IndicNLG Suite: Multilingual Datasets for Diverse NLG Tasks in Indic Languages

Aman Kumar1, Himani Shrotriya2∗, Prachi Sahu3∗, Raj Dabre4
Ratish Puduppully5, Anoop Kunchukuttan6, Amogh Mishra7
Mitesh M. Khapra8 Pratyush Kumar9
IIT Madras1,2,3,8,9 Columbia University7
National Institute of Information and Communications Technology4
Microsoft6,9 University of Edinburgh5 AI4Bharat6,8,9
1cs20m010@gmail.iitm.ac.in 2cs20m024@gmail.iitm.ac.in
3cs20m047@gmail.iitm.ac.in 4raj.dabre@nict.go.jp
5r.puduppully@sms.ed.ac.uk 6ankunchu@microsoft.com
7amogh.mishra@columbia.edu 8miteshk@cse.iitm.ac.in
9pratykumar@microsoft.com

Abstract

In this paper, we present the IndicNLG suite, a collection of datasets for benchmarking Natural Language Generation (NLG) for 11 Indic languages. We focus on five diverse tasks, namely, biography generation using Wikipedia infoboxes (WikiBio), news headline generation, sentence summarization, question generation and paraphrase generation. We describe the process of creating the datasets and present statistics of the dataset, following which we train and report a variety of strong monolingual and multilingual baselines that leverage pre-trained sequence-to-sequence models and analyze the results to understand the challenges involved in Indic language NLG. To the best of our knowledge, this is the first NLG dataset for Indic languages and also the largest multilingual NLG dataset. Our methods can also be easily applied to modest-resource languages with reasonable monolingual and parallel corpora, as well as corpora containing structured data like Wikipedia. We hope this dataset spurs research in NLG on diverse languages and tasks, particularly for Indic languages. The datasets and models are publicly available at https://indicnlp.ai4bharat.org/indicnlg-suite.

1 Introduction

NLG is the process of generating textual output (Gatt and Krahmer, 2018). Initial work on NLG focused on tabular input (Reiter and Dale, 1997) but in a general setting the input can also belong to one or more modalities such as text, images, videos, audio, etc. NLG progress had been hindered by the lack of data for different tasks and languages but, recently, the increasing availability of large scale datasets (Narayan et al. 2018a; Wiseman et al. 2017; Lebret et al. 2016, inter alia), along with the advancements in neural networks pretrained on large amounts of text (Lewis et al., 2020; Raffel et al., 2020) have led to substantial progress in NLG.

Progress in NLG, or any research topic, often goes hand in hand with the availability of datasets and benchmarks. Consider the example of the Europarl (Koehn, 2005) dataset, which enabled researchers to make substantial progress for machine translation. This also motivated the collection and dissemination of additional datasets spanning several languages (Tiedemann, 2012; Zhang et al., 2020; El-Kishky et al., 2020). Another example is the extreme summarization (XSum) dataset (Narayan et al., 2018a) which has led to significant improvements in summarization (Lewis et al., 2020; Zhang et al., 2019). This also motivated the creation of the multilingual extreme summarization (XL-Sum) (Hasan et al., 2021b) dataset spanning 44 languages, as well as the cross-lingual summarization dataset (Hasan et al., 2021a).

Most of the aforementioned progress is for European languages and especially for English, mainly because it is the lingua franca, making it easy to obtain data for it (Bender, 2019). However, English is not the native language for a vast majority of the world’s population, which tend to use their country’s or region’s native languages on a daily basis. India, with its population of 1.4 billion people1 (18% of the world population) is a quintessential example where only 10% of the population speaks English whereas a significant portion of the remain-

∗ Equal contribution.
Table 1: A summary of the datasets and languages covered by IndicNLG suite. The communicative intent, inputs and total corpora sizes are given.

| Dataset               | Languages                        | Communicative Intent                  | Input Type             | Total Size |
|-----------------------|----------------------------------|---------------------------------------|------------------------|------------|
| Biography Generation  | as, bn, hi, kn, ml, or, pa, ta, te | One-sentence biographies              | key-value pairs        | 55K        |
| Headline Generation   | as, bn, gu, hi, kn, ml, mr, or, pa, ta, te | News article headlines                | news article           | 1.43M      |
| Sentence Summarization| as, bn, gu, hi, kn, ml, mr, or, pa, ta, te | Compacted sentence with same meaning  | sentence               | 431K       |
| Paraphrase Generation | as, bn, gu, hi, kn, ml, mr, or, pa, ta, te | Synonymous sentence                   | sentence               | 5.57M      |
| Question Generation   | as, bn, gu, hi, kn, ml, mr, or, pa, ta, te | Question leading to answer given context | context-answer pairs   | 1.08M      |

The rest of the paper is organized as follows: Section 2 describes related work. Section 3 describes the IndicNLG suite where the tasks, datasets and their creation are explained along with some important quantitative and qualitative statistics. This is followed by Section 4 where we describe the experimental settings for benchmarking the performance...
of our NLG models for various tasks. Section 5 contains results and analyses. We end the paper with general and task specific summaries in Section 6 and outline several future directions we plan to pursue. The appendix (Section A) contains additional results and analyses for interested readers.

2 Related Work

This paper focuses on data creation, modeling and benchmarking for NLG, which span an array of related topics. In this section, we attempt to contextualize our work, which should serve as a short survey of NLG.

2.1 NLG Benchmarks

Gehrmann et al. (2021) create a benchmark for NLG tasks such as extreme summarization (Hasan et al., 2021b; Scialom et al., 2020; Narayan et al., 2018a), data-to-text generation (Gardent et al., 2017; Parikh et al., 2020; Nan et al., 2021; Dušek et al., 2020), and cross lingual summarization (Ladhak et al., 2020). In addition, they aim to establish baseline models along with automatic and human evaluations. However, the focus of GEM benchmark is predominantly English (7 out of 11 tasks). Cahyawijaya et al. (2021) propose a NLG benchmark for three Indonesian languages. Concurrent with our work, Guan et al. (2021) propose a NLG benchmark for Chinese long text NLG including two tasks. In addition, Chen et al. (2021) propose an NLG benchmark for three languages ‘fr’, ‘de’ and ‘es’ along with ‘en’, for three tasks of story generation, headline generation and question generation. In contrast, our IndicNLG benchmark covers 11 Indic languages and five tasks, making it the first for Indic languages as well as the most linguistically diverse NLG benchmark to the best of our knowledge. IndicNLG suite complements the IndicGLUE benchmark (Kakwani et al., 2020) for natural language understanding (NLU) of Indian languages.

2.2 NLG Tasks

2.2.1 Data-to-text Generation

Data-to-text generation refers to the task of generating textual output from non-linguistic input (Reiter and Dale, 1997; Gatt and Krahmer, 2018). The input can be in the form of spreadsheets, databases, tables, etc. Recently, large scale datasets for data-to-text generation have been proposed (Lebret et al., 2016; Wiseman et al., 2017; Puduppully et al., 2019; Parikh et al., 2020). Specifically, Lebret et al. (2016) introduced a dataset in English for generating one-sentence biographies from Wikipedia Infobox. An infobox is a table containing a list of attribute-value pairs in a wikipedia page. The biographies are obtained from the first sentence from the corresponding wikipedia page. In this work, we extend the approach to Indic languages by extracting infobox paired with their one-sentence biography for Indic languages. Closely related, and concurrent to our work, is the cross lingual fact to text alignment task for Indic languages (Abhishek et al., 2022) where the goal is to generate sentences in one language which describe facts in another language.

2.2.2 Headline Generation

The task of headline generation involves generation of the headline of a news article. Such a headline is often shorter than a sentence (Banko et al., 2000). Recently, approaches making use of neural models for headline generation have been proposed (see Shen et al. 2017 for a survey). Tilk and Alumäe (2017) pretrain encoder and decoder of the neural model to improve headline generation results in low resource conditions.

2.2.3 Sentence Summarisation

Rush et al. (2015) propose a large scale dataset which pairs the first sentence of news articles with their headline. With regard to methodology, Li et al. (2020) focus on keywords to be retained in a summary and try to strike a balance between extractive and abstractive sentence summarization. Similarly, Zhou et al. (2017b) design an attention mechanism which learns to focus on words that form an integral part of the summary. In general, any document summarization approach should work for sentence summarization, except that sentence summaries are highly compact and span a few words.

2.2.4 Paraphrase Generation

In Paraphrase generation, rule-based approaches (McKeown, 1979; Meteer and Shaked, 1988) and data-driven methods (Madnani and Dorr, 2010) have been explored. In most cases, paraphrasing is treated as a machine-translation problem (Barnard and Callison-Burch, 2005; Barzilay and McKeown, 2001; Pang et al., 2003), often performed using a pivoting approach with a bilingual corpus (Madnani and Dorr, 2010; Prakash et al., 2016; Mallinson et al., 2017). Recently, deep reinforce-
ment learning (Li et al., 2018), supervised learning using sequence-to-sequence models (Gupta et al., 2018; Prakash et al., 2016), and unsupervised approaches (Bowman et al., 2016; Roy and Grangier, 2019) have also been proposed.

A major limiting factor for paraphrasing research is the availability of datasets. Most work focuses on datasets for English paraphrasing (Wieting and Gimpel, 2018). Recently, datasets are being made available for other languages such as Finnish (Kanerva et al., 2021). OpusParcus (Creutz, 2018) is a multilingual paraphrase dataset focusing on European languages such as French, Romanian, German etc. For Indic languages, Anand Kumar et al. (2016) create paraphrasing corpora for four languages including Tamil, Punjabi, Hindi and Malayalam with a total size of around 30k examples. In this paper, we curate a paraphrase dataset for 11 Indic languages with a total of 5.5M examples.

2.2.5 Question Generation

Some early works in Question Generation focused on linguistic annotation tools and statistical methods to generate questions for vocabulary assessment, as well as complex query generation (Brown et al., 2005; Ali et al., 2010). Another influential work in this area is by Heilman (2011), which generates questions given an input of an article of text. The progress in question generation has picked pace with the advent of sequence-to-sequence neural models and the availability of new datasets (Song et al., 2018; Du et al., 2017; Kumar and Black, 2020; Chen et al., 2018). Specifically, Du et al. (2017) repurpose the SQuAD question-answering dataset to serve as a question generation dataset. Each example in the dataset comprises a context sentence, answer word(s) and its question. We build upon this dataset for creating our Indic language equivalent question generation datasets. Examples of works that use an entire paragraph as context are Xiao et al. (2021); Bao et al. (2020); Back et al. (2021). We leave for future work the exploration of the whole paragraph as context.

2.3 Other Related Work

Although this paper does not focus directly on pre-training, machine translation, and document summarization, we cover some relevant works for completeness.

2.3.1 Pre-trained seq2seq models

The availability of pre-trained models, typically trained using unsupervised approaches and monolingual data, help reduce the requirement of large amounts of (supervised) fine-tuning data for a given downstream task. In this context, T5 (Raffel et al., 2020), mT5 (Xue et al., 2021), BART (Lewis et al., 2020), mBART-25 (Liu et al., 2020) and mBART-50 (Tang et al., 2020) are most commonly used for fine-tuning. T5 and BART are trained with data in English language, and thus are not particularly useful for non-English tasks. mT5 supports over 100 languages and is available in a variety of sizes ranging from 200M to 1B parameters. On the other hand, mBART-25/50 focus on 25/50 languages and are available in only 1 size, a model with over 600M parameters. The limitation of these models is that they focus on too many languages, of which not all may be relevant for a downstream task. Additionally, language family specific models are rather attractive since they need not be very large. More recently Dabre et al. (2021b) introduce a pre-trained sequence-to-sequence model for Indic languages.

