PAGES: 242-247
9/22/23
JOURNAL OF SCIENTIFIC RESEARCH AND TECHNOLOGY(JSRT) VOLUME-1 ISSUE-6 SEPTEMBER
Registered under MSME Government of India ISSN: 2583-8660

Text Summarisation And Translation Across Multiple Languages

Dr. Sameena Banu[1], Syeda Ummayhani[2]

[1]HOD, CSE Department, KBN University, Kalaburagi, India

[2]M. Tech Student, CSE Department, KBN University, Kalaburagi, India

syedaummayhani99@gmail.com

ABSTRACT

Multilingual text summarization using Natural Language Processing (NLP) stands as a vital research field, with its primary focus on condensing vital information from documents composed in diverse languages. Multilingual text summarization using Hugging Face's Transformers framework represents a cutting-edge endeavor in Natural Language Processing (NLP), addressing the challenge of distilling crucial information from documents written in various languages. The objective is to generate concise summaries that encapsulate the essential ideas while retaining the original context. This abstract explores the landscape of multilingual text summarization through the lens of Hugging Transformers, delving into methodologies and techniques facilitated by this advanced framework. This research endeavors to push the boundaries of NLP within the realm of multi-language text summarization and translation. By synergizing cutting-edge NLP techniques with the intricacies of language diversity, our work aims to cultivate effortless cross-cultural communication in an increasingly interconnected global landscape.

Keywords- Summarization, Languages, NLP, Pretrained models.

I. INTRODUCTION

In our ever-connected global landscape, the significance of seamless cross-language communication is more pronounced than ever. As language diversity continues to present barriers, the demand for automated resolutions within the domains of multilingual translation and text summarization has reached a critical juncture. This introduction serves as a prelude to the examination of how progressions in natural language processing (NLP) have opened avenues for inventive methodologies in tackling these intricate issues. In today's fast-paced exchange of information, content created in one language often needs to be understood, shortened, or changed into another language for people who speak different languages. This is not easy because each language has its own special ways of saying things, and it's important to keep the original meaning, cultural sensitivities, and purpose of the content. This research delves deeply into the murky field of multilingual text translation and condensing. It intends to use computers and linguistic technology to improve communication across various languages. Computers' ability to comprehend and generate human language has grown tremendously over the years, with advancements ranging from simple principles to cutting-edge technologies like neural networks. The problems we're facing span a broad spectrum of complexity, from reducing a lengthy document to its essentials while preserving its meaning to fluidly translating material across languages while taking into account the distinctive nuances of each. Using data and learning from past projects is crucial since these difficulties apply to languages with varying degrees of accessible linguistic resources. By looking at how the current developments in language technology mix with the complexity of utilizing numerous languages, this research intends to provide insights into new approaches, techniques, and models that enable translating and summarizing content in many languages successful. These insights have important effects in areas like global business, international relations, journalism, and sharing information, where smooth communication across languages can improve understanding and make interactions better in our diverse world. Our setup will consist of two key parts: firstly, the machine translation part, which will change the original text from its language to English, and secondly, the summarization part, which will make a shorter version of the translated text. We will test how well our system works using various texts in different languages and compare it to other very advanced methods already out there.

Our main goals for this project are to create a system that's both effective and precise for translating and summarizing in many languages, and to add to the ongoing progress in deep learning models and methods for understanding human language. We strongly believe that our system will be highly useful in fields like journalism, international business, and diplomacy, as it will make accurate and efficient communication across languages possible. In today's world of sharing information and connecting globally, it's really important to communicate well across different languages. But sometimes, the many languages and cultures can make this difficult. This is where multilingual text summarization comes in – it's a way to make things easier by condensing information from different languages. This introduction is about exploring how to do this, especially using a cool tool called Hugging Face Transformers. Hugging Face Transformers is like a super tool in the world of understanding and creating human-like text. By using this tool for multilingual text summarization, we can make sure that we don't lose the main points when translating between languages. This also helps make communication faster and more accurate. Our study looks closely at how Hugging Face Transformers can work together with multilingual text summarization. Our main goal is to use the power of this tool to make short yet clear summaries of text from many languages. We do this by adjusting these tools using different texts from different languages, so the summaries turn out accurate and make sense, no matter the language.

II. METHODOLOGY

There are different parts for this project and it can be broken down into different parts. There are different components that are in place which perform different actions. We use streamlit for deployment, PyTorch for model training on the GPU, Lang detect for detection of different languages from given text, and hugging face transformers models for the translation and summarization aspects of the project, following is the brief of all the different parts of this system.

MarianMTModel:

The MarianMTModel stands as a machine translation innovation crafted by the Hugging Face team, drawing from the Marian architecture. This model possesses the ability to convert text from one language into another. Its foundational training relies on a parallel corpus, a compilation of sentences in distinct languages conveying equivalent meanings. In the training process, the model assimilates the art of mapping sentences by source language to target language. Its architecture features an encoder-decoder network, wherein it ingests a source sentence and produces a corresponding target sentence in the desired language. Rooted in the Transformer framework, the MarianMTModel employs self-attention mechanisms to grasp the intricate word relationships within sentences. Operating under an autoregressive principle, it crafts target sentences word by word, using prior generated words as reference points. Notably, this model offers multi-lingual translation capabilities, enabling seamless translation across a diverse array of language pairs.

