlight the trade-offs and possibilities in achieving automated news content extraction from diverse web sources.

2.1.3 Visual Based Approaches

In recent years, there has been increasing interest in news extraction methods that simulate human perceptions and investigate visual features. A notable study [5] introduces an extraction approach that leverages human sense simulation through bottom-up adaptive clustering of user perceptions. This method identifies news areas based on content function, space continuity, and formatting continuity of news information. It further identifies detailed news content by considering the position, format, and semantics of the detected news areas.

Furthermore, the news extraction problem has been framed as a classification problem [44] by assigning weights to different features based on their importance. Researchers have explored a wide range of visual and structural features, including position, width, height, background color, padding, margin, border, and font size, among others.

Another proposed technique for webpage segmentation is Vision-based Page Segmentation (VIPS) [63]. This approach defines visual blocks as visible rectangular regions on web pages with fixed positions and nonzero sizes. A vision-based wrapper, known as V-Wrapper, is learned from these visual blocks, distinguishing it from conventional DOM tree-based wrappers, referred to as T-Wrappers.

2.1.4 Machine Learning Based Approaches

Ziegler et.al [67], introduced an automated approach for extracting relevant textual content from HTML pages using machine learning. By analyzing linguistic and structural features, the system classifies text blocks as either signal or noise. The approach is evaluated using a dataset of 600 labeled news documents in multiple languages, comparing its performance against a human gold standard and two benchmark systems. The input HTML pages are transformed into XHTML, and operations are applied

to remove unnecessary elements and prune the DOM tree. Each text block is represented by a vector of 18 features, encompassing linguistic features (e.g., average sentence length, stop-word ratio) and structural features (e.g., anchor ratio, text structuring ratio). The evaluation reveals that the proposed approach achieves a high level of accuracy comparable to human judgment, surpassing the performance of benchmark systems.

Yao et.al [57] addresses the challenge of extracting the main content from webpages by treating it as a classification problem and employing machine learning techniques. The goal is to remove boilerplate elements like navigation panels, advertisements, and comments, which are unrelated to the actual content and can hinder the user's reading experience. The approach involves using an SVM classifier to predict whether each text block in an HTML document is content or non-content based on selected features. The results indicate that the proposed approach achieves comparable performance to existing algorithms in the field. The approach is evaluated using two datasets, one manually classified from Google News articles and the other comprising webpages from RSS feeds, news websites, and blogs. The datasets provide a gold standard for evaluating the extraction of content.

In Peter's research work [38], preliminary results showing improved performance through the combination of feature sets and a method to incorporate semantic information using id and class attributes in HTML5. The authors suggest further enhancements by adding more features or refining existing ones due to limited features in the existing work.

Zhou et.al [65] focuses on web content extraction, specifically targeting less structured content like news articles on noisy web pages. The approach combines visual and language-independent features to classify text blocks. A pipeline is developed for automated labeling through clustering, selecting the best cluster based on relevance to web page descriptions. Visual features, such as font size, color, style, layout, and text density, are used to train an SVM classifier for content extraction without manual labeling. The dataset is collected from popular news websites, and features like block size, position, text content, and tag path are extracted. The study highlights the effectiveness of the DBSCAN clustering algorithm in handling unknown cluster numbers, shapes, and noise. The contribution of features like tag path and CSS selectors is noted, along with their limitations.

Yunis et.al [60] focuses on the task of separating the main content, such as news articles, from noisy elements like advertisements and navigation links on web pages. Instead of operating at a block level, the approach applies content extraction at the level of HTML elements. A dataset of webpages with manually labeled elements as main content or noisy content is created, and machine learning is used to induce rules for the separation. The challenge lies in the close intermingling of main and noisy content in the HTML markup or DOM. The classifier is trained on a set of 30 diverse web pages, including news articles, product descriptions, forum discussions, and videos. The evaluation is performed on a separate

set of 30 web pages, along with 10 web pages from the same websites. Features used include spatial features, content features, stopWordRatio, domHeight, headerAround, text length, tag path, and CSS properties.

In the context of news classification based on individual requirements, Rao et.al [41], developed a web crawler to extract content from news websites and employed machine learning techniques such as Random Forests, Naive Bayes, and SVM classifiers. The approach involves data retrieval through web scraping, data preprocessing for training and testing, and the evaluation of classifier accuracy. In this approach, only the POS tags which contain the Noun Phrases are considered features for model training.

2.1.5 Graph Embeddings based Approaches

Hausner et.al [16] presents a novel method for extracting news articles from diverse news webpages by identifying the main article content and removing irrelevant elements such as advertisements and navigation components. The approach leverages the hierarchical structure of the DOM tree underlying webpages and applies graph representation learning to compute graph embeddings. These embeddings are then used for classifying webpage elements as content or non-content, followed by a refinement step to extract the main article text and eliminate remaining noise. The evaluation on a hand-annotated dataset from German news outlets demonstrates the superior performance of the proposed method compared to baselines. The paper also discusses the use of graph convolutional networks to build low-dimensional vector representations that capture information about the node's neighborhood in the DOM tree. The model's architecture allows for generalization to various classification tasks on webpages, with potential applications in detecting rare patterns or dynamic content. However, the model is limited to static graph representations and may not be as suitable for cases involving dynamic content.

2.1.6 Limitations in Existing Approaches of News Contents' Extraction:

1. Every method of news content(s) extraction uses statistical approaches.

2. Machine learning approaches (extraction as classification problem) extract the content with better results. These approaches are trained on less data.

3. Even though, some of the python libraries (news-fetch, pygooglenews, newscatcher, feedparser, newspaper3, news-please, news-one, goose3) are implemented to extract the content from news articles, but are limited to the RSS feed, and not for the Indian languages.

4. There is a need for semi-automated scraper for news content(s) extraction for data annotation.

2.2 Data Creation Frameworks

Navigating the expansive field of Natural Language Processing (NLP) applications demands robust data creation frameworks that transcend the limitations of task-specific tools. These frameworks form the bedrock for training diverse NLP models. This section delves into an exploration of various prominent frameworks, including but not limited to large-scale frameworks: Label-Studio, Inception, and Shoonya.

2.2.1 General Purpose Tagging Tools

Multiple strategies and tools have been suggested for text annotation, each bringing unique capabilities to the domain. A pioneering approach by [9] introduces a novel method for the selective annotation of extensive corpora, utilizing machine learning to address shortcomings in traditional linguistic search engines. Another notable tool, DoTAT [27], is designed for domain-oriented information extraction, offering collaborative annotation, event annotation, visual specification, and enhanced annotation efficiency. Open-source taggers, NameTag, and MorphoDiTa [45], specialize in named entity recognition and morphological analysis, demonstrating high performance in the Czech language. GATE Teamware [4], a collaborative web-based framework for text annotation, provides user-friendly interfaces, customizable workflows, automatic preprocessing, and project evaluation. Additionally, SLATE [22] stands out as a lightweight terminal-based annotation tool, featuring a simple interface and flexible annotation options.

2.2.2 Named Entity Tagging Tools

AlpacaTag [26], an open-source web-based annotation framework, focuses on sequence tagging tasks like named entity recognition (NER). It incorporates active intelligent recommendation, automatic crowd consolidation, and real-time model deployment, providing a comprehensive solution for sequence labeling tasks. The extended version of WebAnno [58] enhances manual text document annotation with support for multiple annotation layers, a machine learning component for automatic suggestions, and reduced annotation time. Open Annotation (OA) model [39] is a web interface model based on web standards, improving interoperability between online annotation tools and resources. TALEN [31], designed for named entity annotation in low-resource settings, integrates lexicon integration, token statistics, internet search, and entity propagation, achieving higher precision and recall. WebAnno [59] stands as a versatile web-based tool supporting linguistic annotations, offering project management, customizable tagsets, user management, visualization, and editing capabilities. T-NER [49] is a Python library facilitating NER language model (LM) finetuning, enabling cross-domain and cross-lingual generalization studies. APLenty [35] integrates active and proactive learning for high-quality sequence labeling datasets. CroAno [61] addresses label consistency issues in Chinese NER through a web-based crowd annotation platform. INCEpTION [20], a highly configurable open-source labeling tool, offers fine-

grained control over the annotation process. Doccano [34], emphasizing simplicity and ease of use, provides a streamlined web-based interface for annotating text data. Label Studio [47] offers both a free open source community edition and a paid enterprise edition, supporting ML and active learning. Prodigy[8], a closed-source labeling tool developed by the creators of spaCy, caters to data scientists, with an intuitive frontend and advanced functionalities accessible through the command line.

2.2.3 Summarization Tools

The landscape of text summarization has been enriched by an array of innovative tools and frameworks designed to address various aspects of this complex task. In this section, we delve into a diverse collection of summarization tools, each contributing unique perspectives and methodologies to the broader field.

MDS Writer: The MDS Writer (Multi-document Summarization Corpora) is a system designed to simplify the complex task of summarization by breaking it down into intermediate sub-tasks. This modular approach facilitates easier evaluation at each step of the summarization process.

MUSEEC: MUSEEC is a versatile summarization tool supporting extractive techniques across multiple languages. It offers various summarization methods, including the supervised MUSE, unsupervised POLY, and the extended POLY called WECOM. Successfully evaluated on benchmark document collections in English, Arabic, Hebrew, and other languages, MUSEEC boasts a flexible architecture and API. While it utilizes pre-processing tools for summarization quality assessment, users can enhance coherency through techniques like automatic rewriting (AR) and named entity (NE) tagging. Advanced post-processing operations can further improve the overall user experience.

Summarization Integrated Development Environment (SIDE): SIDE provides an infrastructure for personalized summaries, recognizing the subjective nature of a perfect summary. Users can determine the structure and content they find valuable, making it an educational tool for Summarization and Personal Information Management. Tested successfully with a class of 21 students, SIDE serves as a framework to explore and define the user-specific summarization needs.

MEAD: MEAD stands as a platform for multi-document multilingual text summarization, available as open source. Widely adopted, it has found applications in summarization for mobile devices, web page summarization in search engines, and novelty detection.

PKUSUMSUM (A Java Platform for Multilingual Document Summarization): PKUSUMSUM, a Java-based platform, streamlines multilingual document summarization. Supporting multiple languages and integrating ten automatic summarization methods, it addresses common summarization tasks and is

[8] https://prodi.gy/

available for public use.

Interactive Abstractive Summarization for Event News Tweets: This system, based on abstractive summarization and consolidated knowledge representation, offers a bullet-style summary for event news tweets. It enhances text exploration through an interactive user interface, providing key information first and allowing users to delve into specific details gradually.

SUMMARY EXPLORER (Visualizing the State of the Art in Text Summarization): SUMMARY EXPLORER is a tool designed for the manual inspection of text summarization systems. It compiles outputs from 55 state-of-the-art single document summarization approaches, facilitating qualitative assessment through visual exploration. Incorporating three summary quality criteria, it supports close and distant reading analysis, particularly for examining abstractive summarization models.

SUMMARY WORKBENCH (Unifying Application and Evaluation of Text Summarization Models): SUMMARY WORKBENCH is a versatile tool for developing and evaluating text summarization models. It allows easy integration of new models and evaluation measures as Docker-based plugins, supporting assessment of summary quality and providing visual analyses to identify strengths and weaknesses. Accessible online or through local deployment, it streamlines the development and evaluation processes.

SummerTime (Text Summarization Toolkit for Non-experts): SummerTime stands as a comprehensive toolkit for text summarization, offering a wide array of models, datasets, and evaluation metrics. Seamlessly integrating with NLP libraries and providing user-friendly APIs, it enables users, even non-experts, to find pipeline solutions, explore model performance with their data, and visualize differences. The toolkit includes explanations for models and evaluation metrics, aiding users in understanding model behavior and choosing the most suitable options for their specific needs.

Having explored a rich array of existing tools and frameworks in the realm of text summarization and Tagging tools, the subsequent sections of this thesis delve into the development and integration of key components aimed at advancing the field. This thesis comprises three core components, each playing a pivotal role in streamlining the process of data creation, annotation, and evaluation for natural language processing applications. The first component focuses on the **Indic News Scraper**, a sophisticated tool designed for the systematic extraction of content from news articles, offering a structure-aware approach that eliminates the need for site-specific scraping code. The second component delves into the strategic **Leveraging of Large Language Models (LLMs)** for pre-annotations, employing prompt engineering techniques to integrate advanced language models into the annotation process. Lastly, the third component involves the development of dedicated interfaces tailored for various **NLP applications**, providing purpose-built platforms for tasks such as summarization, headline generation, contract

Named Entity Recognition (NER) tagging, machine translation evaluation, semantic similarity assessment, paraphrasing, and question-answering. This section serves as a comprehensive roadmap for the subsequent detailed exploration and analysis of each component.

Chapter 3

Structure Aware Indic News Scraper

Hassanian et.al [15], broadly defined the news article into two subsections namely:- "**Fields**" and "**Zones**".
Fields: It is a part of the content with a predefined length and may differ from the length with respect to the website. The majority of these fields' information can be fetched from the article's webpage using the $< meta >$ tag content.
Zones: It constitutes a component of the content characterized by variable length, which may differ from page to page within articles. Identification of the zone areas (locating the content of each zone) is a challenging task due to the lack of standard structure of the article's webpage. This Chapter, detail the proposed Indic News Scraper architecture with data collection process for model training and improvement (codebase is available at `https://github.com/pavanbaswani/Indic_News_Scraper`)

3.1 Indic News Scraper (INewS) Architecture

Figure 3.1 illustrates the proposed system, which consists of two major modules: **URL Extractor** and **News Extractor**. The *URL Classifier (URL-CLS)* is employed by the URL Extractor to categorize the URL into one of the pre-defined classes (category URL, page URL, article URL, file, and misc), while the *Content Classifier (Cont-CLS)* is utilized by the News Extractor to categorize the news content to pre-defined classes (article, tags, external links, headline, publish date, author, highlights, sub title, category, and images). In the subsequent sections of this paper, we use the terms *URL-CLS* for the URL Classifier and *Cont-CLS* for the Content Classifier.

3.1.1 URL Extractor

The URL Extractor plays a crucial role in fetching article URLs from a designated base URL, typically a news website link. This extraction is achieved through the iterative crawling of URLs and and classifying with the well-trained URL classifier, denoted as URL-CLS. The classifier undergoes training with annotated data to enhance its accuracy in classifying URLs. Once the extraction process is com-

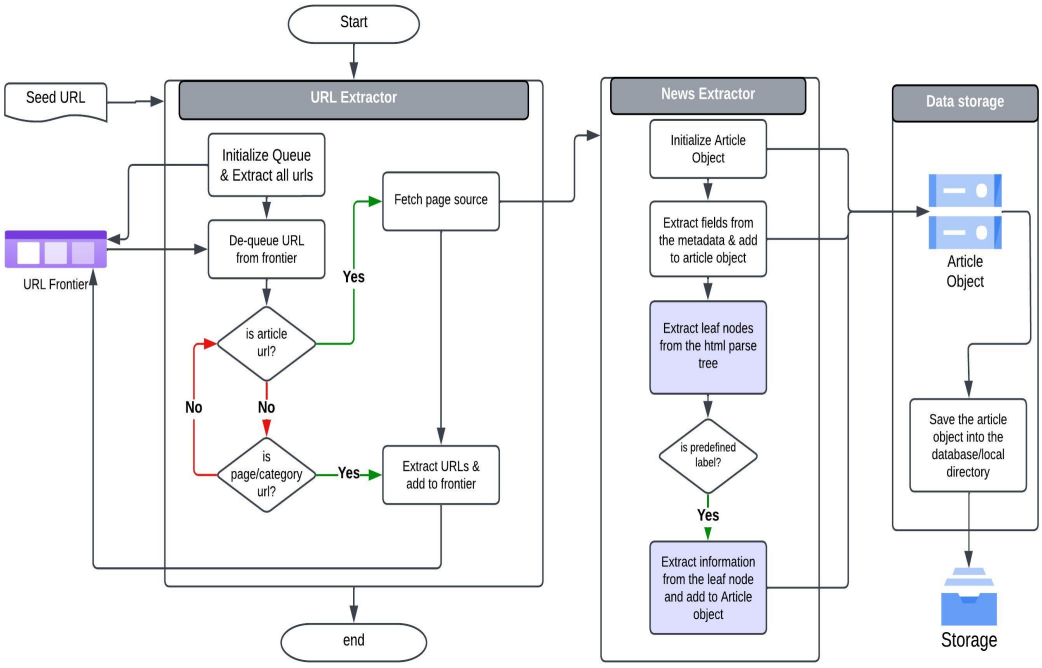

Figure 3.1 INewS Model Architecture

plete, the identified article URLs are then fed to the News Extractor for further processing. To train the URL-CLS, the classification dataset is created with dedicated interface for classifying the URL into the pre-defined classes.