2.3.2 Machine Translation

Machine translation is perhaps the most commonly benchmarked NLG task for testing neural sequence-to-sequence models (Bahdanau et al., 2015; Vaswani et al., 2017). In the context of Indic languages, the Samanantar dataset (Ramesh et al., 2022) is the largest general domain dataset spanning 11 languages and multilingual models trained on it give high translation quality. This dataset and accompanying models are integral for the creation of the IndicNLG suite, in particular for the paraphrasing and question generation tasks. Dabre et al. (2021a) also focus on machine translation as one of the downstream tasks to evaluate their pre-trained models. Indic language translation is also the focus of shared tasks in several workshops, such as Workshop on Asian Translation (WAT) (Nakazawa et al., 2021) and Workshop on Machine Translation (WMT) (Barrault et al., 2021).

2.3.3 Document Summarization

Document summarization (Nenkova and McKeown, 2011) is done either using extractive (Cheng and Lapata, 2016; Narayan et al., 2018b) or abstractive approaches (See et al., 2017; Paulus et al., 2018). While the former focuses on extracting sentences that act as summaries, the latter focuses
on rephrasing the original content. With regard to datasets, some popular document summarization datasets are CNN-Dailymail (Hermann et al., 2015), BBC XSum (Narayan et al., 2018a) and XL-Sum (Hasan et al., 2021b). Pre-trained models significantly improve abstractive summarization quality (Lewis et al., 2020; Rothe et al., 2020), an observation that applies to the NLG tasks in this paper.

3 IndicNLG Suite

The IndicNLG suite is a collection of datasets which we use to benchmark NLG performance for 5 NLG tasks spanning 11 Indic languages. In this section, we describe the creation of the datasets that are a part of the suite. Wherever applicable, we give examples from the datasets. Text written in any Indic script is accompanied by its transliteration and translation into English in round brackets for ease of understanding.

Criteria for Choosing Tasks: Our choice of tasks depended on language coverage, task coverage and practical applications. With respect to language coverage, our priority is to include as many languages as possible, which currently we limit to 11 but plan to extend to all 22 official Indian languages. We intend to cover a variety of tasks including data-to-text, headline generation, sentence summarization, question generation and paraphrasing. These aforementioned tasks have a variety of practical applications where for example, headline generation and sentence summarization can be used to generate catchy captions for articles, paraphrasing can be used for diversity of written content and data-to-text can be useful in summarizing tabular or graphic data.

Task Performance Evaluation: Throughout the paper, we report Rouge scores (Lin, 2004), a metric that is commonly used to evaluate natural language generation quality. For simplicity, we only report Rouge-L scores. The only exception is for paraphrasing, where we use the self-BLEU and iBLEU (Sun and Zhou, 2012) metrics, which are more suitable. These are further explained in Section 5.5

3.1 Biography Generation (WikiBio)

The WikiBio task was first proposed for English, where, given the Wikipedia infobox of a person, the objective is to generate the first sentence of its Wikipedia page (Lebret et al., 2016). An infobox is a table containing facts in a key-value format. This essentially summarizes the infobox in a sentence. We create WikiBio datasets for our 9 out of 11 Indian languages of interest namely, Assamese (as), Bengali (bn), Hindi (hi), Kannada (kn), Malayalam (ml), Odia (or), Punjabi (pa), Tamil (ta) and Telugu (te). In order to create the datasets, we crawl the Wikipedia pages of the aforementioned languages, preprocess and filter them, and finally split them into the train, test, and validation sets. Table 2 contains example from Hindi WikiBio along with their English transliterations and translations. Here, we use a token “<TAG>” to enclose a field which contain the name of the attribute.

3.1.1 Dataset Creation

The dataset creation process involves collection of raw data, noise removal, serialization, filtration, cross-lingual overlap removal and splitting.

Raw Data Collection: We download the Wikipedia\(^5\) dumps for each language, parse them and save the information (metadata) of pages that have infoboxes. Next, we use the wptools API\(^6\) to

| Input |
|---|
| N | <TAG> name </TAG> मनोज कुमार पाण्डेय <TAG> office </TAG> विधायक - ऊँचाहार विधान सभा निर्वाचन क्षेत्र, उत्तर प्रदेश | ऊँचाहार, उत्तर प्रदेश <TAG> term </TAG> 2012 से 2017 <TAG> nationality </TAG> भारतीय |
| R | <TAG> name </TAG> Manoj Kumar Pandey <TAG> office </TAG> Vidhayaak - Unchaahar Vidhan Sabha Nirvachana Chhetra, Uttar Pradesh | Unchaahar, Uttar Pradesh <TAG> term </TAG> 2012 se 2017 <TAG> nationality </TAG> Bharatiya |
| T | <TAG> name </TAG> Manoj Kumar Pandey <TAG> office </TAG> MLA - Unchahar Assembly Constituency, Uttar Pradesh. Unchaahar, Uttar Pradesh <TAG> term </TAG> <TAG> nationality </TAG> Indian |

| Output |
|---|
| N | मनोज कुमार पाण्डेय, भारत के उत्तर प्रदेश की सोलहवीं विधानसभा सभा में विधायक रहे। |
| R | Manoj Kumar Pandey, bharat ke uttar pradesh ki solahnvi vidhaansabha me vidhayak rahe. |
| T | Manoj Kumar Pandey was an MLA in the 16th Legislative Assembly of Uttar Pradesh, India. |

Table 2: The table shows an example of WikiBio input-output fed to the model. Here, N stands for native script, R for Romanization, and T for translation.

\(^4\)The number of Wikipedia pages in Gujarati (gu) and Marathi (mr) were too low to ensure a substantial number of validation and testing examples, and thus these languages were excluded.

\(^5\)https://dumps.wikimedia.org

\(^6\)wptools.page(title, lang = lang_code, silent = True).get().data
extract the infoboxes and the first lines of the pages about people. The first sentence is supposed to be a simple biography of the person whom the page is about.

**Dataset Preprocessing:** The extracted data is rather unclean, as it contains spurious newline characters\(^7\), special characters\(^8\), and the values of some fields of the infobox are inside double square brackets([[]]). This necessitates cleaning of the infoboxes and the single sentences we extract. Following cleaning, we serialize the infoboxes to convert it into a text sequence. Thus, we transform data-to-text generation into a text-to-text generation setup (Kale and Rastogi, 2020; Puduppully and Lapata, 2021). We separate attribute names from the values by enclosing the attributes names within special tokens. For example, the infobox snippet represented in Table 3 (excluding the transliteration column) gets converted into “<TAG> name</TAG> अमिताभ बच्चन” where “<TAG>” and “</TAG>” are tokens indicating that the content enclosed in them are fields.

We perform spelling normalization of the words in the infoboxes and the sentences describing them. Normalizing text written in Indic scripts helps to handle texts that display a lot of quirky behavior on account of varying input methods, multiple representations for the same character, etc. There is a need to canonicalize text representation so that NLP applications can consistently handle the data.

**Dataset cleaning and splitting:** We clean the dataset by discarding examples which satisfy the following criteria: output sentence containing fewer than 5 tokens, name field is not present in the input infobox, duplicate examples, and person name is in English. Furthermore, we clean the dataset to ensure there is no leakage during training of models in a multilingual setting. Otherwise, an example in one language in the training set may have its equivalent in another language in the validation or test set.

### 3.1.2 Dataset Statistics

The number of examples for each language is given in Table 4. We compare the counts of examples with that of English WikiBio dataset (Lebret et al., 2016). We see that the Indic language WikiBio is low resource as compared to English WikiBio, with total number of examples ∼6% of the size of English WikiBio.

In Table 5, we present some quantitative statistics where, we can see that the average count of words in input, output, count of attribute-value pairs, common words, and overlap percentage are comparable between Indic and English WikiBio.

### 3.2 Headline Generation

Headline generation is a task in which, given an article, the objective is to generate an appropriate sentence that accurately depicts the article (Banko et al., 2000). The headline should be able to draw the reader’s attention while compressing information from several hundreds of words into a single phrase. While datasets for headline generation are available for English (Ao et al., 2021; Napoles et al., 2012), we are not aware of any for Indic languages, which motivated us to create a multilingual dataset for 11 languages. Table 6 shows one example from hi data set along with its transliteration (Romanization), and translation (in English) for input and output as well. We only show a snippet (first two

### Table 3: A WikiBio infobox snippet for the Indian actor Amitabh Bachchan. The complete infobox will contain several facts as key-value pairs.

| Field     | Value                                     |
|-----------|-------------------------------------------|
| name      | अमिताभ बच्चन                            |
| spouse    | जया बच्चन                                |
| father    | हरिवंश राय बच्चन                         |

### Table 4: Number of examples in the WikiBio dataset for 9 Indian languages. The total number of examples (last row) does not include examples in English, the statistics for which are obtained from Lebret et al. (2016).

| Languages | Train | Test  | Validation |
|-----------|-------|-------|------------|
| en        | 582,659 | 72,831 | 72,831     |
| as        | 1,300  | 391   | 381        |
| bn        | 4,615  | 1,521 | 1,567      |
| hi        | 5,684  | 1,919 | 1,853      |
| kn        | 1,188  | 389   | 383        |
| ml        | 5,620  | 1,835 | 1,896      |
| or        | 1,687  | 558   | 515        |
| pa        | 3,796  | 1,227 | 1,331      |
| ta        | 8,169  | 2,701 | 2,632      |
| te        | 2,594  | 854   | 820        |
| total     | 34,653 | 11,395| 11,378     |

---

\(^7\)If a sentence like “I am a boy” were to be split across 2 lines in the Wikipedia page with “I am” on one line and “a boy” on another, then the extracted data would look like “I am a boy.”

\(^8\)Words that are in bold in a Wikipedia page are in between a pair of asterisks. This is the result of markup information in the original Wikipedia page which the wpolve API does not get rid of completely.
Table 5: Quantitative statistics of the WikiBio dataset for English and 9 Indian languages. We use statistics such as the average number of words in input (\(W_{avg}^I\)), the average number of words in output (\(W_{avg}^O\)), the average number of field-value pairs (\(W_{avg}^{FV}\)), the average number of words common between input and output (\(W_{avg}^{Common}\)), and the percentage of words common between input and output (% overlap). Here, average statistics (last row) does not include English.

| Languages | \(W_{avg}^I\) | \(W_{avg}^O\) | \(W_{avg}^{FV}\) | \(W_{avg}^{Common}\) | % overlap |
|-----------|----------------|----------------|----------------|------------------|-----------|
| en        | 112.06         | 26.06          | 19.67          | 11.57            | 44.40     |
| as        | 144.48         | 16.50          | 13.27          | 4.82             | 29.21     |
| bn        | 178.24         | 18.94          | 16.95          | 5.87             | 30.99     |
| hi        | 146.50         | 17.46          | 17.62          | 6.29             | 36.03     |
| kn        | 136.30         | 13.87          | 13.83          | 3.73             | 26.89     |
| ml        | 148.89         | 14.24          | 13.83          | 3.47             | 24.37     |
| or        | 131.89         | 14.31          | 14.02          | 4.14             | 28.93     |
| pa        | 146.74         | 20.57          | 13.84          | 6.30             | 30.63     |
| ta        | 163.55         | 16.92          | 19.79          | 5.32             | 31.44     |
| te        | 119.78         | 11.69          | 12.75          | 3.69             | 31.57     |
| average   | 132.13         | 14.10          | 13.01          | 4.26             | 30.21     |

Table 5: Quantitative statistics of the WikiBio dataset for English and 9 Indian languages. We use statistics such as the average number of words in input (\(W_{avg}^I\)), the average number of words in output (\(W_{avg}^O\)), the average number of field-value pairs (\(W_{avg}^{FV}\)), the average number of words common between input and output (\(W_{avg}^{Common}\)), and the percentage of words common between input and output (% overlap). Here, average statistics (last row) does not include English.

