Abstractive Urdu Summarization for News Article Dataset Absolute Sum

Ammar Amjad, Mohammad Uzair Fasih, Mohammad Anas, Michael Pérez
ammar.amjad@ufl.edu, mfasih@ufl.edu, mo.anas@ufl.edu, michaelperez012@ufl.edu

Abstract

This paper delves into the underexplored challenge of abstractive summarization of Urdu and evaluated the performance of large language models applied to this task. Fine-tuned models that achieve improved results compared to pre-trained versions are proposed and their performance including a baseline extractive model, mT5-vanilla, XL-Sum-vanilla, and mT5-fine-tuned, is evaluated using both ROUGE scores and human evaluation. Despite preliminary results that suggested the baseline model has superior performance, fine-tuned models like mT5 and XL-Sum are shown to produce better abstractive summaries when evaluated by humans. The research highlights the insufficiency of ROUGE scores for accurately evaluating Urdu summarization models and emphasizes the need for improved metrics that account for the language's unique characteristics, such as the abundance of synonyms. This study contributes to natural language processing in low-resource languages and underscores the potential of deep learning techniques for abstractive Urdu summarization, paving the way for further exploration in this domain.

1 Introduction

It is often unproductive to read lengthy articles, and using summarization techniques can be very beneficial in such situations. Extractive summarization produces summaries by extracting important sentences and words from the input article, while abstractive summarization aims to learn the semantic meaning of input articles and summarize them using mined keywords. Abstractive summarization has become increasingly important in the age of information overload, as it helps users quickly grasp the essence of long texts. While this technology has advanced significantly for many widely spoken languages, the Urdu language, spoken by 200 million people, has not seen the same level of progress. As the 10th most spoken language globally, Urdu deserves attention in the field of abstractive summarization. This paper aims to address this gap and contribute to the ongoing development of natural language processing tools for the Urdu language.

Abstractive summarization is particularly important for Urdu which is a morphologically rich language and involves complex grammar and diverse vocabulary. Syntactically it has a free word order, or in other words, it is a Subject-Object-Verb language. The language's unique features make it challenging to develop efficient summarization algorithms that can accurately capture the nuances and convey the main ideas without losing the context. By investing in the development of abstractive summarization for Urdu, new opportunities for research, education, and communication can be opened within the Urdu-speaking community, making information more accessible and promoting the growth of the language in the digital age.

Existing methods for abstractive Urdu summarization have relied on outdated sequence-to-sequence models that are limited in their ability to understand and generate contextually accurate summaries. The rapid advancements in natural language processing technologies, particularly with deep learning models, provide us with an opportunity to develop more effective and accurate abstractive summarization models for the Urdu language.

In this paper current solutions for abstractive Urdu summarization are explored, their limitations are reviewed, and a model that leverages state-of-the-art deep learning techniques to deliver improved performance in generating contextually accurate and concise summaries is introduced. A comprehensive evaluation of our model is presented, demonstrating its effectiveness and superiority over existing methods and highlighting the potential impact of our work in advancing Urdu language processing technologies.

2 Related Work

This section reviews recent influential language models and then covers prior approaches for abstractive and extractive summarization of Urdu. Although this paper's focus is on abstractive summarization, extractive summarization is reviewed because this approach is often used as a first step toward abstractive summarization and there has been sparse work in Urdu abstractive summarization.

2.1 Large Language Models

The transformer (Vaswani et al., 2017) is an influential network architecture for sequence-to-sequence mapping that dispenses with the convolutions and recurrence relations used in prior work and shows that the self-attention mechanism is all you need. Devlin et. al. (Devlin et al., 2019) built upon the transformer by introducing BERT (Bidirectional Encoder Representations from Transformers), a pre-trained language representation model that can be fine-tuned for downstream tasks using just one additional output layer.

T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2020) is a sequence transduction model developed by Google that closely follows the original transformer architecture (Vaswani et al., 2017). It achieves state-of-the-art results on numerous NLP tasks by pre-training a novel "Colossal Clean Crawled Corpus". mT5 (Xue et al., 2021) is a multilingual version of T5 that was pre-trained on the multilingual Common Crawl corpus (mC4) which covers 101 languages including Urdu.

Hasan et al. (Hasan et al., 2021) proposes a fine-tuned model based on mt5. They retrained mt5 on their own dataset called "XL-Sum". This dataset is comprised of 1 million article-summary pairs in 44 low to high-resource languages including Urdu. The model shows competitive performance in both multilingual and low-resource summarization tasks. The model is released along with the dataset to encourage further research on multilingual abstractive summarization.