3.1.1.1 URL Classification Dataset

Developing the classification dataset involves several key stages, including web scraping, categorizing URLs, creating an annotation tool, and annotating the URLs.

URL Scraping: In the dataset creation process, the initial phase involved crawling URLs from a comprehensive set of 558 Indian news websites across various Indian languages (see Table 3.1). The scraping process included visiting the first page of each website, usually the homepage, and extracting all URLs found on that page. Ensuring diversity in sources and languages was a priority during dataset construction.

URL Classes: The URLs are broadly divided into five classes as follows:

1. *article URL*: URL that provide the complete content of a single article, along with related article URLs located at the bottom.

Language	# Websites
Assamese (as)	9
Gujarati (gu)	53
Hindi (hi)	87
Kannada (kn)	30
Malayalam (ml)	74
Marathi (mr)	35
Oriya (or)	30
Panjabi (pa)	35
Tamil (ta)	111
Telugu (te)	54
Urdu (ur)	40
Total	**558**

Table 3.1 URL Scraping Data statistics

Example: https://www.andhrajyothy.com/2023/national/health-insurance-obc-quota-ipl-team-in-congress-manifesto-for-mp-elections-avr-1155391.html

2. *page URL*: contains list of article URLs that reside within a single page, with or without any pagination or infinite scroll functionality.
   Example: `https://www.andhrajyothy.com/national/page/2`

3. *category URL*: contains article URLs with pagination, infinite scroll, or "Load more" options. Some of these category URLs were embedded in the menu bar of the website.
   Example: `https://www.andhrajyothy.com/national`

4. *file*: URLs with file extensions such as images, videos, HTML, CSS, JavaScript, ASPX, and others, excluding article, page, or category URLs are classified as file.

5. *misc*: URLs that did not fit into any of the predefined categories.

URL Annotation Tool Development: To streamline the annotation process, a custom URL Annotation Tool was developed. This tool facilitated annotation across all languages (refer Table 3.1), handling a substantial volume of samples. It presented necessary information to annotators by loading the URL webpage for visualization, allowing accurate assignment of each URL to predefined classes (refer Table 3.2). The tool's development was crucial for ensuring consistent and efficient annotation.

URL Annotation: Initially, we computed the Fleiss' kappa[1] score by randomly sampling 2 URLs per website, each annotated by 3 annotators. Given the simplicity of the annotation task, which requires only basic knowledge of news websites, we achieved an Inter-Annotator Agreement score of **0.96**. Using the developed tool (refer to Figure 3.2), each URL in the dataset was annotated with its corresponding class,

[1] `https://tinyurl.com/3ux4hvfw`

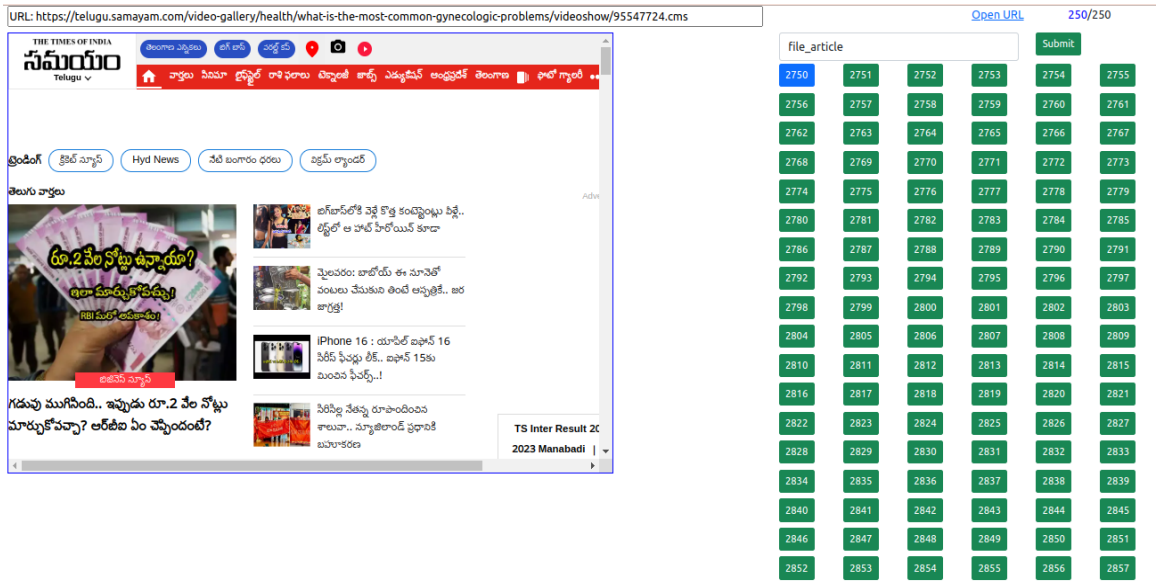

Figure 3.2 URL Annotation Interface

Category	Train	Dev	Test
article_url	30544	1697	1697
category_url	11670	648	648
misc	8256	458	459
page_url	3832	213	213
file	1101	61	61
base_url	495	27	28
# Samples	**55898**	**3104**	**3106**

Table 3.2 URL Classification Annotation Dataset

with only one annotation per sample due to the high agreement rate. This process covered all languages represented in the dataset, ensuring a comprehensive analysis of news articles across diverse linguistic contexts. Accuracy in annotations was emphasized, as they form the foundation for subsequent tasks reliant on the labeled dataset. After eliminating duplicate URLs, the resulting dataset comprises more than 60,000 samples (refer to Table 3.2), offering a substantial and diverse collection for further analysis and training.

3.1.2 News Extractor

The responsibility of the News Extractor is to retrieve the HTML content of an article and categorize it into predefined labels (refer to Table 3.3). Employing a combination of web scraping and natural language processing techniques, the system extracts article content. To classify the text within HTML pages obtained from specific URLs, we developed the Content Classifier (Cont-CLS). To identify important labels for content classification, we conducted a manual review, assessing label presence on web

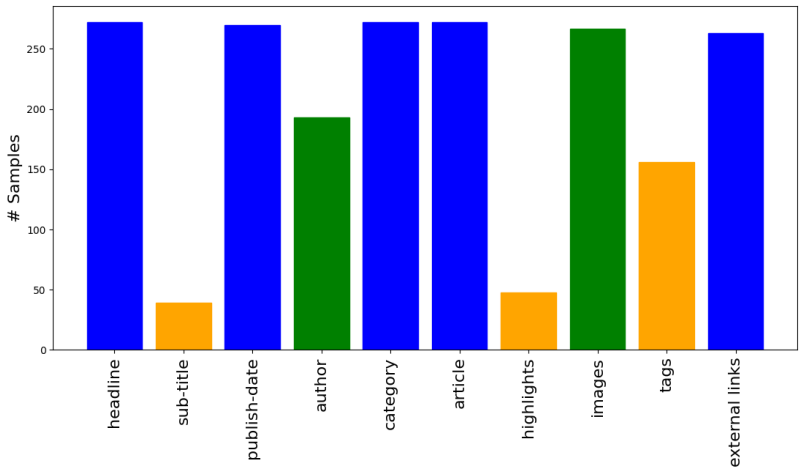

Figure 3.3 Presence of labels in diverse URLs set collected from all languages.

Label	Train	Val	Test	# Samples
article	28915	4933	5782	**39630**
tags	27505	11183	7451	**46139**
external_links	11386		453	**11839**
headline	7711	2327	2544	**12582**
publish_date	7522	2319	2503	**12344**
images	1227			**1227**
author	1170			**1170**
highlights	998		8	**1006**
sub_title	409			**409**
category	360	56	154	**570**
Split Size	**87203**	**20818**	**18895**	

Table 3.3 News Extractor Experimental Data Statistics

pages with diverse URLs across all languages. Figure 3.3 illustrates the importance of the considered labels for content classification. The Cont-CLS is designed to categorize leaf nodes in parsed HTML, containing textual content, into predefined labels.

Website-Specific Content Scraping: To generate necessary training data, we crafted website-specific content scraping code tailored to extract relevant content for predefined labels. For experiments, the scraped data is organized source-wise (distinct websites in train, test, and dev splits) rather than label-wise (the missing values in val and test sets due to source distribution differences). Table 3.3 displays label statistics from all scraped websites, covering Indic languages mentioned in Table 3.1.

Leaf Node Extraction and Content Classification: Parsed HTML pages from the URLs were analyzed to identify leaf nodes, serving as the classification targets. By focusing on the HTML structure, Cont-CLS prioritizes relevant text elements, avoiding extraneous information. The identified leaf nodes are then input to Cont-CLS for predicting one of the predefined labels.

	LR			Naive Bayes			SVM			Random Forest			Boosting			MLP		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
article url	0.80	0.90	0.85	0.87	0.79	0.83	0.84	0.86	0.85	0.96	**0.96**	0.96	0.96	**0.96**	0.96	0.96	0.91	0.94
category url	0.43	0.68	0.52	0.42	0.73	0.54	0.42	0.77	0.54	0.84	0.85	0.84	0.82	**0.90**	0.86	0.69	0.89	0.78
page url	0.00	0.00	0.00	0.65	0.06	0.11	0.00	0.00	0.00	0.85	**0.86**	0.86	0.86	**0.86**	0.86	0.74	0.82	0.78
base url	0.00	0.00	0.00	0.16	0.96	0.27	0.00	0.00	0.00	0.40	**0.14**	0.21	0.67	**0.14**	0.24	0.58	0.25	0.35
file	0.00	0.00	0.00	0.23	0.61	0.34	1.00	0.02	0.03	0.86	0.92	0.89	0.85	**0.93**	0.89	0.89	0.89	0.89
misc	0.71	0.27	0.39	0.78	0.14	0.23	0.60	0.26	0.36	0.82	**0.82**	0.82	0.86	0.78	0.82	0.86	0.68	0.76

Table 3.4 URL Classification Experimental Results

3.2 Experiments & Results

3.2.1 URL Classifier (URL-CLS)

The URL-CLS was designed to categorize URLs into different classes based on their relevance to news articles.

3.2.1.1 Feature Extraction:

To capture pertinent information from URLs, syntactic and visual features were extracted using rule-based methods. These features offered valuable insights into the structure and characteristics of the URLs, including *URL length, host length, path length*, and various other attributes such as the *presence of dots, at symbol, percentage symbol, underscore, tilde, ampersand, hash, hyphen, slash, equal sign, semicolon, comma, period, parameters, queries, fragments, port information, digits in the host, IP-based host, and the presence of a protocol*.

3.2.1.2 Model Implementation:

The URL-CLS was implemented as a text classification model, utilizing the annotated dataset with URL features for training (see Table 3.2). This dataset comprised manually labeled URLs assigned to predefined labels (category URL, page URL, article URL, file, or misc). The classifier learned patterns and relationships between URL features and their corresponding labels, enabling accurate categorization of new URLs. Experimental results using traditional ML models (Logistic Regression (LR), Naive Bayes, Support Vector Machine (SVM), Random Forest, Boosting, and Multi-Layer Perceptron (MLP)) are presented in Table 3.4. Among these models, Random Forest and Boosting demonstrated significant results and were further employed in the proposed architecture.

	precision	recall	f1-score	support
article	0.99	0.88	**0.94**	5782
category	0.00	0.00	0.00	154
external_links	0.78	1.00	**0.88**	453
headline	0.91	0.98	**0.94**	2544
highlights	0.00	0.00	0.00	8
publish_date	1.00	1.00	**1.00**	2503
tags	0.94	1.00	**0.97**	7451

Table 3.5 News Content Extraction Experimental Results on unseen sources

3.2.2 Content Classifier (Cont-CLS)

3.2.2.1 Feature Extraction

The classification of content heavily relies on stylistic features rather than the actual textual content. This dependence is observed in fonts, length, and CSS style features, with content often distinguished at the source through various CSS class and id tags. The primary goal of Cont-CLS is to harness structural and stylistic cues embedded in HTML and CSS tags, combining them with textual representation. To achieve this, we leverage the fusion of HTML and text embeddings for comprehensive feature extraction (depicted in Figure 3.4). The process involves replacing textual content in the leaf nodes of the DOM with a special token, [TEXT]. The modified HTML is then processed through MarkupLM[2], a specialized language model adept at capturing HTML-specific features. Simultaneously, plain text content is fed into the multilingual BERT (mBERT[3]) model, pre-trained with Indic language data to capture semantic nuances. The resulting embeddings from these streams are mean-pooled and then added, creating a representation that amalgamates distinct HTML and plain text features. This enriched representation forms a robust foundation for enhanced classification of leaf nodes in our model.

3.2.2.2 Model Implementation

Cont-CLS is implemented using an ensemble BERT model capable of encoding both HTML and text content (see Figure 3.4). The model is trained using the extracted HTML content and associated plain text for each leaf node[4]. The annotated dataset, comprising scraped markup data from Indic news websites, serves as the training data. The classifier learns patterns and relationships between textual content and HTML structure, enabling accurate categorization of new leaf nodes into predefined labels. Table 3.5 presents experimental results of Cont-CLS on Test data, demonstrating significant improvements for most labels. This model is subsequently employed in the proposed architecture.

[2]microsoft/markuplm-base

[3]bert-base-multilingual-uncased

[4]Sequence length=512, epochs=4, batch size=96, Adam optimizer with learning rate=$1e-4$

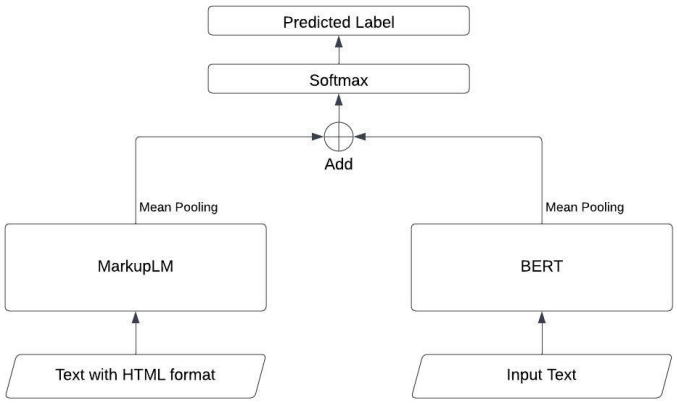

Figure 3.4 Content Extractor Model Architecture

3.3 Conclusion

We introduced an innovative architecture that significantly improves the effectiveness of web scraping for extracting news content and metadata. Our approach efficiently captures crucial data elements, including *headline, article text, publication dates, related article URLs (external links), and tags*. Leveraging the base URL of news websites, we iteratively extract previously unseen article URLs, ensuring a comprehensive retrieval of essential information from each article. In addition to enhancing web scraping techniques, we contribute valuable resources to the research community. We release a URL classification dataset comprising approximately 60,000 instances, facilitating the classification of URLs into categories such as category URL, page URL, article URL, file, or miscellaneous links. Alongside the URL classification dataset, we provide the Content Classifier (Cont-CLS) trained on site-specific crawled data, addressing challenges specific to Indic news websites.

