Cross-Lingual Approaches for Text Generation Tasks in Low-Resource Languages

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

Master of Science in ***Computer Science and Engineering*** *by Research*

by

Shivprasad Sagare
2020701015
shivprasad.sagare@research.iiit.ac.in

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

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Cross-Lingual Approaches for Text Generation Tasks in Low-Resource Languages**" by **Shivprasad Sagare**, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Advisor: Prof. Vasudeva Varma
Date	Co-advisor: Dr. Manish Gupta

To *Kunal and Abhi, for making me believe in the beauty of companionship.*

Acknowledgments

The past few years have been an incredible journey, filled with challenges and triumphs as I worked towards my Master's degree. Pursuing research at such a prestigious institution was a dream come true. I am deeply grateful for the guidance and support of everyone who was a part of it.

First and foremost, I would like to extend my sincere gratitude to my advisors Professor Vasudeva Varma and Dr. Manish Gupta. Vasudeva Sir provided invaluable guidance in steering me in the right direction. He connected me with the right mentors and colleagues. He promptly replied to the smallest of my concerns and ensured that I was making continuous progress. Regarding the research paper deadlines, Manish Sir pushed us regularly to focus on the tasks at hand. His speed, precision, energy, and enthusiasm during the meetings are unparalleled and would motivate us to work even more. His attention to detail and meticulous insights taught me a lot about conducting experiments and writing research papers. I could not have imagined looking up to a better set of advisors.

Next, I thank Tushar Abhishek, my senior, co-author and friend. You are one of the most helpful, talented, and humble people I have met. We worked on many things together, and I have learned much from you. I am also thankful to my wonderful co-authors, Bhavyajeet, Anubhav, Dhaval, and more. Next, I would also like to thank my batchmates. Thank you, Sagar, for helping me with everything on campus. I felt a little more at home every time we spoke in Marathi. Thank you, Sai, for being such a caring friend. I'll cherish the walks we frequently had on campus. Thanks, Gayathri, for being such a respectful and genuine friend.

Most importantly, I am thankful to my family and friends back home. Firstly, I am grateful to Sukhada, my very talented friend, who motivated me to pursue higher education. I want to thank Shubham and Rohit for being beside me at every step and believing in me. Thank you, Kunal and Abhi, for tolerating me.

My parents deserve endless gratitude. My father supported me in every decision and accompanied me wherever I went. My mother gave me the vision and strength to lead my life. I love you both!

Abstract

Text generation has shown tremendous promise recently, mainly attributed to the use of transformer architecture and the models pretrained on vast amounts of data. Multiple business scenarios today deploy neural-network-based models for natural language generation(NLG) tasks. However, this progress is limited to English and other high-resource(HR) languages, with NLG systems in low-resource(LR) languages far behind in terms of accuracy and fluency of generated text. This is due to several factors, such as lack of training data, lack of robust models supporting native script, and lack of linguistic resources as well. In this work, we extensively study an approach of cross-lingual NLG to tackle these challenges. Cross-lingual NLG implies exploiting the widely available data in HR language to generate the desired text in LR language. We focus on two significant tasks, i.e., fact-to-text and summarization, with a larger goal of generating Wikipedia article text in LR languages. We propose novel ways to build the datasets for the above tasks and also the approaches to generate text in LR languages.

Firstly, we propose a novel task of cross-lingual fact-to-text generation(XF2T). Given the Wikidata facts in English, the system is expected to generate a sentence describing these facts in the desired language. To build a parallel dataset to train a model for the same, we explore several methods to link a fact from Wikidata to a sentence from Wikipedia, such as unsupervised, distantly supervised, and zero-shot learning-based approaches. We use the best approach to create the dataset XAlign of 0.55M instances across 12 Indian languages. Further, we implement transformer encoder-decoder and mT5 model as baselines using this dataset. In addition, we also explore the impact of task-specific pretraining, bilingual and monolingual models. We experiment with techniques to improve efficiency, such as structure-aware encoding of facts and fusing role-specific embeddings. We show that these approaches generate fluent and highly accurate sentences.

Further, intending to generate longer text, we propose one more novel idea to generate the Wikipedia article section text using summarization. We leverage the citations available for each section on Wikipedia pages and build a parallel dataset for cross-lingual, multi-document, aspect-based summarization in 8 domains and 15 languages. In the first stage, i.e., extractive summarization, we aim to filter relevant sentences from a set of reference articles, for which we use saliency and graph-based methods. We experiment with recent SOTA models mT5 and mBART in the abstractive stage. Despite high noise in the input reference articles, we

show that the system generates fluent and meaningful outputs. Although, in terms of content coverage and text coherency, models have a lot of scope for improvement.

Overall, we work on various methods using cross-lingual NLG to advance the datasets and models in LR languages. We hope this work will boost more research in these critical areas in the future.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

1.1.1 Disparity in the textual content available across the languages

Although Wikipedia has been the primary choice of encyclopedic reference for millions of users, unfortunately, Wikipedia is extremely sparse for low-resource (LR) languages. English Wikipedia exhibits abundance with ∼6.56M articles expressed in 54.2 GB of text, low resource Wikipedia is poor with only ∼90K articles expressed using 7.5 GB of text on average across seven low resource languages as shown in Fig. 1.1. Further, as illustrated in Fig. 1.2, manual efforts towards enriching LR Wikipedia over the years have also not been as encouraging as in

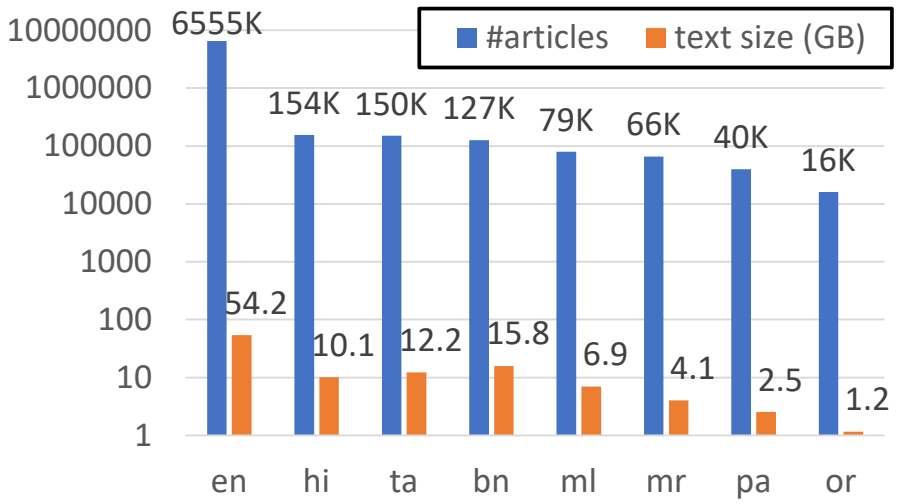

Figure 1.1: Number of Wikipedia articles and text size in GBs across eight languages, using 20220926 Wikipedia dump. Note that the Y axis is in log scale.

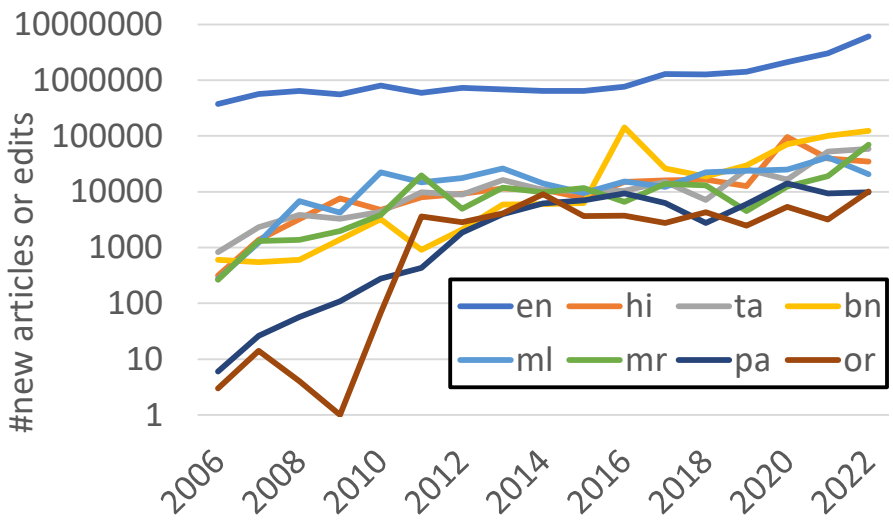

Figure 1.2: Number of new articles or edits on Wikipedia across eight languages from 2006 to 2022. This is obtained using a publication date from the 20220926 Wikipedia dump. Note that the Y axis is in the log scale.

the case of English. These observations indicate that automated text generation for low-resource Wikipedia is critical.

1.1.2 Need of cross-lingual approaches for text generation in LR languages

A possible naïve approach for the automated generation of articles in low-resource Wikipedia is translating text from equivalent English Wikipedia articles. Unfortunately, several low-resource entities of interest tend to be local in nature, leading to a lack of equivalent English Wikipedia pages for ∼42.1% entities on average across seven low-resource languages. In particular, the following are percentages of Wikipedia entities with no equivalent English Wikipedia page: Hindi (50.60%), Tamil (46.70%), Bengali (31.50%), Malayalam (36.30%), Marathi (42.00%), Punjabi (38.70%), Oriya (39.40%). Thus, we need to explore other inputs for LR Wikipedia text generation.

Another approach is to leverage generic Web content for LR Wikipedia text generation. A challenge to this approach is that such web content is itself very sparse in low-resource languages, as can be observed in publicly available large dumps like CommonCrawl [69]. Hence, it is impossible to build monolingual parallel datasets in LR languages. This motivates us to explore the use of cross-lingual approaches for our task.

Lang.	WikiData entries	Facts	Average facts per entity	Wikipedia articles
hi	26.0K	271.0K	10.43	22.9K
mr	16.5K	174.0K	10.56	15.9K
te	12.4K	142.2K	11.49	7.8K
ta	26.0K	280.4K	10.77	25.6K
en	1.3M	30.2M	22.8	627.9K
gu	3.5K	37.8K	10.88	1.9K
bn	36.2K	501.9K	13.87	29.0K
kn	7.5K	83.6K	11.1	4.5K

Table 1.1: WikiData+Wikipedia statistics for the person entities across languages

1.2 Cross-lingual fact-to-text generation (XF2T)

Fact-to-text(F2T) generation [71] is the task of transforming structured data into natural language. F2T generation systems are vital in many downstream Natural Language Processing (NLP) applications like automated dialog systems [82], domain-specific chatbots [64], open domain question answering [16], authoring sports reports [14], etc.

Most of the existing F2T datasets focus on the English language. This is because much of the structured data is available in English only, especially for fresh content like sports reports or information about new entities like newly-launched products. Table 1.1 shows that structured Wikidata entries for person entities in LR languages are minuscule in number compared to that in English. Also, the average facts per entity in LR languages are much smaller than in English. Thus, monolingual F2T for LR languages suffers from data sparsity. Hence, in this work, we

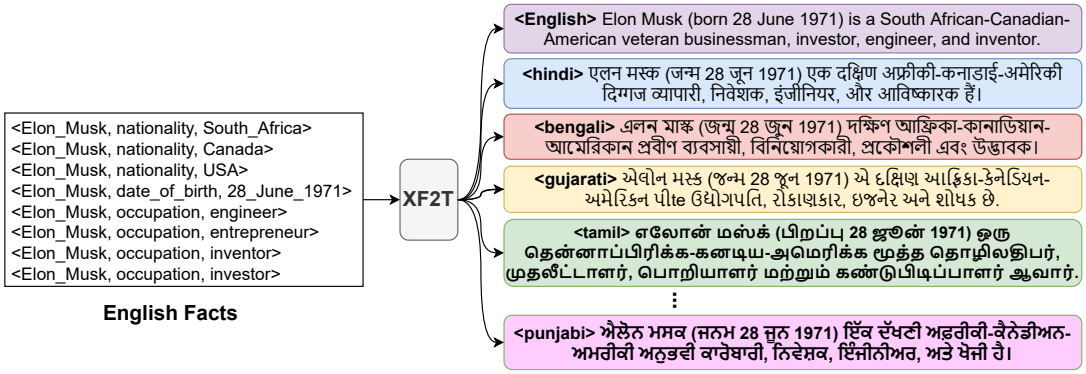

Figure 1.3: XF2T Example: Generating English, Hindi, Bengali, Gujarati or Tamil sentences to capture semantics from English facts.

rigorously investigate the XF2T problem of aligning English structured data with sentences in *multiple LR languages* and contribute a new dataset, XAlign.

The XF2T generation task takes a set of English facts as input and generates a sentence capturing the fact semantics in the specified language. Fig. 1.3 shows an example where a set of English Wikidata facts are used to generate a sentence across various languages. We model this as a multi-lingual text generation task and hence experiment with multiple multilingual deep learning models. Instead of working on a single text generation model for each language, we can leverage language relatedness to build single model that can produce sentences in multiple languages.

1.3 Cross-lingual, multi-document, aspect-based summarization (XWikiGen)

As shown in Fig. 1.4, the input for XWikiGen is a set of reference URLs, a target section title, and a target output language. The expected output is then the text suitable for that Wikipedia section in the target language. Analogous to generic summarization versus query-based summarization, XWikiGen involves section-wise text generation rather than the generation of the entire Wikipedia page. Unlike existing work on monolingual (English-only) Wikipedia text generation, XWikiGen is cross-lingual in nature. Lastly, unlike some existing work that generates cross-lingual text using English Wikipedia pages, XWikiGen focuses on generating cross-lingual text using reference URLs in multiple languages.

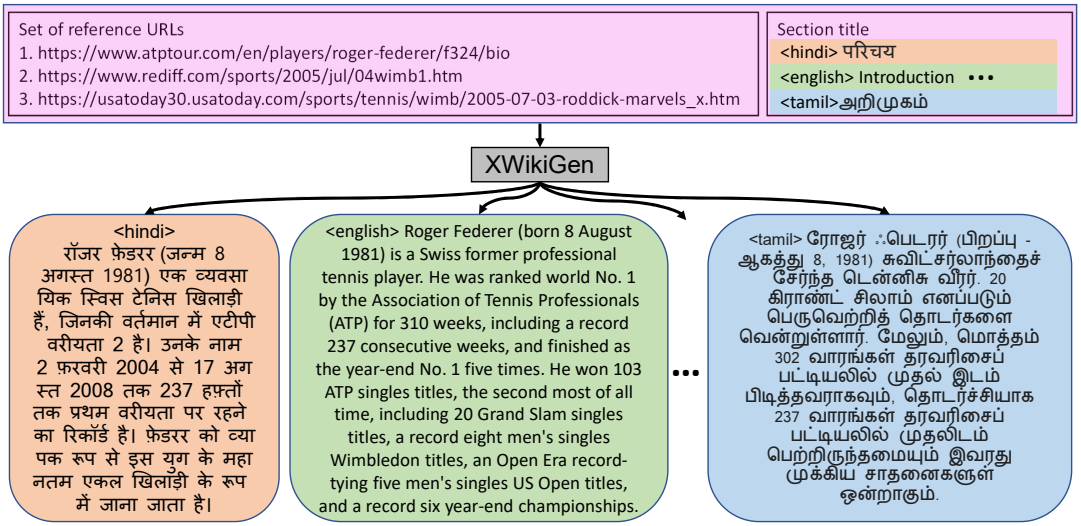

Figure 1.4: XWikiGen examples: Generating Hindi, English, and Tamil text for the Introduction section from cited references.

Our first contribution is a novel dataset, XWikiRef towards the XWikiGen task. The dataset is obtained from Wikipedia pages corresponding to eight languages and five domains. Languages include Bengali (bn), English (en), Hindi (hi), Malayalam (ml), Marathi (mr), Oriya (or), Punjabi (pa) and Tamil (ta). Domains include books, films, politicians, sportsmen, and writers. The dataset spans ∼69K Wikipedia articles with ∼105K sections. Each section has 5.44 cited references on average.

XWikiGen is an extremely challenging task because it involves long text generation, and that too in a cross-lingual manner. Handling long text input is difficult. Hence, we follow a two-stage approach. The first extractive stage identifies important sentences across several reference documents. The second abstractive stage generates the section text. Both stages involve neural models. We experiment with unsupervised methods like salience [87] and hiporank [28] for the extractive stage, and mT5 [84] and mBART [59] for the abstractive stage. We experiment with several training setups like (1) multi-lingual, (2) multi-domain, and (3) multi-lingual-multi-domain. We report results using standard text generation metrics like ROUGE-L, METEOR, and chrF++.

1.4 Key contributions and thesis outline

Overall, we make the following contributions in this work.

1. We highlight the existing lack of resources across the low-resource languages, and propose the use of cross-lingual approaches for text generation. We propose the problem of XF2T alignment and generation for low-resource languages. We motivate and propose the XWikiGen problem where the input is (set of reference URLs, section title, language) and the output is a text paragraph. More details about these tasks are covered in the **Chapter 1** of the thesis. Summary of the related work regarding above tasks is covered in the **Chapter 2**.

2. We propose the creation of XF2T dataset, XAlign, consisting of English WikiData triples/facts mapped to sentences from LR Wikipedia. We introduce solid baselines for the cross-lingual alignment task and propose two novel approaches: NER-based filtering with Semantic Similarity and Key-phrase Extraction with Relevance Ranking. Later, we scale the dataset across multiple languages and propose better approaches for the same, like 1) zero-shot learning and 2) distant supervision. We introduce a large collection of high-quality XF2T datasets in 12 languages: Hindi, Marathi, Gujarati, Telugu, Tamil, Kannada, Bengali, Punjabi, Assamese, Malayalam, Odiya, and monolingual dataset in English. We have also collected 5402 human-labeled gold test dataset spanning all these languages to evaluate alignment methods. More details of the processes involved in building such datasets are covered in **Chapter 3** of the thesis.

3. We report strong baseline results by adopting popular natural language generation (NLG) methods for our proposed novel XF2T task. First, we experiment with standard existing Transformer-based multi-lingual encoder-decoder models like the vanilla Transformer, IndicBART and mT5. Next, we explore performance across various training setups: bi-lingual, translate-output, translate-input and multi-lingual. Further, we systematically explore various strategies for improving XF2T generation like multi-lingual data-to-text pre-training, fact-aware embeddings, and structure-aware encoding. Detailed results and analysis of the above experiments is covered in **Chapter 4** of the thesis.

4. After proposing the XWikiGen task in Chapter 1, we contribute a large dataset, XWikiRef, with ∼105K instances covering eight languages and five domains. We model XWikiGen as a multi-document cross-lingual summarization problem and propose a two-stage extractive-abstractive system. Our multi-lingual-multi-domain models using HipoRank (extractive) and mBART (abstractive) lead to the best results. Detailed results and analysis can be found in **Chapter 5**.

5. We summarize our key ideas, the experimental setups, and the results in the final chapter of the thesis, **Chapter 6**. We discuss the significance of each approach as well as the limitations of our work. We end the chapter by discussing some possible threads for future work in these critical research areas.

Chapter 2

Related work

2.1 Cross-lingual text generation

Recently there has been a lot of work on cross-lingual NLG tasks like question generation [19], news title generation [56], summarization [89], and machine translation [59, 17] thanks to models like XNLG [19], mBART [59], mT5 [84], etc.

Machine Translation(MT) based pipeline methods were used from early days for cross-lingual generation[79][29]. Even in recent times, the creators of WebNLG shared task [12] have used MT systems that convert English sentences to Russian, for the purpose of generating pseudo-training data for cross-lingual data-to-text from English RDF triples to Russian sentences. However these MT based models are not suitable for low resource languages as they do not share parameters across-languages and generated translations are error-prone. Following the recent advancements in the field of multilingual transformer models such as XLM-R[23] and cross-lingual transfer learning in NLP [56] [43], models like MT5[84] and mBART[59] are now successfully used for NLG tasks like Machine Translation, Automatic Summarization, and data-to-text tasks in cross-lingual manner[72]. Improving on these models that are pre-trained on massive monolingual tasks, some recent works include cross-lingual data in their pretraining tasks. mT6[17] improves cross-lingual transfer over mT5 by pre-training on translation pairs. [18] used parallel data to train a sequence-to-sequence model for zero-shot cross-lingual abstractive text summarization and question generation.