3.2.1 Dataset Creation

For headline generation, the dataset creation process involves: raw data crawling, preprocessing and splitting.

**Crawling raw data:** The raw data for Hindi is crawled from HTML web pages of various domains like Dainik Bhaskar\(^9\), Naidunia\(^10\), NDTV\(^11\), Business Standard\(^12\) and IndiaTV\(^13\) to ensure content diversity. For other languages we used the Headline Prediction dataset from ‘IndicGLUE’ (Kakwani et al., 2020)\(^14\) which is a NLU dataset. This dataset has a ‘Document’ field paired with four candidate headlines, and the task is to choose the correct one. We repurpose this dataset for the task of generation of the correct headline given input of a news article.

**Dataset Preprocessing:** The web crawled data contains noise like irrelevant details about the publisher and the timeline of the news in multiple formats, embedded advertisements, content in multiple languages, embedded hyperlinks, and content in bullet format. Study of few examples revealed that the noisy sentences contain some specific words that can be used as keywords to detect and remove such sentences, and some sentences have a common format for date and timestamp. Most of the cleaning is done via a combination of automatic processing via regular expressions, language detection and keyword searching\(^16\). The IndicGLUE dataset sometimes contains noise in the form of data in a different language. We used gcld3\(^17\) and langId (Lui and Baldwin, 2011)\(^18\) for language detection to identify and then remove such examples. In addition, we enforce length constraints, such as the length of the document should be greater than the length of the title and the minimum length of the document should be 100 tokens.

**Dataset cleaning and splitting:** The dataset created using the above process sometimes contains noise in the form of a document matched with an incorrect headline. We notice that such examples have low percentage of words in common between title and document. In order to remove such examples, we compute the overlap between the document content and the title by an “overlapping ratio”. Suppose \(D\) and \(T\) represent the set of words in the document and title, respectively. The overlapping ratio is computed as \(\frac{|D \cap T|}{|T|}\). Table 29 and Table 30 (in Appendix) shows the statistics of the division of the dataset into buckets of different overlap ratios. We inspect few examples in each bucket for each language and accordingly decide a threshold. For Hindi, we select all sentences with ratio of overlap higher than 40% and for other languages with ratio higher than 10%. Thereafter, we split the dataset into train, dev and test sets, with the example ratio as 2:1:1.

---

\(^9\)The input article size being very large, as from Table 8, the average article length is 18.45 sentences for hi dataset.

\(^10\)https://www.bhaskar.com/

\(^11\)https://www.naidunia.com/

\(^12\)https://ndtv.in/

\(^13\)https://hindi.business-standard.com/

\(^14\)https://www.indiatv.in/

\(^15\)https://indicnlp.ai4bharat.org/indic-glue/

\(^16\)We found that keywords such as "published", "updated", "read more", "click here", ".com" etc. helped eliminate most advertisements and hyperlinks.

\(^17\)https://pypi.org/project/gcld3/

\(^18\)https://pypi.org/project/langid/
We followed Narayan et al. (2018a) to compute statistics about the corpora. Table 7 shows the dataset split in train, test and validation sets for all languages. For comparing with a benchmark English language dataset, we make use of the English portion of the XL-sum (Hasan et al., 2021b). XL-sum is originally proposed as a multilingual extreme summarization dataset. But in this scenario, we include XL-sum as it contains the headline of the news articles too.\footnote{The corresponding XSum (Narayan et al., 2018a) dataset doesn’t contain the headlines of the news articles.} The train set of \textit{bn}, \textit{gu}, \textit{hi}, \textit{kn} and \textit{mr} have samples more than 100K. The \textit{ml} train set is the smallest with 10K samples, and other language train sets fall between 20K to 60K. The average train size of the corpora is 90K. All these are smaller than XL-sum (Hasan et al., 2021b) English.

Table 8 shows the quantitative statistics of the data set. The column ‘avg. document length’ shows the average length of document in terms of words and sentences. The vocabulary size gives the average length of title in terms of words and sentences for all the languages, the column ‘vocabulary size’ gives the vocabulary size of title and the document of the data set. We see that for the Indic headline generation dataset, titles generally have only one sentence. The count of sentences ranges from 8 to 18 with an average of 12 sentences per document, corresponding to an average of 185 words. In comparison, the XL-Sum English dataset has 24 sentences and 460 words on average per document. The document vocabulary size varies from 194k for \textit{gu} to 1.2M for \textit{hi} with an average of 540k. On average, the title vocabulary size is 59k, and it is reasonable for the titles whose average length in the corpora is 9 tokens. In contrast, the XL-sum dataset has longer documents, but comparable titles. The vocabulary size is comparable to the Hindi dataset. For other Indic languages, the vocabulary sizes are smaller than XL-sum.

Table 9 shows the percentage of novel n-grams, which indicates the percentage of n-grams of title not present in the document. It is a measure of the abstractive nature of the task, as the model will be required to predict words not present in the input. We find that the Indic Headline Generation dataset is comparable with XL-Sum dataset in terms of novel n-grams.

**Baselines:** We use the LEAD-1 and EXT-ORACLE Rouge-L (R-L) scores as baselines, which also serve as an indicator of task difficulty. LEAD-1 Rouge-L scores are calculated between the first sentence of the document as system summary and title as reference summary. EXT-ORACLE scores are computed by selecting the sentence from the document as summary that give the highest rouge scores with the reference sum-
Table 8: Quantitative statistics of the headline generation dataset, focusing on document-title lengths and vocabulary sizes.

| Dataset | avg. document length | avg. title length | vocabulary size |
|---------|----------------------|-------------------|-----------------|
| XL-Sum En | 460.42 | 24.38 | 1,465,523 |
| as | 185.70 | 8.37 | 246,387 |
| bn | 199.83 | 15.19 | 614,374 |
| gu | 192.26 | 12.30 | 246,387 |
| hi | 354.07 | 18.45 | 1,465,523 |
| kn | 150.33 | 11.18 | 324,994 |
| ml | 133.79 | 13.96 | 399,927 |
| mr | 165.38 | 13.60 | 463,949 |
| or | 148.23 | 12.33 | 480,236 |
| pa | 215.42 | 8.86 | 343,036 |
| ta | 143.83 | 13.79 | 633,888 |
| te | 150.60 | 13.79 | 467,958 |
| average | 185.40 | 12.86 | 540,781 |

Table 9: Qualitative statistics for headline generation dataset, focusing on n-gram overlaps between document and title. Standard baseline scores such as LEAD and EXT-ORACLE are also included.

| Language | % of novel n-gram | LEAD-1 Rouge-L | EXT-ORACLE-1 Rouge-L |
|----------|-------------------|----------------|---------------------|
| n=1 | n=2 | n=3 | n=4 | R-L | R-L |
| XL-Sum En | 32.22 | 80.99 | 94.57 | 98.06 | 8.24 | 18.82 |
| as | 36.86 | 71.53 | 86.98 | 93.97 | 13.95 | 28.56 |
| bn | 46.38 | 78.92 | 90.39 | 94.77 | 13.05 | 27.83 |
| gu | 40.49 | 73.17 | 86.18 | 91.69 | 16.50 | 29.13 |
| hi | 25.93 | 65.62 | 83.67 | 91.54 | 21.67 | 30.51 |
| kn | 39.42 | 72.76 | 87.29 | 93.59 | 17.89 | 29.42 |
| ml | 44.24 | 78.79 | 91.77 | 96.12 | 17.93 | 30.42 |
| mar | 39.39 | 73.49 | 88.33 | 93.89 | 11.32 | 31.94 |
| or | 45.60 | 79.61 | 90.23 | 93.49 | 12.79 | 27.33 |
| pa | 24.67 | 57.82 | 73.10 | 81.39 | 28.01 | 35.41 |
| te | 42.99 | 70.84 | 82.25 | 87.43 | 31.37 | 37.98 |
| average | 38.70 | 71.68 | 85.08 | 90.65 | 19.50 | 31.72 |

3.3 Sentence Summarization

Sentence summarization involves compressing the information of a reasonably long sentence into a shorter, compact sentence (Rush et al., 2015; Chopra et al., 2016). Following Rush et al. (2015), we create a sentence summarization dataset where the input is the first sentence of a news article and the output is its headline. The intuition is that the first sentence in a news article often expands upon the information in the headline. Table 10 contains an example from Hindi sentence summarization dataset along with its English transliteration and translation.
3.3.1 Dataset Creation
We extract the first sentence of the documents in the headline generation dataset (from Section 3.2) as the input and its headline as the output. When we study some examples extracted in this manner, we notice that there are cases where the input sentence contains unrelated, or partially related information compared to the headline (henceforth the sentence summary). This is also partly attested to by the lower LEAD-1 scores for Headline Generation in Table 9. We compute ratio of overlap between words in sentence and summary similar to that in Section 3.2.1 and include examples which are above certain threshold. Table 31 and Table 32 (in Appendix) contains the division of the dataset into buckets of different overlap ratios. We study few examples in each bucket to define the threshold for overlap ratio. We include examples with more than 50% overlap for Hindi and more than 60% overlap for other languages. In addition, we ensure that the count of words in the sentence is higher than that of the summary.

3.3.2 Dataset Statistics
Table 11 shows the count of examples in train/test/validation split for sentence summarization for different languages. The count of examples in the training set ranges from 2.8k for Malayalam to 78.8k for Hindi, with a total count of 326k for all languages. The Gigaword\textsuperscript{20} corpus for sentence summarization for English (Rush et al., 2015) corpus is an order of magnitude larger than the total count of examples in our sentence summarization dataset. Table 12 shows some quantitative statistics for the sentence summarization dataset. The count of words in title and sentence is comparable to that of English Gigaword corpus. The size of vocabulary of sentence and summary is smaller than that of English Gigaword corpus.

We compute the percentage of n-grams which are novel in summary as compared to the sentence in Table 13. We see that the languages in our dataset have much higher percentage of novel n-grams as compared to the English Gigaword corpus. We calculate LEAD Rouge-L scores for sentence summarization as, the rouge score between the summary and the first k words from the corresponding sentence, where k is the average length of the summary in terms of words in each dataset.

Table 11: Size of train, test and validation sets in terms of number of samples for sentence summarization. The total size of the dataset, excluding English, is also included.

| Language | Train | Test | Validation |
|----------|-------|------|------------|
| Gigaword En | 3,803,957 | 1,951 | 189,651 |
| as | 10,812 | 5,452 | 5,232 |
| bn | 17,035 | 2,384 | 2,355 |
| gu | 54,788 | 8,460 | 8,720 |
| hi | 78,876 | 16,935 | 16,935 |
| kn | 61,220 | 1,485 | 9,024 |
| ml | 2,855 | 1,580 | 1,520 |
| mr | 27,066 | 3,309 | 3,249 |
| or | 12,065 | 1,440 | 1,539 |
| pa | 31,630 | 3,967 | 4,004 |
| ta | 23,098 | 2,948 | 2,874 |
| te | 7,119 | 862 | 878 |
| total | 326,564 | 48,722 | 56,330 |

Table 12: Quantitative statistics for sentence summarization, focusing on the lengths of the sentence-title pairs in terms of words, as well as the vocabulary sizes.

| Languages | Average words | Vocabulary Size |
|-----------|---------------|-----------------|
| Gigaword En | 31.35 | 8.23 |
| as | 38.43 | 7.61 |
| bn | 27.88 | 9.27 |
| gu | 31.69 | 10.05 |
| hi | 30.70 | 10.04 |
| kn | 27.56 | 8.11 |
| ml | 18.33 | 9.02 |
| mr | 22.90 | 6.93 |
| or | 21.78 | 7.56 |
| pa | 37.04 | 12.88 |
| ta | 19.79 | 9.10 |
| te | 19.55 | 6.02 |
| average | 26.88 | 8.78 |

The average LEAD score for the corpora is 25.87 which is comparable to the LEAD Rouge-L scores for the Gigaword corpus.