Chapter 4

Large Language Models as Pre-Annotators/Evaluators

The emergence of LLMs has brought about a significant shift in NLP, fundamentally changing how machines understand and generate human-like text. Models like GPT-3 have shown remarkable abilities in grasping context and producing coherent language. However, their applications go beyond mere text generation. In this section, we explore a novel use of LLMs – employing them as pre-annotators or evaluators/raters in different NLP tasks.

This involves leveraging the inherent capabilities of LLMs, such as GPT-3, for tasks like Named Entity Recognition (NER) tagging or evaluating how well machine-generated outputs align with human-like language. By tapping into the pre-existing knowledge within these models, researchers and practitioners gain a valuable resource for kick-starting the annotation process or evaluating the quality of machine-generated content. This section takes a closer look at the methods, benefits, and challenges of incorporating LLMs into these roles, shedding light on their evolving role in shaping the field of NLP applications.

4.1 Fine-grained Contract NER using instruction based model

Contracts serve as legally binding agreements that delineate the rights and responsibilities of parties involved, overseeing interactions among companies, employees, contractors, customers, and suppliers. Unlike the corpora commonly used for pre-training deep models, contracts exhibit distinct composition and terminology. Typically following specific template formats for clarity, precise word selection and sentence structure in contracts are paramount due to the potential ramifications of even minor ambiguities. Therefore, meticulous drafting and comprehensive reviews are essential, as contracts are vital instruments for managing business relationships and mitigating risks. The development of automated tools and applications is pivotal in streamlining the time-consuming processes of contract understanding, drafting, and review.

One critical aspect in facilitating contract review is entity extraction, particularly through named entity

recognition, which plays a foundational role in extracting and processing information from contracts. While systems designed for recognizing named entities typically identify individuals, organizations, dates, locations, and currency terms, legal texts present nuanced differences that necessitate a more nuanced analysis. Manual extraction of named entities or contract elements can be labor-intensive, costly, and repetitive, fueling the demand for automation from both legal professionals and their clients.

With these challenges in mind, this work endeavors to address the automatic identification of crucial contract elements. These elements encompass parties involved, specific dates, monetary values, explicit rights and obligations, and relevant governing laws—all of which bear significant importance within the context of a contract.

Automating the identification of these elements stands as a crucial strategy to streamline the contract analysis process, cut costs, and improve overall efficiency within the legal domain. In the context of contracts, this paper employs the terms "Named Entity Recognition" and "Contract Element Extraction" interchangeably. While prior studies have primarily focused on identifying fine-grained named entities in judgment documents [18, 24, 3], efforts concerning contracts have faced challenges due to limited coverage of entity types [2, 7] and contract categories [25, 36].

This work presents the development of a prompt-based corpus for contract Named Entity Recognition (NER), covering eighteen fine-grained entity types from seven commonly encountered contract types. The study encompasses the creation of baseline models for sequence labeling, parameter-efficient learning, and prompt-based learning using Language Model Models (LLMs), alongside a comparative analysis of LLMs' performance in information extraction tasks.

ContractNER[1] dataset comprises diverse legal contracts sourced from SEC EDGAR [2](Electronic Data Gathering, Analysis, and Retrieval). It is a comprehensive online database maintained by the U.S. Securities and Exchange Commission (SEC). It serves as a centralized repository for a wide range of finan- cial and business-related documents submitted by publicly traded companies, investment firms, and other entities regulated by the SEC.

4.1.1 Contract Processing

In the process of curating our legal contract dataset and making the contracts amenable to further analysis, we extracted plain text from raw documents sourced. In the scraped dataset, we observed a diverse range of contract titles, but not all titles were equally represented. To address this imbalance, we employed heuristics to extract the most common contract titles based on their frequency of occurrence. Table 4.1 outlines the contract titles extracted and the counts of contracts extracted for each title. By

[1] `https://github.com/pavanbaswani/ContractNER`
[2] https://www.sec.gov/edgar/search-and-access

including a diverse set of frequently titled documents in the training data, the model gains a deeper understanding of legal contract structures and their terminology. Leveraging a rich and varied training dataset enables our model to become a powerful tool for handling contract-related entity extraction, and streamlining contract analysis efficiently with precision and efficiency.

Contract Type	Train	Dev	Test
Employment	113	19	15
Credit	4	2	2
Purchase	53	6	6
Loan	15	4	4
Lease	35	4	4
Indemnification	21	2	2
Consulting	16	2	2

Table 4.1 Legal Contract types and their documents' distribution

4.1.2 Pre-Annotations

Manual annotation using entity recognition taggers is a crucial and labor-intensive process. It involves human annotators carefully examining the text data and marking specific words or phrases that represent named entities, such as names of people, organizations, locations, and other proper nouns. Entity taggers are NLP tools that extract mentions of entities (such as people, places, or objects of interest) from a document. They are used for various purposes including information extraction, and question-answering. Different entity recognition taggers are available based on their purpose and scope. General-purpose taggers are versatile annotation tools used for various tasks, such as classification, span detection, entity tagging, and part-of-speech tagging. Some commonly used tools for generic tagging tasks include GATE Teamware [4], NameTag [45], SELECTIVE ANNOTATION [9], SLATE [22], and DoTAT tool [27]. They were largely utilized to perform the generic tagging tasks mentioned above. On the other hand, there are named entity taggers like WebAnno [59], [58], Open Annotation (OA)[39], TALEN[31], APLenty [35], AlpacaTag [26], CroAno [61], Doccano [34], Label Studio [47] and INCEpTION [20] that work well with entity tagging. However, some of these taggers are not open-sourced, and few lack support for pre-loaded annotations using available entity taggers like Spacy[3] or LexNLP[4].

To enhance the annotation process and enable pre-annotations from available pre-trained models, we used in-house named entity tagger that can serve our specific purposes effectively. We leverage the few-shot predictions capability of ChatGPT[5] and predictions from LexNLP to auto-populate annotations related to predefined entity categories. Entity extraction using ChatGPT involves providing context from the contract and posing an instruction. An example of the prompt we used is in Figure 4.1. The

[3]https://spacy.io/

[4]https://github.com/LexPredict/lexpredict-lexnlp

[5]https://chat.openai.com/

model's few-shot capability enables it to extract various entities, such as dates, parties, acts, governing laws, and amounts, from the contract. This flexibility and adaptability make it a valuable tool for automated analysis of legal documents. For other specific entities like generic dates, addresses, courts, and acts, we utilize the LexNLP python library, which employs trained models, heuristics, and dedicated functions to identify and extract entities. For example, LexNLP offers a function to extract generic dates by scanning the input text and retrieving all date-related entities. The output of the extraction process presents well-structured representations of the identified entities, typically in lists or dictionaries, ready for further processing or analysis to meet the application's specific requirements.

```
PROMPT = (
    "Entity Definition:\n"
    "1. ContractTitle: Short name or full name of the contract document.\n"
    "2. ContractParties:  Names of the two or more parties who signed the contract.\n"
    "3. EffectiveDate: The date from when the contract is effective.\n"
    "4. SalaryCompensation: Salary or compensation mentioned for the employee in Employee type Agreements.\n"
    "5. GoverningLaw: The state/country's law that governs the interpretation of the contract.\n"
    "6. EmploymentRole: The role for which an employee is employed.\n"
    "\n"
    "Output Format:\n"
    """{{"ContractTitle": [list of entities present], "ContractParties": [list of entities present], "EffectiveDate": [list of entities present],"SalaryCompensation": [list of entities present],
    "GoverningLaw": [list of entities present],"EmploymentRole": [list of entities present]}}\n"""
    "If no entities are presented in any categories keep it None\n"
    "\n"
    "Examples:\n"
    "\n"
    """1. Sentence: THIS PURCHASE AND SALE AGREEMENT (this "Agreement") made as of the ____ day of October, 2015 between AMERCO REAL ESTATE COMPANY, a Nevada corporation, having an address at 2727
    North Central Avenue, Phoenix, Arizona 85004 ("Seller") and 23RD AND 11TH ASSOCIATES, L.L.C., a Delaware limited liability company, having an address c/o The Related Companies, L.P., 60 Columbus
     Circle, New York, New York 10023.\n"""
    """Output: {{"ContractTitle": ["PURCHASE AND SALE AGREEMENT"], "ContractParties": ["AMERCO REAL ESTATE COMPANY","23RD AND 11TH ASSOCIATES, L.L.C"], "EffectiveDate": ["None"],
    "SalaryCompensation":["None"],"GoverningLaw":["None"],"EmploymentRole":["None"]}}\n"""
    "\n"
    """2. Sentence: TO EXECUTIVE EMPLOYMENT AGREEMENT ("Amendment") is entered into as of August 8, 2016, to be effective as of July 1, 2016, by and between Aqua Metals, Inc., a Delaware corporation
    ("Company"), and Thomas Murphy ("Executive").\n"""
    """Output: {{"ContractTitle": ["EXECUTIVE EMPLOYMENT AGREEMENT"], "ContractParties": ["Aqua Metals, Inc.","Thomas Murphy"], "EffectiveDate": ["July 1, 2016"],"SalaryCompensation":["None"],
    "GoverningLaw":["None"],"EmploymentRole":["None"]}}\n"""
    "\n"
    """3. Sentence: The Employee will be paid an annual salary of Three Hundred Eighty Thousand Dollars ($380,000).\n"""
    """Output: {{"ContractTitle": ["None"], "ContractParties": ["None"], "EffectiveDate": ["None"],"SalaryCompensation":["$380,000"],"GoverningLaw":["None"],"EmploymentRole":["None"]}}\n"""
    "\n"
    """4. Sentence: 2.7.Applicable Law.  This Guaranty shall be construed in accordance with and governed by the laws of the State of New York, without regard to conflict of laws principles.\n"""
    """Output: {{"ContractTitle": ["None"], "ContractParties": ["None"], "EffectiveDate": ["None"],"SalaryCompensation":["None"],"GoverningLaw":["State of New York"],"EmploymentRole":["None"]}}\n"""
    "\n"
    """5. Sentence: Employee serves as its Executive Vice President & Chief Commercial Banking Officer responsible for managing and coordinating the Bank's commercial banking activities.\n"""
    """Output: {{"ContractTitle": ["None"], "ContractParties": ["None"], "EffectiveDate": ["None"],"SalaryCompensation":["None"],"GoverningLaw":["None"],"EmploymentRole":["Executive Vice President
    & Chief Commercial Banking Officer"]}}\n"""
    "\n"

    """6. Sentence: {}\n"""
    """Output: """
)
```

Figure 4.1 Prompt For Few-Shot Learning in ChatGPT

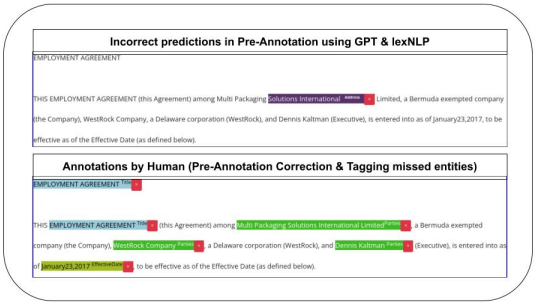

Figure 4.2 Add Missed Annotation

4.1.3 Experiments and Analysis

In this section, we detail the exhaustive experiments on fine-grained named entities found in contracts and verify the effectiveness of instruction models for named entity recognition tasks. Although

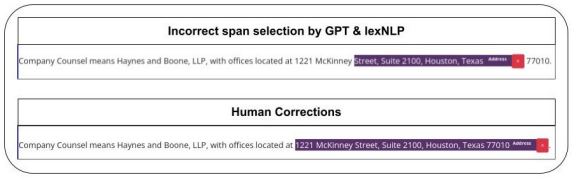

Figure 4.3 Rectify Pre Annotation

LLMs (Large Language Model-based Models) have achieved remarkable success in various NLP tasks like text generation, summarization, and sentiment analysis, their performance in information extraction tasks, particularly in Named Entity Recognition (NER), is still lacking compared to supervised approaches. Additionally, LLMs encounter the issue of hallucination, which limits their usability in critical information retrieval tasks, where controlaccuracy is crucial. To overcome these limitations, a promising approach is to harness the strengths of both LLMs and supervised models through a combination strategy. When fine-tuned on NER-specific data, LLMs can effectively learn to recognize and extract named entities, surpassing the zero-shot and few-shot capabilities of LLMs.

4.1.3.1 Models

In our experiments, we compare popular NER model architectures including prompt-based methods.
1) Sequence labeling models: We apply the traditional sequence labeling method for named entity recognition with the token classification method of BERT [8]. We extend BERT (LEGAL-BERT-BASE) for sequence labeling in order to identify phrases of interest. It enables fine-grained entity recognition at the token level, allowing for precise localization and classification of entities.
2) Parameter Efficient models: Parameter-efficient models [28] have become increasingly popular in recent times. These models focus on updating only a small subset of parameters during the adaptation of a pre-trained model to downstream tasks. A notable example of parameter-efficient tuning is Low-Rank Adaptation (LoRA) [17], which aims to reduce the number of trainable parameters by employing low-rank representations. We fine-tune our dataset with the token classification method of roberta-large [30] model. LoRA was applied on the large model to attain efficiency in storage and training. With significantly fewer parameters, LoRA allows for a more streamlined and resource-efficient model, making it a favorable option.
3) Prompt based models Having observed the benefits of few-shot learning in our pre-annotations, we decided to explore the potential of prompt-based models, which have gained significant importance in the field. These models reframe the sequence labeling task as a generation problem, providing a fresh perspective to tackle the NER task. To align our dataset with this innovative approach, we transformed it into an instruction-based generative framework inspired by NER model based on instructions [51]. By combining source sentences with descriptive task instructions and limited answer options, we crafted a setup that enhances the model's ability to understand and generate relevant entities. Finally, we fine-tuned the T5-small model [40] on this modified dataset, capitalizing on the power and versatility of

prompt-based learning to further improve our NER results. We opted for T5-small due to its architecture, which includes both encoder and decoder components. Information extraction tasks tend to benefit from architectures that incorporate both encoder and decoder, as opposed to models that only feature a decoder. The combination of prompt-based techniques and T5-small fine-tuning improved the performance of our NER system.

4.1.3.2 Experimental Setup

Hyper Parameters: To train the model, we maintained uniform hyperparameters, including a sequence length of 512, a learning rate of 5e-5, Adam optimizer, and a batch size of 4. Additionally, we set the number of beams to 3 for the Prompt-based Model. The training process took place on a machine equipped with the following hardware specifications: an Nvidia P100 GPU with 16GB memory, operating at a GPU clock speed of 1.32GHz, supported by 2 CPU cores and 12GB RAM. The entire setup was hosted on the Kaggle platform.

	Stage-1			**Stage-2**			**Stage-3**		
	Train	**Dev**	**Test**	**Train**	**Dev**	**Test**	**Train**	**Dev**	**Test**
# Tokens	243706	29801	23167	384439	68084	78427	531293	92685	92543
# Unique tokens	10813	4091	3710	17777	7405	8936	25464	9592	10134
# Paras	2986	327	267	4744	770	929	6882	1059	1113
Avg para length	81.61	90.81	86.51	81.02	88.07	84.25	77.2	87.45	83.14
Max para length	641	656	542	947	1557	2452	2725	1557	2452

Table 4.2 Data distribution Statistics

4.1.4 Results

The outcomes obtained from our experiments are summarized in Table 4.4, showcasing the performance of three baseline models: sequence labeling token classification, a parameter-efficient model fine-tuned on a large pre-trained language model coupled with Low-Rank Adaptation (LoRA), and an instruction-based model fine-tuned using T5-small coupled with LoRA on our comprehensive dataset. Our observations reveal that instruction-based models have surpassed both sequence labeling and parameter-efficient models, affirming our hypothesis that supervised learning on large language models (LLMs) leads to enhanced accuracy.