In this work, we propose a new cross-lingual NLG task: XF2T. Further, from a knowledge graph (KG) and text linking perspective, our work is related to tasks like entity linking (link mention in a sentence to a KG entity) [11] and fact linking (linking sentence to a set of facts) [49]. As against this, XF2T is the problem of generating a sentence given a set of facts.

2.2 Fact-to-text

2.2.1 Building the datasets for fact-to-text task

Recently, there has been a lot of effort in creating automated structured data to text datasets in various domains. [54] introduced a WikiBio dataset by aligning opening sentences with infoboxes in English Wikipedia articles on person's biographies. Several extensions of this method of aligning Wikipedia text with infoboxes have been proposed to create a dataset in different languages [62] and domains [67]. Datasets created using these methods are constrained to a specific domain. [34] alleviates this limitation by aligning knowledge graph triples in Wikidata with opening sentences in Wikipedia. It uses lexical overlap between the name entities present in a sentence, and Wikidata triples for alignment. In addition to using triples available in Wikidata (Wikipedia's Knowledge Graph), [2] introduced a dataset that also incorporates sub-property information in the form of quadruples. These datasets focus on aligning either knowledge graph triples or infoboxes with sentences present in Wikipedia articles. [15] introduced a dataset that combined the structured information residing in Wikidata and infoboxes with a given sentence. To scale alignment of structured data with natural text across various domains [31, 44] introduced sequential pipeline strategy consisting of data collection, data filtering, entity linking, and alignment. Additionally, it also suggests incorporating a human-annotated test dataset to evaluate the different alignment methods.

All of the previous approaches depend upon lexical overlap between structured and textual data. These approaches are ineffective for cross-lingual alignment where structured data and textual data are in different languages. Although, we can utilize previously proposed strategies for dataset creation by translating either structured data or textual data to other languages.

Table 2.1 shows basic statistics of popular F2T datasets. There exists a large body of work on generic structured data to text, but here we list only F2T datasets. These datasets contain text from various domains like person, sports, restaurant, airport, politician, artist, etc. Also, these datasets vary widely in terms of statistics like number of instances, number of facts per instance, number of unique predicates and average number of tokens per instance. Unlike other datasets which are mostly on English only, our dataset contains multiple Indian languages and is a cross-lingual dataset.

Training F2T models require aligned data. Some previous studies like WebNLG [35] collected aligned data by crowdsourcing while others have performed automatic alignment by heuristics like TF-IDF. We explore two different unsupervised methods to perform cross-lingual alignment.

Initial methods for F2T were template-based and were therefore proposed on domain-specific data like medical [10], cooking [20], person [30], etc. They first align entities in RDF triples with entities mentioned in sentences. Then, they extract templates from the aligned sentences by replacing the entity mentions with a unique token. Such templates are then used to generate sentences given facts for new entities. Template-based methods works well on RDF triples in

Dataset	Languages	A/M	|I|	|F|	|P|	|T|	X-Lingual
WikiBio	en	A	728K	19.70	1740	26.1	No
E2E	en	M	50K	5.43	945	20.1	No
WebNLG 2017	en	M	25K	2.95	373	22.7	No
fr-de Bio	fr, de	A	170K, 50K	8.60, 12.6	1331, 1267	29.5, 26.4	No
TREX	en	A	6.4M	1.77	642	79.8	No
WebNLG 2020	en, ru	M	40K, 17K	2.68, 2.55	372, 226	23.7	Yes
KELM	en	A	8M	2.02	663	21.2	No
WITA	en	A	55K	3.00	640	18.8	No
WikiTableT	en	A	1.5M	51.90	3K	115.9	No
GenWiki	en	A	1.3M	1.95	290	21.5	No
XAlign	en + 7 LR	A	0.45M	2.02	367	19.8	Yes

Table 2.1: Statistics of popular F2T datasets: WikiBio [54], E2E [64], WebNLG 2017 [35], WebNLG 2020 [12], fr-de Bio [62], KELM [3], WITA [34], WikiTableT [15], GenWiki [44], TREX [31], and XAlign (ours). Alignment method could be A (automatic) or M (manual). |I|=number of instances. |F|=number of facts per instance. |P|=number of unique predicates. |T|=average number of tokens per instance.

a seen domain but fail on RDF triples in a previously unseen domain. Seq-2-seq attention based neural methods [54, 61] gained popularity for F2T around 5-6 years back. Vougiouklis et al. proposed a method which uses feedforward neural networks to encode RDF triples and concatenate them as the input of the LSTM decoder [78]. Variations of LSTM encoder-decoder model with copy mechanism [74] or with hierarchical attentive encoder [62] have also been proposed. Recently, pretrained Transformer based models like BART [55] and T5 [69] have been applied for mono-lingual English F2T [72]. Richer encoding of the input triples has also been investigated using a combination of graph convolutional networks and Transformers [88], triple hierarchical attention networks [16], or Transformer networks with special fact-aware input embeddings [16]. Some recent work also explores specific F2T settings like plan generation when the order of occurrence of facts in text is available [88] or partially aligned F2T when the text covers more facts than those mentioned in the input [34]. However, all of these methods focus on English fact to text only. WebNLG 2020 [12] shared task presents one such cross-lingual aligned dataset where [75] performs automatic translation and post editing of English sentences to Russian. Final dataset consists of English triples aligned with Russain sentences verbalizing those triples. Such approaches do incur the loss due to automatic translation though.

2.2.2 Approaches for fact-to-text generation

Initial F2T methods were template-based and were therefore proposed on domain-specific data like medical [10], cooking [20], person [30], etc. They align entities in RDF triples with entities mentioned in sentences, extract templates from the aligned sentences, and use templates to generate sentences given facts for new entities. Template-based methods are brittle and do not generalize well.

Recently, seq-2-seq neural methods [54, 61] have become popular for F2T. These include vanilla LSTMs [78], LSTM encoder-decoder model with copy mechanism [74], LSTMs with hierarchical attentive encoder [62], pretrained Transformer based models [72] like BART [55] and T5 [69]. Vougiouklis et al. [78] proposed a method which uses feedforward neural networks to encode RDF triples and concatenate them as the input of the LSTM decoder. Variations of LSTM encoder-decoder model with copy mechanism [74] or with hierarchical attentive encoder [62] have also been proposed. Recently, pretrained Transformer based models like BART [55] and T5 [69] have been applied for mono-lingual English Fact-to-Text [72]. Richer encoding of the input triples has also been investigated using a combination of graph convolutional networks and Transformers [88], triple hierarchical attention networks [16], or Transformer networks with special fact-aware input embeddings [16]. Some recent work also explores specific F2T settings like plan generation when the order of occurrence of facts in text is available [88] or partially aligned F2T when the text covers more facts than those mentioned in the input [34]. However, all of these methods focus on English fact to text only. Only recently, the XF2T problem was proposed in [1] but their focus is on problem formulation and dataset contribution. In this paper, we extensively evaluate multiple methods for the XF2T generation task.

2.3 Summarization

In this section, we discuss related work on generating both short and long Wikipedia text. We also briefly discuss work on multi-lingual and cross-lingual summarization.

2.3.1 Generating Short Wikipedia Text

Automated generation of Wikipedia text has been a problem of interest for the past 5–6 years. Initial efforts in the fact-to-text (F2T) line of work focused on generating short text, typically the first sentence of Wikipedia pages using structured fact tuples.

Training F2T models require aligned data with adequate content overlap. Some previous studies like WebNLG [35] collected aligned data by crowdsourcing, while others have performed automatic alignment by heuristics like TF-IDF. Seq-2-seq neural methods [54, 61] have been popularly used for F2T. These include vanilla LSTMs [78], LSTM encoder-decoder model with

copy mechanism [74], LSTMs with hierarchical attentive encoder [62], pretrained Transformer based models [72] like BART [55] and T5 [69].

Most of the previous efforts on F2T focused on English fact-to-text only. Only recently, the Cross-lingual F2T (XF2T) problem was proposed in [1]. Compared to all of these pieces of work which have focused on short text generation, the focus of the current paper is on generating longer text. Unlike F2T literature, where the input is structured, the input is a set of reference URLs in our case.

2.3.2 Generating Long Wikipedia Text

Besides generating short Wikipedia text, there have also been efforts to generate Wikipedia articles by summarizing long sequences [58, 37, 42, 4, 38, 76] as shown in Table 2.2. For all of these datasets, the generated text is either the full Wikipedia article or text for a specific section. Most of these studies [58, 42, 4, 37] have been done on English only. Further, these studies use different kinds of input: single document (existing Wikipedia article in the same or another language) or multi-document (set of citation URLs, review pages).

Liu et al. [58] introduced the WikiSum dataset, which contains article text paired with cited reference articles. Ghalandari et al. [37] introduced a large dataset for multi-document summarization created by leveraging the Wikipedia current events portal. Antognini et al. [4] extended a similar idea to a specific domain of games by contributing the GameWikiSum dataset.

Most of these works fail to include the section-specific intent during summarization and generate an article on the whole. Hence, to capture the section-specific intent while summarization, Hayashi et al. [42] introduced section-specific summarization, which recognizes the main topics in the input text and then creates a summary for each. Although authors rely on the model to figure out the latent subtopics, the content selection step is challenging. We tackle this challenge by providing section-specific citations as input in our dataset, which avoids the noisy references belonging to other sections, and allows us to study the summarization capabilities of the model better.

Interestingly, none of the existing datasets perform cross-lingual multi-document summarization for Wikipedia text. However, as motivated in the previous section, this setup is critical for Wikipedia text generation for LR languages. Hence, we fill this gap in this paper by proposing the XWikiRef dataset.

There has been no previous work in multi-lingual and cross-lingual settings for long input sequences and aspect-based summarization. We mainly focus on the combination of the above in our problem formulation.

Dataset	#Summaries	XL?	ML?	#Langs	MD?	SS?	Input	Output
WikiSum [24]	~2.3M articles	No	No	1	Yes	No	Set of citation URLs	Whole Wiki article
WikiAsp [16]	~400K sections	No	No	1	Yes	Yes	Set of citation URLs	One section in same language
GameWikiSum [2]	~26K gameplay Wikipedia sections	No	No	1	Yes	No	Professional video game reviews	Gameplay Wikipedia sections
Wiki Current Events Portal (WCEP) [12]	~10.2K WCEP event summaries	No	No	1	Yes	No	Set of news articles	WCEP Summary
MultiLing'15 [13]	~1.5K paragraphs	No	Yes	38	No	No	Whole Wikipedia article	First few Wikipedia sentences in same language
WikiMulti [34]	~150K intro paragraph	Yes	Yes	15	No	No	Whole Wikipedia article	Intro paragraph in other language
XWikiRef (Ours)	~105K sections	Yes	Yes	8	Yes	Yes	Set of citation URLs	One section in another language

Table 2.2: Statistics of popular Wikipedia Summarization datasets. XL=Cross-lingual. ML=Multi-Lingual. MD=Multi-document. SS=Section-specific.

2.3.3 Multi-lingual and cross-lingual summarization

Recently there has been a lot of work on cross-lingual NLG tasks like machine translation [17, 59], question generation [19], news title generation [56], and summarization [89] thanks to models like XNLG [19], mBART [59], mT5 [84], etc.

Limited work has been done in the past on summarization for low-resource languages. MultiLing'15 [38] introduced a novel task for multi-lingual summarization in 30 languages. In the past 2–3 years, a few datasets have been proposed for cross-lingual summarization mainly in the news domain: XLSum [41], MLSum [73], CrossSum [40], Global Voices [63], WikiLingua [52], WikiMulti [76]. XL-Sum [41] comprises ~1.35 million professionally annotated article-summary pairs from BBC, extracted using a set of carefully designed heuristics. It covers 44 languages ranging from low to high resource. Hasan et al. [40] extend the multi-lingual XL-Sum dataset by releasing CrossSum, a cross-lingual summarization dataset with ~1.7 million instances. However, both XL-Sum and CrossSum are specific to the news domain only. WikiLingua [52] is a multi-lingual dataset with ~770K summaries where the article and summary pairs are extracted in 18 languages from WikiHow. MLSum and GlobalVoices are also cross-lingual summarization datasets based on news articles with around ~1.5M and ~300K summaries covering 5 and 15 languages, respectively. We enrich this line of work by contributing a new cross-lingual multi-document summarization dataset, XWikiRef, and also proposing a two-stage system for the associated XWikiGen task.

Chapter 3

Building the datasets for XF2T in low-resource languages

3.1 Overview

F2T generation requires structured data that is well-aligned with semantically equivalent textual data. The manual creation of such a high-quality F2T dataset requires human supervision and is quite challenging to scale. Recently various automatic alignment approaches have been proposed like pairing up Wikipedia sentences with Infobox [54], using distant supervision [31], finding the lexical overlap between textual and structural entities [44, 34, 3], etc.

In this work, we propose the creation of the XF2T dataset, XAlign, consisting of English WikiData triples/facts mapped to sentences from LR Wikipedia. We introduce an extensive high-quality XF2T dataset in 12 languages: Hindi, Marathi, Gujarati, Telugu, Tamil, Kannada, Bengali, Punjabi, Malayalam, Assamese, Oriya, and the monolingual dataset in English. Following guidelines for unsupervised fact-to-text dataset creation [44], we have also collected a human-labeled gold test dataset spanning all these languages to evaluate alignment methods.

3.2 Unsupervised approaches for linking facts to sentences

Our alignment model seeks to match Hindi sentences with the most appropriate English triples. We present two innovative methods for tackling cross-lingual challenges requiring fact and sentence alignment. Key-phrase Extraction with Relevance Ranking and NER-based filtering with Semantic Similarity are the two methods.

Named Entity Disambiguation is a novel concept that is included in NER-based filtering with semantic similarity. By projecting Hindi and English words into the same vector space, we employed nearest neighbor-based search to identify the most pertinent English words for the supplied Hindi words in the sentence. We use Multilingual Unsupervised and Supervised Embeddings (MUSE) [53] to obtain multilingual vector representation and then perform the Nearest Neighbor Search to obtain the top-k candidates. Semantic similarity is used to filter the selected candidates further, improving the model's accuracy. We test a number of cutting-edge

multilingual transformer-based models to identify semantic connections between sentences and facts.

In Key-phrase Extraction with Relevance Ranking, key phrases are extracted from a Hindi sentence using straightforward POS-tag-based heuristics, and key phrases are then ranked according to how relevant they are to the sentence's corresponding constituent article. We propose a new multilingual variant of EmbedRank [9] to obtain rankings. Based on similarity scores with the sentence's key phrases, the top-k relevant triples are then chosen.

3.2.1 Data collection and test data annotation

For retrieving English triples, we use Wikidata as our Knowledge Graph (KG), and for retrieving corresponding sentences, we use Hindi Wikipedia. Wikidata entities and Wikipedia articles have a clear one-to-one mapping that enables us to gather comprehensive data for numerous entities. First, we looked through every domain and subdomain of Wikipedia pages. We decided to choose the *person* domain in Hindi Wikipedia as it contains the maximum number of entities within a domain (~16% of Hindi Wikipedia), allowing us to create a larger dataset. Each entity with a Hindi Wikipedia page has its article content and English triples retrieved and preprocessed. Triples containing useless predicates, such as URLs and external identifiers, are eliminated. We use Hindi sentence tokenization to extract the first three sentences from each article. Our alignment models use this information as their input, and they use the whole candidate set of triples for each entity to forecast a set of triples that are pertinent to each sentence. We generate a total of 29224 English triple and sentence pairs covering 12429 entities using our best-proposed methodology.

In addition to the training and validation sets, we additionally gathered a test set of 460 structured data and text pairs that was human-annotated. From the aforementioned data, we provide some of them to the user in a specifically designed web-based user interface (UI). The sentence and all the candidate triples related to that entity are visible to the user. These samples have each been annotated independently by two authors. The inter-annotator agreement, or Cohen's Kappa score, for the annotations was found to be 0.74. The final test data samples were chosen from the annotation responses of both authors with the assistance of a linguist. 350 data examples are chosen as test dataset, on which we present the metrics results for our methods. To fine-tune the hyperparameters, such as threshold values, the remaining 110 samples are used as an internal validation set. The distribution of sentences and other statistics across different domains can be found in table 3.1.

Domain	Entity count	Sentence count	Sentence count (in test data)	Avg sentence length (in test data)	Avg fact count (in test data)
Actors	2106	5469	50	14.32	3.60
Cricketers	2316	4694	100	21.19	4.70
Politicians	3906	8916	100	18.64	3.47
Writers	2755	6629	50	15.65	1.78
Singers	739	1944	25	18.04	2.92
Journalists	607	1572	25	17.32	2.12
Total	12429	29224	350	17.52	3.08

Table 3.1: Table contains entity count and sentence count for final aligned dataset across different domains. It also presents statistics of manually annotated test data for each domain.

3.2.2 Approaches

3.2.2.1 NER-based filtering with Semantic Similarity

The goal is to filter the English triples using named entity recognition before matching them based on semantic similarity in order to find matching English triples for a given Hindi sentence (s). Our presumption is based on the observation that if a triple contains a Named Entity, the sentence it aligns with will also contain the same Named Entity or a variant thereof. We take into account a triple for determining semantic similarity with the sentence if it lacks a Named Entity.

Each word in the triple is concatenated before being used to extract named entities. Finding the overlap between the Hindi sentence's words and the named entities listed in the triple is our aim. In an Indian language like Hindi, there are numerous ways to write a Named Entity. Therefore, the alignment target would not be met by utilising a straight translation. Furthermore, there can be a translation loss involved.

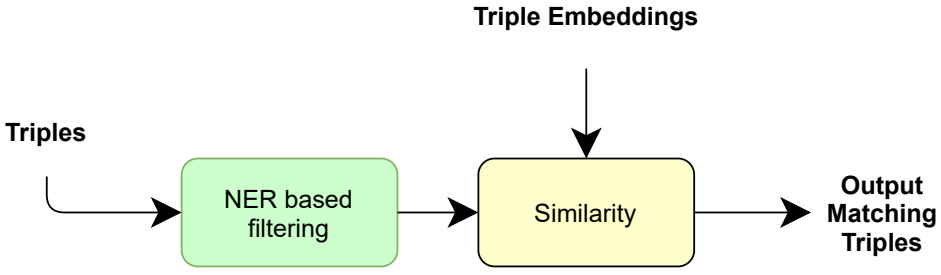

Figure 3.1: NER-based filtering + Semantic Similarity

We employed a pipeline technique with two phases to get around this issue: 1) Triples are filtered using the bucket method, and 2) semantic similarity is used.

By collecting the top k nearest English terms for each word in the given Hindi sentence, s, from the common multilingual vector space produced using MUSE [53], the filtering of triples based on bucket method provides a bucket of English words. The intersection of the named entities identified in triple with the bucket of English terms previously constructed for that Hindi sentence s is then calculated. By dividing the intersection by the total number of words present across all the specified entities, we can finally get a score for each triple. We save facts that score higher than a particular threshold and move on to the next level of semantic similarity.

By computing the inner product between the Hindi sentence representation and fact representation, the semantic similarity approach further refines the triples acquired from the previous stage. As described in Section 3.2.2.4, both sentence-level and fact-level representations are derived from multilingual transformer models. Last but not least, we keep triples over a defined threshold (a different threshold from the previous stage). We have illustrated the pipeline approach in Figure 3.1.

3.2.2.2 Key-phrase Extraction with Relevance Ranking

For this method, we extract the Hindi key phrases from the Hindi Wikipedia page using straightforward POS-tag-based heuristics. A phrase meets our definition of a key phrase if it starts with zero or more adjectives and ends with one or more nouns. Based on how semantically close they are to the source Hindi Wikipedia article, these key phrases are ranked. Key-phrase Extraction with Relevance Ranking is the name of this procedure. A multilingual version of the EmbedRank [9] technique serves as the basis for the ranking system. EmbedRank works by embedding potential phrases and the associated article in the same high-dimensional vector space. The key terms are then ordered inside the same vector space according to how closely they relate to the article. Our variant is explained in Algorithm 1.

The process of obtaining similar triples from ranked key phrases is explained in Figure 3.2. We extract n-grams for each key phrase after ranking them for article A. We determine the vector embeddings for each triple and n-gram. Now a semantic similarity score is calculated for each triple and n-gram pair. For each n-gram, we retain the best matching triple. Next, we find the triples for a key phrase that are most similar across all n-grams. We choose the top-k triples from among them. The top-k triples for a key phrase are those that are most pertinent. We combine the outcomes from each key phrase in a Hindi sentence to find matches at the sentence level.