3.4 Paraphrase Generation
Paraphrase generation or paraphrasing (McKeown, 1983; Barzilay and Lee, 2003) is the task of transforming a sentence into a different sentence in the same language while preserving meaning and semantics. A paraphrasing system is important as it enables generation of alternatives for a given sentence, and this can be especially useful to language learners so that they may freely express themselves (Mayhew et al., 2020). Table 14 contains an example from Hindi paraphrase generation dataset along with its English transliteration and translation.

3.4.1 Dataset Creation
Approach: We use the pivoting approach, which is the simplest approach for paraphrase data creation (Zhao et al., 2008). In this approach, we assume the

\textsuperscript{20}https://huggingface.co/datasets/gigaword
Table 13: Qualitative statistics for sentence summarization, focusing on n-gram overlaps between the sentence-summary pairs. Baseline scores using the first “k” words of the sentence as a summary are also computed.

| Languages | % of novel n-gram | LEAD |
|-----------|-------------------|-----|
|           | n=1   | n=2   | n=3   | n=4   | R-L  |
| Gigaword En | 21.46 | 40.26 | 46.52 | 48.67 | 23.14 |
| as        | 35.63 | 66.47 | 83.21 | 91.93 | 17.82 |
| bn        | 40.48 | 66.66 | 81.36 | 89.15 | 32.33 |
| gu        | 38.07 | 65.19 | 79.42 | 86.39 | 29.63 |
| hi        | 31.36 | 66.15 | 82.41 | 90.49 | 25.73 |
| kn        | 37.27 | 67.31 | 83.28 | 91.31 | 16.40 |
| ml        | 41.82 | 67.00 | 81.93 | 89.87 | 33.67 |
| mr        | 38.23 | 71.54 | 87.64 | 94.15 | 32.33 |
| or        | 39.67 | 69.34 | 84.38 | 91.60 | 18.95 |
| pa        | 32.54 | 60.19 | 73.29 | 81.01 | 31.61 |
| ta        | 40.55 | 66.78 | 80.75 | 88.12 | 37.41 |
| te        | 39.79 | 71.65 | 88.19 | 94.62 | 24.64 |
| average   | 37.76 | 67.12 | 82.35 | 89.88 | 25.87 |

Table 14: The table shows an example of paraphrase generation input-output example fed to the model. Here, N stands for native script, R for romanization, and T for translation.

Table 15: Language-wise splits for the Samanantar corpus, which we use for extracting paraphrases.

| Language Pair       | #Sentence Pairs |
|---------------------|-----------------|
| en-as               | 0.14M           |
| en-bn               | 8.52M           |
| en-gu               | 3.05M           |
| en-hi               | 8.56M           |
| en-kn               | 4.07M           |
| en-mi               | 5.85M           |
| en-mr               | 3.32M           |
| en-or               | 1.00M           |
| en-pa               | 2.42M           |
| en-ta               | 5.16M           |
| en-te               | 4.82M           |

Initial extraction: We use the Samanantar corpus (Ramesh et al., 2022) which contains parallel corpora between English and all 11 Indic languages of interest\(^2\). Table 15 contains the count of parallel pairs for each language with en. We use the pivoting approach to extract the initial set of paraphrases. Prior to pivoting, we normalize and tokenize the English sentences using Sacremoses\(^2\) along with removing white spaces to ensure that the same sentences with differing spaces between words become identical. A single paraphrase example is a tuple consisting of \(M\) sentences which are all considered to be paraphrases of each other. We subject this initial set of paraphrases to rigorous filtering/cleaning to ensure a high quality paraphrasing dataset.

Dataset cleaning and splitting: After paraphrase extraction, we clean it to remove noise. First, we remove the sentences in an example which are exact duplicates that only differ due to tokenization and spelling. For this, we first normalize and tokenize the sentences using the IndicNLP library (Kunchukuttan, 2020)\(^2\). Then, we remove the punctuation and white spaces from the sentence. If this results in a single sentence in the example, then the example is discarded.

We then randomly select one paraphrase as the input from each group of paraphrases and calculate n-gram overlap for \(n=1, 2, 3,\) and 4 between it and the remaining sentences which are considered as references. We eliminate the paraphrases which have an n-gram overlap ratio greater than 0.8 to ensure that the paraphrases are not very similar. The ratio is calculated using the formula:

\[
a_n = \frac{O_n}{I_n}
\]

\[
b_n = \frac{O_n}{R_n}
\]

\[
\text{ratio} = \frac{\sum_{n=1}^{4} a_n + b_n}{4}
\]

where \(O_n\) is n-gram overlap between reference and input, \(I_n\) is total n-grams in input and \(R_n\) is total n-grams in reference. This formula computes the average of 1-, 2-, 3-, and 4-gram overlaps. These overlaps are computed as the harmonic mean between the overlapping n-grams relative to the input.

---

\(^1\)https://indicnlp.ai4bharat.org/samanantar
\(^2\)https://github.com/alvations/sacremoses
\(^2\)https://github.com/anoopkunchukuttan/indic_nlp_library
and the reference \((b_n)\). This ensures that the overlap information is not biased towards either the input or the reference.

Next, for each input, we select up to five references. We first sort the references based on the n-gram overlap scores with respect to the input, then divide the scores into 5 equal intervals and finally, select the example corresponding to the first score in each interval. If for a particular input, the number of references is less than 5, we keep all the references. Following this, we split the collection of examples into train, validation and test sets. We first sort examples in the descending order of the number of paraphrases in them, the top 10,000 go into the test set, the next 10,000 into the validation set and the remaining into the training set. Except Assamese, all languages have 10,000 examples with 5 references per input in the validation and test set. The training set has anywhere between 1 and 5 references. Assamese, due to its low-resource nature, could only be split into validation and test sets with 4,420 examples each.

The per language statistics are given in Table 16. The number of training examples range from 105k for Oriya to 929k for Hindi, with a total of 5.3M examples across all languages. Note that if an instance contains \(M+1\) sentences (1 input and \(M\) references) then, all are paraphrases of each other and there is no restriction on using any pair of sentences as input and output. However, for simplicity, we designate a sentence as input and the remaining as references. Furthermore, the references are ordered according to the n-gram overlap with respect to the input and the first reference is the most dissimilar while the last one is the most similar. We compare our statistics with an English language paraphrase corpus. Specifically, we compare with the English portion of OpusParcus corpus (Creutz, 2018). In the English portion of OpusParcus corpus, each instance consists of only two paraphrases, out of which one is chosen as the input and the other as the reference. Dataset statistics: Table 17 gives the percentage of novel n-grams, i.e. n-grams in the reference not present in the input. The statistics show high percentage of novel n-grams for higher n-grams \((n=2,3,4)\). We compare the statistics with the English portion of the OpusParcus Test set (Creutz, 2018). The statistics are comparable between English and Indic languages.

### 3.5 Question Generation

In this work, we adopt the setup of given a sentence as context along with a phrase/word as answer, the task is to generate a question which can produce the answer (Du et al., 2017; Zhou et al., 2017a). Table 18 contains an example from Hindi question generation dataset along with its English translation and translation. Since the input is the concatenation of the answer and the sentence context, we use the token “[SEP]” to separate them.

**Dataset creation:** Following earlier work (Du et al., 2017; Zhou et al., 2017a; Dong et al., 2019 inter alia), we start with the SQuAD (Rajpurkar et al., 2016) question answering dataset repurposed to serve as a question generation dataset. Specifically, we extract the question, its answer and the sentence containing the answer. We construct the input to question generation as the sentence along with the answer. The question serves as the output. We then translate the data into Indic languages using the IndicTrans (Ramesh et al., 2022)25 English to Indic model (the details are in Appendix A.2). The number of examples in train, validation and test sets in each language is 69979, 17495, and 10553 respectively. One limitation of the dataset is that the testset is machine translated given the lack of QA benchmarks for Indian languages. IndicTrans is a high-quality MT system, which should make the testset acceptable to benchmark performance of question generation systems.

**Dataset statistics:** Table 19 gives the novel n-gram percentage between question and context. We can see that Indic languages have higher novel n-grams when compared to the English dataset.

### 3.6 Summary of Datasets

To summarize, the IndicNLG Suite contains diverse NLG tasks for multiple Indic languages that vary in their linguistic characteristic and resource availability. While the corpora are smaller than their English counterparts in some cases, they are of reasonable size for building a standard dataset. Moreover, the relatedness of Indic languages opens up the possibility of training multilingual generation models. To ensure fairness of the dataset in the multilingual scenario, we believe that there should be few to none overlaps between train and test sets.

---

24 Except for Oriya’s validation set which has 4 or fewer references for a few examples.

25 [https://github.com/AI4Bharat/indicTrans](https://github.com/AI4Bharat/indicTrans)
Table 16: Language-Wise Statistics for paraphrase generation dataset. Each instance consist of up to 6 paraphrases, of which one is chosen as the input and the rest are references, ordered according to the least n-gram overlap.

| Language | #Instances | #Total Paraphrases | #Instances | #Total Paraphrases | #Instances | #Total Paraphrases |
|----------|------------|--------------------|------------|--------------------|------------|--------------------|
| OpusParcus En | - | - | 1,445 | 2,890 | - | - |
| as | 890,445 | 2,837,641 | - | - | 4,420 | 12,965 |
| bn | 397,202 | 1,145,079 | 10,000 | 60,000 | 10,000 | 60,000 |
| gu | 929,507 | 2,690,315 | 10,000 | 60,000 | 10,000 | 60,000 |
| hi | 522,148 | 1,683,599 | 10,000 | 60,000 | 10,000 | 60,000 |
| kn | 761,933 | 2,426,155 | 10,000 | 60,000 | 10,000 | 60,000 |
| ml | 406,003 | 1,263,050 | 10,000 | 60,000 | 10,000 | 60,000 |
| mr | 266,704 | 734,573 | 10,000 | 60,000 | 10,000 | 60,000 |
| or | 497,798 | 1,631,560 | 10,000 | 60,000 | 10,000 | 60,000 |
| pa | 596,283 | 1,977,942 | 10,000 | 60,000 | 10,000 | 60,000 |

Table 17: Percentage of novel n-grams in references not present in the input for paraphrase generation.

| Language | % of novel n-grams |
|----------|---------------------|
| OpusParcus En | n=1 | n=2 | n=3 | n=4 |
| bn | 63.22 | 92.23 | 98.24 | 99.72 |
| gu | 397,202 | 1,145,079 | 10,000 | 60,000 | 10,000 | 60,000 |
| hi | 929,507 | 2,690,315 | 10,000 | 60,000 | 10,000 | 60,000 |
| kn | 522,148 | 1,683,599 | 10,000 | 60,000 | 10,000 | 60,000 |
| ml | 761,933 | 2,426,155 | 10,000 | 60,000 | 10,000 | 60,000 |
| mr | 406,003 | 1,263,050 | 10,000 | 60,000 | 10,000 | 60,000 |
| or | 266,704 | 734,573 | 10,000 | 60,000 | 10,000 | 60,000 |
| pa | 497,798 | 1,631,560 | 10,000 | 60,000 | 10,000 | 60,000 |

Table 18: The table shows an example of question generation input-output example fed to the model. Here, N stands for native script, R for romanization, and T for translation.