In scenarios involving certain entity categories such as *Rent* and *Shares*, where token-based classification in both sequence-based and parameter-efficient models fell short in producing results due to limited samples, prompt-based models exhibited superior performance. This underscores the significance of thoughtfully crafted prompts in guiding models to generate accurate responses, particularly in data-scarce situations. This principle extends to other entities, where we observed higher precision and recall values. Tasks encompassing a diverse range of inputs and outputs are more effectively managed

	Stage-1			Stage-2			Stage-3		
Labels	**Train**	**Dev**	**Test**	**Train**	**Dev**	**Test**	**Train**	**Dev**	**Test**
Act	5631	651	327	8111	1072	976	9664	1368	1186
Address	1230	147	133	4277	750	2197	9921	1544	2874
Court	727	101	109	1123	209	202	1305	217	205
EffectiveDate	468	83	59	786	148	196	1179	188	236
PII_Ref	41	20	11	270	40	57	445	96	72
Parties	2764	344	262	3747	895	711	5627	1145	833
Percentage	446	42	20	518	89	81	550	94	84
Price	23	1	2	96	19	13	96	22	14
Principal	-	-	-	150	21	24	244	31	47
Ratio	26	4	4	92	16	13	151	21	13
Regulation	1085	144	104	1212	153	143	1484	181	188
RenewalTerm	120	20	8	120	20	8	120	20	8
Rent	-	-	-	-	-	-	32	4	6
Role	1534	135	114	1534	135	114	1756	148	126
Salary	288	33	15	288	33	15	317	35	17
Shares	61	7	7	104	19	13	108	22	16
TerminationDate	233	34	25	250	40	28	302	48	34
Title	1054	92	95	1439	256	183	2293	338	240
O	227962	27836	21804	360225	63879	73294	495693	87075	86334

Table 4.3 Label-wise data distribution statistics

through the strategic use of prompts.

Our findings underscore the robustness of instruction-based models, emphasizing their adaptability and performance in scenarios with sparse data and novel contract categories. This versatility enhances the applicability of such models in real-world settings where access to extensive training data is often challenging. Notably, the instruction-based T5-small model trained on the entire dataset achieved a higher recall value compared to other baseline models, further highlighting its desirable feature of improved recall.

4.1.5 Conclusion

Our exploration of named entity taggers like Spacy, legal entity taggers such as LexNLP, and few-shot instruction models like ChatGPT revealed their valuable capabilities, but also brought to light significant limitations. One prominent drawback was the lack of fine-grained classification in existing models. Currency terms and dates, for instance, often received broad categorizations without specific distinctions, hindering precise information extraction. Moreover, the zero-shot and few-shot learning capabilities of GPT models proved insufficient, necessitating further fine-tuning on task-specific data. To address these issues, our paper focused on providing fine-grained classification for general entities

	Token Classification (LegalBERT)			RoBERTa-Large + LoRa			T5-small Instruction Model + LoRa		
Entity Name	**precision**	**recall**	**f1-score**	**precision**	**recall**	**f1-score**	**precision**	**recall**	**f1-score**
Act	0.50	0.64	0.56	0.32	0.18	0.23	0.83	**1.00**	0.91
Address	0.35	0.33	0.34	0.19	0.17	0.18	1.00	**0.67**	0.80
Court	0.64	0.70	0.67	0.42	0.44	0.43	1.00	**1.00**	1.00
EffectiveDate	0.71	0.72	0.72	0.62	0.53	0.57	0.94	**0.94**	0.94
PII_Ref	1.00	**1.00**	1.00	0.08	0.06	0.07	1.00	**1.00**	1.00
Parties	0.43	0.64	0.52	0.24	0.18	0.21	1.00	**0.74**	0.85
Percentage	0.81	0.73	0.77	0.72	0.81	0.76	0.78	**1.00**	0.88
Price	0.93	**1.00**	0.96	0.90	0.45	0.60	1.00	0.80	0.89
Principal	0.41	0.44	0.43	0.12	0.12	0.12	0.80	**0.67**	0.73
Ratio	0.50	0.62	0.56	0.30	0.50	0.37	0.25	**0.67**	0.36
Regulation	0.60	**0.88**	0.71	0.21	0.22	0.22	0.66	0.75	0.70
RenewalTerm	0.50	**0.50**	0.50	0.29	0.29	0.29	0.75	0.43	0.54
Rent	-	-	-	-	-	-	0.50	**1.00**	0.67
Role	0.66	0.76	0.70	0.8	0.88	0.83	0.33	**1.00**	0.50
Salary	0.52	**0.88**	0.65	0.24	0.23	0.24	0.67	0.67	0.67
Shares	-	-	-	0.39	0.63	0.48	1.00	**1.00**	1.00
TerminationDate	0.71	**0.92**	0.80	0.44	0.58	0.50	1.00	0.67	0.80
Title	0.68	0.79	0.73	0.37	0.22	0.28	1.00	**0.84**	0.91

Table 4.4 Model Comparisions on Overall Test Dataset (Stage-3 Dataset).

like amounts and dates, while acknowledging the need for continuous improvement and adaptation.

Looking ahead, our commitment extends to exploring fine-grained classification for additional entities, such as percentages, and expanding the scope of contract categories addressed in our research. By openly sharing our fine-tuned instruction-based models and dataset, we aim to contribute to the advancement of entity extraction from contracts. We aspire to inspire further research and improvements, fostering a more comprehensive and practical approach to entity extraction in the ever-evolving landscape of legal documents.

4.2 Large Language Models as Evaluators/Raters

The recent surge in NLP research, propelled by the introduction of APIs for LLMs such as ChatGPT and the open-source availability of models like LLaMA variants, has led to the development of LLM-based metrics [6]. Examples include GEMBA [21], which explores the use of prompts with ChatGPT and GPT4 directly as metrics, and Instructscore [55], which fine-tunes an LLaMA model for detailed error diagnosis in machine-translated content.

However, the current research landscape lacks a systematic evaluation of potential prompts and prompting techniques for metric usage, encompassing instructing a model or having the model explain a task independently. Additionally, there is a scarcity of assessments regarding the performance of recent open-source LLMs, despite their pivotal role in enhancing the reproducibility of metric research compared to closed-source alternatives.

To address these gaps, this work leverages open-source, pre-trained LLMs provided by the Eval4NLP shared task [23] for assessing machine translations and summaries. We focus on prompting techniques without LLM fine-tuning, aiming to improve alignment with human evaluations and enhance metric interpretability while identifying promising models for future fine-tuning.

Among the provided LLMs, orca_mini_v3_7b was selected due to its smaller size, accommodating resource constraints. Challenges were encountered when attempting to load other LLMs. Prompts were curated using a blend of fine-grained and chain-of-thought prompting strategies. Additionally, utilizing bitsandbytes[6], 4-bit quantization was employed to enhance model loading efficiency, considering MAX TOKENS as 512 during inference.

This work contributes summary-level quality scores for all documents in the task and segment-level quality scores for language pairs (en-de, en-zh, en-es) in the MT or Summarization evaluation task, without relying on references. Scores range from 0-100, where 0 signifies the lowest score for a poor translation/summary, and 100 represents the highest score for a perfect translation/summary (codebase is available at `https://github.com/pavanbaswani/Eval4NLP_SharedTask`).

4.2.1 Model Description

Table 4.5 illustrates the provided test sample statistics. The reported token counts were computed using bert tokenizer[7].

		# Entries	min tokens	max tokens	average tokens
summarization	*source (en)*	825	144	818	279.413
	target (en)		9	402	51.697
en_de	*source (en)*	1425	18	137	37.935
	target (de)		17	156	41.297
en_es	*source (en)*	1834	15	137	37.472
	target (es)		19	149	41.683
en_zh	*source (en)*	1297	18	137	37.856
	target (zh)		21	212	51.436

Table 4.5 Test Data Statistics

4.2.2 Our Prompting Strategies

We outline our prompting strategies for this shared task as follows.

[6]https://huggingface.co/blog/4bit-transformers-bitsandbytes#advanced-usage
[7]https://huggingface.co/bert-base-multilingual-cased

Instruction
The task is to provide the overall score for the given summary with reference to the given article on a continuous scale from 0 to 10
along with explanation in JSON format with "score" and "explanation" keys as follows: {"score": <float-value>, "explanation": <explanation-text>}.
Where a score of 0 means the summary is "irrelevant, factually incorrect and not readable" and score of 10 means "relevant, factually correct, good readability".
You must justify the score that you provided with clear and concise reason within 2 sentences interms of justifying the relevance, readability, factuality metrics.
The article text and summary text is given in triple backticks ‘‘‘ with ### Article: and ### Summary: as prefix respectively.
Note: The generated response must be in json format without any missed braces or incomplete text. Also, it should not provide any additional information other than JSON output.

Article: ‘‘‘{}‘‘‘
Summary: ‘‘‘{}‘‘‘
Response:

Table 4.6 Zero-shot prompting for evaluating Summary

Instruction:
The task is to score a translated text from {English} to {German} with respect to the source sentence on a continous scale from 0 to 100,
along with explaination in JSON format with "score" and "explanation" keys as follows: {"score": <float-value>, "explanation": <explanation-text>}.
Where a score of zero means "no meaning preserved and poor translation quality" and score of one hundred means "excellant translation quality with perfect meaning and grammar".
You must justify the score that you provided with clear and concise reason within 2 sentences interms of justifying the adequacy, fluency, faithfulness metrics.
The source sentence and target sentence is given in triple backticks with ### source sentence: and ### target sentence: as prefix respectively.
Note: The generated response must be in json format without any missed braces or incomplete text. Also, it should not provide any additional information other than JSON output.

source sentence: ‘‘‘{}‘‘‘
target sentence: ‘‘‘{}‘‘‘
Response:

Table 4.7 Zero-shot prompting for evaluating MT

4.2.2.1 Approach-1 (Zero-shot W/o explanation)

"Zero-shot prompting without explanation" means prompting the LLM to generate a response without providing any additional information or context to clarify or support the prompt. It relies solely on the initial instruction without further elaboration.

4.2.2.2 Approach-2 (Zero-shot w/ explanation)

"Zero-shot prompting with explanation" involves providing a prompt or instruction to a system and supplementing it with additional information or context to clarify or support the prompt (refer Table 4.6 & 4.7). This approach aims to enhance the system's understanding of the task or request by offering more details or background information alongside the initial instruction.

4.2.2.3 Approach-3 (CoT + Fine-grained w/ explanation)

To facilitate a deeper understanding and enhance the LLM's ability to provide improved responses, we incorporate a strategic approach involving a combination of chain of thought (CoT) prompting and fine-grained analysis. Specifically, we focus on the aspects of Relevance, Consistency, Coherence, and Fluency for Summarization, and emphasize Adequacy, Faithfulness, and Fluency for Machine Translation (MT).

- **Fine-grained Analysis for Summarization:** Firstly, the LLM is instructed to provide individual scores for Relevance, Consistency, Coherence, and Fluency. These individual scores are then used to prompt the model to provide a final overall summary score, ensuring a comprehensive

assessment of the summarization quality (refer Table 4.8). This approach enables a more detailed and nuanced evaluation of the summary's performance in each aspect.

- **Fine-grained Analysis for MT:** Initially, the LLM generates separate scores for Adequacy, Faithfulness, and Fluency. Subsequently, using these scores, the model is prompted to produce a final translation quality score, ensuring a comprehensive evaluation of the translation's performance in each dimension (refer Table 4.9). This approach enhances our ability to assess translation quality thoroughly.

```
### Instruction
You will be given one summary written for a news article.

Your task is to assign the single score for the summary on continuous scale from 0 to 10 along with explanation.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing,
and refer to it as needed. You must justify the score that you provided with clear and concise reason within 2 sentences in
terms of justifying the relevance, fluency, coherence and consistency metrics.

The article text and summary text is given in triple backticks “‘ with ”Source Text:” and ”Summary:” as prefix respectively.

Evaluation Criteria:
1) Relevance (1-5) - selection of important content from the source. The summary should include only important information
from the source document. Annotators were instructed to penalize summaries which contained redundancies and excess information.
Here, 1 is the lowest and 5 is the highest.
2) Consistency (1-5) - the factual alignment between the summary and the summarized source. A factually consistent summary
contains only statements that are entailed by the source document. Annotators were also asked to penalize summaries that contained
hallucinated facts. Here, 1 is the lowest and 5 is the highest
3) Coherence (1-5) - the collective quality of all sentences. We align this dimension with the DUC quality question of structure and
coherence whereby ”the summary should be well-structured and well-organized. The summary should not just be a heap of related
information, but should build from sentence to a coherent body of information about a topic.”. Here, 1 is the lowest and 5 is the highest.
4) Fluency (1-3): the quality of the summary in terms of grammar, spelling, punctuation, word choice, and sentence structure.
- 1: Poor. The summary has many errors that make it hard to understand or sound unnatural.
- 2: Fair. The summary has some errors that affect the clarity or smoothness of the text, but the main points are still comprehensible.
- 3: Good. The summary has few or no errors and is easy to read and follow.

Evaluation Steps:
1. Read the summary and the source document carefully.
2. Compare the summary to the source document and identify the main points of the article.
3. Assign scores for Relevance, Consistency, Coherence and Fluency based on the Evaluation Criteria.
4. By utilizing the generated scores of Relevance, Readability, Coherence and Fluency, aggregate these scores to assign the single score
for the summary on continuous scale from 0 to 10 along with explanation in JSON format with ”score” and ”explanation” keys as follows:
{”score”: <float-value>, ”explanation”: <explanation-text>}.

### Source Text: “‘{}“‘
### Summary: “‘{}“‘
### Response:
```

Table 4.8 CoT + fine-grained prompting for evaluating summaries

4.2.3 Results

Table 4.10 depicts the summary-level Kendall correlation scores for the summarization evaluation task. Our submission (LTRC) ranks 4th, with a slight difference of 0.06 compared to the top submission. Initially utilizing zero-shot prompting resulted in a leaderboard correlation of 0.41. After employing

Instruction
You will be given one translated sentence in {Spanish} for a source sentence in {English}.

Your task is to assign the single score for the translation on continuous scale from 0 to 100 along with explanation.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed. For explanation, you must justify the score that you provided with clear and concise reason within 2 sentences interms of justifying the adequacy, fluency and faithfulness metrics.

The source text and translation text is given in triple backticks "' with "Source Text:" and "Translation:" as prefix respectively.

Evaluation Criteria:
1) Adequacy (1-5) - the correspondence of the target text to the source text, including the expressive means in translation. Annotators were instructed to penalize translation which contained misinformation, redundancies and excess information. Here, 1 is the lowest and 5 is the highest.
2) Faithfulness (1-5) - translation faithfulness to the meaning depends on how the translator interprets the speaker's intention and does not imply that one should never or always translate literally. Here, 1 is the lowest and 5 is the highest.
3) Fluency (1-3): the quality of the translation in terms of grammar, spelling, punctuation, word choice, and sentence structure.
- 1: Poor. The translation has many errors that make it hard to understand or sound unnatural.
- 2: Fair. The translation has some errors that affect the clarity or smoothness of the text, but the main points are still comprehensible.
- 3: Good. The translation has few or no errors and is easy to read and follow.

Evaluation Steps:
1. Read the translation and the source document carefully.
2. Compare the translation to the source text.
3. Assign scores for Adequacy, Faithfulness and Fluency based on the Evaluation Criteria.
4. By utilizing the generated scores of Adequacy, Faithfulness and Fluency, aggregate these scores to assign the single score for the translation on continuous scale from 0 to 100 along with explanation in JSON format with "score" and "explanation" keys as follows: {"score": <float-value>, "explanation": <explanation-text>}.

Source Text: "'{}"'
Translation: "'{}"'
Response:

Table 4.9 CoT + fine-grained prompting for evaluating MT

CoT + Fine-grained prompting, the Kendall correlation improved to 0.44. This indicates a positive impact on system performance with strategic prompting.