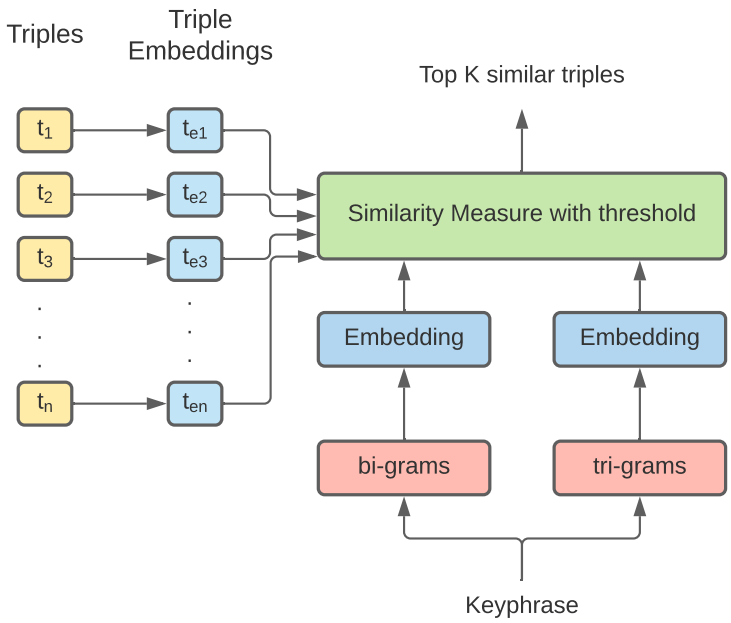

Figure 3.2: Method to return Top K triples from key phrases

Algorithm 1: Ranking key phrases with respect to Article Relevance

1. Let $N = \{$ set of all key phrases in article $A\}$.
2. Concatenate all the key phrases in N and let $Nv \leftarrow$ vector representation of the concatenated key phrases.
3. For a sentence s in the article A, $M \leftarrow$ set of all extracted key phrases from s. So, $M \subseteq N$.
4. For each key phrase K in M, let $Kv \leftarrow$ vector representation of K.
5. Assign a *score* to K , where *score* = similarity between Kv and Nv.
6. Rank all the key phrases in M based on the *score*.

3.2.2.3 Baselines

We experimented with the following baselines:

Multilingual Universal Sentence Encoder [85] is a general-purpose sentence embedding model for applications including retrieving semantic information from texts and transfer learning. It uses a typical dual-encoder neural network with shared weights that has been trained in a multi-task environment with an additional translation task. To eliminate fact triples, we employ the same method as for mBERT.

Word Overlap selects the K-most pertinent English words for each Hindi word that appears in the Hindi sentence using the K Nearest Neighbor search method. Using MUSE [53], a multilingual vector space was built where the word search takes place. These top K English words are all kept in a bucket. The overlap between the triple and the English words in the bucket is then calculated for each sentence. We categorize that triple as being aligned with that sentence if the overlap exceeds a predetermined threshold.

Static Sentence Similarity use MUSE [53] to obtain multilingual word embeddings. In order to represent a Hindi sentence, we find the average of these word embeddings. To obtain fact-level representation for a triple, we average all the word embeddings within the triple. In order to keep triples above a predetermined threshold for a given Hindi sentence, we finally calculate the cosine similarity between sentence level and fact level representation.

mBERT [25] (multilingual Bidirectional Encoder Representations from Transformers) encodes the Hindi text as well as a list of related facts. Concatenating the subject, predicate, and object allows for the verbalization of facts. By averaging the sub-word representations from the final layer of the mBERT, we are able to create vector representations (mean pooling). Next, we calculate the cosine similarity score between the fact-level representation and the text. Finally, we keep fact triples whose similarity score above a predetermined cutoff.

MuRIL [47] (Multilingual Representations for Indian Languages) has a big vocabulary for Indian languages and has been pre-trained on a sizable amount of Indian text corpus. We employ the same method for MuRIL to filter out fact triples as we did for mBERT.

LaBSE [32](Language-Agnostic BERT Sentence Embedding) is a multilingual embedding model that has been pre-trained utilizing the translation language modeling, and masked language modeling aims to encode text from many languages into a common embedding space. We employ the same method for LaBSE to filter out fact triples as we did for mBERT.

XLM-R (STS) and XLM-R (Paraphrase) are sentence transformers that fine-tune XLM-Roberta [21] on semantic text similarity (STS) [13] and on multilingual paraphrase dataset [86] respectively.

3.2.2.4 Experimental settings

We fixed k=5 in k closest neighbor retrieval and set the cutoff value for the Word Overlap method to 1. We translate words that are not commonly used. We employ the most recent base model for all multilingual transformer-based techniques, including mBERT, MuRIL, LaBSE, multilingual universal sentence encoder, and XLM-R. (consists 12 layers) on Huggingface [83].

The threshold value is set to 0.45 for cosine similarity after hyperparameter tuning on our internal validation dataset. We tried various pooling strategies like [CLS] token representation, sum pooling, and mean pooling for sentence-level representation. We found that mean pooling consistently performs the best.

Threshold value	F1-Score
0.35	0.48
0.45	**0.55**
0.55	0.52
0.65	0.38

Table 3.2: Threshold values for sentence-triple semantic similarity on internal validation set for XLM-R (base)

K	F1-Score
3	0.65
4	0.72
5	**0.74**
6	0.66
7	0.67
8	0.68
9	0.66
10	0.63

Table 3.3: K value for K-Nearest neighbor for NER-based filtering with Semantic Similarity method (tested on internal validation set)

By adjusting these hyperparameters on the 110-instance internal validation set, we get the ideal K for K-Nearest Neighbors and the ideal similarity threshold. In Tables 3.2 and 3.3, we present the comprehensive findings of our hyperparameter search. K-Nearest Neighbors' ideal value for K is found to be 5. Similar to this, we see that 0.45 is the ideal value for the similarity threshold. Since XLM-R (base) is the best-performing baseline, we utilise it as the reference transformer-based model.

For recognizing named entities, we use a BERT-CRF tagger trained on the OntoNotes dataset [81]. We use AllenNLP [36] for our NER implementation.

We utilise Stanford coreNLP [60] to detect POS-tags for Key-phrase Extraction with Relevance Ranking and set n-gram values $\in [2, 3]$. With a similarity threshold of 0.45, we employed the multilingual transformer encoder XLM-R (Paraphrase).

S.no	Approaches	Precision	Recall	F1-Score
1	mBERT (mean pooling)	0.37	0.31	0.33
2	Static Sentence Similarity	0.38	0.48	0.42
3	Multilingual Universal Sentence Encoder	0.62	0.38	0.47
4	Word Overlap	0.50	0.52	0.51
5	LaBSE (mean pooling)	0.49	0.56	0.52
6	XLM-R (STS)	0.57	0.48	0.52
7	MuRIL (mean pooling)	0.55	0.51	0.53
8	XLM-R (paraphrase)	0.52	0.58	0.55
9	Key-phrase Extraction with Relevance Ranking	0.78	0.72	0.75
10	NER based filtering with Semantic similarity	**0.79**	**0.83**	**0.81**

Table 3.4: Precision, Recall and F1-score across different approaches.

3.2.3 Evaluation metrics, results, and analysis

Our evaluation measures include micro-averaged *Precision*, *Recall*, and *F1-Score*. As can be seen from the findings in Table 3.4, MuRIL outperforms mBERT since it is only pre-trained on Indian languages with large vocabulary sizes. Surprisingly, word overlap, a straightforward method, had higher recall than MuRIL. The rationale is that, as described in section 3.2.2.3, it finds k-nearest neighbours in a multilingual vector space. As a result, this method retrieves the information while capturing more word variations. In baselines, the XLM-R model outperforms other multilingual transformers since it has been optimised for the downstream tasks related to text similarity. On the translation language modelling loss, LaBSE has prior training. The semantic similarity between facts and sentences from various languages is so successfully captured.

We note the remarkable precision of key-phrase extraction with relevance ranking. The procedure makes sure to maintain only the key phrases that are extremely essential to the content by capturing the relation of each key phrase with its corresponding article. N-gram matching with triples is used to further hone the matches.

Surprisingly, the best performance in terms of precision and recall is provided by NER-based filtering with Semantic Similarity. This finding demonstrates a considerable bias toward named entities serving as the principal source of factual information in the most pertinent fact triples. Therefore, the NER-based model still outperforms our Key-phrase Ranking technique even though it takes the complete context of an article into account to find relevant terms.

Figure 3.3: Error analysis of incorrect predictions

Only the top-ranked triples that are the most relevant are kept for Key-phrase Extraction with Relevance Ranking. The model does not always capture all of the pertinent triples, especially when numerous triples transmit the same information. Due to the fact that only triples with the highest rank are taken into account, triples with a comparable ranking may be overlooked. The first example in Figure 3.3 is a sample that was predicted using the Key-phrase extraction model. We note that the ranking process causes the model to ignore the occupations of author and novelist.

We observe that fact triples without named entities are sometimes ignored by the model for NER-based filtering with Semantic Similarity. A predicted sample by the NER-based model is shown in the second example in Figure 3.3. Since "politician" is not a named item, we note that the model has ignored the occupation.

3.2.4 Limitations

We implement the above-mentioned unsupervised approaches in the Hindi language and rely on multilingual word embeddings, MUSE, that are static in nature. These factors can prove to be a limitation while deploying a multilingual system with sentences in multiple target languages simultaneously. There have been efforts in the community in implementing semi-supervised, distantly supervised, as well as few-shot learning for this task. Above work fails to take into consideration these paradigms. Considering the recent advances in the contextual representations and better approaches for cross-lingual information retrieval, we experiment with such methods, across 7 more languages at once. We discuss this work in upcoming section.

3.3 Scaling the fact-sentence alignment to multiple languages

3.3.1 Overview

Following the previous work, we now formulate our problem in a more challenging and useful way. We aim to align English facts with sentences in any of the 7 target languages, using a single modelling approach. For every (entity e, language l) pair, the dataset has a set F_{el} of English Wikidata facts and a set of Wikipedia sentences S_{el} in that language. But the sentences and facts are not aligned with each other. The goal of this section is build an automatic aligner that associates a sentence in S_{el} with a subset of facts from F_{el}. The Wikidata facts for a particular entity can grow in number for some entities. Hence, we propose a two stage system, where the first stage aims to reduce the search space, as well as achieving high recall. Second stage aims at filtering the irrelevant facts and precisely selecting the ones relevant to the sentence. The two stages are Candidate Generation and Selection. The first stage generates (facts, sentence) candidates based on automated translation and syntactic+semantic match. The second stage retains only those candidates which are strongly aligned using transfer learning and distant supervision. Fig. 3.4 shows the overall F2T alignment flow.

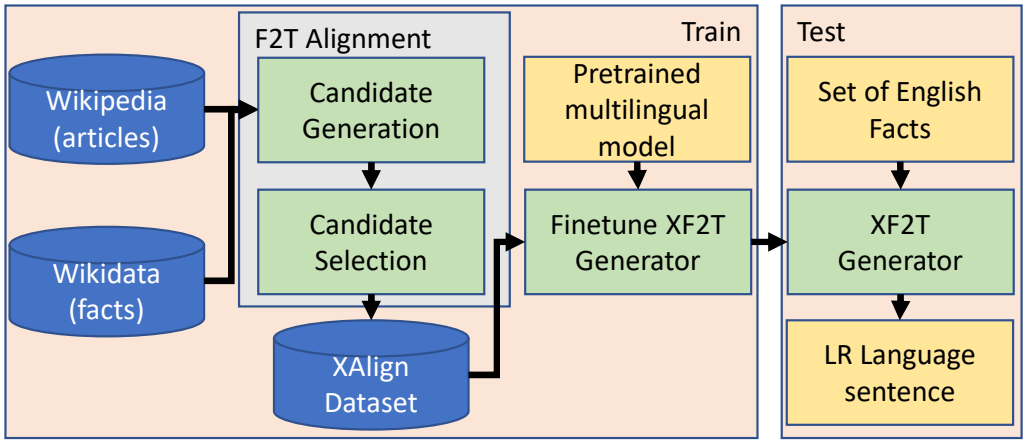

Figure 3.4: XAlign F2T Alignment System Architecture

3.3.2 Data collection and preprocessing

The XAlign is an XF2T dataset that consists of sentences from LR language Wikipedia mapped to English fact triples from WikiData. It contains data for the following languages: Hindi (hi), Telugu (te), Bengali (bn), Gujarati (gu), Marathi (ma), Kannada (kn), Tamil (ta), Malayalam(ml), Assamese(as), Oriya(or), Punjabi(pa), and English (en) that are included in recent research papers on Indian languages like Samanantar [70], IndicNLPSuite [46] and

MuRIL [47]. In this section, we discuss the broad data collection and basic pre-processing steps for the dataset.

To start, we compile a list of Wikidata person entities with links to relevant Wikipedia pages in at least one of our 11 LR languages. We chose the person entity type because it has the most Wikidata entities connected to at least one of our 11 LR languages, and Wikidata has a high quantity of facts per entity for the person type. This leads to a dataset D where every instance d_i contains a tuple ⟨entityID, English Wikidata facts, LR language, and LR-language Wikipedia URL for the entityID⟩.

The forward (subject-centric) and backward (object-centric) facts for each entity in D for all 12 languages were extracted using the 20201221 WikiData dump. For the entity "Michael Faraday", ⟨Michael Faraday, occupation, Physicist⟩ is an example of a forward fact, while ⟨Humphry Davy, student, Michael Faraday⟩ is an example of a backward fact. Using the WikiData API[1], we filtered out backward facts if the corresponding forward fact was present. Also, we gathered facts corresponding to only these Wikidata property (or relation) types which capture most of the useful factual information for person entities and ignore non-informative properties like unique_resource_ids: WikibaseItem, Time, Quantity, Monolingualtext. If there exists additional supporting information associated with the fact triple, we retain it as a fact qualifier.

The LR-language Wikipedia document for each instance in dataset D is parsed to produce a list of clean phrases that may be aligned with the English Wikidata facts in the instance using the methods below. We extracted text from the 20210520 Wikipedia XML dump using the Wikiextractor[5] for each language. Wikiextractor produces clean main content by automatically removing tables, photos, links, Infoboxes, references, etc. Using the Indic NLP Library, [51], and a few extra heuristics to take into consideration Indic punctuation characters, sentence delimiters, and non-breaking prefixes, we divided the articles' primary material into sentences. On Wikipedia, sentences written in different languages can occasionally be found. We prune out such sentences using Polyglot language detector[2]. Sentences containing fewer than five words or more than one hundred words were eliminated. We use POS tagging to weed out sentences that might be devoid of factual information and only keep the sentences that have at least one noun or verb. For POS tagging, we used Stanza [68] for en, hi, ma, te, ta, ml, pa; LDC Bengali POS Tagger [6] for Bengali; and [65] for Gujarati. We compiled a list of entities for these languages as a backup. This is accomplished by keeping track of all Wikipedia articles that cite or are cited in another page. After that, this list is combined to produce a comprehensive list of entities for the target language. Finally, we took WikiData's native language labels for each of these entities. We manually generate a set of pronouns for the target language because the above method will overlook factual sentences that contain pronouns. If the entity-pronoun list

[1] `https://query.wikidata.org/`

[2] `https://polyglot.readthedocs.io/en/latest/Detection.html`

and the words in the provided phrase overlap, we keep the sentences. We additionally keep the section information for each sentence per Wikipedia URL.

3.3.3 Candidate generation

Given a set of English facts $\{f_i\}_{i=1}^{|f|}$ and set of sentences $\{s_j\}_{j=1}^{|s|}$ in language l, we compute a similarity score that captures syntactic as well as semantic similarity between a (fact f_i, sentence s_j) pair. For syntactic match, we use TFIDF by translating either the fact to language l or the sentence to English [70]. For semantic match, we compute cosine similarity between MuRIL [47] representations of the fact and the sentence, or between their translations. Besides MuRIL, we also experimented with mBERT [26], XLM-R [23] and LaBSE [33], but found MuRIL to perform the best on a small dataset of 500 examples, separately annotated for Stage-1 quality evaluation.

The similarity score $sim(f_i, s_j)$ is thus obtained as an average of the following 4 scores: MuRIL(f_i, s_j), TFIDF-cos(translate(f_i, l), s_j), TFIDF-cos(f_i, translate(s_j, English)), MuRIL(translate(f_i, l), translate(s_j, English)). For translating sentences, we use IndicTrans [70]. When translating the facts, we retain the label of entities within the fact tuple for which Wikidata multi-lingual label is present in LR language, and we translate the remaining parts of the fact.

We obtain $sim(f_i, s_j)$, i.e. a score between 0 and 1, for every (fact, sentence) pair. We filter out sentences if the most similar fact has similarity score less than a threshold τ. By manual inspection, we fix $\tau = 0.65$. For every remaining sentence, we retain at most top-K facts sorted according to their scores. Empirically we observe that most sentences can be covered by less than 10 facts. Hence, we fix K=10.

3.3.4 Manual Annotations for Ground-Truth Data

We performed manual annotation in two phases. For both the phases, the annotators were presented with (LR language sentence, K facts) which were output by Stage 1. In short, the annotators were asked to do the following: "The task is to mark English facts that are present in the given LR language sentence. You should choose all the facts that can be inferred from the given sentence by selecting the checkbox against it. Also, mention if the set of selected facts partially/completely cover the semantic information mentioned in the sentence." There were also specific guidelines to ignore redundant facts, how to handle abbreviations, etc. More detailed annotation guidelines are mentioned in the Appendix.

We received 60 examples per language for labelling in the initial round. An skilled team of eight annotators who were trusted graduate students and had a thorough understanding of the assignment completed the annotations. In phase 2, the labelled data served as the ideal control set for quality assurance. Using the identical set of 60 cases per language as in phase 1, we first evaluated annotators. We chose eight annotators each language for this test in phase

2, totaling 64 annotators drawn at random from the National Register of Translators.[3]. After that, we requested the top four annotators from each language (based on Kappa score with golden annotations) to annotate 1000 occurrences. So, in phase 2, a group of 25 crowdsourced workers completed our final annotations. Table 3.6 provides the average Kappa scores for the annotators by language.

3.3.4.1 Instructions related to platform

- When you select a question, you will see a sentence in low resource (LR) language and a list of English facts.
- Please read the LR sentence carefully. Although English translated sentence is provided for the reference, don't rely entirely on it. The translated sentence may not be accurate all the time.
- You will find list of English facts below the sentence. Please choose the facts that can be inferred from the given sentence by selecting the checkbox against it.
- If the sentence is grammatically incorrect, incomplete or erroneous for any other reason, please mention the reason in the textbox at the bottom.

3.3.4.2 Instructions related to annotations

- Exact Fact Matching: Information should exactly match what is present in the sentences (some exceptions are mentioned later; other than them, follow this rule strictly). For example,
  - Sentence: टीना मुनीम (जन्मः 11 फरवरी, 1955) हिन्दी फ़िल्मों की एक अभिनेत्री हैं।
  - English Translation: Tina Munim (DOB: 11 Feb 1955) is an actress who acts in Hindi movies.
  - Fact: Date of Birth | 11 February 1957.
  - Although the fact mentions that date of birth is 11 Feb 1957 but we won't consider it as a valid alignment for the sentence.
- Implied Information in facts
  - If information is related to language related inference and does not require external world knowledge (a piece of knowledge not embedded in language itself), we mark that fact.