| Input | Output |
|-------|--------|
| Denver Broncos [SEP] American Football Conference (AFC) champion Denver Broncos defeated National Football Conference (NFC) champion Carolina Panthers 24-10 to win their third Super Bowl title. | Which NFL team represented the AFC in Super Bowl 50? |
| सुपर बाउल 50 में आंध्र प्रदेश टीम ने एएफसी का प्रतिनिधित्व किया? | सुपर बाउल 50 में किस एनएफएल टीम ने एएफसी का प्रतिनिधित्व किया? |

4.1 Models Compared

We compare the following models for all languages and tasks:

- Monolingual model (no pre-training)
- Multilingual model (no pre-training)
- Monolingual fine-tuned models
- Multilingual fine-tuned models

We chose these baselines as they provide insights into the performance of multilingual models and
Table 19: The table shows the percentage of novel n-gram for question generation. We give the statistics of novel n-gram in question string compared to context string.

| Language | Question And Context Overlap |
|----------|-----------------------------|
|          | n=1 | n=2 | n=3 | n=4 |
| en       | 63.33 | 85.35 | 92.40 | 95.61 |
| as       | 78.21 | 93.01 | 97.51 | 98.96 |
| bn       | 74.22 | 91.20 | 96.75 | 98.66 |
| gu       | 76.95 | 92.18 | 97.04 | 98.78 |
| hi       | 62.61 | 85.17 | 93.15 | 96.53 |
| kn       | 78.10 | 92.91 | 97.46 | 98.95 |
| ml       | 79.80 | 93.33 | 97.68 | 99.11 |
| mr       | 76.66 | 93.04 | 97.45 | 98.98 |
| or       | 74.22 | 92.53 | 97.24 | 98.87 |
| pa       | 66.76 | 87.57 | 94.63 | 97.47 |
| ta       | 80.15 | 93.69 | 97.83 | 99.16 |
| te       | 76.58 | 92.22 | 97.21 | 98.90 |
| average  | 75.33 | 91.53 | 96.72 | 98.38 |

finetuning of pre-trained models - strategies that are important for low-resource languages.

4.2 Pre-trained Models Used

For our experiments with finetuning of pre-trained models, we compare IndicBART, a language-group specific model trained specifically for Indic languages with mT5, a general pre-trained model for 100+ languages.

**IndicBART**: IndicBART (Dabre et al., 2021a) is an encoder-decoder pre-trained model that focuses only on Indic languages. In fact, it covers all 11 languages we focus on in this paper. It is trained in the same way as mBART (Liu et al., 2020), using the text-infilling based denoising objective. IndicBART has two variants, unified script and separate script:

1. **Unified script**: This model is trained on Indic language data, where all the text is transliterated into the Devanagari script. This model is the official IndicBART model, abbreviated as IB henceforth. We convert our data into Devanagari script before fine-tuning and convert the model outputs back to original script. We make use of the Indic NLP library (Kunchukuttan, 2020) for the script conversion.

2. **Separate script**: This model is equivalent to IB except that the data with original Indic language scripts were used. This model is the called the separate script IndicBART model, abbreviated as SSIB henceforth. For fine-tuning, no special processing is needed.

**mT5**: mT5 (Xue et al., 2021) is the multilingual version of the T5 model. It is an encoder-decoder model, trained using the span-prediction denoising approach. mT5 covers 101 languages and is available in a variety of sizes. In the absence of a smaller model, we choose the mT5-small model containing 300 million parameters as we wish to have a fair comparison with the IndicBART model which contains 244 million parameters.

Currently, we do not experiment with mBART (Liu et al., 2020) as it does not cover as many languages as IndicBART or mT5. It also contains over 600M parameters, which would make comparison against IndicBART and mT5 unfair given that they contain fewer parameters compared to mBART.

4.3 Implementations Used

We use the YANMTT toolkit (Dabre and Sumita, 2021) to train monolingual and multilingual models from scratch, as well as for fine-tuning using IndicBART. For fine-tuning mT5, we use the Huggingface (Wolf et al., 2020) implementation for summarization.

4.4 Training Settings

We describe the settings for training models from scratch, as well as for fine-tuning IndicBART and mT5 models. As much as possible, we give details about hyperparameter settings and training convergence. In general, we tuned hyperparameters for models trained from scratch for a given task according to corpora sizes, where we used larger models for tasks with larger amounts of data.

4.4.1 WikiBio

**Training from scratch**: Given the relatively smaller corpora size for WikiBio, we use Transformer (Vaswani et al., 2017) with a smaller capacity when training from scratch. Specifically, we employ 4 encoder and decoder layers with hidden and filter sizes of 512 and 2048 respectively, and 8 attention heads. We use dropouts of 0.1, label smoothing of 0.1, learning rate of 0.0003 and weight decay of 0.00001 with the ADAM optimizer. We use maximum source and target length of 512 and 64 respectively. We train till convergence on

---

26 [https://github.com/anoopkunchukuttan/indic_nlp_library](https://github.com/anoopkunchukuttan/indic_nlp_library)

27 [https://github.com/prajdabre/yanmtt](https://github.com/prajdabre/yanmtt)

28 [https://github.com/huggingface/transformers/tree/master/examples/pytorch/summarization](https://github.com/huggingface/transformers/tree/master/examples/pytorch/summarization)
the validation set scores, which are computed via greedy decoding every 1,000 batches. For decoding the test sets, we use beam search with a beam of size 4, length penalty of 1.0 and penalize translations in the beam where 4-grams are repeated. In case of multilingual models, the convergence is determined by the average of validation set scores for all languages.

Training via fine-tuning: The settings for fine-tuning IndicBART are the same as for scratch models except that, IndicBART uses 6 encoder and decoder layers with hidden and filter sizes of 1024 and 4096 respectively, and 16 attention heads. On the other hand, we train mT5 for 10 epochs and choose the checkpoint with the highest validation set scores.

4.4.2 Headline Generation

Training from scratch: We use 6 encoder and decoder layers with hidden and filter sizes of 512 and 2048 respectively, and 8 attention heads. We use label smoothing, dropouts, attention dropout, and activation dropout 0.1 for all. The learning rate of 0.0003 and weight decay of 0.00001 with the ADAM optimizer. We train all the models till convergence on the validation set scores, which are computed via greedy decoding every 4096 batches. For decoding the test sets, we use beam search with a beam of size 5, length penalty of 1.0 and penalize translations in the beam where 3-grams are repeated.

Training via fine-tuning: The settings for fine-tuning IndicBART are the same as for scratch models except that, we use 16 attention heads and hidden and filter sizes of 1024 and 4096 respectively. On the other hand, we train monolingual mT5 for 10 epochs with weight decay of 0.01 and choose the checkpoint with the highest validation set scores. We train multilingual mT5 for 10 epochs and choose the last checkpoint.

4.4.3 Sentence Summarization

Training from scratch: Given the larger corpora sizes, we use 6 encoder and decoder layers with hidden and filter sizes of 512 and 2048 respectively, and 8 attention heads. We use dropouts of 0.1, label smoothing of 0.1, learning rate of 0.0005 and weight decay of 0.00001 with the ADAM optimizer. We train all the models till convergence on the validation set scores, which are computed via greedy decoding every 1,000 batches. Similar to WikiBio, in the case of multilingual models, the convergence is determined by the average of validation set scores for all languages. For decoding the test sets, we use beam search with a beam of size 5, length penalty of 0.8 and penalize translations in the beam where 3-grams are repeated.

Training via fine-tuning: The settings for fine-tuning IndicBART are the same as for scratch models except that, IndicBART uses 6 encoder and decoder layers with hidden and filter sizes of 1024 and 4096 respectively. On the other hand, we train monolingual mT5 for 10 epochs with weight decay of 0.01 and choose the checkpoint with the highest validation set scores. We train multilingual mT5 for 10 epochs and choose the last checkpoint.

4.4.4 Paraphrase Generation

Most settings are the same as for sentence summarization. We train monolingual and multilingual mT5 for 5 and 10 epochs, respectively. For decoding the test sets, we use beam search with a beam of size 4, length penalty of 1.0.

4.4.5 Question Generation

Training from scratch: Other than using maximum source and target length of 256 and 32 respectively, we used the same settings as for WikiBio. Training via fine-tuning: The settings for fine-tuning IndicBART were the same as for scratch models except that, IndicBART uses 6 encoder and decoder layers with hidden and filter sizes of 1024 and 4096 respectively, and 16 attention heads. On the other hand, we train mT5 for 10 epochs and choose the checkpoint with the highest validation set scores.

4.5 Evaluation Metrics

The Rouge metric (Lin, 2004) is the most popular evaluation metric for natural language generation tasks, and we follow suit in this paper. We report the Rouge-L F1 score. In order to compute Rouge scores for the decoded test sets, we use the multilingual rouge scoring implementation (Hasan et al., 2021b) which enables segmentation, stem-
Table 20: The table shows Rouge-L scores for the WikiBio test sets. We compare between models without pre-training (no PT), IndicBART (IB), separate script IndicBART (SSIB) and mT5 models in monolingual and multilingual settings.

| Language | Rouge-L | Monolingual | Multilingual |
|----------|---------|-------------|--------------|
|          | No PT   | mT5 | SSIB | IB | No PT | mT5 | SSIB | IB |
| as       | 13.67   | 49.53 | 55.21 | 55.68 | 49.16 | 56.48 | 56.50 | 56.28 |
| bn       | 15.32   | 57.51 | 56.83 | 56.84 | 51.90 | 57.99 | 56.58 | 57.42 |
| hi       | 57.49   | 67.08 | 67.16 | 65.86 | 52.79 | 57.57 | 67.34 | 67.48 |
| kn       | 4.57    | 36.13 | 37.27 | 38.46 | 33.04 | 42.47 | 39.37 | 40.01 |
| ml       | 6.05    | 38.79 | 37.42 | 37.95 | 33.34 | 57.99 | 38.42 | 38.84 |
| or       | 23.33   | 61.87 | 69.82 | 65.79 | 62.69 | 69.49 | 70.71 | 67.13 |
| pa       | 15.69   | 52.85 | 50.40 | 49.57 | 46.40 | 53.33 | 52.78 | 52.88 |
| ta       | 44.14   | 51.60 | 51.14 | 51.69 | 47.26 | 52.36 | 51.11 | 51.82 |
| te       | 10.39   | 51.53 | 50.89 | 50.25 | 43.79 | 52.22 | 51.72 | 51.43 |
| average  | 21.18   | 51.88 | 52.90 | 52.45 | 47.82 | 54.61 | 53.84 | 53.70 |

5 Results and Analysis

In this section, we present the results obtained using models trained for a variety of tasks and analyze them from various perspectives. We compare between models without pre-training (no PT), IndicBART (IB), separate script IndicBART (SSIB) and mT5 models in monolingual and multilingual settings. We compute the Rouge-L scores for the model outputs.

5.1 Research Questions

In the analysis presented below, we try to answer the following research questions:

| Impact of pre-training: Are pre-trained models better compared to models trained from scratch? |
| Impact of multilingualism: How do monolingual models compare with multilingual models? There are 3 types of models to compare: (a) monolingual models trained from scratch, (b) monolingual models fine-tuned using multilingual pre-trained models, and (c) multilingual models fine-tuned using multilingual pre-trained models. |
| Impact of language family: Are language family specific pre-trained models (IndicBART) better than universal pre-trained models (mT5)? |
| Impact of script unification: How do single script pre-trained models compare to multi-script pre-trained models? |
| Impact of task nature on performance: What are the determiners of the task performance? For this, we try to compare across tasks and languages and provide insights. |

5.2 WikiBio

We report the Rouge-L scores on the test set for 9 languages (as, bn, hi, kn, ml, or, pa, ta, te) in Table 20. In general, we observe high scores for all languages, especially when pre-trained models are used in a multilingual setting.