Tables 4.11, 4.12, and 4.13 illustrate segment-level Kendall correlations for machine translation (MT) on en-de, en-zh, and en-es language pairs, respectively. Our submissions consistently rank 2nd (in the small models track) across language pairs. For the en-de language pair, zero-shot prompting yielded a correlation of 0.11, significantly improving to 0.19 with CoT + Fine-grained prompting. Conversely, for en-zh, the correlation score dropped to 0.09 with CoT + Fine-grained prompting. Hence, we submitted with zero-shot prompting for en-zh and en-es. An intriguing observation is that our submissions have outperformed most submissions in the large model track, except NLLG for en-de and en-es, and MysteryTest for en-es.

4.2.4 Error Analysis

In our manual analysis of English-German MT samples, we uncovered a minor scoring issue related to language compatibility, as orca_mini_v3_7b was initially trained on English text. The provided examples in Table 4.14 illustrate this issue. Notably, the zero-shot prompting strategy yielded high scores

Track	Team Name	Summ
Small	DSBA	0.5
	iML	0.49
	IUST_NLP_Lab	0.48
	LTRC	0.44
	CompetitionEntrants	0.44
	Beginners	0.38
	ManCity	0.25
Large	NLLG	0.35

Table 4.10 Summary-level Kendall Correlation for Summarization Task

Track	Team Name	en-de
Small	HIT-MI&T Lab	0.49
	LTRC	0.19
	uOttawa	0.12
	TaiwanSenior	0.04
Large	NLLG	0.24
	MysteryTest	0.17
	Eval4NLP	0

Table 4.11 Segment-level Kendall Correlation for MT on English-German pairs.

but overlooked translation accuracy and generated inaccurate explanations in certain cases. Conversely, CoT + fine-grained prompting penalized accuracy in the first example and provided an incorrect assessment of text fluency in its explanation. This highlights the necessity for a nuanced evaluation approach, considering not only final scores but also the accuracy and reliability of the model's explanations.

4.2.5 Conclusion

This paper outlines our contribution to the Eval4NLP shared task, specifically tailored to the small model track. Utilizing the "orca_mini_v3_7b" model, we evaluated machine translation (MT) and summarization test data. Employing both zero-shot and fine-grained + CoT prompting strategies, we generated scores accompanied by explanations. Our submission ranked 2nd in the machine translation task and 4th in the summarization task. Our findings suggest that further exploration of strategic prompting methods has the potential to enhance both scoring and explainability in large language models.

Track	Team Name	en-zh
Small	HIT-MI&T Lab	0.32
	LTRC	0.13
Large	NLLG	0.13
	MysteryTest	0.1
	Eval4NLP	0.01

Table 4.12 Segment-level Kendall Correlation for MT on English-Chinese pairs.

Track	Team Name	en-es
Small	HIT-MI&T Lab	0.42
	LTRC	0.11
Large	NLLG	0.18
	MysteryTest	0.12
	Eval4NLP	-0.02

Table 4.13 Segment-level Kendall Correlation for MT on English-Spanish pairs.

		Zero-Shot Prompting		CoT + Fine-grained Prompting		Human Evaluation	
English (en)	**German (de)**	**Score**	**Explanation**	**Score**	**Explanation**	**Score**	**Explanation**
Samson Lawal (born 25 April 2004) is a Nigerian professional footballer who plays as a midfielder for Livingston.	Samson Lawal (* 25. April 2004 in Livingston) ist ein nigerianischer Fußballspieler.	90	The translation is adequate, with the main information about the player and his birthdate preserved. However, the sentence structure is not as fluent as it could be, and the use of the German word "ist" instead of "ist in" could be improved.	70	The translation is adequate as it conveys the main information from the source text, including the birthdate, nationality, and position of the footballer. However, the sentence structure is not as fluent as it could be, and there are some minor grammatical errors.	60	The translation misses some information like \textit{Samson Lawal plays as a midfielder for Livingston} and instead mentions Livingston as his birthplace.
Andrii Ivanovych Dubchak (born 19 October 1976, Kalynivka, Vinnytsia Oblast) is a Ukrainian photographer, videographer, photojournalist, and war correspondent.	Andrii Ivanovych Dubchak (* 19. Oktober 1976 in Kalynivka, Vinnytsia Oblast) ist ein ukrainischer Fotograf, Videograf, Fotojournalist und Kriegskorrespondent.	90	The translation is very fluent and maintains the meaning of the original sentence. However, there are some minor grammatical errors, such as the use of 'ist' instead of 'ist ein' for the first occurrence of 'Andrii Ivanovych Dubchak' and the missing comma after 'Kalynivka, Vinnytsia Oblast).	80	The translation is adequate, as it conveys the main information about Andrii Ivanovych Dubchak's profession and birthplace. The fluency is good, with no major grammatical or spelling errors. However, the faithfulness could be improved, as the sentence structure is not entirely faithful to the original source text.	95	hethetheThe translation is accurate and preserves source meaning. The only minor issue is that letter 'U' should be capitalized in \textit{ukrainischer}.

Table 4.14 Analysis on en-de MT pairs.

Chapter 5

Annotation/Evaluation Tools

In the process of crafting datasets of exceptional quality, human annotation emerges as a pivotal factor. This section meticulously explores the landscape of human annotation, shedding light on both academic collaborations and specific annotation tools that play a substantial role in augmenting the quality and diversity of annotated data. These collaborative endeavors and specialized tools have demonstrated their significance in elevating the overall standard and variety of annotated data, meeting the varied requirements of diverse natural language processing applications. Through a thorough examination of these collaborative initiatives and annotation tools, we unveil the intricate methodologies employed in the annotation process, emphasizing their profound impact on the broader domain of NLP research and applications.

5.1 Academic Collaborations on Annotation Tools

5.1.1 Semantic Textual Relatedness (STR)

The intricate interplay of semantic relatedness between language units serves as a cornerstone in comprehending meaning within textual content, a notion well-established in linguistic studies [14, 32]. Automatic determination of semantic relatedness has proven instrumental across diverse applications, spanning sentence representation evaluation, question answering, and summarization [1].

The Semantic Textual Relatedness (STR) [1] shared-task focuses on predicting the degree of relatedness between pairs of sentences. The assessment considers sentences to be semantically similar if they exhibit paraphrasal or entailment relations. However, the scope of relatedness extends beyond these specific relationships, encapsulating broader aspects such as shared topics, aligned perspectives, temporal concordance, and causal relationships.

The STR dataset, comprising instances in training, development, and test sets, features sentence pairs

[1] https://semantic-textual-relatedness.github.io/

labeled with scores indicative of their semantic textual relatedness. These scores range from 0, representing maximal unrelatedness, to 1, signifying maximal relatedness. Manual annotation, employing a comparative approach, was adopted to derive these gold label scores, mitigating biases associated with traditional rating scale annotation methods. This comparative annotation strategy contributes to the high reliability of the final relatedness rankings. This subsection particularly delves into the data creation process for Telugu, Hindi, and Marathi languages, detailing the design and development of an annotation interface tailored for Semantic Textual Relatedness.

In the pursuit of refining semantic relatedness assessments between sentence pairs, a dedicated Semantic Text Relatedness (STR) Annotation Tool has been designed and developed. This versatile tool caters to four distinctive user roles, ensuring a systematic and collaborative approach to the annotation process. It also incorporates intrinsic evaluation metrics, including golden samples, enhancing the reliability and quality of annotations. Coordinators can track annotator performance, identifying instances where golden samples were incorrectly marked, thereby ensuring ongoing quality control.

1. **Annotator:** Annotators in the STR tool are pivotal contributors responsible for evaluating sentence pairs. Their primary role involves selecting the most and least related pairs from the annotation page, directly influencing the creation of a robust dataset for semantic relatedness.

2. **Task Coordinator:** Coordinators play a crucial role in managing the annotation process. Their responsibilities encompass overseeing annotators, user management, and maintaining file assignments. By ensuring organizational efficiency, coordinators contribute to the smooth execution of the annotation task.

3. **Project Manager:** Project managers hold a supervisory position, providing oversight to coordinators and annotators. With access to annotation statistics, they contribute to the strategic direction of the annotation efforts, ensuring alignment with project goals.

4. **Admin:** Admins wield administrative control over the tool, managing functionalities, user roles, and configurations. Their contribution extends to maintaining the overall stability and functionality of the tool, addressing both technical and administrative aspects.

This tool encompasses various pages designed to cater to specific roles and tasks, ensuring a well-organized and productive workflow. Each page serves a unique purpose in managing annotation tasks, tracking progress, and maintaining user integrity. Let's delve into a brief overview of the key pages that constitute this robust annotation tool.

Tasks View: This page serves as a centralized hub visible to coordinators, providing a comprehensive overview of the annotation progress (refer Figure 5.1). Coordinators can monitor the completion status of assigned files, download completed samples, and assess annotator performance. This page

also facilitates quality control by highlighting instances where annotators may have misjudged golden samples, ensuring the reliability of the annotations.

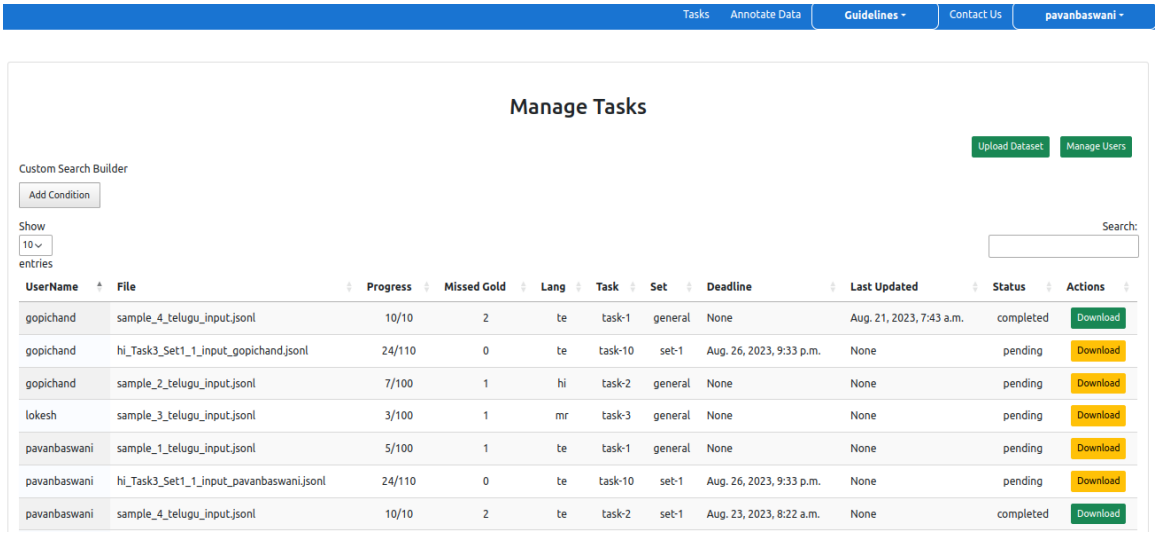

Figure 5.1 STR: Tasks View

Annotation View: Accessible to all users, who assigned with annotation files. This view offers a quick snapshot of the ongoing annotation status as shown in Figure 5.2. Completed files are visually indicated with a green marker, and the status of each file is clearly displayed. This page acts as a real-time dashboard, keeping all stakeholders informed about the progress of the annotation task.

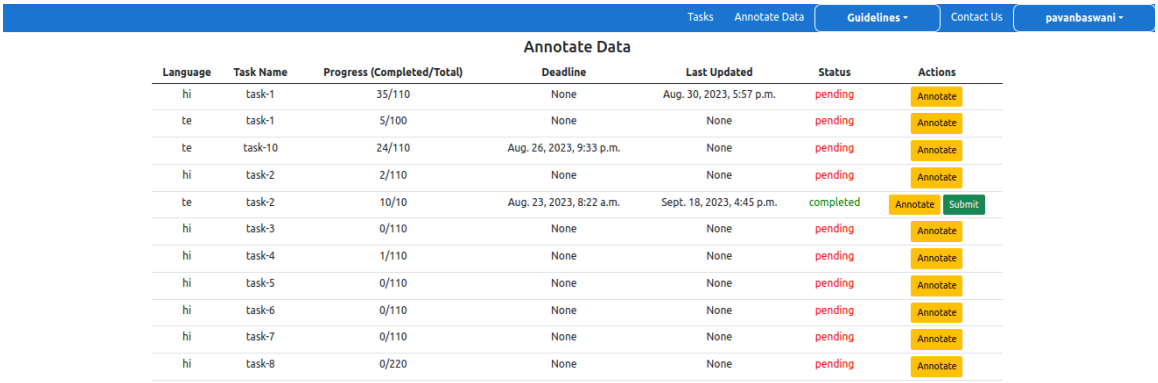

Figure 5.2 STR: Annotation View

Data Annotation View: The *Data Annotation Page* view is the core interface where annotators engage with the STR task. This user-friendly interface presents samples for annotation, allowing annotators to select the most and least related pairs. It streamlines the annotation process, ensuring annotators can focus on the task at hand while maintaining accuracy and consistency in their judgments.

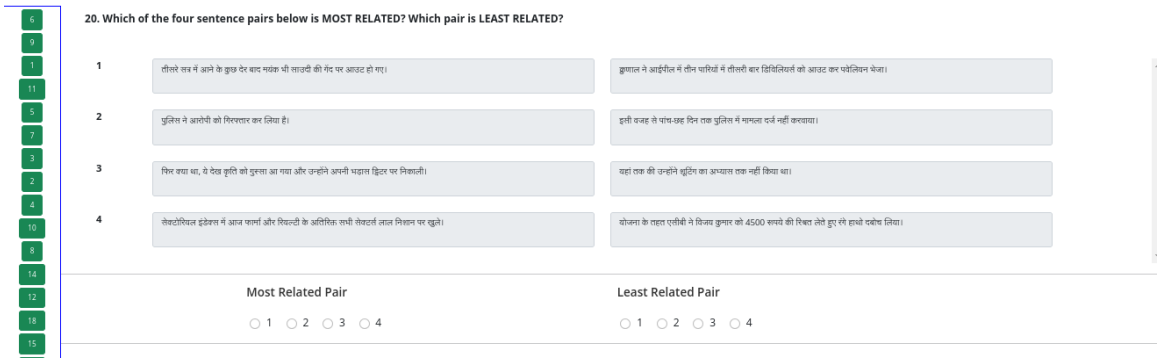

Figure 5.3 STR: Data Annotation Page View

Manage Users: Designed for coordinators, project managers, and admins, the *Manage Users* page provides a centralized space to oversee user management activities (refer Figure 5.4). Coordinators can activate or deactivate users based on their activity, ensuring a dynamic and responsive user ecosystem. This page enhances the administrative efficiency of the annotation tool.

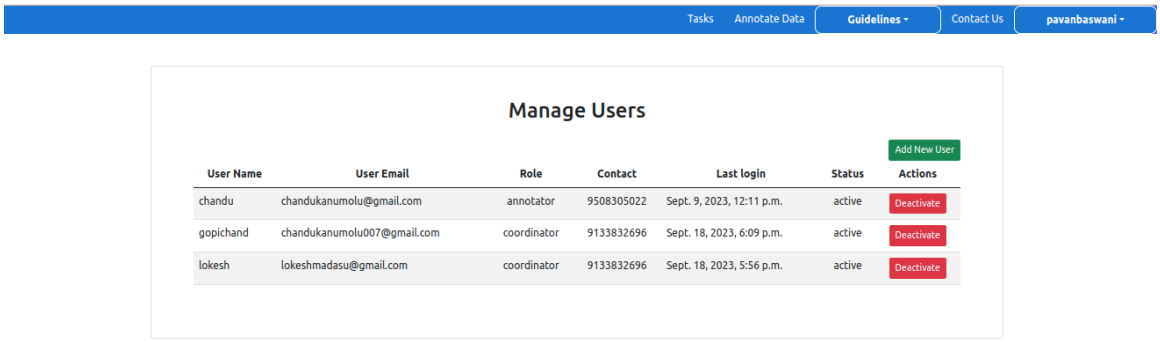

Figure 5.4 STR: Manage Users Page View

This tool represents a significant advancement in facilitating the annotation process for semantic textual relatedness tasks. The introduction of four distinct user roles – annotator, coordinator, project manager, and admin – contributes to a well-organized and collaborative environment. The tool's multi-faceted design includes key features like the Tasks page, Annotation View, Annotation Page view, and Manage Users, each catering to specific needs in the annotation workflow. The Tasks page allows coordinators to monitor progress and download completed samples, while the Annotation View provides a quick overview of file annotation statuses. The Annotation Page view serves as the user interface for actual annotation tasks, and the Manage Users page enables efficient user management. Also, the integration of intrinsic evaluation metrics, such as tracking annotators' handling of golden samples, enhances the tool's quality control mechanism.