[3] https://www.ntm.org.in/languages/english/nrtdb.aspx

    * Sentence: पी॰ नागराजन भारत की सोलहवीं लोकसभा में सांसद हैं ।
    * English Translation: P. Nagarajan is a Member of Parliament in India's 16th Lok Sabha.
    * Facts: P Nagarajan | position held | Member of the 16th Lok Sabha : P Nagarajan | occupation | politician.
    * For the given sentence, the information that the subject is a politician (राजनेता) isn't written, but we can say that a Member of Parliament will be a politician, hence we mark it.
    * As another example, consider a sentence that says that a person did her Masters in Geography but doesn't explicitly mention her occupation directly. Still, we can mark the occupation= geographer fact as valid.
  - If information in the fact requires external world knowledge, we DO NOT mark that fact.
    * Sentence: अमृता मलयाली माँ और पंजाबी पिता की संतान हैं और वह मुंबई में पैदा हुई थी।
    * English Translation: Amruta's mother is a Malayali and her father is a Punjabi, and she was born in Mumbai.
    * Fact: Place of Birth | Chembur.
    * Even if you know that Chembur is in Mumbai, please don't mark it.
- If some facts contain redundant information , then dont mark it.
- Abbreviations: If the part of the sentence is abbreviated in the facts or if the part of fact is abbreviated in the sentence, we don't consider those facts.
  - Sentence: फील्ड मार्शल आर्किबाल्ड पेर्सियल वेवेल , पहले अर्ल वावेल , जीसीबी , जीसीएसआई , जीसीआईई , सीएमजी , वीएम , केएसटीजे , पीसी (5 मई 1883 – 24 मई 1950) , ब्रिटिश सेना के एक वरिष्ठ अधिकारी और भारत के वाइसराय थे ।
  - English Translation: Field Marshal Archibald Percival Wavell, 1st Earl Wavell, GCB, GCSI, GCIE, CMG, MC, KStJ, PC (5 May 1883 — 24 May 1950) was a senior officer of the British Army and an Indian Viceroy.
  - Facts: Archibald Wavell, 1st Earl Wavell | award received | Virtuti Militari
- Fact Generalisation

- If specific information is present in the sentence but there isn't an exact match in the fact list, then select the apt synonyms.
  - Sentence: उन्होने अपनी कविताओं से एक अच्छे साहित्यकार की छवि स्तापित कर ली थी
  - English Translation: He had established the image of a good litterateur through his poems.
  - Now if the fact list contains occupation as poet, and there is no other fact with occupation as litterateur, we consider the apt synonym and mark this fact as valid.
- If facts contain more specific terms as compared to the term present in the sentence then consider that fact for annotation (facts can contain more specific information).
  - Sentence: राजगोपाल चिदम्बरम (जन्म 12 नवम्बर 1936) जिन्हें सामान्यतः आर॰ चिदम्बरम के नाम से जाना जाता है , पद्मविभूषण सम्मानित भारतीय वैज्ञानिक हैं ।
  - English Translation: Rajagopal Chidambaram (born 12 November 1936), commonly known as R. Chidambaram, is a Padma Vibhushan honored Indian scientist.
  - Fact: Rajagopala Chidambaram | occupation | nuclear physicist
  - We mark this fact as a nuclear physicist is also a वैज्ञानिक (scientist). The fact has more specific information and we mark it as valid.

3.3.5 Candidate selection

For every entity and language pair, Stage 1 outputs sentences each associated with a maximum of K facts. To maintain only strongly aligned (fact, sentence) pairs, we employ two alternative methods: distant supervision from another English-only F2T dataset and transfer learning from an NLI (Natural language Inference) task.

3.3.5.1 Zero shot learning based approaches

The NLI problem and the (fact, sentence) alignment problem are semantically related since the fact and the sentence can be thought of as the premise and the hypothesis, respectively. We test the XLM-R, mT5, and MuRIL multilingual NLI models. We utilise Huggingface's Xtreme-XNLI fine-tuned checkpoints[4] and evaluate them for the (fact, sentence) alignment problem as follows. We feed these models "sentence<SEP>fact" at the moment of inference. The (fact, sentence) pair is thought to be aligned if the model predicts entailment; otherwise, it is not. We choose a subset of facts from among the K candidate facts for each sentence. To select the

[4]We finetuned MuRIL only for en, hi, and ur because it does not support vocabulary for all XNLI languages.

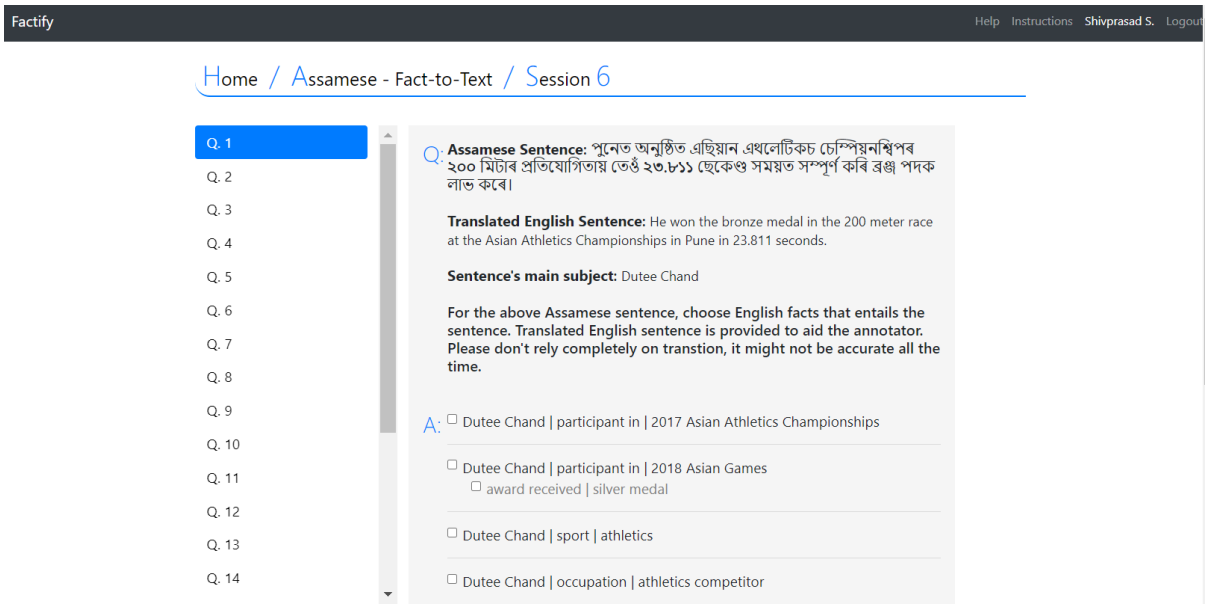

Figure 3.5: Annotation interface provided to the user. The user has to select the facts relevant to the given sentence. English translation is given to aid the user.

optimal model, the chosen fact list is then contrasted with the golden fact list for the given sentence.

We have investigated models tuned on several methodologies in order to examine the effectiveness of various models for cross-lingual knowledge transfer:

(1) mT5-large finetuned on translate-train XNLI dataset and MNLI dataset.<native language premise, native language hypothesis> (15 languages which includes two Indian language Hindi and Urdu) [4.5M training instances]

(2) MuRIL finetuned on cross-lingual premise-hypothesis for one language pair: <Hindi premise, English hypothesis> (google released translate train dataset on TF.) [0.3M training instances]

(3) MuRIL finetuned on cross-lingual premise hypothesis for 13 Indian language pairs: <Native language premise, English hypothesis> (used IndicTrans for translating English premise to 11 Indian languages). [3.9M training instances]

(4) MuRIL finetuned on a combination of different premise-hypothesis language pairs for Hindi: <English premise, Hindi hypothesis>, <English premise, English hypothesis >, <Hindi premise, Hindi hypothesis >. [0.9M training instances]

(5) MuRIL trained on mixed cross-lingual premise-hypothesis for one language pair. Here Hindi premise is concatenated with the English premise, and the hypothesis is English. <Hindi premise: English premise, English hypothesis>

(6) MuRIL trained on mixed cross-lingual premise hypothesis for 13 language pairs. <Native language premise: English premise, English hypothesis>

3.3.5.2 Distant supervision-based approaches

In this method, given an (English fact, LR language sentence) pair, we train a binary classifier to predict whether the fact is associated with the LR language sentence or not. The (fact, sentence) input is expressed as the input string "sentence<SEP>subject|predicate|object". For this task, we leverage the Knowledge Enhanced Language Modelling (KELM) [3] dataset.

KELM is a distantly supervised dataset with automatically aligned (Wikipedia sentence, Wikidata facts) for English language. For a Wikipedia page corresponding to Wikidata entity e, a sentence s is aligned with a Wikidata fact $f = \langle e, r, e' \rangle$ if s contains subject e and object e'.

It focuses on subject-centric facts only where candidate facts belonging only to the Wikidata of the entity of interest. Authors further shortlist the facts based on lexical overlap between the sentence and the object aliases from the fact. They also apply specific heuristics for data types like date and quantity.

For a given english sentence, it has list of triples and quadruples. We filter out the quadruples (conveys the qualifier information) for the given sentences and retained only the triples. An additional of having more than one facts is imposed to get data-instances that has high coverage of factual information within the sentence. We included 2,70,880 data-instance out of 7,96,981 present in kelm test dataset. We used FAISS[5] maximum inner product search (MIPS) package to find sentence similar to the given sentence. As sentences are in English, we used Distil Bert-Base to obtain the vector representation of sentence used by FIASS.

For every sentence in the dataset, we create a positive instance for every fact aligned with the sentence. For example, if sentence s has two aligned facts f_1 and f_2, we create two positive instances. For every positive instance, we also create a negative instance as mentioned next. We order all the other sentences on the same Wikipedia page (which contains s) in decreasing order of semantic similarity and choose a sentence s' randomly from top 10. We skipped top two sentences as they can be very similar to sentence s. We then use a fact extracted from sentence s' along with the original sentence s as a negative instance. We split the dataset in 90:10 for training and validation. Overall, the dataset contains 1,177,636 (54% positive, 46% negative) training instances and 130,849 (54% positive, 46% negative) validation instances.

Since our dataset is cross-lingual in nature, for inference on output of the Stage 1 data, we experiment with cross-lingual, translate-test and translate-train settings. We observe that the translate-train setting performs the best and hence report results in Table 3.5 using this setting.

We observe that mT5 constantly outperforms XLM-Roberta MuRIL across all the languages.

[5] `https://github.com/facebookresearch/faiss`

	hi	mr	te	ta	en	gu	bn	kn	Avg.
Baselines									
KELM-style [3]	0.493	0.426	0.368	0.451	0.41	0.372	0.436	0.338	0.411
WITA-style [34]	0.507	0.574	0.517	0.459	0.602	0.500	0.535	0.530	0.528
Stage-1 + TF-IDF	0.750	0.685	0.693	0.718	0.737	0.701	0.787	0.647	0.715
Distant supervision based methods									
MuRIL	0.763	0.684	0.74	0.755	0.705	0.785	0.624	0.677	0.717
XLM-Roberta	0.781	0.69	0.765	0.739	0.765	0.785	0.669	0.724	0.740
mT5	0.79	0.714	0.776	0.786	0.766	0.8	0.698	0.705	0.754
Transfer learning based methods									
MuRIL	0.716	0.717	0.765	0.751	0.734	0.787	0.795	0.718	0.748
XLM-Roberta	0.772	0.767	0.78	0.812	0.79	0.805	0.831	0.727	0.786
mT5	0.902	0.831	0.841	0.886	0.845	0.851	0.751	0.785	0.837

Table 3.5: Stage-2 (Fact, Sentence) Candidate Selection F1 Scores across different methods. For TF-IDF based aligner, we used candidates generated from the stage-1 process. For KELM and WITA-style aligners, we followed the ranking algorithm mentioned in their paper and didn't apply the stage-1 aligner.

Table 3.5 shows candidate selection F1 scores across all the languages on our golden annotated dataset. Besides our proposed transfer learning and distant supervision based models, we also compare with the KELM-style [3] and WITA-style [34] alignment baselines. All experiments were run on a machine with four 10GB RTX 2080 GPUs. We finetune for 5 epochs with L2-norm weight decay of 0.001 and dropout of 0.1. We set the learning rate of 1e-5, 2e-5 and 1e-3 for XLM-RoBERTa, MuRIL and mT5 respectively. We use batch size set of 32, 32 and 16 for XLM-RoBERTa, MuRIL and mT5 resp. We observe that mT5 with transfer learning performs the best.

3.3.6 XAlignV2, and dataset analysis

In first phase of the work, we worked on English and 7 LR languages, i.e., Hindi (hi), Telugu (te), Bengali (bn), Gujarati (gu), Marathi (ma), Kannada (kn), Tamil (ta). We run mT5-transfer-learning Stage 2 aligner on Stage 1 output to get Train+Validation part of XAlign. In later phase of our work, we extend above methodology to 4 more LR languages, i.e., Malayalam(ml), Assamese(as), Oriya(or), and Punjabi(pa). We call this extended version of the XAlign as XAlignV2.

Table 3.6 shows dataset stats. Figs. 3.6 and 3.7 show fact count distribution. We observe that a large percent of sentences contain more than one fact across languages. Also, the distribution

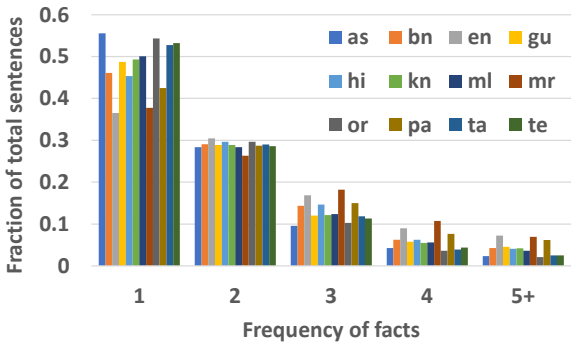

Figure 3.6: Fact Count Distribution across languages

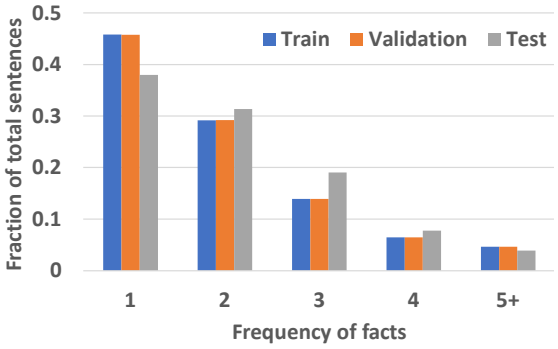

Figure 3.7: Fact Count Distribution across data subsets

is similar across languages and data subsets. Finally, Table 3.7 shows top 10 frequent fact relations across all the languages.

	|V|	Train+Validation			Manually Labeled Test				
		|I|	|T|	|F|	κ	|A|	|I|	|T|	|F|
hi	75K	57K	25.3/5/99	2.0	0.81	4	842	11.1/5/24	2.1
mr	50K	19K	20.4/5/94	2.2	0.61	4	736	12.7/6/40	2.1
te	61K	24K	15.6/5/97	1.7	0.56	2	734	9.7/5/30	2.2
ta	121K	57K	16.7/5/97	1.8	0.76	2	656	9.5/5/24	1.9
en	104K	133K	20.2/4/86	2.2	0.74	4	470	17.5/8/61	2.7
gu	35K	9K	23.4/5/99	1.8	0.50	3	530	12.7/6/31	2.1
bn	131K	121K	19.3/5/99	2.0	0.64	2	792	8.7/5/24	1.6
kn	88K	25K	19.3/5/99	1.9	0.54	4	642	10.4/6/45	2.2
pa	59K	30K	32.1/5/99	2.1	0.54	3	529	13.4/5/45	2.4
as	27K	9K	19.23/5/99	1.6	-	1	637	16.22/5/72	2.2
or	28K	14K	16.88/5/99	1.7	-	2	242	13.45/7/30	2.6
ml	146K	55K	15.7/5/98	1.9	0.52	2	615	9.2/6/24	1.8

Table 3.6: Basic Statistics of XAlignV2. |I|=# instances, |T|=avg/min/max word count, |F|=avg #facts, |V|=Vocab. size, κ=Kappa score, |A|=#annotators. For Train+Validation, min and max fact count is 1 and 10 resp across languages.[6]

3.4 Summary

We extensively covered in this chapter the methodology to build the datasets for the task of our interest, i.e. cross-lingual fact-to-text generation(XF2T). We focused on the data collection,

hi	occupation, date of birth, position held, cast member, country of citizenship, award received, place of birth, date of death, educated at, languages spoken written or signed
mr	occupation, date of birth, position held, date of death, country of citizenship, place of birth, member of sports team, member of political party, cast member, award received
te	occupation, date of birth, position held, cast member, date of death, place of birth, award received, member of political party, country of citizenship, educated at
ta	occupation, position held, date of birth, cast member, country of citizenship, educated at, place of birth, date of death, award received, member of political party
en	occupation, date of birth, position held, country of citizenship, educated at, date of death, award received, place of birth, member of sports team, member of political party
gu	occupation, date of birth, cast member, position held, award received, date of death, languages spoken written or signed, place of birth, author, country of citizenship
bn	occupation, date of birth, country of citizenship, cast member, member of sports team, date of death, educated at, place of birth, position held, award received
kn	occupation, cast member, date of birth, award received, position held, date of death, performer, place of birth, author, educated at
pa	occupation, date of birth, place of birth, date of death, cast member, country of citizenship, educated at, award received, languages spoken, written or signed, position held
as	occupation, date of birth, cast member, position held, date of death, place of birth, country of citizenship, educated at, award received, member of political party
or	occupation, date of birth, position held, cast member, member of political party, place of birth, date of death, award received, languages spoken, written or signed, educated at
ml	occupation, cast member, position held, date of birth, educated at, award received, date of death, place of birth, author, employer

Table 3.7: Top-10 frequent fact relations across languages.

data pre-processing to ensure high quality of samples, and analyzed the data for key parameters. We then implemented two unsupervised approaches, and later moved on to more sophisticated approaches for aligning facts to sentences. We found out that mt5 with transfer learning finetuned on NLI task performs the best, and hence we use it to prepare our final dataset XAlignV2. This dataset is indeed a key contribution to the field of text generation in NLP, and can be used as a benchmark for several research problems. In upcoming section, we leverage this dataset and experiment with several text generation models in multilingual setting. We systematically explore various strategies for improving XF2T generation like multi-lingual data-to-text pre-training, fact-aware embeddings, and structure-aware encoding.

Chapter 4

Approaches for cross-lingual fact-to-text generation

4.1 Problem formulation

The XF2T generation task takes a set of English facts as input and generates a sentence capturing the fact-semantics in the specified language. Fig. 1.3 shows an example where a set of English Wikidata facts are used to generate a sentence across various languages.

We model this as a multi-lingual text generation task and hence experiment with multiple multilingual deep learning models. For all cross-lingual fact-to-text generation models except mT5 and translation baseline, we use a vocabulary size of 64K subword learnt from training corpus using SentencePiece [50] tokenizer. We use Pytorch-lightning and Huggingface for training all the models. For the transformer model, we use 6 encoder and decoder layers, input embeddings of size 512 with 8 attention heads and feedforward dimension of 2048. We optimized the cross entropy loss using the AdamW optimizer. We use an initial learning rate of 1e-4, 4000 warmup steps and the learning rate annealing schedule as proposed in Vaswani et al. [77]. We finetune the transformer with batch size of 64 for 100 epochs and early stopping with patience of 15. We finetune mT5-small model with constant learning rate of 3e-5, batch size of 24, weight decay 0.001 and dropout of 0.1. We optimize cross entropy loss using the Adafactor optimizer for 30 epochs. For all models, we use beam search with a beam size of 5 and length penalty set to 1.

Evaluation Metrics: We use overall BLEU scores [70] for evaluating the multi-lingual models for English-Indic fact-sentence pairs. Following previous work, we also use METEOR [7] and chrF++ [66]. BLEU, METEOR and CHRF++ were originally designed to evaluate machine translation systems. PARENT [27] relies on the word overlap between input and the prediction text. Since the input and prediction in XF2T are in different languages, we cannot compute PARENT scores.

	BLEU	METEOR	chrF++
Vanilla Transformer	21.93	50.21	50.89
IndicBART	23.78	50.80	53.88
mT5	**28.13**	**53.54**	**57.27**

Table 4.1: XF2T scores on XAlignV2 test set using standard Transformer-based encoder-decoder models. The best results are highlighted.

4.2 Efficient encoding of input facts

Each input instance consists of multiple facts $F = \{f_1, f_2, \ldots, f_n\}$ and a section title t. A fact f_i is a tuple composed of subject s_i, relation r_i, object o_i and m qualifiers $Q = q_1, q_2, \ldots, q_m$. Each qualifier provides more information about the fact. Each of the qualifiers $\{q_j\}_{j=1}^{m}$ can be linked to the fact using a fact-level property which we call as qualifier relation qr_j. For example, consider the sentence: "Narendra Modi was the Chief Minister of Gujarat from 7 October 2001 to 22 May 2014, preceded by Keshubhai Patel and succeeded by Anandiben Patel." This can be represented by a fact where subject is "Narendra Modi", relation is "position held", object is "Chief Minister of Gujarat" and there are 4 qualifiers each with their qualifier relations as follows: (1) q_1="7 October 2001", qr_1="start time", (2) q_2="22 May 2014", qr_2="end time", (3) q_3="Keshubhai Patel", qr_3="replaces", and (4) q_4="Anandiben Patel", qr_4="replaced by".