5.2.1 Analysis

| Impact of pre-training: We see that models without pre-training (No PT) obtains lower results as compared to pretrained models (IB, SSIB and mT5). This is especially the case of languages such as as, bn, kn, ml, or, pa, te where the count of examples is relatively low. The language with larger count of training examples such as hi and ta perform reasonably well in comparison. |
| Impact of multilingualism: We see that multilingual models outperform the corresponding monolingual models. Thus, we can see benefits of cross-lingual knowledge transfer. This is more noticeable in the case of models without pretraining where multilingual models outperform monolingual models of some languages (as, bn, kn, ml, or, pa, te) by a huge margin. |
| Impact of language family: For the monolingual models, IB performs better than mT5 in as, hi, kn, or, and ta. But, for the multilingual models, mT5 has the upper hand in all the languages except as and or. |
| Impact of script unification: We compare between unified script IndicBART (IB) and separate script IndicBART (SSIB). For the monolingual case, script unification helps in as, kn, ml, and ta. In case of multilingual models, script unification helps in bn, hi, kn, ml, pa, and ta. |
| Impact of task nature on performance: The high rouge scores are a strong indicator that the models |
are able to perform well on this task. Furthermore, even without pre-training, the multilingual models perform well with rouge scores above 30. The most mostly involves generating a sentence from a limited subset of the key-values in the infobox. A more challenging version of this task may be the generation of multiple sentences from the infobox, which we leave for future work.

5.3 Headline Generation

We report Rouge-L scores of all monolingual and multilingual experiments in Table 21. Similar to biography generation, we observe high scores for all languages except Oriya. We do not see benefits from multilingualism for this task.

5.3.1 Analysis

Impact of pre-training: Pre-training helps both monolingual and multilingual models. But in the case of monolingual models, the difference between the average rouge score of No PT and SSIB is 14.57, and in the case of multilingual models, it is 5.4. Thus, pre-training helps more in the case of monolingual models than multilingual ones.

Impact of multilingualism: The multilingual mT5 model outperforms the monolingual mT5 model by a significant margin of 7.03 rouge score on average. But for some datasets, the difference is extremely high. For example, as has a difference of 25.45 in their rouge score between multilingual mT5 and monolingual mT5. On the other hand, for some datasets like gu, hi, or, and pa, the difference is less than 1.0 rouge point.

Multilingual no PT also outperforms monolingual no PT by a difference of 3.54 rouge point on average. The examples of languages where multilingual No PT is better than monolingual No PT include ml with a difference of 34.13 Rouge L and or with a difference of 11.05 Rouge L. For as, bn and kn, the rouge score of monolingual No PT is better than multilingual No PT, with a difference of 23.28 Rouge L for as.

In both cases, multilingual SSIB and IB models, underperforms monolingual SSIB and IB. For SSIB the difference in the rouge scores varies across individual datasets. The difference is 21.36 rouge points for as, whereas for ta both are almost equal. Similarly, for IB, except or and ta, the rouge scores of multilingual IB are less than monolingual IB. The difference is highest for as that is 26.6 rouge points.

Impact of language family: The monolingual IB outperforms monolingual mT5 with a margin of 9.65 rouge points on average. However, multilingual IB underperforms multilingual mT5 by a smaller margin of 3.01 rouge point on average. We see that monolingual IB is consistently better for all the languages in the task of headline generation.

Impact of script unification: The unified-script IB outperforms the SSIB model in almost all cases, showing that script-mapping has a strong impact. On average, the scores do not have high differences, but the differences are noticeable individually for some languages. For example, in a monolingual setting, as and bn show higher improvements with IB.

Impact of task nature on performance: The task is to generate news headlines using long news articles. Technically, the task is a summarization task from a document to a single sentence. In the recent times, there is active research in
summarization approaches in English language. But this task is challenging given the diversity and complexity of Indian languages compared to English. We see that the models perform generally well for almost all the languages, with an average rouge score above 30. We also see that pre-training helps, resulting in even higher rouge scores.

5.4 Sentence Summarization

We report the Rouge-L scores on the test set for all the languages in Table 22. In this case, we observe that multilingualism and pre-training have a large impact on the final Rouge-L scores.

5.4.1 Analysis

Impact of script unification: In monolingual settings, script unification helps in hi, ml, and pa. In multilingual settings, script unification helps in kn, ml, and pa.

Impact of pre-training: The pre-trained models outperform the model trained from scratch by a vast margin, especially in monolingual settings. The model trained from scratch underperforms even for high resource languages such as gu, hi, and kn. Thus, we see that pre-training has a strong positive impact on this task.

Impact of language family: Monolingual IB outperforms the corresponding mT5 model by a considerable margin for all the languages except kn. In multilingual settings, IB obtains better results for seven of the languages, and mT5 fares better for the remaining languages.

Impact of multilingualism: We see that multilingualism has a positive impact. Multilingual training helps highly in the case of the No PT and mT5 models.

Impact of task nature on performance: The task is to generate the title given the first sentence of the news article. The task is less challenging than that of Headline Generation, as the length of the input is smaller. The monolingual models trained from scratch perform well, with an average score of above 30. A combination of pre-training and multilingualism helps, resulting in high rouge scores.

5.5 Paraphrase Generation

We compute the self-BLEU and iBLEU (Sun and Zhou, 2012) following earlier work (Hosking and Lapata, 2021). We report the self-BLEU and iBLEU scores on the test set for all the languages in Table 23 and 24. In order to compute the BLEU scores on the decoded test sets, we use sacreBLEU (Post, 2018). For self-BLEU we compute BLEU(output, input). The iBLEU scores are calculated as: $iBLEU = \alpha \ast BLEU(O, R) - (1 - \alpha) \ast BLEU(O, I)$, where $O =$output, $R =$references, $I =$input and, $\alpha = 0.7$. For calculating the BLEU(O, R) scores, we compare against all five references, except Assamese. Since we had no training data for Assamese, we use Bengali model to decode Assamese test set as both are similar languages. Note that lower self-BLEU and higher iBLEU indicates that paraphrasing quality is better.

Overall, we see that multilingual fine-tuning leads to paraphrases with substantial differences from the original inputs while preserving meaning.

---

Table 22: The table shows Rouge-L scores for the sentence summarization test sets. We compare between models without pre-training (no PT), IndicBART (IB), separate script IndicBART (SSIB) and mT5 models in monolingual and multilingual settings.
5.5 Analysis

We use iBLEU as the primary metric for analysis. 

Impact of script unification: The unified script IndicBART model outperforms the separate script IndicBART in almost all cases, showing that script-mapping has a strong impact. 

Impact of pre-training: The unified script IB performs better than No PT, especially in the monolingual settings. In monolingual settings, SSIB gives better results than No PT for half of the languages, whereas in the Multilingual settings, it performs better for all the languages except as and ml. 

Impact of language family: IndicBART outperform mT5 by a considerable margin, showing that language family-specific models work better than universal models for paraphrase generation. IndicBART focuses only on Indic languages, whereas mT5 covers 101 languages. The relatedness between the Indic languages helps. 

Impact of multilingualism: Multilingual training helps in the case of IB and mT5. 

Impact of task nature on performance: The task is to generate an output sentence with the same meaning as the input sentence, but lexically different. The task is challenging as the model should learn to produce diverse paraphrases and not just make trivial changes to the input. The low iBLEU scores indicate the challenging nature of the task.

5.6 Question Generation

We report the Rouge-L scores on the test set for all the languages in Table 25. Once again, multilingualism and pre-training shows its strong impact on the quality of questions generated. Compared to...
Table 25: Rouge-L scores for question generation. We compare between models without pretraining (no PT), IndicBART (IB), separate script IndicBART (SSIB) and mT5 models in monolingual and multilingual settings.

| Language | Rouge-L |
|----------|---------|
|          | Monolingual | Multilingual |
|          | No PT | mT5 | SSIB | IB | No PT | mT5 | SSIB | IB |
| as       | 11.89 | 19.69 | 20.33 | 20.21 | 15.36 | 20.90 | 20.73 | 20.48 |
| bn       | 14.46 | 29.56 | 26.61 | 26.63 | 20.63 | 29.64 | 30.38 | 26.63 |
| gu       | 15.92 | 26.31 | 25.24 | 25.24 | 22.53 | 26.19 | 28.13 | 27.71 |
| hi       | 21.67 | 34.58 | 33.60 | 32.24 | 29.14 | 34.14 | 34.42 | 35.38 |
| kn       | 11.31 | 23.32 | 21.32 | 22.40 | 18.18 | 23.06 | 23.77 | 23.56 |
| ml       | 9.94  | 21.82 | 19.87 | 19.71 | 16.93 | 21.55 | 22.24 | 22.17 |
| mr       | 12.18 | 22.81 | 21.13 | 20.61 | 18.37 | 22.67 | 23.62 | 23.52 |
| or       | 14.87 | 20.34 | 25.70 | 24.29 | 20.30 | 20.90 | 27.53 | 25.25 |
| pa       | 18.91 | 29.72 | 30.74 | 30.59 | 26.08 | 29.96 | 32.53 | 32.10 |
| ta       | 5.09  | 22.84 | 21.24 | 21.24 | 12.01 | 22.61 | 23.49 | 22.98 |
| te       | 12.38 | 25.63 | 23.15 | 24.46 | 19.71 | 25.01 | 25.81 | 25.67 |
| average  | 13.51 | 25.15 | 24.45 | 24.23 | 19.98 | 25.15 | 26.60 | 25.95 |

headline generation and sentence summarization, the scores are lower, showing that the task is more difficult.

5.6.1 Analysis

Impact of pre-training: The No PT model underperforms, especially the monolingual models, with the lowest score for ta. Pre-trained models perform better than models trained from scratch. This is clear, seeing how the monolingual pre-trained model outperforms the No PT model considerably.

Impact of multilingualism: We see that multilingual models outperform monolingual models. The examples in other languages also help the model learn more about the data. This is especially noticed in the case of No PT models, where multilingual No PT model output performs monolingual No PT considerably.

Impact of language family: In case of monolingual setting, IB performs better in hi, or, and pa. In case of multilingual setting, IB performs better in all the languages except as.

Impact of script unification: In monolingual setting, IB is worse than SSIB in languages other than gu, kn, and te. In multilingual settings, IB is worse than SSIB in all languages other than hi. Thus, script unification doesn’t help for the task of question generation.

Impact of task nature on performance: The task is to generate questions using the answer and the context, which is essentially the sentence containing the answer. The task is challenging because the models often have to rely on background knowledge not mentioned in the context. Indeed, the low rouge scores are a good indicator that the model is not able to perform very well.

6 Conclusion and Future Work

In this work, we created IndicNLP Suite, a collection of datasets for 5 diverse NLG tasks in 11 Indic languages, with the aim of creating much-required standard benchmarks to drive NLG research for Indic languages. In addition, this is the largest multilingual NLG dataset to the best of our knowledge. Our methods are simple enough to create similar datasets for modest resourced languages.

We trained a variety of monolingual and multilingual models and showed the impact of the combination of multilingualism and pre-trained models. From our experiments and results (summarized in Table 26), we can draw the following conclusions to the high-level research questions we posed in Section 5.1:

- In general, multilingual models outperform their monolingual counterparts.
- Fine-tuning of pre-trained models is better than training models from scratch, particularly for low-resource languages and for monolingual fine-tuning.
- Paraphrase generation is the one task where the benefits from pre-training and fine-tuning are limited as compared to other tasks. The training data for paraphrase generation is larger compared to other tasks, which might explain this behavior. In addition, the performance of mT5 is very poor on this task.
- On an average, language language-family specific pre-trained models like IndicBART perform better than general models like mT5. However, in the multilingual fine-tuning sce-
mT5 is comparable or better than IndicBART. This observation seems to be an outlier, and we will investigate the results to understand this further.