5.1.2 BBMTE: Machine Translation Evaluation

The BBMTE - Machine Translation Evaluation tool is a dedicated platform designed for evaluating machine translations by assessing both source and translated texts. The primary objective is to identify and rectify any inaccuracies or improvements required in the translations generated by state-of-the-art machine translation systems. Much like the Semantic Text Relatedness (STR) tool, this tool incorporates a user-friendly interface and diverse functionalities to enhance the efficiency and collaboration of annotators.

This tool encompasses four key user roles, namely annotator, coordinator, project manager, and admin, each assigned specific responsibilities to streamline the evaluation process. The Tasks page, accessible to coordinators, facilitates progress monitoring and the downloading of completed samples. Figure 5.5 depicts the tasks view of the annotations.

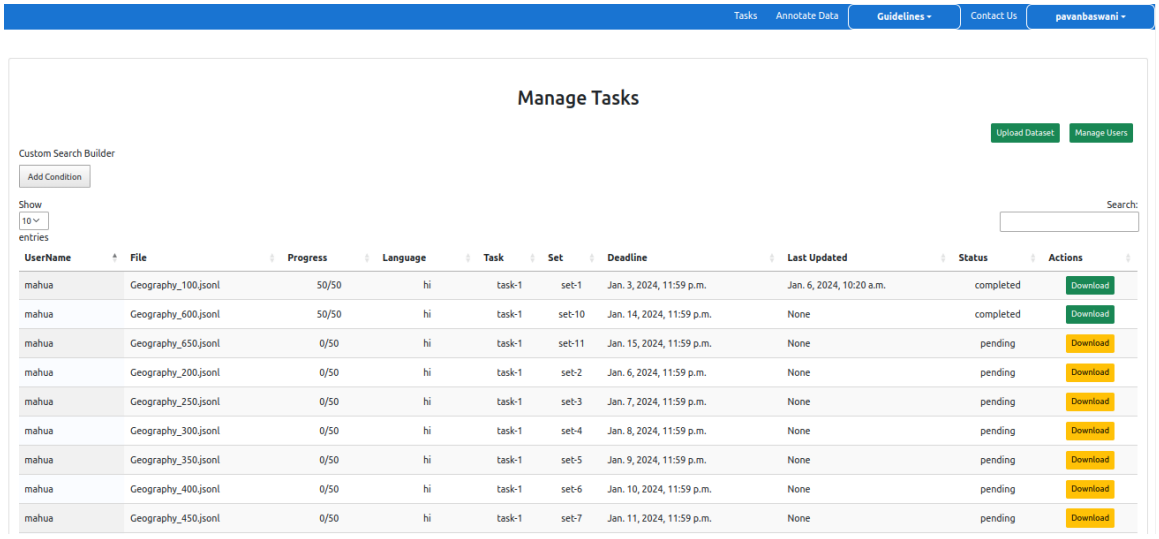

Figure 5.5 BBMTE: Task Page View

The Annotation View (refer Figure 5.6) provides an overview of file annotation statuses, while the Annotation Page view (refer Figure 5.7) serves as the interface for actual evaluation tasks. Additionally, the Manage Users page is available for coordinators, project managers, and admins to efficiently handle user management aspects.

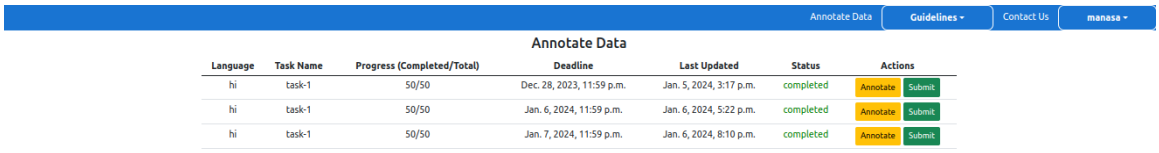

Figure 5.6 BBMTE: Annotation View

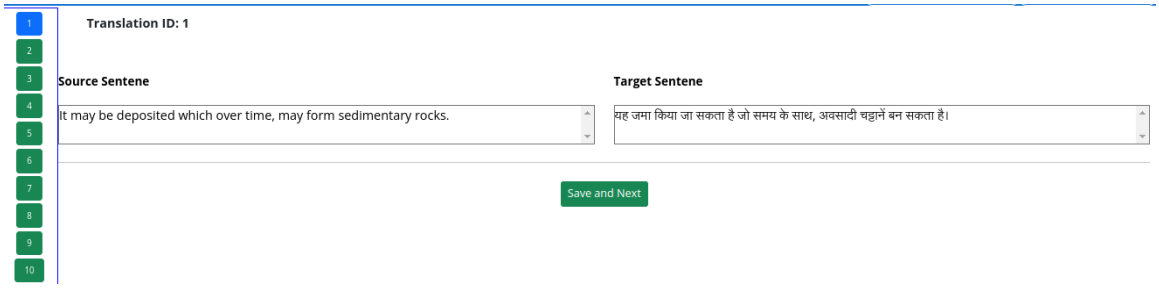

Figure 5.7 BBMTE: Data Annotation Page View

In the following section, we provide a brief overview of the human annotation tools designed and developed specifically for NLP tasks.

5.2 Human Annotations Tools

The creation and evaluation of high-quality datasets play a pivotal role in advancing research and model performance. This section introduces a suite of carefully designed annotation tools tailored to specific NLP tasks. These tools encompass diverse applications, ranging from abstractive summarization and headline classification to question-answering, paraphrasing, and contract Named Entity Recognition (NER) tagging. Each tool is meticulously crafted to facilitate efficient and accurate annotation processes, harnessing the power of human annotators and, in some cases, integrating Large Language Models (LLMs) as pre-annotators or evaluators. The following sub-sections delve into the unique characteristics, functionalities, and impacts of each annotation tool, highlighting their contributions to the development and evaluation of datasets.

5.2.1 Abstractive Summarization

Abstractive Summarization [48], a pivotal aspect of natural language processing (NLP), involves the creation of concise and coherent summaries that convey the essential information of a given text while potentially introducing novel expressions. In the realm of annotation tools, the development of a robust Abstractive Summarization system is paramount for generating informative and succinct summaries across diverse content domains. This section delves into the intricacies of the Abstractive Summarization Annotation Tool, elucidating its design, functionalities, and contributions to the broader landscape of NLP applications. Through a meticulous annotation process, this tool aims to enhance the efficiency and accuracy of abstractive summarization, catering to the evolving demands of information extraction and comprehension in varied textual contexts.

5.2.1.1 Annotation

The Summarization Annotation Tool (refer Fig.5.8) follows a comprehensive five-step process to enhance and refine article contents. Each step contributes to the creation of a concise, meaningful summary, ultimately producing a well-structured output.

1. **Content Review and Modification:** In the initial step of the Summarization Annotation Tool, the article undergoes a meticulous review to identify and eliminate noisy or irrelevant content. This process aims to enhance the overall clarity and relevance of the article, ensuring that the subsequent summarization is based on a refined foundation.

2. **Article Sentencification:** Following the content review, the modified article content is structured into individual sentences. Each sentence is placed on a new line, facilitating improved readability and providing a foundation for the subsequent steps in the summarization process. This segmentation ensures that the article's content is organized in a way that aligns with natural language structure.

3. **Writing Relevant & Concise Summary:** The summarization process involves crafting a concise summary of the modified article content. This step incorporates two key metrics to assess the quality of the summary: Compression Ratio and Abstractivity. The Compression Ratio is calculated by evaluating the token count of the summary relative to the article, adhering to a predetermined threshold. Simultaneously, the Abstractivity metric ensures that the summary captures essential information while maintaining a balanced ratio, contributing to an informative yet succinct summary.

4. **Summary Sentencification:** After generating the summary, the text is segmented into individual sentences. Similar to the treatment of the article content, each sentence of the summary is placed on a new line. This step enhances the coherence and readability of the summary, preparing it for further analysis or presentation.

5. **Title Generation:** The final step involves generating a relevant and concise title for the modified article content. This process leverages key content points extracted from the modified article to construct a title that encapsulates the essence of the information. The title serves as a succinct representation of the content, providing readers with a quick understanding of the article's focus and key takeaways.

Impact of Annotation Tool: This tool significantly influences the efficiency and quality of the manual annotation process. The introduction of this tool, along with the incorporation of relevant metrics, has brought about notable improvements, as reflected in the provided Table. 5.1. This tool led to a reduction in the number of annotators required, enhanced efficiency in evaluating longer sentences, and an overall increase in the percentage of high-quality data, culminating in a more streamlined and effective annotation process (refer Figure. 5.9).

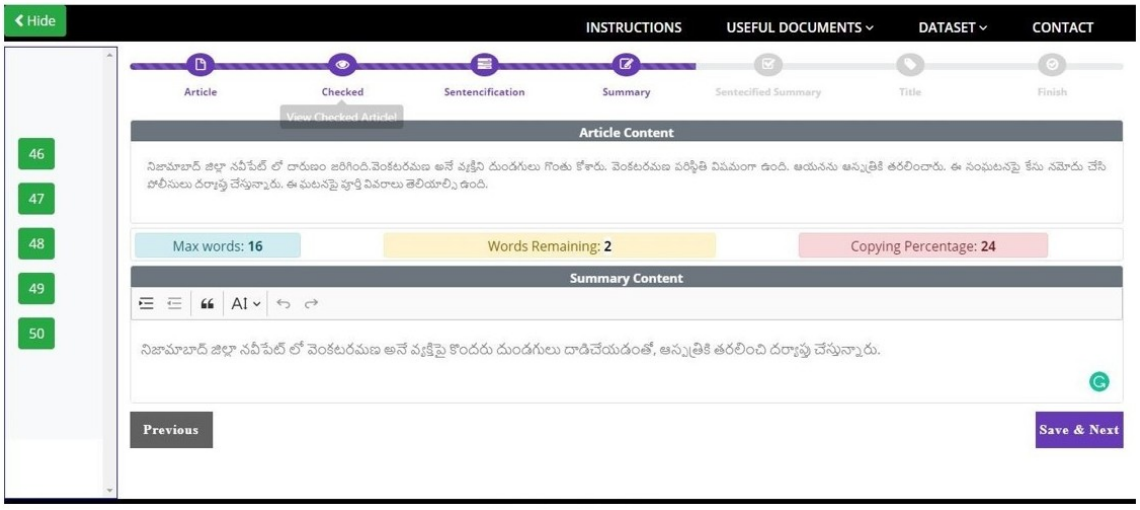

Figure 5.8 Abstractive Summarization Annotation Tool

From Figure 5.9 it is evident that even when the article's length increases (sentence ranges from 6 to 9), most annotators managed to finish the task in approximately similar duration (10 to 13 hours). But on the other hand, we obtained only 22.9% of quality data. To increase the percentage of quality data, we integrated the intrinsic evaluation metrics in both interfaces. As a result, we have obtained 61.66% quality data and the majority of the annotators expressed that the complexity of the task is moderate. However, most of the annotators had to spent 15+ hours to finish the task, due to an increase in the number of sentences in the articles ranging from 10 to 17. We also observed that more than 70% of annotators preferred to use Google input tools offline to type the Telugu text while creating the summary. Table 5.1 also presents the average minimum and maximum time consumption for random evaluation of 12-16 samples in a set of 50 samples and with the corresponding feedback.

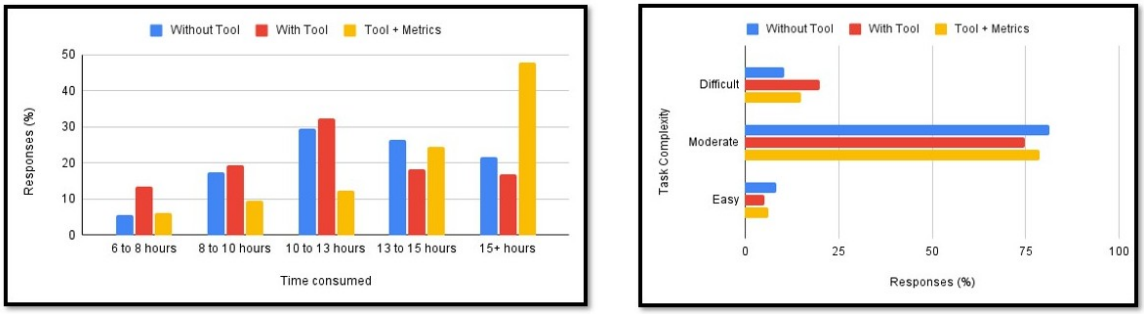

Figure 5.9 End-User Feedback (Manual vs Interface)

5.2.1.2 Evaluation

The manual evaluation process using Evaluation tool involves a carefully designed methodology to assess the quality of generated summaries. After successfully passing the intrinsic evaluations focusing on compression ratio and abstractivity, the selected samples undergo a manual assessment by human evaluators/raters. The process follows these key steps:

1. **Intrinsic Evaluation:** A curated set of samples, meeting the intrinsic evaluation criteria, is selected for manual evaluation. These samples represent diverse content and linguistic complexities.

2. **Manual Evaluation:** Each evaluator/rater assigns scores ranging from 0 to 4 for each metric, where 0 represents poor quality, and 4 denotes a perfect score. The three metrics, Relevance, Readability, and Creativity, are individually assessed for every sample. The scores of a set of samples (30 samples) are aggregated to derive a collective evaluation for each metric, ensuring a comprehensive and diversified perspective.

3. **Pruning Low Quality Sets:** Based on the aggregated score, the group will be selected/rejected.

To ease the manual evaluation process, the Evaluation tool (refer Figure. 5.10) carefully designed and developed. Also, conducted the survey on this tool with the evaluators/raters and tabulated in Table 5.1.