Each fact f_i is encoded as a string and the overall input consists of a concatenation of such strings across all facts in F. The string representation for a fact f_i is "$\langle S\rangle s_i \langle R\rangle r_i \langle O\rangle o_i \langle R\rangle qr_{i_1} \langle O\rangle q_{i_1} \langle R\rangle qr_{i_2} \langle O\rangle q_{i_2} \ldots \langle R\rangle qr_{i_m} \langle O\rangle q_{i_m}$" where $\langle S\rangle$, $\langle R\rangle$, $\langle O\rangle$ are special tokens. Finally, the overall input with n facts is obtained as follows: "generate [language] f_1 f_2 ... f_n $\langle T\rangle [t]$" where "[language]" is one of our 12 languages, $\langle T\rangle$ is the section title delimiter token, and t is the section title.

4.3 Approaches

4.3.1 Baseline sequence-to-sequence models

For XF2T generation, we train multiple popular multi-lingual text generation models on the Train+Validation part of our XAlign dataset. We use a basic Transformer model, mT5-small finetuned on Xtreme XNLI and the IndicBART [24] for the XF2T task. We do not experiment with mBART [59] and Muril [47] since their small-sized model checkpoints are not publicly available. We train these models in a multi-lingual cross-lingual manner. Thus, we train a single model using training data across languages without requiring translation.

	Vanilla Transformer				IndicBART				mT5			
	BLEU	METEOR	chrF++	LaBSE	BLEU	METEOR	chrF++	LaBSE	BLEU	METEOR	chrF++	LaBSE
hi	35.04	63.46	60.85	86.04	40.44	66.41	66.27	88.51	**44.65**	**68.58**	**68.49**	**90.28**
mr	18.28	50.66	49.87	77.73	**28.08**	55.35	57.73	82.79	26.47	**56.85**	**59.17**	**85.25**
te	6.95	36.17	41.70	77.44	**15.67**	41.52	50.40	80.46	14.46	**43.45**	**52.58**	**83.47**
ta	14.67	44.64	53.03	80.63	**19.37**	45.78	56.63	83.07	18.37	**46.15**	**57.42**	**84.53**
en	37.12	65.32	59.69	80.05	10.47	42.35	34.35	66.38	**46.94**	**70.60**	**65.20**	**84.91**
gu	15.66	47.70	46.29	79.46	19.16	47.92	49.30	78.64	**22.69**	**50.31**	**51.36**	**84.36**
bn	48.55	74.18	75.68	87.97	**55.90**	**79.29**	**80.51**	**91.44**	40.38	61.71	68.71	84.17
kn	4.78	28.96	37.60	71.90	10.30	**33.55**	46.65	77.13	**10.66**	32.58	**46.92**	**80.45**
ml	16.29	50.84	47.26	80.13	**27.41**	56.27	56.80	86.08	26.22	**56.71**	**57.01**	**86.53**
pa	17.76	50.27	44.73	77.82	22.32	53.20	50.74	81.34	**26.96**	**54.82**	**52.33**	**84.11**
or	39.94	61.09	62.79	81.33	22.16	53.76	58.30	77.56	**47.17**	**67.82**	**71.20**	**86.05**
as	8.08	29.27	31.24	60.18	**14.07**	**34.25**	**38.87**	63.58	12.61	32.93	36.91	**65.84**
Avg	21.93	50.21	50.89	78.39	23.78	50.80	53.88	79.75	**28.13**	**53.54**	**57.27**	**83.33**

Table 4.2: Detailed results for standard models across all the 12 languages. The best results for a (metric, language) combination are highlighted.

Table 4.1 shows BLEU results across different (model, metric) combinations using three standard Transformer-based encoder-decoder models. Across the 12 languages, on average for each metric, mT5 performs better than IndicBART which in turn is better than vanilla Transformer. We observed that IndicBART performed exceptionally well for Bengali but is exceptionally poor on English. Given that mT5 is better on average amongst the three, we perform further experiments using mT5.

4.3.1.1 Comparison of monolingual, bilingual, multilingual models

Next, we experiment with different training setups. Traditionally in cross-lingual settings, it has been observed that bi-lingual models could be more accurate for some language pairs. Note that in our case, input is always in English while the output could be in any of the 12 languages. Hence, we train bi-lingual models, i.e., one model per language since our input is always in English. A drawback with this approach is the need to maintain one model per language which is cumbersome.

Further, we also train two translation based models. In the "translate-output" setting, we train a single English-only model which consumes English facts and generates English text. The English output is translated to desired language at test time. In the "translate-input" setting, English facts are translated to LR language and fed as input to train a single multi-lingual model across all languages. While translating if mapped strings for entities were present in Wikidata they were directly used. A drawback with these approaches is the need for translation at test time.

Table 4.3 shows results when the mT5 model is trained using various bi-lingual, multi-lingual, and translation-based settings. We observe that across all settings, the initial setting of training a single multi-lingual cross-lingual model is the best on average across all metrics. That said, for Bengali, a bi-lingual model, i.e., a model specifically trained for en→bn, is much better.

	BLEU	METEOR	chrF++
Bi-lingual mT5 (12 models)	25.88	50.91	52.88
Translate-Output mT5 (1 model)	18.91	42.83	49.10
Translate-Input mT5 (1 model)	26.53	52.24	55.32
Multi-lingual mT5 (1 model)	**28.13**	**53.54**	**57.27**

Table 4.3: XF2T scores on XAlignV2 test set using bi-lingual, multi-lingual and translation-based variants of mT5 model. Best results are highlighted.

	Bi-lingual (12 models)				Translate-Output (1 model)				Translate-Input (1 model)				Multi-lingual (1 model)			
	BLEU	METEOR	chrF++	LaBSE	BLEU	METEOR	chrF++	LaBSE	BLEU	METEOR	chrF++	LaBSE	BLEU	METEOR	chrF++	LaBSE
hi	41.07	66.15	65.57	88.40	24.88	55.91	54.48	83.85	41.98	66.14	66.47	89.32	**44.65**	**68.58**	**68.49**	**90.28**
mr	16.74	49.36	48.40	78.00	20.62	46.87	52.23	82.59	24.90	54.56	57.25	83.14	**26.47**	**56.85**	**59.17**	**85.25**
te	12.23	37.85	44.94	78.93	14.13	38.69	50.36	82.78	13.11	40.83	49.64	81.19	**14.46**	**43.45**	**52.58**	**83.47**
ta	18.37	**46.57**	57.10	83.50	8.36	30.41	46.35	74.76	**19.23**	45.68	**57.54**	**84.73**	18.37	46.15	57.42	84.53
en	45.79	69.90	63.79	84.64	**50.81**	70.47	**65.43**	**85.73**	45.12	69.88	64.11	83.55	46.94	**70.60**	65.20	84.91
gu	12.49	38.73	37.01	72.69	18.23	42.25	46.27	79.97	20.84	48.71	49.30	82.10	**22.69**	**50.31**	**51.36**	**84.36**
bn	**53.61**	**75.42**	**78.12**	**89.87**	20.57	46.58	56.60	78.62	40.56	67.75	71.36	86.65	40.38	61.71	68.71	84.17
kn	8.71	31.02	41.16	75.62	7.93	27.58	44.47	78.97	7.75	30.82	41.44	75.96	**10.66**	**32.58**	**46.92**	**80.45**
ml	24.28	55.37	55.49	85.35	18.60	47.39	51.47	82.86	26.16	56.49	**57.22**	**87.36**	**26.22**	**56.71**	57.01	86.53
pa	21.92	51.10	47.82	80.64	26.24	53.18	51.57	83.19	24.42	51.64	49.28	80.95	**26.96**	**54.82**	**52.33**	**84.11**
or	45.53	62.91	65.30	82.09	9.37	29.40	37.80	75.41	43.43	64.12	65.20	83.67	**47.17**	**67.82**	**71.20**	**86.05**
as	9.76	26.48	29.80	56.93	7.15	25.25	32.19	62.38	10.89	30.27	35.00	64.05	**12.61**	**32.93**	**36.91**	**65.84**
Avg	25.88	50.91	52.88	79.70	18.91	42.83	49.10	79.30	26.53	52.24	55.32	81.90	**28.13**	**53.54**	**57.27**	**83.33**

Table 4.4: Detailed scores on the test set using bi-lingual, multi-lingual, and translation-based variants of the mT5 model. The best results for a (metric and language) combination are highlighted.

Translate-output and translate-input settings lead to slightly improved models for English and Tamil, respectively. On average, the translate-output setting performs the worst, while the multi-lingual setting performs the best.

4.3.2 Task-specific pretraining

Pretraining has been a standard method to obtain very effective models even with small amounts of labeled data across several tasks in natural language processing (NLP). Domain and task specific pretraining has been shown to provide further gains [39]. We experiment with the following four pretraining strategies on top of the already pretrained mT5 model before finetuning it on XAlignV2 dataset.

1. Translation-only pretraining: Wang et al. [80] provide a noisy, but larger corpus (542192 data pairs across 15 categories) crawled from Wikipedia for English F2T task. The dataset is obtained by coupling noisy English Wikipedia data with Wikidata triples.

No.	Method	BLEU	METEOR	chrF++
1	Multi-lingual mT5 (No pretraining, no fact-aware embeddings)	28.13	53.54	57.27
2	Multi-stage Pretraining	27.70	51.87	55.32
3	Multi-task Pretraining	28.45	51.87	55.20
4	Translation-only Pretraining	27.53	50.67	53.71
5	Multi-lingual Pretraining	28.71	**53.83**	**57.58**
6	Fact-aware embeddings	**29.27**	53.64	57.30

Table 4.5: XF2T scores on XAlignV2 test set using different pretraining strategies and fact-aware embeddings for the mT5 model. Best results are highlighted.

2. Multi-lingual pretraining: In this method, we translate English sentences from the Wikipedia-based [80]'s data to our LR languages. Thus, the multi-lingual pretraining data contains ∼6.5M data pairs. For translating sentences, we use IndicTrans [70].

3. Multi-stage pretraining: Translation is a preliminary task for effective cross-lingual NLP. Thus, in this method, in the first stage, we pretrain mT5 on translation data corresponding to English to other language pairs with ∼0.25M data instances per language. In the second stage, we perform multi-lingual pretraining as described above.

4. Multi-task pretraining: This method also involves training for both translation as well as XF2T tasks. Unlike the multi-stage method where pretraining is first done for translation and then for XF2T (multi-lingual pretraining), in this method we perform the two tasks jointly in a multi-task learning setup.

Table 4.5 (lines 1 to 5) shows results using different pretraining strategies. Translation-only pretraining is the model obtained using pretraining for translation task only. We observe that multi-lingual pretraining leads to improvements compared to no XF2T specific pretraining across 3 of the 4 metrics. Multi-stage pretraining is slightly better than translation-only pretraining but not as good as multi-lingual pretraining. Finally, multi-task performs better than multi-stage. For English and Bengali, we found that multi-stage pretraining provided best results. However, multi-lingual pretraining is the best on average across languages, with biggest wins for Malayalam and Oriya.

4.3.3 Fusing the fact-aware embeddings

The input to mT5 consists of token embeddings as well as position embeddings. For XF2T, the input is a bunch of facts. Facts contain semantically separate units each of which play a different role: subject, relation, object. We extend the standard mT5 input with specific (fact-aware) role embeddings. Specifically, we use four role IDs: 1 for subject, 2 for relation

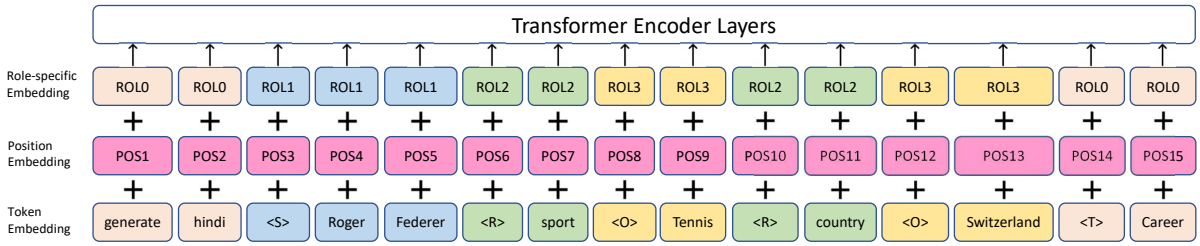

Figure 4.1: English facts being passed as input to mT5's encoder with token, position and (fact-aware) role embeddings.

	Vanilla mT5			Multi-lingual Pretraining			Fact-aware embeddings		
	BLEU	METEOR	chrF++	BLEU	METEOR	chrF++	BLEU	METEOR	chrF++
hi	44.65	68.58	68.49	43.32	68.19	68.21	42.72	67.49	68.03
mr	26.47	56.85	59.17	27.64	56.34	57.74	29.06	55.40	57.97
te	14.46	43.45	52.58	15.94	42.71	52.40	16.21	42.14	51.25
ta	18.37	46.15	57.42	16.68	42.32	54.88	19.07	43.65	56.01
en	46.94	70.60	65.20	46.61	70.45	65.33	48.29	70.75	65.42
gu	22.69	50.31	51.36	21.39	47.98	50.14	23.27	50.00	50.64
bn	40.38	61.71	68.71	50.89	75.62	77.43	49.48	73.03	76.19
kn	10.66	32.58	46.92	11.61	33.00	47.18	11.57	33.44	46.66
ml	26.22	56.71	57.01	27.38	56.63	57.35	29.04	57.15	57.60
pa	26.96	54.82	52.33	26.04	54.17	52.50	28.65	55.19	53.38
or	47.17	67.82	71.20	44.97	66.49	70.64	41.75	63.77	67.96
as	12.61	32.93	36.91	12.00	32.04	37.15	12.16	31.61	36.44
Avg	28.13	53.54	57.27	28.71	53.83	57.58	29.27	53.64	57.30

Table 4.6: XF2T scores on XAlignV2 test set using vanilla mT5, multi-lingual pretrained mT5 and mT5 with fact-aware embedding models.

and qualifier relation, 3 for object and qualifier tokens, and 0 for everything else, as shown in Fig. 4.1. We hope that this explicit indication of the role played by each token in the input facts, will help the model for improved XF2T generation.

We also experimented with (1) separate role embeddings for qualifier relation and qualifier, and (2) adding fact id embeddings, i.e., if the input contains K facts, we have K fact IDs, and all tokens corresponding to a fact gets the same fact ID embedding. However, these did not lead to better results.

Table 4.5 (line 6) shows that fact-aware embeddings lead to improvements over the vanilla mT5 method without fact-aware embeddings (line 1).

In summary, we note that both the proposed methods (multi-lingual pretraining, fact-aware embedding) lead to improvements over the vanilla mT5. We also experimented with combi-

	mT5			Our best model		
	Fluency	Factual correctness	Extra info	Fluency	Factual correctness	Extra info
hi	4.89	4.75	4.37	4.95	4.79	4.62
te	4.65	4.18	4.14	4.30	3.85	3.80
mr	4.70	4.35	4.44	4.75	4.53	4.32
en	4.69	4.17	4.29	4.90	4.68	4.05

Table 4.7: Human Evaluation Results for mT5 and our best model, on selected languages.

nations of these approaches but did not observe better results. Amongst these, multi-lingual pretraining performs the best on two of the metrics (METEOR and chrF++), while fact-aware embeddings perform best on BLEU. Hence, we present a language-wise detailed comparison across these three models in Table 4.6. We observe that all the models perform well on bn, hi, en, and or. On the other hand, performance is poor for te, ta, know, and as.

4.4 Results and analysis

Finally, we obtain human annotations to evaluate the perceived quality of the generated text. Table 4.7 shows comparative results for the mT5 model and our fact-aware embedding model across three metrics: fluency, factual correctness, and the presence of extra information in the generated output. Higher, the better. The evaluation has been done on 100 samples for 3 languages (en, mr, hi) on a 5-point Likert scale per metric. Fluency checks for coherence and grammar correctness of the generated output. Factual correctness verifies if most facts are captured in the sentence correctly. The absence of extra information verifies if the model does not generate any hallucinated information. Fluency, factual correctness, and extra info are 4.76, 4.42, 4.37 for mT5 and 4.87, 4.67, 4.33 for our best model, respectively.

Table 5.9 shows XF2T prediction examples for our fact-aware embedding model. In general, across examples, we observe that the generated text is fluent and correct. Most of the input facts are covered by the generated sentence. Sometimes, though, the model hallucinates and brings in extra information in the output, e.g., for English, “Uttar Pradesh” is not mentioned as part of input facts.

So far, we have presented results using small-scale models. We also train a large-scale checkpoint with 12 encoder and 12 decoder layers for the fact-aware embedding model. We observe that it leads to a BLEU of 30.90, METEOR of 55.12, and chrF++ of 59.17, which is significantly better than the small model, as expected.