2.2 Extractive and Abstractive Summarization

One group (Nawaz et al., 2020) developed an extractive summarization framework for Urdu text by modeling local weights and global weights approaches for computing the weights of words in a sentence. In extractive summarization, sentences with the highest weight are prioritized to be in the produced summaries and sentence weights are sums of the weights of words. In local weights approaches the weights of words depend on the article content so a word can have different weights across different articles. In global weights approaches the weights of words are the same across all articles. The human evaluation showed that local weights approaches were better for Urdu extractive summarization even though global weights approaches are used more often in English. This work also compared the performance of preprocessed input against unprocessed input and found that preprocessed input yielded better summaries.

Another team (Farooq et al., 2021) investigated eight different algorithms for Urdu extractive summarization: Reduction, KL, Edmundson, TextRank, LextRank, LSA, SumBasics, and Luhn Summarizer. The Reduction and Luhn's summarizers worked best for Urdu summarization when compared to human-generated summaries using Recall-Oriented Understudy for Gisting Evaluation scores (ROUGE-1) (Lin, 2004). ROUGE-1 scores measure the overlap of unigrams between the reference and output summaries, ROUGE-2 measures the overlap of bigrams, and ROUGE-L measures the overlap of the longest common subsequence (Lin and Och, 2004). High overlaps suggest high-quality summaries.

Asif et al. (Asif et al., 2022) proposed to produce extractive summaries of Urdu using word frequency, sentence weight, and the TF-IDF (Term Frequency - Inverse Document Frequency) algorithm, then use BERT (Devlin et al., 2019) to process the extractive summaries and produce abstractive summaries.

Another work (Shafiq et al., 2023) proposed a framework for abstractive summarization of Urdu text that processes extractive summaries using an encoder-decoder long short-term memory (LSTM) to produce abstractive summaries.

3 Dataset

The dataset for training our model was carefully chosen. This dataset encompasses an extensive range of Urdu news articles and corresponding summary references. The subsequent sections explain the various attributes of this dataset.d

3.1 Overview

The dataset used for this study is comprised of $84,581$ rows and 5 columns: *id*, *url*, *title*, *summary*,

and *text*. The *id* column serves as a unique identifier for each entry, while the *url* column contains the URL of the article from the source website. The title column contains the title of the *article*, while the *summary* column provides a brief summary or abstract of the content. The *text* column contains the full text of the article, which serves as the main body of data for analysis.

3.2 Training, Validation and Testing splits

To prevent overfitting and enable accurate evaluation, the dataset is split into a training set of $67,665$ rows, a test set of $8,458$ rows, and a validation set of $8,458$ rows. The training set comprises 80% of the data, while both the test and validation sets are 10% each. There are no missing values in the dataset. The dataset was sourced from a publicly available repository on GitHub (Irfan, 2023).

3.3 Data Exploration

The *title* column has $84,023$ unique values and 558 duplicate values. The 'summary' column has $84,281$ unique values and 300 duplicate values. The *text* column has $84,565$ unique values and only 16 duplicate values. The data has a right skew, with the median being lower than the mean in all columns. The average word count in the title is 8.3. The average word length in the title, summary, and text columns are 5.0, 4.7, and 4.6 characters, respectively. The average lexical diversity of the title, summary, and text columns is 1.01, 1.13, and 2.06, respectively.

4 Methodology

Our proposed methodology entails the following steps:

- Acquiring Dataset
- Data Cleaning and Preprocessing
- Importing and Fine-tuning Models
- Automatic Evaluation using ROUGE
- Human Evaluation using Survey

The design of our methodology is shown in Figure 1. The following sections further elaborate on each step.

4.1 Preprocessing

The following preprocessing steps were conducted on the dataset:

1. Duplicate Row Removal: Identify and remove duplicate rows in the title, summary, and text columns from the dataset.

2. Tokenization: A tokenizer function was used to tokenize the text data in the *summary* and *title* columns, separating sentences, phrases, paragraphs, or entire text into particular or distinct expressions.

3. Normalization: Normalizing text by separating digraphs. The normalization removes diacritics and accents and also transforms a phrase into various forms, such as a list of tuples or a list of words.

4. Length limiting: The maximum length of the input text was limited to ensure that the input data adheres to a certain length constraint, which was necessary for our training requirements.

5. Stop-words disposal: Remove the most common words from the text data as stop words. Removing stop words, which are words that have little semantic significance and frequently appear in documents, to focus the text on key information.

6. Lemmatization: Breaking down words into their most fundamental components by shortening prefixes and suffixes, leaving only the word's stem. This is done to understand the context in which the word is used.

7. Data collation: A data collator was used to make the representation of sentences have a uniform length and pad zeroes as necessary to create batches.

4.2 Models

Various models were developed and compared for abstractive text summarization. Two pre-trained models, two fine-tuned models, and one baseline model were compared. The two architectures evaluated were mT5 (Xue et al., 2021) and mt5-XL-sum (Hasan et al., 2021). The baseline model is an extractive summarization method that uses the first three sentences of an article as the summary.