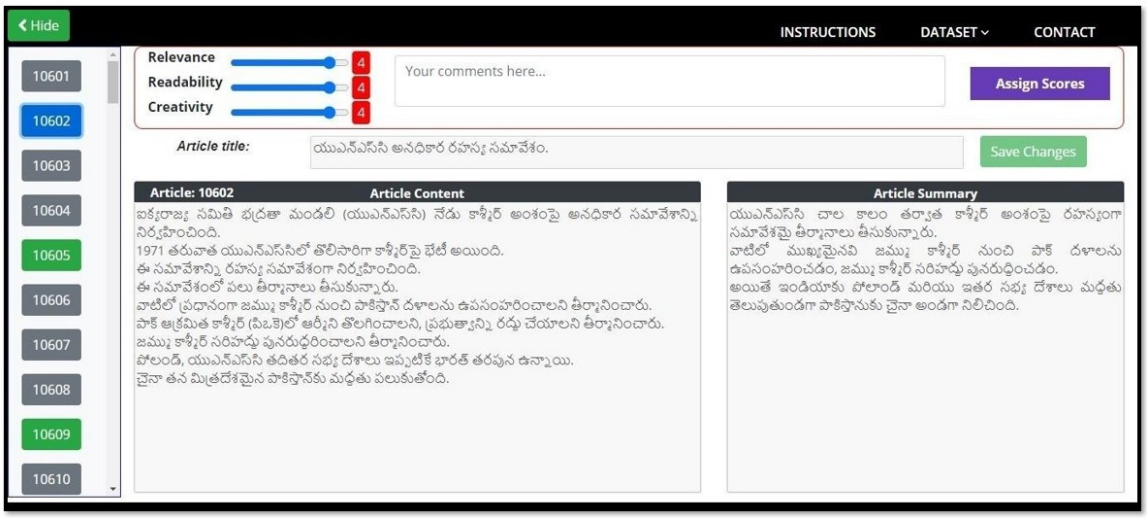

Figure 5.10 Abstractive Summarization Evaluation Tool

	#Annotators	#Sentences	Evaluation Time (m)	#Samples Collected	Quality Data (%)
Without Tool	110	3 - 6	53.8 - 75	30000	39.67
With Tool	120	6 - 9	52.5 - 67.5	40000	22.91
Tool + Metrics	117	10+	60 - 102.5	13370	61.66

Table 5.1 Summarization Data Quality

5.2.2 HeadlineClassication

Headline classification is a crucial aspect of information retrieval, guiding readers to comprehend the core themes of news articles efficiently. The development of the Headline Classification tool involves a systematic approach to efficiently categorize headlines based on their alignment with the content of the corresponding articles. The task is to classify each headline into one of the predefined classes, including Factual main event, Factual secondary event, Strong conclusion, Weak conclusion, Misleading conclusion, Unsupported opinion, Irrelevant, and Inconclusive.

In this annotation tool, headlines are categorized into eight distinct classes, each representing a different facet of headline content. These classes range from accurately portraying the primary event to presenting misleading conclusions, expressing unsupported opinions, or even being entirely irrelevant.

1. **Factual Main Event:** Headlines accurately represent the primary and most significant event covered in the corresponding article.

2. **Factual Secondary Event:** Headlines depict a secondary event covered in the article, providing additional information but not the main focus.

3. **Strong Conclusion:** Headlines deliver a decisive and robust conclusion, summarizing the key outcomes or findings of the article.

4. **Weak Conclusion:** Headlines offer a less definitive or conclusive summary of the article's content, providing information without strong concluding statements.

5. **Misleading Conclusion:** Headlines present a conclusion that may mislead readers, either through ambiguity, partial information, or deliberate distortion.

6. **Unsupported Opinion:** Headlines express an opinion without sufficient supporting evidence from the article content.

7. **Irrelevant:** Headlines are unrelated to the content of the corresponding article, lacking relevance and coherence.

8. **Inconclusive:** Headlines do not provide a clear or definitive message regarding the content of the article, leaving the reader uncertain about the main theme or purpose.

The task involves annotators assigning the most appropriate class to a given headline concerning its alignment with the content of the corresponding article. The classification spans a spectrum, capturing the strength of conclusions, the presence of factual events, and potential misleading elements. This nuanced approach enables a detailed evaluation of headlines, contributing to the overall quality and relevance of news dissemination. Figure 5.11 shows the annotation page view of the developed tool.

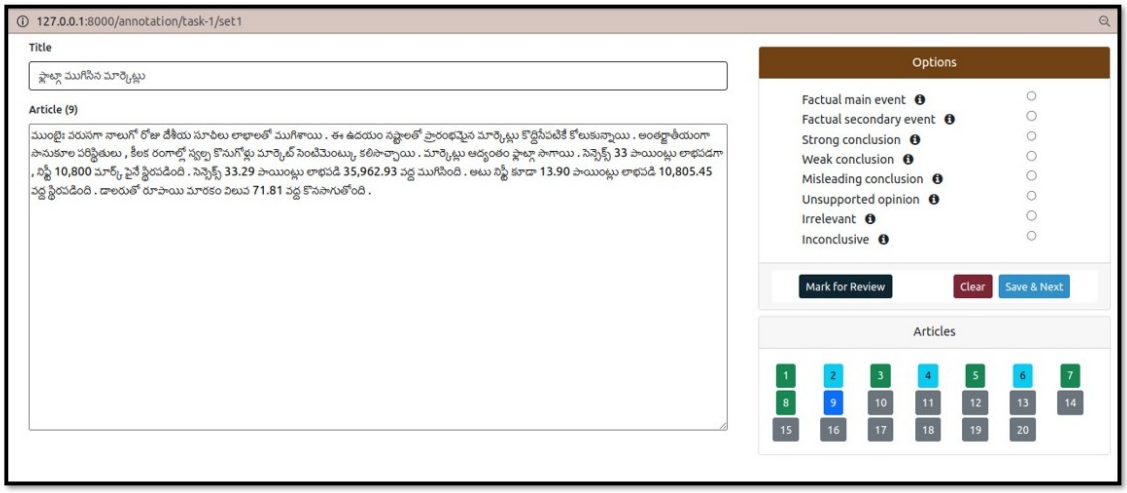

Figure 5.11 Headline Classification Annotation Tool

Intrinsic evaluation is a crucial component of the Headline Classification tool, aiming to assess annotator accuracy in selecting the correct class for each headline. The 2% golden samples, with predetermined correct classifications, are strategically placed within the dataset. Annotators are expected to align their responses with these golden samples, allowing for a quantitative evaluation of their accuracy. This intrinsic evaluation mechanism not only ensures the reliability of the annotation process but also provides valuable insights into annotator proficiency and the overall quality of the generated dataset.

5.2.3 ContractNER

In legal documents and contracts, precise identification of named entities is paramount for extracting meaningful insights. The Contract Named Entity Recognition (contractNER) annotation tool (refer Figure 5.12) is designed to streamline this process, offering a specialized interface for annotators to identify and tag entities specific to legal and contractual contexts.

This tool provides a user-friendly and efficient environment for annotators to annotate named entities in legal documents. Unlike generic NER tools, contractNER is tailored to accept the pre-annotations relevant to legal domains, such as parties involved, contract durations, monetary figures, legal clauses, and more obtained using ChatGPT and LexNLP. This innovative approach leverages the strengths of advanced language models to initiate the annotation process, offering annotators a head start in identifying potential entities. The interface allows annotators to review and modify these pre-annotations, ensuring a collaborative and iterative refinement process. This customized set of labels ensures a finer-grained and domain-specific annotation process, contributing to the creation of high-quality datasets for legal NLP applications.

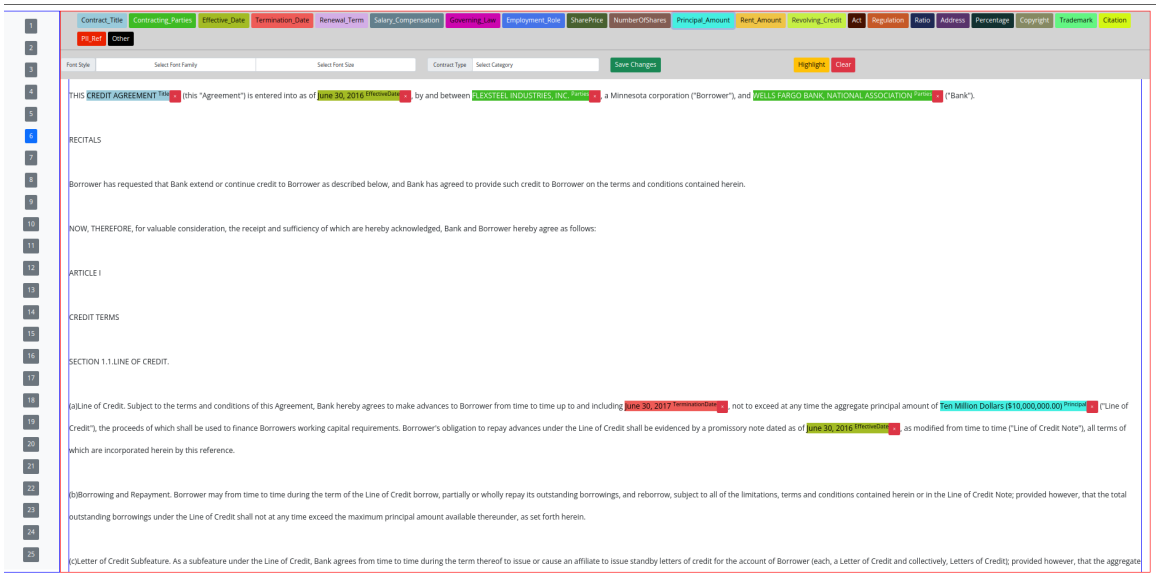

Figure 5.12 ContractNER Data Annotation View

5.2.4 Paraphrasing

Paraphrasing, the art of expressing the same information in different linguistic forms, is integral to natural language processing and understanding. The nuanced variations in how individuals convey information offer a rich dataset for training language models. In the realm of tool development for paraphrasing, the emphasis lies not only on creating diverse and equivalent expressions but also on ensuring the quality and relevance of the generated paraphrases. This section delves into the world of paraphrasing tools, exploring their significance, the intrinsic evaluation metrics. Figure 5.13 depicts the annotaiton page view of the developed paraphrasing tool.

Utilizing this annotation interface for paraphrasing tasks offers several distinct advantages over manual annotation processes. One key advantage lies in the efficiency and speed of the annotation process. The interface streamlines the task, allowing annotators to focus on generating paraphrases swiftly without the administrative overhead associated with manual methods.

5.2.5 TeQuAD Annotation Tool

The development of a comprehensive Telugu SQuAD dataset requires a meticulous annotation process that involves both content modification and question-answering correction. The primary objective is to adapt the existing SQuAD dataset, translated into Telugu, to ensure linguistic accuracy and contextual relevance. The annotation tool encompasses two critical stages: content modification of the translated passage and the correction of question-answering pairs.

In the content modification phase, annotators meticulously adjust the translated text to align with Tel-

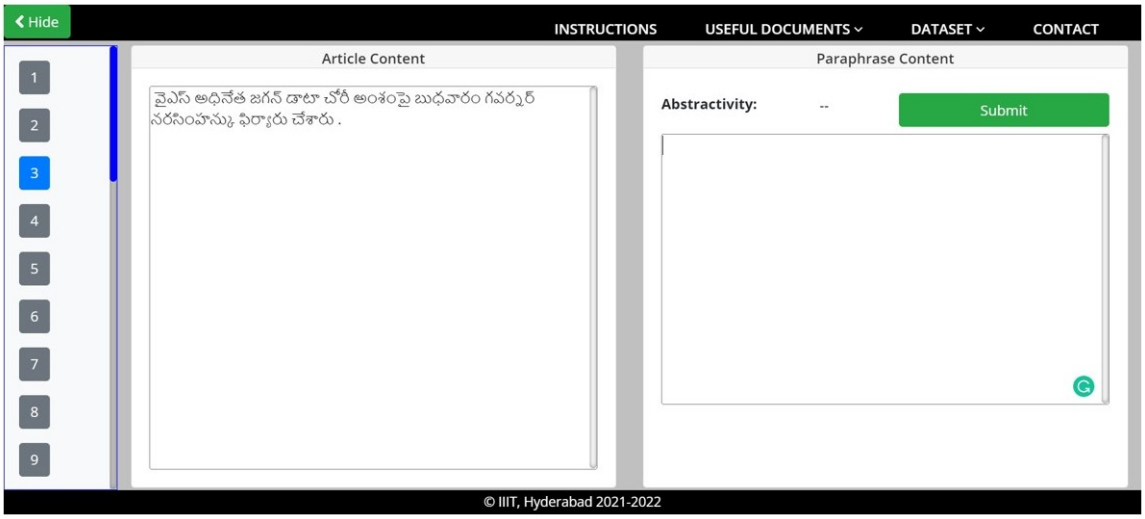

Figure 5.13 Paraphrasing Annotation Tool

ugu language conventions, ensuring fluency and linguistic precision. This step is pivotal to enhance the dataset's linguistic authenticity and facilitate accurate comprehension.

The subsequent question-answering correction process is a multifaceted task. Annotators carefully review each question, correcting language nuances in comparison to the original English questions. Simultaneously, they identify the correct answer within the Telugu passage by referencing the English answer, ensuring consistency across language versions.

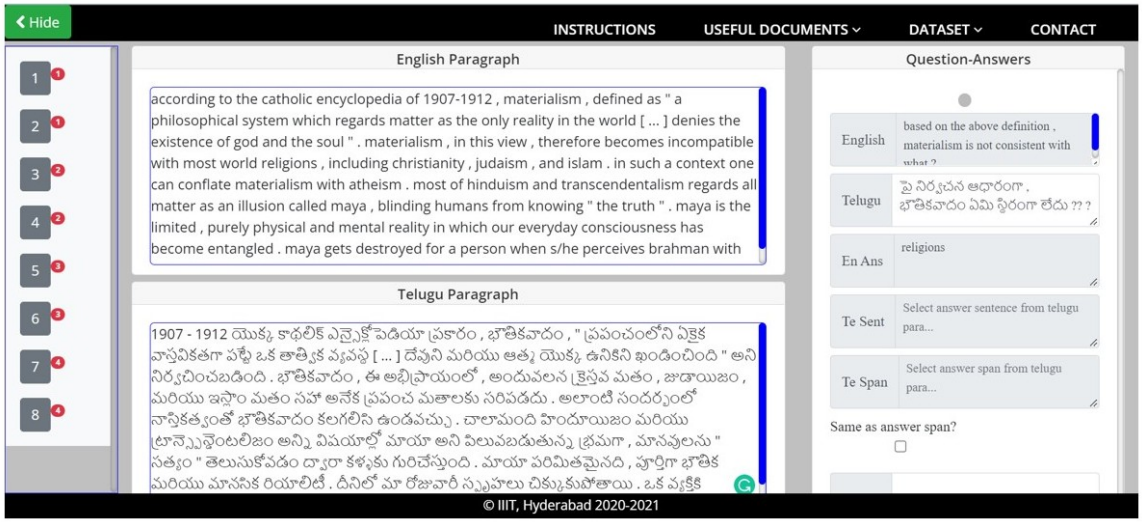

Figure 5.14 TeQuAD Annotation Tool

Figure 5.14 shows the annotation view of the TeQuAD tool. To streamline the annotation process, the tool employs a sophisticated mechanism to mark the start and end indices of sentences containing the answer. This strategic approach serves two key purposes: it eliminates duplicate answers by pinpointing the specific location within the passage, and it facilitates efficient dataset organization for subsequent analysis.

This tool offers distinct advantages over traditional manual annotation methods. One of the primary benefits is the meticulous modification of translated content. By engaging annotators in refining the Telugu passage, the tool ensures linguistic fluency and cultural relevance, addressing potential disparities introduced during the translation process. This step significantly contributes to the creation of a high-quality Telugu SQuAD dataset with improved linguistic authenticity.

Furthermore, the tool's approach to question-answering correction enhances the accuracy and consistency of the dataset. Annotators meticulously align Telugu questions with their English counterparts, rectifying linguistic nuances and ensuring semantic coherence. The identification of correct answers within the Telugu passage is facilitated by referencing the English answers, promoting cross-language consistency and precision.

In essence, the question-answering annotation tool is designed not only to enrich the Telugu SQuAD dataset but also to provide a systematic and effective means of linguistic adaptation and content correction. Its advantages extend beyond traditional manual annotation, offering enhanced accuracy, linguistic coherence, and dataset organization for improved usability and analytical insights.

Chapter 6

Conclusions and Future work

6.1 Conclusion

In this comprehensive exploration of large-scale data creation for NLP applications, each component has contributed distinctively to the overarching goal of enhancing data quality, diversity, and adaptability. The *Indic News Scraper* presented in this thesis emerges as a potent tool for systematically extracting content from news articles. Its structure-aware scraping capabilities, coupled with iterative adaptability across Indic languages, positions it as a robust solution for gathering diverse and multilingual data. As news articles serve as a crucial source for NLP tasks, the scraper's contributions lay the foundation for improved data quality and coverage.