Lang.	Input	Reference Text	Generated Text
hi	generate Hindi <S> Asha Nautiyal <R> member of political party <O> Bharatiya Janata Party <R> date of birth <O> 25 June 1969 <R> occupation <O> politician <R> country of citizenship <O> India <T> introduction	आशा नौटियाल (जन्म २५ जून, १९६९) एक भारतीय राजनीतिज्ञ हैं जो भारतीय जनता पार्टी से हैं ।	आशा नौटियाल (जन्म २५ जून, १९६९) एक भारतीय राजनीतिज्ञ और भारतीय जनता पार्टी की सदस्य हैं ।
en	generate English <S> Kedarnath Singh <R> date of death <O> 19 March 2018 <R> date of birth <O> 07 July 1934 <R> occupation <O> poet <R> languages spoken, written or signed <O> Hindi <R> country of citizenship <O> India <T> introduction	Kedarnath Singh (7 July 1934 - 19 March 2018) was an Indian poet who wrote in Hindi.	Kedarnath Singh (7 July 1934 - 19 March 2018) was a Hindi poet from Uttar Pradesh, India.
mr	generate Marathi <S> Théodore de Banville <R> date of death <O> 13 March 1891 <R> date of birth <O> 14 March 1823 <R> occupation <O> writer <R> country of citizenship <O> France <T> introduction	थेओदोर दि बॅनव्हिल (मार्च १४, इ. स. १८२३ – मार्च १३, इ. स. १८९१) हा फ्रेंच साहित्यिक होता.	थॉडेर द बॅनव्हिल (मार्च १४, इ. स. १८२३ – मार्च १३, इ. स. १८९१) हा फ्रेंच लेखक होता.
te	generate Telugu <S> Sushmita Sen <R> date of birth <O> 19 November 1975 <R> place of birth <O> Hyderabad <T> introduction	ఈమె 1975 నవంబరు 19 న హైదరాబాదులో జన్మించింది.	సుష్మితా సేన్ 1975 నవంబరు 19 న హైదరాబాదులో జన్మించింది.
ta	generate Tamil <S> Kirti Kumari <R> member of political party <O> Bharatiya Janata Party <R> date of birth <O> 13 August 1967 <R> date of death <O> 28 August 2017 <R> occupation <O> politician <R> country of citizenship <O> India <T> introduction	கீர்த்தி குமாரி (13 ஆகத்து 1967 - 28 ஆகத்து 2017) பாரதீய ஜனதா கட்சியின் இந்திய அரசியல்வாதி ஆவார்.	கீர்த்தி குமாரி (13 ஆகத்து 1967 - 28 ஆகத்து 2017) ஓர் இந்திய அரசியல்வாதியும், பாரதிய ஜனதா கட்சியின் முன்னாள் சட்டமன்ற உறுப்பினரும் ஆவார்.
kn	generate Kannada <S> Barry C. Barish <R> award received <O> Henry Draper Medal <R> point in time <O> 2017 <T> awards and honors	ಮತ್ತು ಬ್ಯಾರಿಷ್ ಅವರಿಗೆ ೨೦೧೭ ರ ಹೆನ್ರಿ ಡ್ರೇಪರ್ ಪದಕವನ್ನು ನೀಡಲಾಯಿತು.	೨೦೧೭ ರಲ್ಲಿ ಅವರು ಹೆನ್ರಿ ಡ್ರೆಪರ್ ಪದಕವನ್ನು ಪಡೆದರು.
bn	generate Bengali <S> Jim Pothecary <R> member of sports team <O> South Africa national cricket team <R> occupation <O> cricketer <T> introduction	দক্ষিণ আফ্রিকা ক্রিকেট দলের অন্যতম সদস্য ছিলেন তিনি ।	দক্ষিণ আফ্রিকা ক্রিকেট দলের অন্যতম সদস্য ছিলেন তিনি ।
gu	generate Gujarati <S> Krishnalal Shridharani <R> date of birth <O> 16 September 1911 <R> date of death <O> 23 July 1960 <R> occupation <O> poet <R> occupation <O> playwright <R> languages spoken, written or signed <O> Gujarati <T> introduction	કૃષ્ણલાલ શ્રીધરાણી (૧૬ સપ્ટેમ્બર ૧૯૧૧ – ૨૩ જુલાઇ ૧૯૬૦) ગુજરાતી ભાષાના કવિ અને નાટ્યકાર હતા.	કૃષ્ણલાલ શ્રીધરાણી (૧૬ સપ્ટેમ્બર ૧૯૧૧ – ૨૩ જુલાઇ ૧૯૬૦) ગુજરાતી કવિ, નાટ્યકાર અને નાટ્યકાર હતા.
pa	generate Punjabi <S> Orhan Pamuk <R>award received <O> Nobel Prize in Literature <R> point in time <O> 2006 <R> date of birth <O> 07 June 1952 <R> occupation <O> novelist <R> languages spoken, written or signed <O> Turkish <T> introduction	ਓਰਹਾਨ ਪਾਮੁੱਕ (ਜਨਮ 7 ਜੂਨ 1952) ਇੱਕ ਤੁਰਕੀ ਨਾਵਲਕਾਰ ਹੈ ਜਿਸ ਨੇ 2006 ਵਿੱਚ ਸਾਹਿਤ ਲਈ ਨੋਬਲ ਇਨਾਮ ਹਾਸਿਲ ਕੀਤਾ.	ਓਰਹਾਨ ਪਾਮੁੱਕ (ਜਨਮ 7 ਜੂਨ 1952) ਇੱਕ ਤੁਰਕੀ ਨਾਵਲਕਾਰ ਹੈ ਜਿਸ ਨੂੰ 2006 ਵਿੱਚ ਸਾਹਿਤ ਲਈ ਨੋਬਲ ਪੁਰਸਕਾਰ ਨਾਲ ਸਨਮਾਨਿਤ ਕੀਤਾ ਗਿਆ .
ml	generate Malayalam <S> Naomi Scott <R> date of birth <O> 06 May 1993 <R> place of birth <O> London <R> country of citizenship <O> United Kingdom <T> introduction	1993 മെയ് 6 ന് ഇംഗ്ലണ്ടിലെ ലണ്ടനിലാണ് സ്കോട്ട് ജനിച്ചത്	1993 മെയ് 6 ന് ഇംഗ്ലണ്ടിലെ ലണ്ടനിലാണ് സ്കോട്ട് ജനിച്ചത്
or	generate Odia <S> Ajay Swain <R> award received <O> Odisha Sahitya Akademi Award <R> point in time <O> 2012 <T> introduction	ସେ ୨୦୧୨ ମସିହାରେ ଓଡ଼ିଶା ସାହିତ୍ୟ ଏକାଡେମୀ ପୁରସ୍କାର ଲାଭ କରିଥିଲେ ।	୨୦୧୨ ମସିହାରେ ସେ ଓଡ଼ିଶା ସାହିତ୍ୟ ଏକାଡେମୀ ପୁରସ୍କାର ଲାଭ କରିଥିଲେ ।
as	generate Assamese <S> Harishankar Parsai <R> date of death <O> 10 August 1995 <R> date of birth <O> 22 August 1922 <R> occupation <O> writer <R> country of citizenship <O> British India <R> country of citizenship <O> Dominion of India <R> occupation <O> author <T> introduction	হৰিশংকৰ পৰসাঈ (২২ আগষ্ট, ১৯২৪ - ১০ আগষ্ট, ১৯৯৫) আছিল হিন্দী সাহিত্যৰ এগৰাকী প্রসিদ্ধ লেখক আৰু ব্যংগকাৰ ।	হৰিশংকৰ পৰসাঈ (২২ আগষ্ট, ১৯২২ - ১০ আগষ্ট, ১৯৯৫) এজন ভাৰতীয় লেখক ।

Table 4.8: Test examples with reference text and predictions from our fact-aware embedding model.

4.5 Summary

In this chapter, we worked on the XF2T problem of generating a sentence capturing the input fact semantics in the specified language. We investigated several multi-lingual Transformer methods with different training setups, pretraining setups, and input representations. We discussed how to linearize the input before feeding it to the models and the role-specific embedding layer that helps differentiate the subject, predicate, and object in the triple. We compared translation baselines, monolingual, and bilingual models and showed the efficacy of a single multilingual model based on our human-annotated test data results. We obtained models with the best metrics of 30.90 BLEU, 55.12 METEOR, and 59.17 chrF++ for XF2T.

XAlign deals only with the data instances having a single sentence as the target output. All our methods above are tested on the single sentence generation task. In reality, a Wikipedia article or any other document has several sentences in it in a coherent manner. Thus, outputting longer text is desirable from any text generation system. The approaches implemented above can also be scaled for longer form text generation, and the performance needs to be tested.

In the upcoming chapter, we explore one more way to generate long-form text for Wikipedia article generation: Summarization. We perform cross-lingual multi-document summarization in order to generate Wikipedia article sections in 8 LR languages. We feed as the input a set of reference URLs, a target section title, and a target output language. The expected output is then the text suitable for that Wikipedia section in the target language. This is an extremely challenging task because it involves long text generation, and that too in a cross-lingual manner. Handling long text input is difficult. We discuss the strategies we adopted to tackle these challenges in detail in the upcoming chapter.

Chapter 5

Cross-lingual, multi-document, aspect-based summarization

5.1 Overview

As discussed in previous chapter, although we are able to successfully generate a single sentence output fluently, generating the long form text in LR languages still remains a challenge. We explore the same by adopting a novel idea of cross-lingual, multi-document, aspect-based summarization in this chapter, which we refer to as XWikiGen. As shown in Fig. 1.4, the input for XWikiGen is a set of reference URLs, a target section title, and a target output language. The expected output is then the text suitable for that Wikipedia section in the target language. Analogous to generic summarization versus query-based summarization, XWikiGen involves section-wise text generation rather than the generation of the entire Wikipedia page. Unlike existing work on monolingual (English-only) Wikipedia text generation, XWikiGen is cross-lingual in nature. Lastly, unlike some existing work that generates cross-lingual text using English Wikipedia pages, XWikiGen focuses on generating cross-lingual text using reference URLs in multiple languages.

Our first contribution is a novel dataset, XWikiRef towards the XWikiGen task. The dataset is obtained from Wikipedia pages corresponding to eight languages and five domains. Languages include Bengali (bn), English (en), Hindi (hi), Malayalam (ml), Marathi (mr), Oriya (or), Punjabi (pa) and Tamil (ta). Domains include books, films, politicians, sportsmen, and writers. The dataset spans ∼69K Wikipedia articles with ∼105K sections. Each section has 5.44 cited references on average.

XWikiGen is an extremely challenging task because it involves long text generation, and that too in a cross-lingual manner. Handling long text input is difficult. Hence, we follow a two-stage approach. The first extractive stage identifies important sentences across several reference documents. The second abstractive stage generates the section text. Both stages involve neural models. We experiment with unsupervised methods like salience [87] and hiporank [28] for the extractive stage, and mT5 [84] and mBART [59] for the abstractive stage. We experiment with several training setups like (1) multi-lingual, (2) multi-domain, and (3) multi-lingual-multi-

domain. We report results using standard text generation metrics like ROUGE-L, METEOR, and chrF++.

5.2 Leveraging Wikipedia to build a parallel corpus

In this section, we first discuss the procedure for XWikiRef data collection and pre-processing. Then we present a detailed analysis.

5.2.1 Data collection and pre-processing

XWikiRef contains Wikipedia sections related to five distinct domains (books, films, politicians, sportsmen, writers) spanning across eight languages (bn, en, hi, ml, mr, or, pa, ta). We start by using Wikidata API[1] to filter the domains of interest initially and further fetch the entities that have Wikipedia pages in our set of languages. Later, we use Wikipedia language-specific 20220926 XML dumps to extract the Wikipedia pages of filtered entities. Sections and subsections follow a standard structure in Wikipedia text. We extract sections and subsections from the text. Text in containers with a depth greater than two is merged into parent sub-sections.

We also extract the citation URLs in each section using wiki markup. We use the `MediaWikiParserFromHell`[2] module in Python to clean all the wiki markup in a particular section and gather clean section text. We filter the URLs to remove file formats other than HTML and pdf. For each reference URL, we use `BeautifulSoup`[3] in Python to scrape the <p> paragraph text from the corresponding webpages, and `pdfminer`[4] to extract the text from pdf. Finally, we tokenize the scraped text into individual sentences using a universal sentence tokenizer in IndicNLP [45]. We retain only those sections as part of the dataset with at least one (crawlable) reference URL with non-empty text.

Overall, each sample in the dataset consists of the domain, language, section title, set of reference URLs, and Wikipedia section text. This dataset is then split into train, validation, and test in the 60:20:20 ratio, stratified by domain and language. We make these standard splits publicly available as part of the dataset.

5.2.2 Data analysis

We analyze our prepared dataset across several parameters, the details of which are in the following tables. Table 5.1 shows the total number of articles per domain per language in the

[1] `https://query.wikidata.org/`

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

[3] https://pypi.org/project/beautifulsoup4/

[4] https://pypi.org/project/pdfminer/

Domain/Lang	bn	hi	ml	mr	or	pa	ta	en	Total
Books	313	922	458	87	73	221	493	1467	4034
Film	1501	1025	2919	480	794	421	3733	1810	12683
Politicians	2006	3927	2513	988	1060	1123	4932	1628	18177
Sportsmen	5470	6334	1783	2280	319	1975	2552	919	21632
Writers	1603	2024	2251	784	498	2245	1940	714	12059
Total	10893	14232	9924	4619	2744	5985	13650	6538	**68585**

Table 5.1: XWikiRef: Total number of articles per domain per language

Domain/Lang	bn	hi	ml	mr	or	pa	ta	en	Total
Books	434	987	557	111	88	238	598	2972	5985
Film	2139	1363	3737	676	1351	476	4781	4766	19289
Politicians	3261	4478	3719	1384	1404	1524	6431	4780	26981
Sportsmen	9485	8118	2642	3056	485	2624	3769	2698	32877
Writers	2598	2743	3435	1166	896	3034	3113	2409	19394
Total	17917	17689	14090	6393	4224	7896	18692	17625	**104526**

Table 5.2: XWikiRef: Total number of sections per domain per language

XWikiRef dataset. By the nature of spread of Wikipedia articles across domains, the number of articles differ across domains per language. Overall, there are ∼69K articles from which we extract sections for the dataset.

Next, Table 5.2 shows the distribution of number of sections across various (domain, language) pairs in the XWikiRef dataset. Further, as mentioned earlier, XWikiRef is a multi-document summarization dataset. Table 5.3 shows the average number of references per section for each (domain, language) pair. As can be seen from the table, the dataset contains at least two references on average for every (domain, language) pair, although a large percent of these references are not in the LR language.

Fig. 5.1 shows the distribution of the number of reference URLs across domains in the dataset. The figure shows that there are several samples where the number of reference URLs is 5+ across all domains showing that multi-document summarization is essential.

Finally, we show word clouds of the most frequent Wikipedia section titles for each of the five domains in Fig. 5.2. Each word cloud contains the five most frequent titles per language. Section titles for one language are shown using a single color. Font size indicates relative frequency. The word clouds show the variety of section titles per (language, domain) pair.

Domain/Lang	bn	hi	ml	mr	or	pa	ta	en
Books	3.62	2.61	2.59	2.07	3.46	2.30	2.40	6.34
Film	4.85	7.14	3.34	2.96	3.81	4.10	3.83	12.74
Politicians	4.98	4.09	3.75	3.87	2.07	3.59	3.91	14.21
Sportsmen	6.37	8.30	6.96	4.20	3.93	4.49	6.38	21.88
Writers	5.20	5.46	4.16	3.74	2.85	3.34	4.20	17.61

Table 5.3: XWikiRef: Average number of references per section for each domain and language

5.3 Two stage approach

In this section, we first motivate the need for proposing a two-stage approach for the cross-lingual multi-document summarization task, XWikiGen. Next, we discuss the details of the two stages: extractive and abstractive. Finally, we present multiple training setups.

Table 5.4 shows the average number of sentences in references of a section for each domain and language in our dataset. Combined with the number of references per section as shown in Table 5.3, the overall text input is very large. Given the quadratic complexity of Transformer-based methods, it is infeasible to feed such long inputs to an encoder-decoder model and expect it to be able to output reasonable summaries. Transformers sub-quadratic complexity is an active area of research with models like Longformer [8], Reformer [48], etc. But we plan to explore them as part of future work.

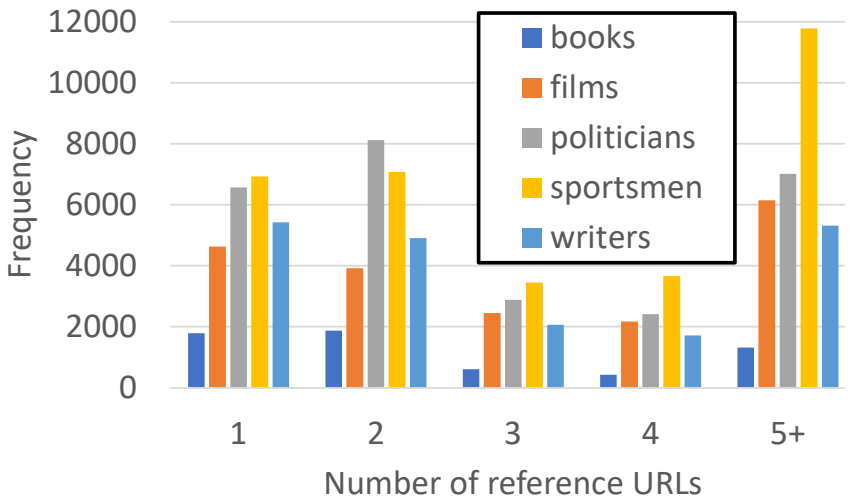

Figure 5.1: Distribution of the number of reference URLs across domains in our XWikiRef dataset

Figure 5.2: Word clouds of most frequent Wikipedia section titles per domain. Each word cloud contains titles across all languages. Section titles for one language are shown using a single color. Font size indicates relative frequency.

Domain	bn	hi	ml	mr	or	pa	ta	en
Books	200.2	117.9	1232.0	225.8	51.9	246.7	302.7	940.8
Films	223.9	320.6	91.9	105.6	345.9	172.6	192.5	1253.6
Politicians	1318.3	467.1	513.3	394.0	54.5	255.4	614.1	1540.9
Sportsmen	335.7	1166.3	406.9	167.5	724.0	253.5	714.0	1535.0
Writers	643.2	2032.5	800.1	385.5	118.5	351.0	1279.0	2061.3

Table 5.4: Average number of sentences in references of a section for each domain and language in XWikiRef.

In order to address the long input problem, we propose a two-stage system where the first stage identifies promising candidate sentences across all the reference citations for a sample. The highest-scoring candidate sentences are passed as input to the second stage, which generates an abstractive summary. In the following, we will discuss the two stages in detail.

5.3.1 Stage 1: Extractive summarization

Given a set of reference URLs, the extractive stage aims to select a subset of sentences from these URLs that best represent a summary of the set of URLs. While earlier methods for extractive summarization were position-based or lexical chains-based, neural methods have become popular in the past decade. We experiment with two different extractive summarization-based techniques: Salience and HipoRank. For both methods, the input consists of the section title and a sentence from the set of reference URLs. The output is a summary worthiness score.

5.3.1.1 Salience-based extractive summarization

The main idea of salience-based extractive summarization is to find the top-K salient sentences from the input references based on the relevance of that sentence relative to a particular section title. Our salience method is inspired by the relevance scoring method in [87], where a language model was used to calculate the relevance score of each answer entity relative to the QA (question-answer) context. We first split the reference text into sentences to extract the top-K sentences. Each sentence is then prepended with a section title and passed as input to a pretrained XLM-RoBERTa[22] language model. We score each sentence based on the likelihood from the language model. Top-K sentences with the highest relevance scores are passed on as output to the next stage.

5.3.1.2 HipoRank-based extractive summarization

Hierarchical and Positional Ranking model (HipoRank) [28] is an unsupervised graph-based model for the extractive summarization of long documents. A document with multiple sections creates a directed hierarchical graph with sentence and section nodes and sentence-sentence and sentence-section edges with asymmetrically weighted edges. The score for a sentence node is then computed based on a weighted sum of edges incident on the node.

We compute sentence node representations using mBERT [25]. We take the mean of all the sentence representations within a section to compute the representation for every section node.

Each sentence node is connected to other nodes via intra-sectional and inter-sectional edges. Intra-sectional connections are between all sentences of the same section, meant to model the local importance of the sentence. The key idea is that sentences similar to most sentences within a section are more critical. On the other hand, inter-sectional connections are between sentences and section nodes, meant to model the global importance of the sentences. Here, the

idea is that sentences most similar to other sections are the most important. For efficiency, edges are not allowed between two sentences in different sections.

Cosine similarity between node embeddings is used to compute edge weights. Based on the hypothesis that essential sentences are near the boundaries (start or end) of a text, intra-sectional edges have higher weight if they are incident on a boundary sentence. Similarly, essential sections are near the boundaries of the document. This hypothesis is used to weigh inter-sectional edges appropriately. Finally, the importance score for a sentence node is computed based on a weighted sum of edges (both intra-sectional and inter-sectional) incident on the node. We then sort these sentences in descending order based on the importance score and greedily select the top-K sentences as our extractive summary.

5.3.2 Stage 2: Abstractive summarization

Note that the output from the extractive stage is in the reference text language itself. Also, since these sentences have been obtained across several documents, they often form an incoherent extractive summary. We need an abstractive stage to generate coherent summaries in the target language. For the abstractive stage, we use two state-of-the-art multi-lingual natural language generation models viz. mBART-large[59] and mT5-base[84]. mT5 and mBART are both multi-lingual encoder-decoder Transformer models and have been shown to be very effective across multiple such NLP tasks like question answering, natural language inference, named entity recognition, etc. Both these models contain 24 layers (12 layers encoder + 12 layers decoder). For both models, we pass the target language id, article title, section title, and top-k sentences from the extractive stage (descending sorted based on score) as input.

mT5 [84] was pretrained on mC4 dataset[5] comprising of web data in 101 different languages and leverages a unified text-to-text format. mBART [59] was pretrained on the CommonCrawl corpus using the BART objective, where the input texts are noised by masking phrases and permuting sentences. A single Transformer model is learned to recover the texts. Specifically, our mT5-base model is an encoder-decoder model with 12 layers each for the encoder and decoder. It has 12 heads per layer, a feed-forward size of 2048, keys and values are 64 dimensional, d_{model}=768, and a vocabulary size of 250112. Overall the model has 582.40M parameters. Our mBART-large-50 model [59] also has 12 layers each for the encoder and decoder. It has 16 heads per layer, a feed-forward size of 4096, d_{model}=1024, and a vocabulary size of 250054. Overall the model has 610.87M parameters. Note that the two models have almost the same size.

Using the training part of our XWikiRef dataset, we fine-tune both these models on the extractive stage output.

[5] `https://www.tensorflow.org/datasets/catalog/c4#c4multi-lingual_nights_stay`

5.4 Multi-lingual, multi-domain, and multi-lingual-multi-domain setups and training configuration

XWikiRef contains data for eight languages and five domains. We could perform training in various ways. We could train one model per (language and domain) pair. Given five domains and eight languages, we must train, maintain and deploy 40 models. Also, the amount of training data per (language, and domain) pair is not very large. Thus, such individual models may not be able to benefit from cross-language or cross-domain knowledge.

Another way of training models is multi-lingual. This means that we train one model per domain using training data across all languages. Thus, there will be five models. A third way is to train models in a multi-domain manner. Thus, we will have one model per language using training data across all domains, leading to eight models.

One last approach is to train a multi-lingual-multi-domain model. We collate training data across all languages and domains and train a single model. This model can exploit cross-language and cross-domain clues and learn robust representations.