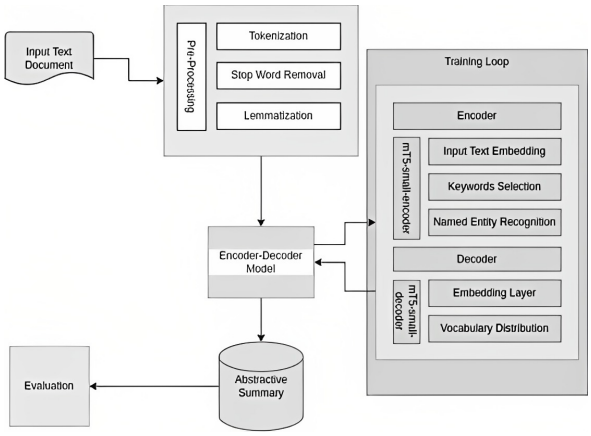

Figure 1: Methodology Design

4.3 Fine Tuning and Implementation

The *AdamW* (Loshchilov and Hutter, 2017) optimizer was used. *AdamW* is a modification of *ADAM* (Kingma and Ba, 2014) that improves generalization performance by decoupling the weight decay factor from the optimization steps taken with respect to the loss function, providing regularization. To improve the performance of the mt5-XL-sum and mt-5 models on the summarization task, the models were fine-tuned using the acquired dataset. The training process lasted for 50 epochs. The aim of this additional training was to fine-tune the models and enhance their ability to generate abstractive summaries of news articles. By training the models on a diverse and comprehensive dataset, it was hypothesized that they would be able to better capture the nuances and complexities of news articles and improve their summarization quality.

The implementation was inspired by a publicly available GitHub repository (Tunstall, 2023). The code was developed on a Jupyter notebook [1] that uses *PyTorch* 2.0.0 with CUDA and Hugging Face's Transformers library. The models were fine-tuned using an NVIDIA Tesla T4 with 16 gigabytes of memory on Google Colab and an NVIDIA A100 GPU with 80 gigabytes of memory on UF's HiPerGator supercomputer.

5 Results and Analysis

The performance of the models was evaluated using two approaches. The first approach involved computing the ROUGE scores, specifically ROUGE-1, ROUGE-2, and ROUGE-L, between the generated summaries and reference summaries. These scores were used to evaluate the quality of the generated summaries and compare the performance of the different models.

In addition to the ROUGE scores, a second evaluation approach was conducted. This involved a survey where 10 volunteers were asked to rate the quality of the generated summaries on a scale of 1 to 5, with 5 indicating the best summary. The study used text taken from recent BBC-Urdu news articles to evaluate the models. By involving human evaluators, this approach provided a more comprehensive evaluation of the effectiveness of the models in producing accurate and high-quality summaries.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Baseline	0.46	0.10	0.52
mt5-vanilla	0.087	0.013	0.080
XL-Sum-vanilla (SOTA)	0.33	0.061	0.27
XL-Sum-fine-tuned	0.36	0.073	0.29
mt5-fine-tuned	0.26	0.11	0.25

Table 1: ROUGE Scores for each Model

Table 1 displays the ROUGE scores for various models, including a baseline model that extracts the first three sentences, mt5-vanilla, XL-Sum-fine-tuned, and mt5-fine-tuned. ROUGE-1, ROUGE-2, and ROUGE-L are metrics used for evaluating the accuracy of output summaries. While the baseline

[1] Project Code: https://github.com/michaelperez023/urdu-abstractive-summarization/blob/main/Urdu_Summarization.ipynb