The strategic Leveraging of *LLM as pre-annotators or Evaluators/Raters* introduces a paradigm shift in the data creation pipeline. Employing prompt engineering techniques, this facet demonstrates the potential of advanced language models to extract pre-annotations for specific NLP tasks. The iterative training process with human-annotated samples enhances model adaptability and efficiency, promising superior performance in subsequent annotation tasks.

The development of *dedicated interfaces* tailored for various NLP applications marks a significant step towards democratizing the annotation and evaluation process. These interfaces serve as purpose-built platforms for tasks such as summarization, headline classification, contract Named Entity Recognition (NER) tagging, machine translation evaluation, semantic text relatedness, paraphrasing, and question-answering. By providing accessible and user-friendly tools, this thesis contributes to streamlining the application of NLP methodologies.

To summarize, this thesis has critically assessed and proposed solutions for key components in the data creation framework. The combined contributions of the Indic News Scraper, LLMs as pre-annotators, and dedicated interfaces lay the groundwork for a more comprehensive and versatile approach to large-scale data creation for NLP applications. As we conclude, the pursuit of refining and expanding these

components remains imperative for advancing the field, promising a more nuanced, adaptable, and unified data creation solution for the dynamic realm of NLP applications.

6.2 Future Work

As we pave the way for future endeavors, the proposed data creation framework represents a starting point for further research and development. Future work involves refining and expanding the framework's capabilities, ensuring adaptability to emerging NLP tasks and challenges. Integrating more advanced instruction-based models, exploring novel annotation techniques, and enhancing user-friendly interfaces are avenues for improvement. Moreover, collaborating with the NLP community to collect insights and feedback will be crucial in refining the framework to align with the changing needs of researchers and practitioners in the field. The pursuit of a comprehensive, adaptable, and unified data creation solution remains a dynamic endeavor, promising continued advancements in the NLP applications.

Related Publications

Relevant Publications

1. Ashok Urlana, Nirmal Surange, **Pavan Baswani**, Priyanka Ravva, and Manish Shrivastava. 2022. TeSum: Human-Generated Abstractive Summarization Corpus for Telugu. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 5712–5722, Marseille, France. European Language Resources Association.

2. Hiranmai Sri Adibhatla, **Pavan Baswani**, Manish Shrivastava, "Fine-grained Contract NER using instruction based model" (PACLIC - 2023)

3. **Pavan Baswani**, Ananya Mukherjee and Manish Shrivastava, "LTRC_IIITH's 2023 Submission for Prompting Large Language Models as Explainable Metrics Task" (Eval4NLP - 2023)

4. Ousidhoum, N., Muhammad, S.H., Abdalla, M., Abdulmumin, I., Ahmad, I.S., Ahuja, S., Aji, A.F., Araujo, V., Ayele, A.A., **Baswani, P**. and Beloucif, M., 2024. "SemRel2024: A Collection of Semantic Textual Relatedness Datasets for 13 Languages" (ACL 2024 Findings)

5. **Pavan Baswani** and Manish Shrivastava. "Structure Aware Indic Language Content Extractor for News" **(yet to submit)**

Other Publications

1. **Pavan Baswani**, Hiranmai Sri Adibhatla, and Manish Shrivastava. 2023. LTRC at SemEval-2023 Task 6: Experiments with Ensemble Embeddings. In Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 841–846, Toronto, Canada. Association for Computational Linguistics.

2. Gopichand Kanumolu, Lokesh Madasu, **Pavan Baswani**, Ananya Mukherjee and Manish Shrivastava, "Unsupervised Approach to Evaluate Sentence-Level Fluency: Do We Really Need Reference?" (IJCNLP-AACL workshop 2023)

Bibliography

[1] M. Abdalla, K. Vishnubhotla, and S. M. Mohammad. What makes sentences semantically related: A textual relatedness dataset and empirical study. *arXiv preprint arXiv:2110.04845*, 2021.

[2] T. W. T. Au, I. J. Cox, and V. Lampos. E-ner–an annotated named entity recognition corpus of legal text. *arXiv preprint arXiv:2212.09306*, 2022.

[3] V. Barriere and A. Fouret. May i check again?–a simple but efficient way to generate and use contextual dictionaries for named entity recognition. application to french legal texts. *arXiv preprint arXiv:1909.03453*, 2019.

[4] K. Bontcheva, H. Cunningham, I. Roberts, A. Roberts, V. Tablan, N. Aswani, and G. Gorrell. Gate teamware: a web-based, collaborative text annotation framework. *Language Resources and Evaluation*, 47:1007–1029, 2013.

[5] J. Chen, S. Shankar, A. Kelly, S. Gningue, and R. Rajaravivarma. An adaptive bottom up clustering approach for web news extraction. In *2009 18th Annual Wireless and Optical Communications Conference*, pages 1–5. IEEE, 2009.

[6] C.-H. Chiang and H.-y. Lee. Can large language models be an alternative to human evaluations? In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada, July 2023. Association for Computational Linguistics.

[7] M. de Almeida, C. Samarawickrama, N. de Silva, G. Ratnayaka, and A. S. Perera. Legal party extraction from legal opinion text with sequence to sequence learning. In *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 143–148, 2020.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[9] E.-L. Do Dinh, R. E. de Castilho, and I. Gurevych. In-tool learning for selective manual annotation in large corpora. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 13–18, 2015.

[10] Y. Dong, Q. Li, Z. Yan, and Y. Ding. A generic web news extraction approach. In *2008 International Conference on Information and Automation*, pages 179–183. IEEE, 2008.

[11] M. Freitag, R. Rei, N. Mathur, C.-k. Lo, C. Stewart, E. Avramidis, T. Kocmi, G. Foster, A. Lavie, and A. F. T. Martins. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In P. Koehn, L. Barrault, O. Bojar, F. Bougares, R. Chatterjee, M. R. Costa-jussà, C. Federmann, M. Fishel, A. Fraser, M. Freitag, Y. Graham, R. Grundkiewicz, P. Guzman, B. Haddow, M. Huck, A. Jimeno Yepes, T. Kocmi, A. Martins, M. Morishita, C. Monz, M. Nagata, T. Nakazawa, M. Negri, A. Névéol, M. Neves, M. Popel, M. Turchi, and M. Zampieri, editors, *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid), Dec. 2022. Association for Computational Linguistics.

[12] F. Hamborg, N. Meuschke, C. Breitinger, and B. Gipp. news-please: A generic news crawler and extractor. 2017.

[13] H. Han and T. Tokuda. A layout-independent web news article contents extraction method based on relevance analysis. In *Web Engineering: 9th International Conference, ICWE 2009 San Sebastián, Spain, June 24-26, 2009 Proceedings 9*, pages 453–460. Springer, 2009.

[14] R. Hasan and M. A. Halliday. Cohesion in english. *London, 1976; Martin JR*, 1976.

[15] R. Hassanian-esfahani and M.-j. Kargar. A survey on web news retrieval and mining. In *2016 Second International Conference on Web Research (ICWR)*, pages 90–101. IEEE, 2016.

[16] P. Hausner and M. Gertz. News article extraction using graph embeddings. In *LWDA*, pages 119–132, 2021.

[17] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[18] P. Kalamkar, A. Agarwal, A. Tiwari, S. Gupta, S. Karn, and V. Raghavan. Named entity recognition in indian court judgments. *arXiv preprint arXiv:2211.03442*, 2022.

[19] S. Kim, S. Joo, D. Kim, J. Jang, S. Ye, J. Shin, and M. Seo. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning, 05 2023.

[20] J.-C. Klie, M. Bugert, B. Boullosa, R. E. de Castilho, and I. Gurevych. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *proceedings of the 27th international conference on computational linguistics: system demonstrations*, pages 5–9, 2018.

[21] T. Kocmi and C. Federmann. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland, June 2023. European Association for Machine Translation.

[22] J. K. Kummerfeld. Slate: a super-lightweight annotation tool for experts. *arXiv preprint arXiv:1907.08236*, 2019.

[23] C. Leiter, J. Opitz, D. Deutsch, Y. Gao, R. Dror, and S. Eger. The eval4nlp 2023 shared task on prompting large language models as explainable metrics. In *Proceedings of the 4th Workshop on Evaluation and Comparison for NLP systems*, 2023.

[24] E. Leitner, G. Rehm, and J. Moreno-Schneider. A dataset of german legal documents for named entity recognition. arxiv 2020. *arXiv preprint arXiv:2003.13016*.

[25] S. Leivaditi, J. Rossi, and E. Kanoulas. A benchmark for lease contract review. *arXiv preprint arXiv:2010.10386*, 2020.

[26] B. Y. Lin, D.-H. Lee, F. F. Xu, O. Lan, and X. Ren. Alpacatag: An active learning-based crowd annotation framework for sequence tagging. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 2019.

[27] Y. Lin, T. Ruan, M. Liang, T. Cai, W. Du, and Y. Wang. Dotat: A domain-oriented text annotation tool. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–8, 2022.

[28] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. A. Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.

[29] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment, 2023.

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

[31] S. Mayhew and D. Roth. Talen: Tool for annotation of low-resource entities. In *Proceedings of ACL 2018, System Demonstrations*, pages 80–86, 2018.

[32] G. A. Miller and W. G. Charles. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28, 1991.

[33] R. Mitchell. *Web scraping with Python: Collecting more data from the modern web*. " O'Reilly Media, Inc.", 2018.

[34] H. Nakayama, T. Kubo, J. Kamura, Y. Taniguchi, and X. Liang. doccano: Text annotation tool for human. *Software available from https://github. com/doccano/doccano*, page 34, 2018.

[35] M.-Q. Nghiem and S. Ananiadou. Aplenty: annotation tool for creating high-quality datasets using active and proactive learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 108–113, 2018.

[36] J. Niklaus, V. Matoshi, P. Rani, A. Galassi, M. Stürmer, and I. Chalkidis. Lextreme: A multi-lingual and multi-task benchmark for the legal domain. *arXiv preprint arXiv:2301.13126*, 2023.

[37] R. Oyri et al. News item extraction for text mining inweb newspapers. In *International workshop on challenges in web information retrieval and integration*, pages 195–204. IEEE, 2005.

[38] M. E. Peters and D. Lecocq. Content extraction using diverse feature sets. In *Proceedings of the 22nd international conference on world wide web*, pages 89–90, 2013.

[39] S. Pyysalo, J. Campos, J. M. Cejuela, F. Ginter, K. Hakala, C. Li, P. Stenetorp, and L. J. Jensen. Sharing annotations better: Restful open annotation. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 91–96, 2015.

[40] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[41] V. Rao and J. Sachdev. A machine learning approach to classify news articles based on location. In *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, pages 863–867. IEEE, 2017.

[42] D. D. C. Reis, P. B. Golgher, A. S. Silva, and A. Laender. Automatic web news extraction using tree edit distance. In *Proceedings of the 13th international conference on World Wide Web*, pages 502–511, 2004.

[43] K. Sattar and S. Iqbal. Multiple leaf nodes based parse trees for news fetching systems. In *2012 15th International Multitopic Conference (INMIC)*, pages 265–268. IEEE, 2012.

[44] A. Spengler and P. Gallinari. Learning to extract content from news webpages. In *2009 International Conference on Advanced Information Networking and Applications Workshops*, pages 709–714. IEEE, 2009.

[45] J. Straková, M. Straka, and J. Hajic. Open-source tools for morphology, lemmatization, pos tagging and named entity recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, 2014.

[46] K. Sundaramoorthy, R. Durga, and S. Nagadarshini. Newsone—an aggregation system for news using web scraping method. In *2017 International Conference on Technical Advancements in Computers and Communications (ICTACC)*, pages 136–140. IEEE, 2017.

[47] M. Tkachenko, M. Malyuk, A. Holmanyuk, and N. Liubimov. Label Studio: Data labeling software, 2020-2022. Open source software available from https://github.com/heartexlabs/label-studio.

[48] A. Urlana, N. Surange, P. Baswani, P. Ravva, and M. Shrivastava. TeSum: Human-generated abstractive summarization corpus for Telugu. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, and S. Piperidis, editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5712–5722, Marseille, France, June 2022. European Language Resources Association.

[49] A. Ushio and J. Camacho-Collados. T-ner: an all-round python library for transformer-based named entity recognition. *arXiv preprint arXiv:2209.12616*, 2022.

[50] J. Wang, X. He, C. Wang, J. Pei, J. Bu, C. Chen, Z. Guan, and G. Lu. News article extraction with template-independent wrapper. In *Proceedings of the 18th international conference on World wide web*, pages 1085–1086, 2009.

[51] L. Wang, R. Li, Y. Yan, Y. Yan, S. Wang, W. Wu, and W. Xu. Instructionner: A multi-task instruction-based generative framework for few-shot ner. *arXiv preprint arXiv:2203.03903*, 2022.

[52] X. Wang, W. Wang, B. Liu, Z. Wang, and X. Wang. A novel approach to automatically extracting main content of web news. In *2009 International Conference on E-Business and Information System Security*, pages 1–4. IEEE, 2009.

[53] G. Wu and X. Wu. Extracting web news using tag path patterns. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 588–595. IEEE, 2012.

[54] G.-Q. Wu, X. Wu, X.-G. Hu, H.-G. Li, Y. Liu, and R.-G. Xu. Web news extraction based on path pattern mining. In *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, volume 7, pages 612–617. IEEE, 2009.

[55] W. Xu, D. Wang, L. Pan, Z. Song, M. Freitag, W. Y. Wang, and L. Li. Instructscore: Towards explainable text generation evaluation with automatic feedback, 2023.

[56] J. Yan, H. Duan, L. Fang, and W. Ying. News web text extraction based on the maximum subsequence segmentation. In *2013 International Conference on Computational and Information Sciences*, pages 619–622. IEEE, 2013.

[57] J. Yao and X. Zuo. A machine learning approach to webpage content exraction, 2013.

[58] S. M. Yimam, C. Biemann, R. E. de Castilho, and I. Gurevych. Automatic annotation suggestions and custom annotation layers in webanno. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 91–96, 2014.

[59] S. M. Yimam, I. Gurevych, R. E. de Castilho, and C. Biemann. Webanno: A flexible, web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–6, 2013.

[60] H. Yunis, B. Stein, J. Kiesel, and A. Jakoby. Content extraction from webpages using machine learning. *Unpublished master's thesis). Bauhaus-Universität Weimar*, 2016.

[61] B. Zhang, Z. Li, Z. Gan, Y. Chen, J. Wan, K. Liu, J. Zhao, S. Liu, and Y. Shi. Croano: A crowd annotation platform for improving label consistency of chinese ner dataset. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 275–282, 2021.

[62] C. Zhang and Z. Lin. Automatic web news content extraction based on similar pages. In *2010 International Conference on Web Information Systems and Mining*, volume 1, pages 232–236. IEEE, 2010.

[63] S. Zheng, R. Song, and J.-R. Wen. Template-independent news extraction based on visual consistency. In *AAAI*, volume 7, pages 1507–1513, 2007.

[64] B. Zhou, Y. Xiong, and W. Liu. Efficient web page main text extraction towards online news analysis. In *2009 IEEE International Conference on e-Business Engineering*, pages 37–41. IEEE, 2009.

[65] Z. Zhou and M. Mashuq. Web content extraction through machine learning. *Standford Univ*, pages 1–5, 2014.

[66] C.-N. Ziegler, C. Vogele, and M. Viermetz. Distilling informative content from html news pages. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 707–712. IEEE, 2009.

[67] C.-N. Ziegler, C.-N. Ziegler, C. Vögele, and M. Viermetz. Distilling informative content from html news pages using machine learning classifiers. *Mining for strategic competitive intelligence: Foundations and applications*, pages 151–166, 2012.

[68] L. Ziyi, S. Beijun, T. Xinhuai, and C. Delai. Automatic web news extraction using blocking tag. In *2009 Second International Conference on Machine Vision*, pages 74–78. IEEE, 2009.