Previous literature in multi-lingual cross-lingual natural language generation has shown that multi-lingual models are better than individual ones, especially for low-resource languages. Since this work is focused on LR languages, we experiment with multi-lingual, multi-domain, and multi-lingual-multi-domain setups.

The two stages in our approach have different computing requirements. We performed extractive steps on a machine with one NVIDIA 2080Ti with 12 GPU RAM. For the abstractive stage, we fine-tuned the model on a machine having NVIDIA V100 having 32GB of GPU RAM with CUDA 11.0 and PyTorch 1.7.1.

For the salience-based extractive stage, we used XLM-RoBERTa-base[22] model for extracting the sentence representation with 512 as the maximum input length. For HipoRank, we used the multi-lingual BERT (mBERT [25]) model to get the sentence representation for building the graph with 512 as the maximum input length. We took a maximum of 50 sentences per sample as output from the extractive stage.

For the abstractive stage, we fine-tuned mBART[59] and mT5[84] models for 20 epochs keeping a batch size of 4. We initialize using google/mt5-base and facebook/mbart-large-50 huggingface checkpoints. We kept the maximum input and output length as 512 across all of our experiments. We used AdamW optimizer with a learning rate of 1e-5. We perform greedy decoding.

5.5 Metrics, results, and analysis

We evaluate our models using standard Natural Language Generation (NLG) metrics like ROUGE-L [57], METEOR [7] and chrF++ [66]. Another popular NLG metric is PARENT.

	Extractive	Abstractive	ROUGE-L	chrF++	METEOR
Multi-lingual	Salience	mBART	15.59	17.20	10.98
	Salience	mT5	14.66	15.45	8.92
	HipoRank	mBART	**16.96**	**19.11**	**12.19**
	HipoRank	mT5	15.98	17.11	10.08
Multi-domain	Salience	mBART	**19.88**	**22.82**	**15.00**
	Salience	mT5	12.13	13.66	7.27
	HipoRank	mBART	18.87	20.79	14.10
	HipoRank	mT5	12.29	13.93	7.36
Multi-lingual-multi-domain	Salience	mBART	20.50	22.32	14.81
	Salience	mT5	17.31	18.77	11.57
	HipoRank	mBART	**21.04**	**23.44**	**15.35**
	HipoRank	mT5	17.65	19.04	11.74

Table 5.5: XWikiGen Results across multiple training setups and (extractive, abstractive) methods on test part of XWikiRef. Best results per block are highlighted in bold. Overall best results are also underlined.

But PARENT [27] relies on the word overlap between input and the prediction text. Since the input and prediction in XWikiGen are in different languages, we cannot compute PARENT scores.

1. **ROUGE-L**: ROUGE-L, or Recall-Oriented Understudy LCS is based on statistics using the longest common subsequence (LCS). The longest common subsequence task automatically determines the longest co-occurring n-grams given a reference sequence and a machine-generated sequence, while taking sentence-level structure similarities into account.

2. **chrf++**: In addition to adding word n-grams, chrF++ is an evaluation measure that uses the F-score statistic for character n-gram matches.

3. **METEOR**: METEOR, an automated metric evaluation, is based on a generalized idea of unigram matching between the text generated by the machine and the reference text created by a human. Based on their meanings, surface forms, and stemmed forms, unigrams can be matched.

Table 5.5 shows results across two extractive methods (salience, HipoRank), two abstractive methods (mBART, mT5), three training setups (multi-lingual, multi-domain, multi-lingual-

	bn	en	hi	mr	ml	or	pa	ta
ROUGE-L	14.49	7.46	29.01	20.67	12.25	25.54	16.89	17.09
chrF++	18.58	10.55	28.38	20.41	15.30	27.31	13.49	21.90
METEOR	9.71	5.90	25.24	13.72	6.42	22.69	10.12	9.87

Multi-lingual HipoRank+mBART

	bn	en	hi	mr	ml	or	pa	ta
ROUGE-L	15.30	12.07	36.16	31.25	14.22	29.53	16.91	15.00
chrF++	19.40	17.41	34.34	32.50	18.34	32.20	14.10	21.65
METEOR	10.34	9.59	31.02	24.86	8.89	26.86	10.01	9.29

Multi-domain Salience+mBART

	bn	en	hi	mr	ml	or	pa	ta
ROUGE-L	15.21	16.32	36.38	22.71	15.50	27.41	18.64	18.87
chrF++	19.50	21.34	34.55	21.93	18.65	28.83	16.27	23.99
METEOR	10.24	12.74	31.24	14.88	8.84	23.93	11.6	11.26

Multi-lingual-multi-domain HipoRank+mBART

Table 5.6: Detailed per-language results on test part of XWikiRef, for the best model per training setup.

multi-domain), and three metrics (ROUGE-L, METEOR, and chrF++) computed as a micro-average across all test instances in XWikiRef.

The table shows that the best results are obtained using the multi-lingual-multi-domain training setup. Also, in this setup, the combination of HipoRank with mBART provides the best overall results. These results are statistically significantly better compared to other rows in the table. The supremacy of the multi-lingual-multi-domain training setup is expected given that it combines learning across all languages and domains in the dataset. Also, HipoRank was expected to perform better since it combines the knowledge of the pretrained (mBERT) model with the hierarchical document structure. Even for the multi-lingual setup, best results are obtained using the HipoRank+mBART combination. However, for the multi-domain setup, we observe that Salience+mBART performs better.

Further, we wish to drill deeper into the performance of the best models for each of the training setups. Hence, for these three models, we show micro-averaged metrics per language and per domain for the test set in Tables 5.6 and 5.7, respectively. We make the following observations from Table 5.6: (1) Multi-domain training is much better than multi-lingual training except for Tamil (ta). (2) Interesting, relatively richer languages like en and hi seem to benefit most when we move from multi-lingual to multi-lingual-multi-domain setup. (3) When comparing multi-domain training with multi-lingual-multi-domain, we observe gains across most languages ex-

cept for losses in mr and or. From Table 5.7, we observe that across all domains, results improve as we move from multi-lingual training to multi-domain training to multi-lingual-multi-domain setup (except for minor drop for sportsmen in the multi-lingual-multi-domain case).

Finally, we present the most detailed per (domain, language) level results for our best model in Table 5.8. We observe that the best results are obtained for the hi-books combination. Overall, the model works best for Hindi across all domains. The model also performs reasonably for mr and or. But more work must be done to improve the model for Bengali and Malayalam.

For a qualitative analysis of our best model outputs, we show some sample outputs in Table 5.9. In general, our model generates fluent text to a certain length. But, as the length of the output grows, we see the repeated patterns in the text, breaking the sentence structure. Pretrained language models usually present this problem of repeating n-grams, and increasing the training dataset size has been shown to alleviate it. Further, we observe the faithfulness of content between the generated text and reference text. Despite generating correct sentence structure, the model is seen to predict value strings incorrectly, like that of date of birth, names of persons, and related entities. This issue of hallucination is also common in pretrained language models, and finetuning on more training data should help.

	writers	books	sportsmen	politicians	films
ROUGE-L	10.12	3.65	20.61	22.01	14.60
chrF++	10.76	3.58	22.94	24.34	18.36
METEOR	5.77	1.93	14.66	17.61	10.04

Multi-lingual HipoRank+mBART

	writers	books	sportsmen	politicians	films
ROUGE-L	14.21	20.17	20.65	22.77	20.82
chrF++	17.24	21.86	22.75	26.14	24.30
METEOR	10.06	16.26	14.71	18.88	14.81

Multi-domain Salience+mBART

	writers	books	sportsmen	politicians	films
ROUGE-L	14.67	22.03	20.44	23.70	21.60
chrF++	16.65	22.81	21.57	25.75	24.51
METEOR	9.81	17.55	13.84	18.92	15.11

Multi-lingual-multi-domain HipoRank+mBART

Table 5.7: Detailed per-domain results on test part of XWikiRef, for the best model per training setup.

	ROUGE-L					chrf++					METEOR				
	writers	books	sports	politi	films	writers	books	sports	politi	films	writers	books	sports	politi	films
bn	10.61	9.43	15.78	17.46	15.75	14.72	14.19	20.28	21.21	20.03	6.13	5.66	10.56	12.99	10.39
en	13.04	15.62	18.53	13.32	20.15	19.71	18.90	22.80	20.00	24.13	10.65	11.62	13.89	11.47	15.09
hi	33.23	58.71	28.48	53.18	21.46	31.05	51.99	26.99	52.05	19.64	28.49	53.78	21.46	51.65	15.30
mr	15.37	17.00	26.77	20.06	24.15	14.68	16.24	26.84	18.12	21.82	7.40	9.50	20.14	10.74	14.30
ml	8.96	10.93	12.97	14.36	24.19	13.35	12.18	15.42	18.01	26.51	3.92	4.77	6.14	7.73	16.16
or	13.15	12.31	9.38	43.76	26.66	14.44	15.16	10.51	44.17	29.27	5.67	9.14	5.28	40.89	23.30
pa	14.96	12.35	24.54	16.59	17.15	13.42	12.39	21.32	14.02	13.82	8.59	7.48	16.54	9.80	9.63
ta	10.62	11.85	18.94	19.18	24.90	16.43	17.63	23.98	23.77	29.94	4.89	6.29	10.03	11.24	17.05

Table 5.8: Detailed results for every (domain, language) partition of the test set of our XWikiRef dataset, for our best XWikiGen model: Multi-lingual-multi-domain HipoRank+mBART. *sports* and *politi* indicate sportsmen and politicians respectively.

Domain	Reference URLs	Entity/Section	Lang	Reference Text	Generated Summary
Books	• https://www.nytimes.com/books/97/07/13/reviews/970713.13polklt.html	കോൾഡ് മൌണ്ടൻ/ആമുഖം	ml	കോൾഡ് മൌണ്ടൻ 1997 ൽ പ്രസിദ്ധീകരിക്കപ്പെട്ട ചാൾസ് ഫ്രേസിയർ എഴുതിയ ഒരു ചരിത്ര നോവലാണ്. ഈ നോവലിന് യു.എസ്. നാഷണൽ അവാർഡ് ഫോർ ഫിക്ഷൻ എന്ന പുരസ്കാരം ലഭിച്ചിട്ടുണ്ട്. അമേരിക്കൻ ആഭ്യന്തരയുദ്ധത്തിൻറെ അവസാനകാലത്ത് യുദ്ധത്തിൽ പങ്കെടുത്ത കോൾഡ്മൗണ്ടറ് സേനയിലെ ഡബ്ല്യു. പി. ഇൻമാൻ എന്ന മുറിവേറ്റ സൈനികനാണ് ഈ കഥയിലെ കേന്ദ്രകഥാപാത്രം. അദ്ദേഹം അഡ മൺറോയെന്ന തൻറെ പ്രണയിനിയുടെ അടുത്തു തിരിച്ചെത്തുവാനായി മാസങ്ങളോളം ഏകനായി നടത്തുന്ന യാത്രയാണ് കഥയുടെ ഇതിവൃത്തം. ഹോമറിൻറെ ഒഡിസിയുമായി ഈ നോവലിന് അനേകം സാദൃശ്യങ്ങളുണ്ട്.	ഒരു അമേരിക്കൻ നോവലിസ്റ്റാണ് കോൾഡ് മൌണ്ടൻ (ജീവികാലം: 1798–1831). അദ്ദേഹത്തിന്റെ ഏറ്റവും പ്രശസ്തമായ നോവലായ കോൾഡ് മൌണ്ടൻ, ന്യൂയോർക്ക് ടൈംസിന്റെ 1997 ലെ മികച്ച നോവലിനുള്ള അവാർഡ് നേടി.
Films	• http://www.thehindu.com/features/cinema/cinema-columns/blast-from-the-past-chandra-mohana-or-samooga-thondu-1936/article5668076.ece	சந்திர மோகனா /திரைப்படம்/அறிமுகம்	ta	சந்திரமோகனா என்பது 1936 ஆம் ஆண்டு வெளிவந்த தமிழ்த் திரைப்படமாகும். முதுரை மோகன் மூவிடோன் நிறுவனத்தினரின் தயாரிப்பில் வெளிவந்த இத்திரைப்படத்தில் எம். கே. ராதா, எஸ். வி. வெங்கட்ராமன் மற்றும் பலரும் நடித்துள்ளனர். இப்படமானது கொத்தமங்கலம் எஸ். எம். சுப்பிரமணியம் என்று அப்போது அழைக்கப்பட்ட கொத்தமங்கலம் சுப்பு எழுதிய சந்திரமோகனா அல்லது சமுதாயத் தோண்டு என்ற புதினத்தை அடிப்படையாகக் கொண்டு எடுக்கப்பட்டது.	சந்திர மோகனா (Chandra Mohana) 1947 ஆம் ஆண்டு வெளிவந்த தமிழ்த் திரைப்படமாகும். எஸ். எஸ். வாசன் இயக்கத்தில் வெளிவந்த இத்திரைப்படத்தில் எம். கே. ராதா, டி. ஆர். ராமச்சந்திரன் மற்றும் பலரும் நடித்திருந்தனர்.
Politicians	• https://timesofindia.indiatimes.com/city/patna/all-eyes-on-4-seats-in-khagaria-district/articleshow/78939785.cms • https://web.archive.org/web/20201204042405/https://www.jansatta.com/lifestyle/bihar-election-ljp-khagaria-candidate-renu-kumari-kushwaha-spent-24-and-half-lakhs-on-jewelry-and-cars-owns-this-much-property/1564879/ • http://164.100.47.194/loksabha/Members/memberbioprofile.aspx?mpsno=537&lastls=13	ਰੇਣੂ ਕੁਸ਼ਵਾਹਾ/ਜੀਵਣ-ਪਛਾਣ	pa	ਰੇਣੂ ਕੁਸ਼ਵਾਹਾ (ਜਿਸ ਨੂੰ ਰੇਣੂ ਕੁਮਾਰੀ ਸਿੰਘ ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ), ਇੱਕ ਭਾਰਤੀ ਸਿਆਸਤਦਾਨ, ਲੋਕ ਜਨਸ਼ਕਤੀ ਪਾਰਟੀ ਦੀ ਆਗੂ ਅਤੇ ਬਿਹਾਰ ਦੀ ਇੱਕ ਸਾਬਕਾ ਰਾਜ ਮੰਤਰੀ ਹੈ। ਉਹ ਖਗੜੀਆ ਵਿੱਚ ਸਥਤ ਪਰਟੀ ਅਤੇ ਜਨਤਾ ਦਲ (ਯੂਨਾਈਟਡ) ਤੋ ਹੈ ਉਹ ਭਾਰਤੀ ਜਨਤਾ ਪਾਰਟੀ ਤੱਕ ਵਤੀ ਸਿਆਸੀ ਪਰਤੀਆਂ ਨਾਲ ਜੁੜੀ ਰਹੀ ਹੈ। 2015 ਵਿੱਚ ਉਸਨੇ ਭਾਜਪਾ ਦੀ ਟਿਕਟ 'ਤੇ ਸਮਸਤੀਪੁਰ ਵਿਧਾਨ ਸਭਾ ਸੀਟ ਤੋ ਚੋਣ ਲੜੀ, ਜਿੱਥੇ ਉਹ ਆਰ.ਜੇ.ਡੀ. ਉਮੀਦਵਾਰ ਤੋ ਹਾਰ ਗਈ।	ਰੇਣੂ ਕੁਸ਼ਵਾਹਾ ਇੱਕ ਭਾਰਤੀ ਸਿਆਸਤਦਾਨ ਹੈ। ਉਹ ਭਾਰਤੀ ਜਨਤਾ ਪਾਰਟੀ ਦਾ ਮੈਂਬਰ ਹੈ ਅਤੇ ਭਾਰਤੀ ਜਨਤਾ ਪਾਰਟੀ ਦੀ ਨੁਮਾਇੰਦਗੀ ਕਰਦਾ ਹੈ।
Sportsmen	• https://www.icc-cricket.com/news/1939383 • https://www.icc-cricket.com/media-releases/1212091 • http://www.espncricinfo.com/story/_/id/25033717/associates-pathway-2023-world-cup-undergoes-major-revamp	2021 पापुआ न्यू गिनी क्रिकेणी सीरीज (मई)/परिचय	hi	2021 पापुआ न्यू गिनी क्रिकेणी सीरीज़ 2019–2023 आईसीसी क्रिकेट विश्व कप लीग 2 क्रिकेट टूर्नामेंट का 8 वां दौर होने वाला था, जो मई 2021 में पापुआ न्यू गिनी में खेला जाना था। यह नामीबिया, पापुआ न्यू गिनी और संयुक्त राज्य अमेरिका की क्रिकेट टीमों के बीच एक क्रिकेणीय राष्ट्र श्रृखला होती, जिसमें मैच एक दिवसीय अंतरराष्ट्रीय (वनडे) जुड़नार के रूप में खेले जाते थे। आईसीसी क्रिकेट विश्व कप लीग 2 2023 क्रिकेट विश्व कप के लिए योग्यता मार्ग का हिस्सा है। हालांकि, 12 फरवरी 2021 को, कोविड-19 महामारी के कारण श्रृखला को स्थगित कर दिया गया था।	2021 पापुआ न्यू गिनी ट्राई-नेशन सीरीज़ एक क्रिकेट टूर्नामेंट था जो सितंबर 2021 में पापुआ न्यू गिनी में खेला गया था। यह पापुआ न्यू गिनी क्रिकेट टीम और पापुआ न्यू गिनी क्रिकेट बोर्ड (पीएनसीबी) के बीच एक क्रिकेणीय राष्ट्र श्रृखला थी, जिसमें पापुआ न्यू गिनी और संयुक्त राज्य अमेरिका के बीच एक दिवसीय अंतरराष्ट्रीय (वनडे) मैच खेला गया था। यह श्रृखला सितंबर 2021 में होने वाली थी, लेकिन कोविड-19 महामारी के कारण इसे स्थगित कर दिया गया था।
Writers	• http://www.columbia.edu/itc/mealac/pritchett/00ambedkar/timeline/graphics/youth.html • http://www.columbia.edu/itc/mealac/pritchett/00ambedkar/timeline/1890s.html • http://www.columbia.edu/itc/mealac/pritchett/00ambedkar/txt_ambedkar_waiting.html	ভীমরাও রামজী শকপাল/প্রথম জীবন এবং শিক্ষা	bn	ভীমরাও রামজী শাকপাল যুবক থাকাকালীন 'মোং' (Mhow) অঞ্চলের (বর্তমান মধ্য প্রদেশ) এবং কেরীয় সামরিক সেনানিবাসে ব্রিটিশ কর্তৃক স্থাপিত শহরে আম্বেদকর অন্তর্ভুক্ত করেছিলেন। তিনি ছিলেন রামজী মালজী শাকপাল (Ramji Maloji Sakpal) এবং ভীমাবাইয়ে (Bhimabai) এরম তম সর্বকনিষ্ঠ পুত্র। তার পরিবার ছিলেন মারাঠী অধুনিক বর্তমান কালের "মহারাষ্ট্র" -এর রত্নগিরি জেলার "আম্বাডাং" (Ambavade) শহরে। তারা হিন্দু সম্প্রদায়ের অন্তির্ভুক্ত ছিল (মহর জাতি), যারা অস্পৃশ্য জাতি হিসেবে এবং প্রচণ্ড আর্থ-সামাজিক বৈষম্যের শিকার হত। আম্বেদকরের পূর্বপুরুষেরা ছিলেন ব্রিটিশ ইস্ট –ইন্ডিয়া কোম্পানির সেনা এবং তার পিতা "রামজী শাকপাল" মোং সেনানিবাসের ভারতীয় সেনা হিসেবে নিযুক্ত ছিলেন, তিনি সেকালের গৎবাঁধা শিক্ষাপদ্ধতিতে মারাঠী এবং ইংরেজিতে তিন্নি লাভ করেছিলেন এবং সেইসাথে তিনি প্রাথমিক বিদ্যালয়ের শিক্ষা লাভে কঠোর পরিশ্রম সহ্যগুণের উদ্বুদ্ধ করেন। কবির পাশের মতে, রামজী শাকপাল তার সন্তানদের হিন্দু সংস্কৃতি সম্পর্কে অধ্যয়ন করতে উদ্বুদ্ধ করতেন। যদিও আম্বেদকর বিদ্যালয়ে যেতেন, তাকে একজন এমন উচ্চতা হতে সেই পানি দেবে পান করাতো, যাতে নিম্নজাতের শিক্ষার্থীরা বা পানি বা পানির গাত্র স্পর্শ না করতে পারে । এই কাজটি সাধারণত আম্বেদকরের জন্য করতো বিদ্যালয়ের চাপরাশী (Peon) এবং যদি পিওন না থাকত বা না আসত, তখন সারাদিন পানি ঘন্টাই কাটাতে হতো, আম্বেদকর এই অবস্থাকে এভাবে আখ	ভীমরাও রামজি আম্বেদকর ১৯৪০ সালের ২৪শে নভেম্বর কলকাতায় জন্মগ্রহণ করেন। তার পিতার নাম মালক আম্বেদকর এবং মাতার নাম অজলিনা আম্বেদকর। তিনি কলকাতা বিশ্ববিদ্যালয় থেকে স্নাতক ডিগ্রি অর্জন করেন। এরপর তিনি কলকাতা বিশ্ববিদ্যালয়ে অধ্যাপনা শুরু করেন। এরপর তিনি কলকাতা বিশ্ববিদ্যালয় থেকে স্নাতকোত্তর ডিগ্রি অর্জন করেন।
Films	• https://www.rogerebert.com/reviews/stranger-than-paradise-1984	Stranger Than Paradise/Introduction	en	Stranger Than Paradise is a 1984 American black-and-white absurdist deadpan comedy film, co-written, directed and co-edited by Jim Jarmusch, and starring jazz musician John Lurie, former Sonic Youth drummer-turned-actor Richard Edson, and Hungarian-born actress and violinist Eszter Balint. It features a minimalist plot in which the main character, Willie, is visited by Eva, his cousin from Hungary. Eva stays with him for ten days before going to Cleveland. Willie and his friend Eddie go to Cleveland to visit her, and the three then take a trip to Florida. The film is shot entirely in single long takes with no standard coverage.	Stranger Than Paradise is a 1984 American drama film directed and co-written by Roger Ebert and starring John Lurie and John Travolta. The film tells the story of Willie (Lurie) and his cousin Eva (Travolta) traveling from Hungary to New York City in search of a place to stay. The film received positive reviews from critics, with praise for Lurie's performance, his direction, and the film's screenplay. It was nominated for the Academy Award for Best Original Screenplay.