- The single-script and separate-script IndicBART variants have similar performance on average. We recommend using the single-script variant to perform better cross-lingual transfer, particularly in zeroshot generation scenarios.

Given these observations, we recommend future baselines to consider multilingual fine-tuning as the starting point. Future works will focus on extending the datasets for additional Indic languages, as well as adding new tasks and stronger baseline models. The limitations of lexical metrics for NLG evaluation are well acknowledged - particularly for morphologically rich target languages (Guzmán et al., 2016), and we would also like to explore embedding based metrics (Sellam et al., 2020) to track progress on these tasks.

Table 26: Summary of results on IndicNLG Suite across tasks. The table shows average scores across languages (iBLEU for paraphrase generation and Rouge-L for other tasks). We compare between models without pretraining (no PT), IndicBART (IB), separate script IndicBART (SSIB) and mT5 models in monolingual and multilingual settings.

| Task                        | Monolingual |           |            |           |            |
|-----------------------------|-------------|-----------|------------|-----------|------------|
|                             | No PT       | mT5       | SSIB       | IB        | No PT       | mT5       | SSIB       | IB        |
| Biography Generation        | 21.18       | 51.88     | 52.90      | 52.45     | 47.82       | 54.61     | 53.84      | 53.70     |
| Headline Generation         | 33.51       | 38.43     | 48.08      | 48.60     | 37.05       | 45.46     | 42.45      | 43.71     |
| Sentence Summarization      | 38.27       | 50.19     | 54.29      | 54.07     | 48.87       | 55.15     | 54.88      | 54.51     |
| Paraphrase Generation       | 9.11        | 4.60      | 9.88       | 10.22     | 8.71        | 5.11      | 10.06      | 10.64     |
| Question Generation         | 13.51       | 25.15     | 24.45      | 24.23     | 19.98       | 25.15     | 26.60      | 25.95     |
| **average**                 | 23.12       | 34.05     | **37.92**  | **37.91** | 32.49       | 37.10     | 37.57      | 37.70     |

References

Tushar Abhishek, Shivprasad Sagare, Bhavyajeet Singh, Anubhav Sharma, Manish Gupta, and Vasudeva Varma. 2022. Xalign: Cross-lingual fact-to-text alignment and generation for low-resource languages. *CoRR*, abs/2202.00291.

Usam Ali, Ylihas Chali, and Sadid A. Hasan. 2010. Automatic question generation from sentences. In *Actes de la 17e conférence sur le Traitement Automatique des Langues Naturelles. Articles courts*, pages 213–218, Montréal, Canada. ATALA.

M Anand Kumar, Shivkaran Singh, B Kavirajan, and KP Soman. 2016. Dpil@ fire 2016: Overview of shared task on detecting paraphrases in indian languages (dpil). In *CEUR Workshop Proceedings*, pages 233–238.

Xiang Ao, Xiting Wang, Ling Luo, Ying Qiao, Qing He, and Xing Xie. 2021. PENS: A dataset and generic framework for personalized news headline generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 82–92, Online. Association for Computational Linguistics.

Seohyun Back, Akhil Kedia, Sai Chetan Chinthakindi, Haejun Lee, and Jaegul Choo. 2021. Learning to generate questions by learning to recover answer-containing sentences. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1516–1529, Online. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Michele Banko, Vibhu O. Mittal, and Michael J. Witbrock. 2000. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 318–325, Hong Kong. Association for Computational Linguistics.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, pages 597–604, Ann Arbor, Michigan. Association for Computational Linguistics.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. *CoRR*, abs/2002.12804.

Loic Barrault, Ondrej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussa, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco

References

Tushar Abhishek, Shivprasad Sagare, Bhavyajeet Singh, Anubhav Sharma, Manish Gupta, and Vasudeva Varma. 2022. Xalign: Cross-lingual fact-to-text alignment and generation for low-resource languages. *CoRR*, abs/2202.00291.

Usam Ali, Ylihas Chali, and Sadid A. Hasan. 2010. Automatic question generation from sentences. In *Actes de la 17e conférence sur le Traitement Automatique des Langues Naturelles. Articles courts*, pages 213–218, Montréal, Canada. ATALA.

M Anand Kumar, Shivkaran Singh, B Kavirajan, and KP Soman. 2016. Dpil@ fire 2016: Overview of shared task on detecting paraphrases in indian languages (dpil). In *CEUR Workshop Proceedings*, pages 233–238.
Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Tom Kocmi, Andre Martins, Makoto Morishita, and Christof Monz, editors. 2021. *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics, Online.

Regina Barzilay and Lillian Lee. 2003. *Learning to paraphrase: An unsupervised approach using multiple-sequence alignment*. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 16–23.

Regina Barzilay and Kathleen R. McKeown. 2001. *Extracting paraphrases from a parallel corpus*. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 50–57, Toulouse, France. Association for Computational Linguistics.

Emily Bender. 2019. *The benderrule: On naming the languages we study and why it matters*.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. *Generating sentences from a continuous space*. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Jonathan Brown, Gwen Frishkoff, and Maxine Eskenazi. 2005. *Automatic question generation for vocabulary assessment*. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 819–826, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Samuel Cahyawijaya, Genta Indra Winata, Bryan Wilie, Karissa Vincentio, Xiaohong Li, Adhiguna Kuncoro, Sebastian Ruder, Zhi Yuan Lim, Syafri Bahar, Masayu Khodra, Ayu Purwarianti, and Pascale Fung. 2021. *IndoNLG: Benchmark and resources for evaluating Indonesian natural language generation*. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8875–8898, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Guanliang Chen, Jie Yang, Claudia Hauff, and Geert-Jan Houben. 2018. *Learningq: A large-scale dataset for educational question generation*. *Proceedings of the International AAAI Conference on Web and Social Media*, 12(1).

Yiran Chen, Zhenqiao Song, Xianze Wu, Danqing Wang, Jingjing Xu, Jiazhe Chen, Hao Zhou, and Lei Li. 2021. *MTG: A benchmarking suite for multilingual text generation*. *CoRR*, abs/2108.07140.

Jianpeng Cheng and Mirella Lapata. 2016. *Neural summarization by extracting sentences and words*. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. *Abstractive sentence summarization with attentive recurrent neural networks*. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California. Association for Computational Linguistics.

Mathias Creutz. 2018. *Open subtitiles paraphrase corpus for six languages*. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Raj Dabre, Aizhan Imankulova, and Masahiro Kaneko. 2021a. *Studying the impact of document-level context on simultaneous neural machine translation*. In *Proceedings of Machine Translation Summit XVIII: Research Track*, pages 202–214, Virtual. Association for Machine Translation in the Americas.

Raj Dabre, Himani Shrotriya, Anoop Kunchukuttan, Ratish Puduppully, Mitesh M. Khapra, and Pratyush Kumar. 2021b. *Indicbart: A pre-trained model for natural language generation of indic languages*. *CoRR*, abs/2109.02903.

Raj Dabre and Eiichiro Sumita. 2021. *YANMTT: yet another neural machine translation toolkit*. *CoRR*, abs/2108.11126.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. *Unified language model pre-training for natural language understanding and generation*. *Advances in Neural Information Processing Systems*, 32.

Xinya Du, Junru Shao, and Claire Cardie. 2017. *Learning to ask: Neural question generation for reading comprehension*. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. *Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge*. *Computer Speech & Language*, 59:123–156.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. *CCAligned: A massive collection of cross-lingual web-document pairs*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. *Creating training corpora for NLG micro-planners*. In *Proceedings
Vaibhav Kumar and Alan W. Black. 2020. ClarQ: A large-scale and diverse dataset for clarification question generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7296–7301, Online. Association for Computational Linguistics.

Anoop Kunchukuttan. 2020. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4034–4048, Online. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880, Online. Association for Computational Linguistics.

Haoran Li, Junnan Zhu, Jiju Zhang, Chengqing Zong, and Xiaodong He. 2020. Keywords-guided abstractive sentence summarization. Proceedings of the AAAI Conference on Artificial Intelligence, 34(05):8196–8203.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3865–3878, Brussels, Belgium. Association for Computational Linguistics.

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.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. Transactions of the Association for Computational Linguistics, 8:726–742.

Marco Lui and Timothy Baldwin. 2011. Cross-domain feature selection for language identification. In Proceedings of 5th International Joint Conference on Natural Language Processing, pages 553–561, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Nitin Madnani and Bonnie J. Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. Computational Linguistics, 36(3):341–387.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 881–893, Valencia, Spain. Association for Computational Linguistics.

Stephen Mayhew, Clinton Bicknell, Chris Brust, Bill McDowell, Will Monroe, and Burr Settles. 2020. Simultaneous translation and paraphrase for language education. In Proceedings of the Fourth Workshop on Neural Generation and Translation, pages 232–243, Online. Association for Computational Linguistics.

Kathleen R. McKeown. 1979. Paraphrasing using given and new information in a question-answer system. In 17th Annual Meeting of the Association for Computational Linguistics, pages 67–72, La Jolla, California, USA. Association for Computational Linguistics.

Kathleen R. McKeown. 1983. Paraphrasing questions using given and new information. American Journal of Computational Linguistics, 9(1):1–10.

Marie Meteer and Varda Shaked. 1988. Strategies for effective paraphrasing. In Coling Budapest 1988 Volume 2: International Conference on Computational Linguistics.

Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Darbe, Shohei Higashiyama, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondjef Bojar, Chenhui Chu, Akiko Eriguchi, Kaori Abe, Yusuke Oda, and Sadao Kurohashi. 2021. Overview of the 8th workshop on Asian translation. In Proceedings of the 8th Workshop on Asian Translation (WAT2021), pages 1–45, Online. Association for Computational Linguistics.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Ashinand Sivaprasad, Chiauhan Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irvanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chenn Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. DART: Open-domain structured data record to text generation. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 432–447, Online. Association for Computational Linguistics.
Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX), pages 95–100, Montréal, Canada. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018a. Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuvan Dhingra, Dyi Yang, and Dianjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1173–1186, Online. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforcement model for abstractive summarization. In International Conference on Learning Representations.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In Proceedings of the Third Conference on Machine Translation: Research Papers, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Aadiya Prakash, Sadid A. Hasan, Kathy Lee, Vivek Dalal, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual LSTM networks. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2923–2934, Osaka, Japan. The COLING 2016 Organizing Committee.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with entity modeling. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2023–2035, Florence, Italy. Association for Computational Linguistics.

Ratish Puduppully and Mirella Lapata. 2021. Data-to-text generation with macro planning. Transactions of the Association for Computational Linguistics, 9:510–527.

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Gowtham Ramesh, Sumanth Doddapaneni, Aravindh Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages. Transactions of the Association for Computational Linguistics, 10:145–162.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. Nat. Lang. Eng., 3(1):57–87.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. Transactions of the Association for Computational Linguistics, 8:264–280.

Aurko Roy and David Grangier. 2019. Unsupervised paraphrasing without translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6033–6039, Florence, Italy. Association for Computational Linguistics.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. MLSUM: The multilingual summarization corpus. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8051–8067, Online. Association for Computational Linguistics.
Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1073–1083. Vancouver, Canada. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7881–7892, Online. Association for Computational Linguistics.

Shi-Qi Shen, Yan-Kai Lin, Cun-Chao Tu, Yu Zhao, Zhi-Yuan Liu, Mao-Song Sun, et al. 2017. Recent advances on neural headline generation. Journal of computer science and technology, 32(4):768–784.

Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. 2018. Leveraging context information for natural question generation. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Volume 2 (Short Papers), pages 569–574, New Orleans, Louisiana. Association for Computational Linguistics.