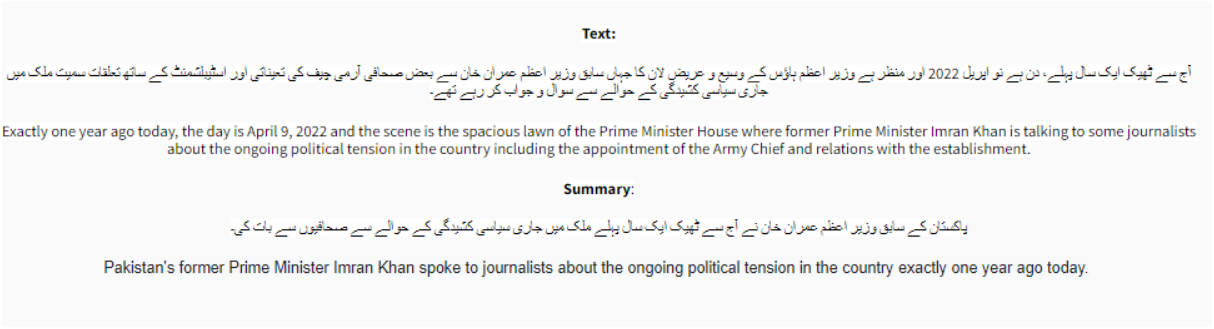

Figure 2: mt5-fine-tuned Summarization Result Example

model initially appears to have the highest scores, it was discovered that this was due to its use of a simple extractive method. This approach yields summaries with high word similarity to the reference summary, resulting in inflated ROUGE scores. In contrast, more sophisticated models such as XL-Sum-vanilla and mt5-fine-tuned achieve intermediate scores. To confirm this, a human evaluation was conducted. Table 2 illustrates that fine-tuned models outperformed the baseline model in the human evaluation, which resulted in better abstractive summaries. This was observed due to the abundance of synonyms in the Urdu language. ROUGE scores rely on word similarity and penalize summaries that do not include the exact words from the article. However, the fine-tuned models used condensed words to express the same meaning, which resulted in better summaries, as confirmed by human evaluators.

Table 2 also indicates that, following fine-tuning, the mt5 model demonstrated superior performance in Urdu text summarization compared to other state-of-the-art models like mT5-XL-sum and mT5-vanilla. The mt5-vanilla and mT5-XL-sum models were pre-trained on Urdu language datasets, which implies that they have learned to understand the language's structure and rules. However, the quality of the dataset used for pretraining was not optimal, leading to lower performance during evaluation. Our method improved the performance of these models. This achievement can be attributed to the utilization of a varied dataset during training and the application of effective preprocessing techniques.

6 Conclusion

This paper presents an investigation into the problem of abstractive Urdu summarization. Various deep learning techniques were evaluated, including mT5 and XL-Sum, and a fine-tuned model that achieves state-of-the-art results when evaluated by humans was proposed.

Model	Mean Score
Baseline	0.26
mt5-vanilla	0.25
XL-Sum-vanilla (SOTA)	0.28
XL-Sum-fine-tuned	0.35
mt5-fine-tuned	**0.45**

Table 2: Human Study Ratings (normalized)

The research presented in this paper makes a significant contribution to the field of natural language processing, particularly in low-resource languages like Urdu. The study emphasizes the importance of fine-tuning models to produce satisfactory abstractive summaries, which can overcome the limitations of existing summarization models. This finding has important implications beyond Urdu, as similar techniques can be applied to improve performance in other low-resource languages such as Persian. By building on the success of this research and using diverse datasets and appropriate preprocessing techniques, natural language processing models can be fine-tuned to provide accurate and effective abstractive summaries in various low-resource languages.

The findings of the project suggest that the evaluation metrics for ROUGE scores may not be sufficient for accurately assessing the performance of text summarization models in the Urdu language. This highlights the need for developing better evaluation metrics that take into account the unique characteristics of the Urdu language, such as the large presence of synonyms, to provide a more accurate assessment of the model's performance.

Considering the limited sample size of ten volunteers, we propose conducting a more comprehen-

sive study in the future to gain a better understanding of summary evaluation and provide comprehensive insights into the effectiveness of summaries. This study shows the need for evaluation metrics like Translation Edit Rate (Snover et al., 2006) and BLEU (Papineni et al., 2002) scores to be implemented for Urdu.

Overall, the study's findings demonstrate the potential of natural language processing techniques in low-resource languages such as Urdu and the need to continue developing and refining these techniques to improve their accuracy and effectiveness.

References

Muhammad Asif, Syed Ali Raza, Javed Iqbal, Nousheen Perwaiz, Tauqeer Faiz, and Shan Khan. 2022. Bidirectional encoder approach for abstractive text summarization of urdu language. In *2022 International Conference on Business Analytics for Technology and Security (ICBATS)*, pages 1–8.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Aman Farooq, Safiyah Batool, and Zain Noreen. 2021. Comparing different techniques of urdu text summarization. In *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*, pages 1–6.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

Mohammad Irfan. 2023. Urdu summary dataset. https://mirfan899.github.io/Urdu.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612, Barcelona, Spain.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Ali Nawaz, Maheen Bakhtyar, Junaid Baber, Ihsan Ullah, Waheed Noor, and Abdul Basit. 2020. Extractive text summarization models for urdu language. *Information Processing Management*, 57(6):102383.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nida Shafiq, Isma Hamid, Muhammad Asif, Qamar Nawaz, Hanan Aljuaid, and Hamid Ali. 2023. Abstractive text summarization of low-resourced languages using deep learning. *PeerJ Computer Science*, 9:e1176.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Lewis Tunstall. 2023. Hugging face course chapter 7 tutorial. https://github.com/huggingface/notebooks/tree/main.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.