Table 5.9: Some examples of XWikiGen using our best model (one example for each domain).

5.6 Summary

This chapter introduced a novel multi-document, cross-lingual, aspect-based summarization approach to generate Wikipedia article sections. We contributed a large dataset towards the same, XWikiRef, with ∼105K instances covering eight languages and five domains. Bengali (bn), English (en), Hindi (hi), Malayalam (ml), Marathi (mr), Oriya (or), Punjabi (pa), and Tamil(ta) are among the languages supported. Books, movies, politicians, athletes, and authors are examples of domains. We term our task XWikiGen, and because it involves the generation of long, multilingual text, XWikiGen is a very difficult task. Long text input can be challenging. As a result, we adopt a two-stage strategy. The first stage identifies promising candidate sentences across all the reference citations for a sample. The highest-scoring candidate sentences are passed as input to the second stage, which generates an abstractive summary.

We experiment with two different extractive summarization-based techniques: Salience and HipoRank. The main idea of salience-based extractive summarization is to find the top-K salient sentences from the input references based on the relevance of that sentence relative to a particular section title. Hierarchical and Positional Ranking model (HipoRank) is an unsupervised graph-based model where the key idea is that sentences similar to most sentences within a section are more critical. We use two state-of-the-art multi-lingual natural language generation models for the abstractive stage, mT5, and mBART. We experimented with different training setups to understand how language-relatedness and domain-relatedness help in the training of multi-lingual and multi-domain models, respectively. We report results using standard text generation metrics like ROUGE-L, METEOR, and chrF++. Our multi-lingual-multi-domain models using HipoRank (extractive) and mBART (abstractive) lead to the best results. The qualitative analysis reflects the issues that still need to be addressed in the above long-form generation, such as hallucination and repeating n-grams. Overall, the performance of the SOTA models on our dataset seems promising and can be used as a benchmark to work upon further by the research community.

Chapter 6

Conclusion and future work

In this work, we highlighted the existing lack of resources across the low-resource languages and proposed using cross-lingual approaches for text generation. We introduce the problem of XF2T alignment and generation for low-resource languages and also motivate and propose the XWikiGen problem where the input is (set of reference URLs, section title, language) and the output is a text paragraph. **Chapter 1** and **Chapter 2** covered the quantitative study of the availability of resources across the languages and explained the tasks and related work around them in detail.

The thesis covered the process of building parallel datasets for both the proposed tasks and explained in detail the novel NLP architectures suited for these tasks. **Chapter 3** covered the creation of the XF2T dataset, XAlign, consisting of English WikiData triples/facts mapped to sentences from LR Wikipedia. Our two unique approachesNER-based filtering with Semantic Similarity and Key-phrase Extraction with Relevance Rankinginclude solid baselines for the cross-lingual alignment challenge. Later, we scale the dataset across several languages and suggest more effective methods, such as 1) transfer learning and 2) remote supervision. We introduce a sizable collection of high-quality XF2T datasets in 12 languages, including English and a monolingual dataset in Hindi, Marathi, Gujarati, Telugu, Tamil, Kannada, Bengali, Punjabi, Assamese, Malayalam, and Odiya. In order to assess alignment techniques, we have also gathered 5402 human-labeled gold test datasets covering all of these languages.

We demonstrate solid baseline results by modifying well-known natural language generation (NLG) techniques for our suggested innovative XF2T task. First, we test widely used multilingual encoder-decoder models based on Transformer, such as the vanilla Transformer, IndicBART, and mT5. The performance of several training configurations, including multilingual, translate-input, translate-output, and bilingual, is then examined. Additionally, we methodically investigate several techniques for enhancing XF2T creation, including multilingual data-to-text pre-training, fact-aware embeddings, and structure-aware encoding. Detailed results and analysis of the above experiments are covered in **Chapter 4** of the thesis.

In **Chapter 5**, the long-form text generation task is modeled as a multi-document, cross-lingual summarising problem. A two-stage extractive-abstractive method is suggested. The best outcomes come from our multilingual, multidomain models using mBART (abstract) and HipoRank (extractive). We also discussed theăproblems with the aforementioned long-form generation that still need to be solved, such as hallucinations and repeated n-grams.

Future work:

1. We focus on some very specific challenges in the process of end-to-end text generation in low-resource languages. Although, our work is still constrained to particular domains and languages, and it would be interesting to explore how this scales up to the rest of the domain-language pairs.

2. We can connect different knowledge graphs with non-encyclopedic text available on the web. It might broaden the uses and diversity of text generation systems. Text creation capabilities for particular domains might be enhanced by including other data like images or videos. For instance, to give more signals (over fact) to the text generation engine, relevant images or videos can be combined with domains like landmarks, cities, chemicals, paintings, etc. But aligning graphics with text in one's own language may provide new challenges.

3. In the task of cross-lingual fact-to-text generation, we still focus on one-sentence generation. It is worth experimenting with generating more sentences in a single inference. In this case, maintaining the coherence between the sentences is an important research problem to explore as well.

4. Current models depict, to a large extent, the problem of hallucination, which is a critical research question to solve in the future. Repetition of tokens or phrases in the generated text is also a challenge fequently associated with large language models, and can be explored as a part of the future work.

5. Coming to the paradigm of experiments, all our methods are end-to-end neural network-based approaches. But, several other problem settings with partial neural and rule-based approaches can also be experimented with. Given these neural models' challenges in factual correctness, the above idea seems a good thread to explore, where rule-based methods will prove to be more interpretable in data selection stages.

Overall, there is a lot of scope to extend this work and modify its parts to develop better text generation models in low-resource languages.

Related publications

- Shivprasad Sagare, Tushar Abhishek, Bhavyajeet Singh, Anubhav Sharma, Manish Gupta, Vasudeva Varma. **XF2T: Cross-lingual Fact-to-Text Generation for Low-Resource Languages** arXiv
- Tushar Abhishek, Shivprasad Sagare, Bhavyajeet Singh, Anubhav Sharma, Manish Gupta, Vasudeva Varma. **XAlign: Cross-lingual Fact-to-Text Alignment and Generation for Low-Resource Languages.** In Companion Proceedings of the Web Conference 2022 (WWW 22 Companion).
- Swayatta Daw, Shivprasad Sagare, Tushar Abhishek, Vikram Puri, Vasudeva Varma. **Cross-lingual Alignment of Knowledge Graph Triples with Sentences.** In Proceeding of ICON-2021 Main Conference.

Bibliography

[1] T. Abhishek, S. Sagare, B. Singh, A. Sharma, M. Gupta, and V. Varma. Xalign: Cross-lingual fact-to-text alignment and generation for low-resource languages. In *The World Wide Web Conference*, pages 171–175, 2022.

[2] O. Agarwal, H. Ge, S. Shakeri, and R. Al-Rfou. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. *arXiv preprint arXiv:2010.12688*, 2020.

[3] O. Agarwal, H. Ge, S. Shakeri, and R. Al-Rfou. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565, 2021.

[4] D. Antognini and B. Faltings. Gamewikisum: a novel large multi-document summarization dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6645–6650, 2020.

[5] G. Attardi. Wikiextractor. `https://github.com/attardi/wikiextractor`, 2015.

[6] K. Bali, M. Choudhury, and P. Biswas. Indian language part-of-speech tagset: Bengali. *Linguistic Data Consortium, Philadelphia, LDC2010T16*, 2010.

[7] S. Banerjee and A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[8] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[9] K. Bennani-Smires, C. Musat, A. Hossmann, M. Baeriswyl, and M. Jaggi. Simple unsupervised keyphrase extraction using sentence embeddings. *arXiv preprint arXiv:1801.04470*, 2018.

[10] K. Bontcheva and Y. Wilks. Automatic report generation from ontologies: the miakt approach. In *International conference on application of natural language to information systems*, pages 324–335. Springer, 2004.

[11] J. A. Botha, Z. Shan, and D. Gillick. Entity linking in 100 languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7833–7845, 2020.

[12] T. Castro Ferreira, C. Gardent, N. Ilinykh, C. van der Lee, S. Mille, D. Moussallem, and A. Shimorina. The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76, Dublin, Ireland (Virtual), 12 2020. Association for Computational Linguistics.

[13] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*, 2017.

[14] D. L. Chen and R. J. Mooney. Learning to sportscast: a test of grounded language acquisition. In *Proceedings of the 25th international conference on Machine learning*, pages 128–135, 2008.

[15] M. Chen, S. Wiseman, and K. Gimpel. Wikitablet: A large-scale data-to-text dataset for generating wikipedia article sections. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 193–209, 2021.

[16] W. Chen, Y. Su, X. Yan, and W. Y. Wang. Kgpt: Knowledge-grounded pre-training for data-to-text generation. *arXiv preprint arXiv:2010.02307*, 2020.

[17] Z. Chi, L. Dong, S. Ma, S. H. X.-L. Mao, H. Huang, and F. Wei. Mt6: Multilingual pretrained text-to-text transformer with translation pairs, 2021.

[18] Z. Chi, L. Dong, F. Wei, W. Wang, X.-L. Mao, and H. Huang. Cross-lingual natural language generation via pre-training, 2019.

[19] Z. Chi, L. Dong, F. Wei, W. Wang, X.-L. Mao, and H. Huang. Cross-lingual natural language generation via pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7570–7577, 2020.

[20] P. Cimiano, J. Lüker, D. Nagel, and C. Unger. Exploiting ontology lexica for generating natural language texts from rdf data. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 10–19, 2013.

[21] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

[22] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

[23] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale, 2020.

[24] R. Dabre, H. Shrotriya, A. Kunchukuttan, R. Puduppully, M. M. Khapra, and P. Kumar. Indicbart: A pre-trained model for natural language generation of indic languages. *arXiv preprint arXiv:2109.02903*, 2021.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, 2019.

[27] B. Dhingra, M. Faruqui, A. Parikh, M.-W. Chang, D. Das, and W. Cohen. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, 2019.

[28] Y. Dong, A. Mircea, and J. C. Cheung. Discourse-aware unsupervised summarization of long scientific documents. *arXiv preprint arXiv:2005.00513*, 2020.

[29] X. Duan, M. Yin, M. Zhang, B. Chen, and W. Luo. Zero-shot cross-lingual abstractive sentence summarization through teaching generation and attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3162–3172, Florence, Italy, July 2019. Association for Computational Linguistics.

[30] D. Duma and E. Klein. Generating natural language from linked data: Unsupervised template extraction. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*, pages 83–94, 2013.

[31] H. Elsahar, P. Vougiouklis, A. Remaci, C. Gravier, J. Hare, F. Laforest, and E. Simperl. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[32] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*, 2020.

[33] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*, 2020.

[34] Z. Fu, B. Shi, W. Lam, L. Bing, and Z. Liu. Partially-aligned data-to-text generation with distant supervision. *arXiv preprint arXiv:2010.01268*, 2020.

[35] C. Gardent, A. Shimorina, S. Narayan, and L. Perez-Beltrachini. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[36] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. F. Liu, M. Peters, M. Schmitz, and L. S. Zettlemoyer. Allennlp: A deep semantic natural language processing platform. 2017.

[37] D. G. Ghalandari, C. Hokamp, J. Glover, G. Ifrim, et al. A large-scale multi-document summarization dataset from the wikipedia current events portal. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1302–1308, 2020.

[38] G. Giannakopoulos, J. Kubina, J. Conroy, J. Steinberger, B. Favre, M. Kabadjov, U. Kruschwitz, and M. Poesio. Multiling 2015: multilingual summarization of single and multi-documents, on-line

fora, and call-center conversations. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 270–274, 2015.

[39] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. Dont stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, 2020.

[40] T. Hasan, A. Bhattacharjee, W. U. Ahmad, Y.-F. Li, Y.-B. Kang, and R. Shahriyar. Crosssum: Beyond english-centric cross-lingual abstractive text summarization for 1500+ language pairs. *arXiv preprint arXiv:2112.08804*, 2021.

[41] T. Hasan, A. Bhattacharjee, M. S. Islam, K. Samin, Y.-F. Li, Y.-B. Kang, M. S. Rahman, and R. Shahriyar. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. *arXiv preprint arXiv:2106.13822*, 2021.

[42] H. Hayashi, P. Budania, P. Wang, C. Ackerson, R. Neervannan, and G. Neubig. Wikiasp: A dataset for multi-domain aspect-based summarization. *Transactions of the Association for Computational Linguistics*, 9:211–225, 2021.

[43] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, and M. Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization, 2020.

[44] Z. Jin, Q. Guo, X. Qiu, and Z. Zhang. Genwiki: A dataset of 1.3 million content-sharing text and graphs for unsupervised graph-to-text generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2398–2409, 2020.

[45] D. Kakwani, A. Kunchukuttan, S. Golla, N. Gokul, A. Bhattacharyya, M. M. Khapra, and P. Kumar. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, 2020.

[46] D. Kakwani, A. Kunchukuttan, S. Golla, N. Gokul, A. Bhattacharyya, M. M. Khapra, and P. Kumar. inlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4948–4961, 2020.

[47] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, et al. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*, 2021.

[48] N. Kitaev, L. Kaiser, and A. Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2019.

[49] K. Kolluru, M. Rezk, P. Verga, W. W. Cohen, and P. Talukdar. Multilingual fact linking. In *3rd Conference on Automated Knowledge Base Construction*, 2021.

[50] T. Kudo and J. Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP (Demos)*, 2018.

[51] A. Kunchukuttan. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf, 2020.

[52] F. Ladhak, E. Durmus, C. Cardie, and K. McKeown. Wikilingua: A new benchmark dataset for cross-lingual abstractive summarization. *arXiv preprint arXiv:2010.03093*, 2020.

[53] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*, 2017.

[54] R. Lebret, D. Grangier, and M. Auli. Neural text generation from structured data with application to the biography domain. *arXiv preprint arXiv:1603.07771*, 2016.

[55] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. 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, 2020.

[56] Y. Liang, N. Duan, Y. Gong, N. Wu, F. Guo, W. Qi, M. Gong, L. Shou, D. Jiang, G. Cao, X. Fan, R. Zhang, R. Agrawal, E. Cui, S. Wei, T. Bharti, Y. Qiao, J.-H. Chen, W. Wu, S. Liu, F. Yang, D. Campos, R. Majumder, and M. Zhou. Xglue: A new benchmark dataset for cross-lingual pretraining, understanding and generation, 2020.

[57] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[58] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.

[59] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer. Multilingual denoising pre-training for neural machine translation, 2020.

[60] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[61] H. Mei, T. UChicago, M. Bansal, and M. R. Walter. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. In *Proceedings of NAACL-HLT*, pages 720–730, 2016.

[62] P. Nema, S. Shetty, P. Jain, A. Laha, K. Sankaranarayanan, and M. M. Khapra. Generating descriptions from structured data using a bifocal attention mechanism and gated orthogonalization. *arXiv preprint arXiv:1804.07789*, 2018.

[63] K. Nguyen and H. Daumé III. Global voices: Crossing borders in automatic news summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 90–97, 2019.

[64] J. Novikova, O. Dušek, and V. Rieser. The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*, 2017.

[65] C. Patel and K. Gali. Part-of-speech tagging for gujarati using conditional random fields. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, 2008.

[66] M. Popović. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618, 2017.

[67] R. Qader, K. Jneid, F. Portet, and C. Labbé. Generation of company descriptions using concept-to-text and text-to-text deep models: dataset collection and systems evaluation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 254–263. Association for Computational Linguistics, 2018.

[68] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.

[69] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[70] G. Ramesh, S. Doddapaneni, A. Bheemaraj, M. Jobanputra, R. AK, A. Sharma, S. Sahoo, H. Diddee, D. Kakwani, N. Kumar, et al. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *arXiv preprint arXiv:2104.05596*, 2021.

[71] E. Reiter and R. Dale. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87, 1997.

[72] L. F. R. Ribeiro, M. Schmitt, H. Schütze, and I. Gurevych. Investigating pretrained language models for graph-to-text generation, 2021.

[73] T. Scialom, P.-A. Dray, S. Lamprier, B. Piwowarski, and J. Staiano. Mlsum: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, 2020.

[74] H. Shahidi, M. Li, and J. Lin. Two birds, one stone: A simple, unified model for text generation from structured and unstructured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3864–3870, 2020.

[75] A. Shimorina, E. Khasanova, and C. Gardent. Creating a corpus for Russian data-to-text generation using neural machine translation and post-editing. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 44–49, Florence, Italy, Aug. 2019. Association for Computational Linguistics.

[76] P. Tikhonov and V. Malykh. Wikimulti: a corpus for cross-lingual summarization. *arXiv preprint arXiv:2204.11104*, 2022.

[77] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[78] P. Vougiouklis, H. Elsahar, L.-A. Kaffee, C. Gravier, F. Laforest, J. Hare, and E. Simperl. Neural wikipedian: Generating textual summaries from knowledge base triples. *Journal of Web Semantics*, 52:1–15, 2018.

[79] X. Wan, H. Li, and J. Xiao. Cross-language document summarization based on machine translation quality prediction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 917–926, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[80] Q. Wang, S. Yavuz, X. V. Lin, H. Ji, and N. Rajani. Stage-wise fine-tuning for graph-to-text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 16–22, 2021.

[81] R. M. Weischedel, E. H. Hovy, M. P. Marcus, and M. Palmer. Ontonotes : A large training corpus for enhanced processing. 2017.

[82] T.-H. Wen, M. Gasic, N. Mrksic, L. M. Rojas-Barahona, P.-H. Su, D. Vandyke, and S. Young. Multi-domain neural network language generation for spoken dialogue systems. *arXiv preprint arXiv:1603.01232*, 2016.

[83] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics.

[84] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel. mt5: A massively multilingual pre-trained text-to-text transformer, 2021.

[85] Y. Yang, D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. H. Abrego, S. Yuan, C. Tar, Y.-H. Sung, et al. Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307*, 2019.

[86] Y. Yang, Y. Zhang, C. Tar, and J. Baldridge. PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification. In *Proc. of EMNLP*, 2019.

[87] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, 2021.

[88] C. Zhao, M. Walker, and S. Chaturvedi. Bridging the structural gap between encoding and decoding for data-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2481–2491, 2020.

[89] J. Zhu, Q. Wang, Y. Wang, Y. Zhou, J. Zhang, S. Wang, and C. Zong. Ncls: Neural cross-lingual summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3054–3064, 2019.