Hong Sun and Ming Zhou. 2012. Joint learning of a dual SMT system for paraphrase generation. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 38–42, Jeju Island, Korea. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen,纳米 Goyal, Vishvar Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. CoRR, abs/2008.00401.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12), pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Ottokar Tilk and Tanel Alumäe. 2017. Low-resource neural headline generation. In Proceedings of the Workshop on New Frontiers in Summarization, pages 20–26, Copenhagen, Denmark. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Proceedings of the Advances in Neural Information Processing Systems 30, pages 5998–6008. Curran Associates, Inc.

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 451–462, Melbourne, Australia. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierrick Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Dongling Xiao, Han Zhang, Yukun Li, Yu Sun, Hao Tian, Hua Wu, and Haipeng Wang. 2021. Ernie-gcn: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI’20.

Liting 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.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1628–1639, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Shiqi Zhao, Haipeng Wang, Ting Liu, and Sheng Li. 2008. Pivot approach for extracting paraphrase patterns from bilingual corpora. In Proceedings of ACL-08: HLT, pages 780–788, Columbus, Ohio. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanki Tan, Hangbo Bao, and M. Zhou. 2017a. Neural question generation from text: A preliminary study. In NLPCC.
Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017b. Selective encoding for abstractive sentence summarization. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1095–1104, Vancouver, Canada. Association for Computational Linguistics.
A Appendix

A.1 Paraphrasing

**Results:** We report the BLEU scores on the test set for all the languages in Table 27.

A.2 Question Generation

**Dataset:** Table 28 shows the original sample, in which 'Context' is a paragraph which is associated with multiple question and answer pairs. Each 'Answer' has an 'Answer Text' which is the actual answer and a 'Answer Start Index' which is the index of the first character of 'Answer Text' in 'context'. The bold text in the Context row of the Original English Sample is the sentence which contains the answer to the question. This sentence is extracted with the help of 'Answer Start Index' for each question-answer pair.

Finally, each sample of the English dataset has <Context, Answer, Question, ID> and then this is translated using IndicTrans\(^{32}\) to the required Indic language dataset. The example of the translated dataset in Hindi and Marathi is shown in the same table.

\(^{32}\)https://github.com/AI4Bharat/indicTrans
| Language | Monolingual | | | | Multilingual | | | |
|---|---|---|---|---|---|---|---|---|
| No PT | IB | SSIB | mT5 | No PT | IB | SSIB | mT5 |  |
| as | 1.02 | 1.12 | 0.69 | 0.94 | 1.61 | 1.66 | 1.19 | 1.11 |  |
| bn | 10.60 | 11.30 | 10.15 | 4.44 | 8.85 | 11.57 | 10.04 | 4.45 |  |
| gu | 18.48 | 21.14 | 16.80 | 7.36 | 18.53 | 22.10 | 18.69 | 8.66 |  |
| hi | 24.94 | 27.55 | 25.35 | 12.50 | 22.94 | 27.29 | 25.05 | 13.77 |  |
| kn | 13.56 | 15.25 | 12.83 | 7.11 | 12.93 | 15.40 | 13.14 | 7.52 |  |
| ml | 9.96 | 10.36 | 8.68 | 7.81 | 9.48 | 10.57 | 8.71 | 7.75 |  |
| mr | 17.86 | 17.74 | 18.56 | 7.52 | 17.48 | 20.38 | 18.50 | 8.64 |  |
| or | 11.21 | 15.34 | 20.84 | 0.50 | 16.59 | 19.26 | 23.02 | 1.77 |  |
| pa | 12.93 | 14.44 | 18.01 | 7.39 | 12.40 | 14.87 | 17.61 | 8.72 |  |
| ta | 15.51 | 18.04 | 16.03 | 12.20 | 15.52 | 18.52 | 16.25 | 12.39 |  |
| te | 15.16 | 16.78 | 13.69 | 7.48 | 13.54 | 16.70 | 14.16 | 8.16 |  |

Table 27: The table shows BLEU scores for the paraphrase generation test sets. We compare between models without pre-training (no PT), Indic BART (IB), separate script Indic BART (SSIB) and mT5 models in monolingual and multilingual settings.

---

**Original English sample**

**Context**
Architecturally, the school has a Catholic character. Atop the Main Building’s gold dome is a golden statue of the Virgin Mary. Immediately in front of the Main Building and facing it, is a copper statue of Christ with arms upraised with the legend Venite Ad Me Omnes. Next to the Main Building is the Basilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a Marian place of prayer and reflection. It is a replica of the grotto at Lourdes, France, where the Virgin Mary reputedly appeared to Saint Bernadette Soubirous in 1858. At the end of the main drive (and in a direct line that connects through 3 statues and the Gold Dome), is a simple, modern stone statue of Mary.

**Question**
To whom did the Virgin Mary allegedly appear in 1858 in Lourdes, France?

**Answer Text**
Saint Bernadette Soubirous

**Answer Start index**
515

**ID**
5733be284776f41900661182

---

**Converted English sample**

**Context**
It is a replica of the grotto at Lourdes, France, where the Virgin Mary reputedly appeared to Saint Bernadette Soubirous in 1858.

**Answer**
Saint Bernadette Soubirous

**Question**
To whom did the Virgin Mary allegedly appear in 1858 in Lourdes, France?

**ID**
5733be284776f41900661182

---

**Hindi Translated sample**

**Context**
यह लूर्दै, फ्रांस में स्थित ग्रोटो की प्रतिकृति है, जहां 1858 में सेंट बनिडैट आर्थिस को विश्व वैश्विक मैरी दिखाई दी थी।
(R:yah loordes, phraans mein shhit groto kee pratikrti hai, jahaan 1858 mein sent banaaude saubiras ko varjin mairee dikhaee dee thee.)

**Answer**
सेंट बनिडैट साबिरौस
(R:sant banaaude saabiros)

**Question**
सन् 1858 में लूर्दै स्थान में कुंवारी मारियम दिखित तौर पर किसके सामने प्रकट हुई?
(R:saan 1858 mein loordas phraans mein kunvaaree mariyam kathit taur par kisake saamane prakat huce?)

**ID**
5733be284776f41900661182

Table 28: Translation example for question generation
| Languages | Total size | 1.0 to 0.9 | 0.9 to 0.8 | 0.8 to 0.7 | 0.7 to 0.6 |
|-----------|------------|------------|------------|------------|------------|
| as        | 60,148     | 5.286      | 10.495     | 9.377      | 11,046     |
| bn        | 163,371    | 6.426      | 12,814     | 17,822     | 23,415     |
| gu        | 277,834    | 18.767     | 31,806     | 43,090     | 48,667     |
| hi        | 341,400    | 47.219     | 75,837     | 63,901     | 54,701     |
| kn        | 160,397    | 9.432      | 23,469     | 22,508     | 30,013     |
| ml        | 27,472     | 3.897      | 1.757      | 2,373      | 3,456      |
| mr        | 147,960    | 10,266     | 21,900     | 23,064     | 26,500     |
| or        | 79,612     | 5,623      | 7,258      | 7,652      | 11,315     |
| pa        | 63,802     | 13,692     | 14,268     | 14,511     | 12,730     |
| ta        | 79,414     | 2,442      | 10,782     | 12,292     | 12,730     |
| te        | 28,800     | 1,354      | 3,209      | 2,582      | 5,217      |
| **average** | **130019.36** | **5.135** | **3.475** | **2,260** | **2,254** | **71** |

Table 29: Overlapping fraction statistics for headline generation. These are the number of samples that fall into a particular range of overlapping between title and its document.

| Languages | 0.6 to 0.5 | 0.5 to 0.4 | 0.4 to 0.3 | 0.3 to 0.2 | 0.2 to 0.1 | 0.1 to 0.0 |
|-----------|------------|------------|------------|------------|------------|------------|
| as        | 9,877      | 5,135      | 3,475      | 2,260      | 2,254      | 71         |
| bn        | 28,108     | 16,705     | 17,255     | 10,426     | 10,574     | 4,755      |
| gu        | 51,123     | 28,199     | 23,299     | 13,031     | 7,030      | 920        |
| hi        | 30,118     | 15,683     | 10,803     | 7,146      | 6,488      | 307        |
| kn        | 4,428      | 2,914      | 3,332      | 1,928      | 1,329      | 148        |
| ml        | 26,352     | 13,559     | 9,979      | 5,665      | 5,665      | 323        |
| or        | 13,379     | 8,478      | 7,757      | 6,604      | 6,387      | 429        |
| pa        | 3,930      | 2,027      | 1,083      | 426        | 298        | 23         |
| ta        | 13,898     | 8,732      | 7,776      | 4,628      | 7,384      | 554        |
| te        | 5,283      | 3,065      | 2,671      | 2,104      | 1,919      | 117        |
| **average** | **20560.27** | **11248** | **9605.72** | **6294.81** | **5174.63** | **851.18** |

Table 30: Overlapping fraction statistics for headline generation. This table is the remainder of Table 29.

| Languages | Total size | 0.6-0.5 | 0.5-0.4 | 0.4-0.3 | 0.3-0.2 | 0.2-0.1 | 0.1-0.0 |
|-----------|------------|---------|---------|---------|---------|---------|---------|
| as        | 60,148     | 5.135   | 3.475   | 2,260   | 2,254   | 71      |
| bn        | 28,108     | 16,705  | 17,255  | 10,426  | 10,574  | 4,755   |
| gu        | 51,123     | 28,199  | 23,299  | 13,031  | 7,030   | 920     |
| hi        | 30,118     | 15,683  | 10,803  | 7,146   | 6,488   | 307     |
| kn        | 4,428      | 2,914   | 3,332   | 1,928   | 1,329   | 148     |
| ml        | 26,352     | 13,559  | 9,979   | 5,665   | 5,665   | 323     |
| or        | 13,379     | 8,478   | 7,757   | 6,604   | 6,387   | 429     |
| pa        | 3,930      | 2,027   | 1,083   | 426     | 298     | 23      |
| ta        | 13,898     | 8,732   | 7,776   | 4,628   | 7,384   | 554     |
| te        | 5,283      | 3,065   | 2,671   | 2,104   | 1,919   | 117     |
| **average** | **20560.27** | **11248** | **9605.72** | **6294.81** | **5174.63** | **851.18** |

Table 31: Overlapping fraction statistics for sentence summarization.

| Languages | Total size | 0.6-0.5 | 0.5-0.4 | 0.4-0.3 | 0.3-0.2 | 0.2-0.1 | 0.1-0.0 |
|-----------|------------|---------|---------|---------|---------|---------|---------|
| as        | 7746       | 3127    | 4648    | 4239    | 6177    | 1417    |
| bn        | 9621       | 8032    | 13506   | 13929   | 25183   | 18802   |
| gu        | 2034       | 16873   | 23639   | 25859   | 43079   | 23892   |
| hi        | 36718      | 30458   | 38247   | 40877   | 56081   | 25822   |
| kn        | 18115      | 14013   | 13452   | 12549   | 19399   | 3740    |
| ml        | 2249       | 1835    | 3467    | 2928    | 4109    | 1601    |
| mr        | 9171       | 7523    | 9158    | 11620   | 21199   | 5835    |
| or        | 4537       | 3741    | 4708    | 5519    | 11664   | 2906    |
| pa        | 7851       | 5163    | 5787    | 4773    | 4986    | 2412    |
| ta        | 9918       | 7688    | 9310    | 7940    | 10480   | 3484    |
| te        | 2527       | 1953    | 2066    | 2691    | 4878    | 975     |
| **average** | **11706.09** | **9309.64** | **11635.27** | **12084.00** | **18832.27** | **8262.36** |

Table 32: Overlapping fraction statistics for sentence summarization. This is the remainder of Table 31.