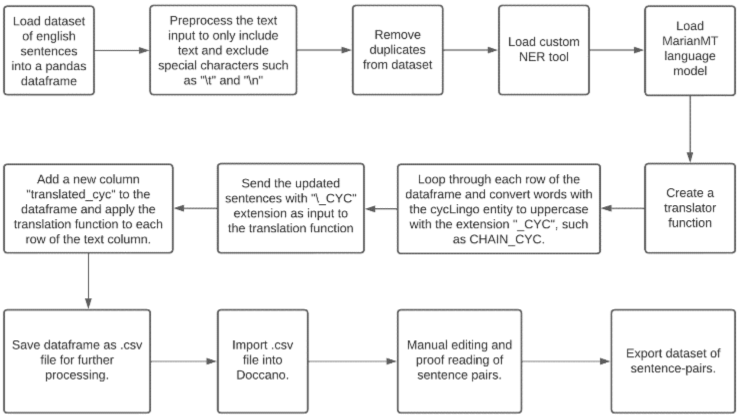

Load dataset of english sentences into a pandas dataframe
Preprocess the text input to only include text and exclude special characters such as "\t" and "\n"
Remove duplicates from dataset
Load custom NER tool
Load MarianMT language model
Create a translator function
Loop through each row of the dataframe and convert words with the cycLingo entity to uppercase with the extension "_CYC", such as CHAIN_CYC.
Send the updated sentences with "_CYC" extension as input to the translation function
Add a new column "translated_cyc" to the dataframe and apply the translation function to each row of the text column.
Save dataframe as .csv file for further processing.
Import .csv file into Doccano.
Manual editing and proof reading of sentence pairs.
Export dataset of sentence-pairs.

Figure 1: MarianMT Translator Function

T5

T5 is like a super-smart computer program created by Google. It already knows a lot about words and sentences. People can teach it to do different jobs, such as translating languages, making summaries, answering questions, and sorting out types of writing. T5's design is based on the ideas from the Transformer model, which is explained in a paper called "Attention is All You Need" by Vaswani and friends. Imagine T5 as a translator: you give it a bunch of words, and it gives you a different bunch of words that mean the same thing but in another language. T5 learned this by looking at tons of texts while training. It's like learning how to talk by listening to people. T5 uses tricks like paying attention to words in sentences and understanding how they're connected. When it's learning, it tries really hard to guess what the right words are in the new language based on the words it already knows. It's like a puzzle where it tries to put the pieces together in the best way. This way, T5 becomes good at translating or doing other language tasks. The creation of this model was presented in a research paper called " The Bounds of Transfer Learning: A Unified Text-to-Text Approach." In this paper, the researchers investigated how well transfer learning could work by developing a single framework that can transform all sorts of language-related challenges into a simple "text in, text out" style. T5 model relies on an encoder-decoder design. Essentially, it takes text as input & produces new text as output.

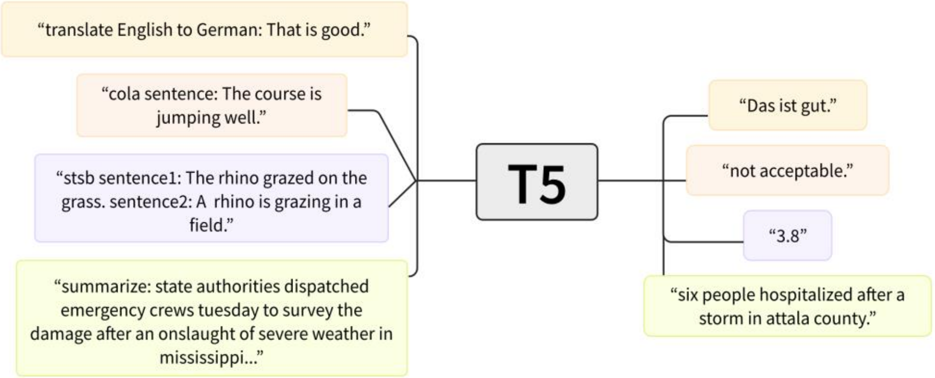

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi..."
T5
"Das ist gut."
"not acceptable."
"3.8"
"six people hospitalized after a storm in attala county."

Figure 2: Using T5 to summarize text

LANGDETECT

Langdetect is a Python library used for language detection in text data. It allows you to identify the language of a given text or document based on its content. Langdetect is a language detection tool created by Nakatani Shuyo. Built on Java, it can figure out the language of a given text using statistical techniques. This tool employs things called character n-grams and language profiles to determine the text's language. It was trained on a bunch of text in many different languages, and it makes educated guesses about the language using probabilities. Langdetect is handy because it can identify over 55 languages quite well. It's really useful when you're dealing with texts in multiple languages and you're not sure what languages they are.

Huggingface Transformers

Hugging Face Transformers stands as a flexible toolkit that simplifies the incorporation and utilization of advanced natural language processing (NLP) models. Hugging Face Transformers serves as an open-source hub and library having aim of democratizing NLP & machine learning. It grants users access to an array of over 20,000 pre-trained models built upon the transformer architecture. This empowers data scientists and developers to seamlessly engage with text, speech, vision, tabular data, and reinforcement learning. The library is compatible with diverse deep learning frameworks, including PyTorch and TensorFlow, and streamlines tasks via user-friendly interfaces called Pipelines. Additionally, the platform boasts a rich collection of datasets and thrives on a vibrant community, effectively broadening the accessibility and practicality of advanced AI across various domains.It particularly excels in offering a robust foundation for constructing systems that can summarize text in multiple languages. The Hugging Face Transformers library has become widely renowned within the NLP community. This can be attributed to its intuitive user interface, comprehensive documentation, and the ease of access to robust pre-trained models. The library has significantly lowered barriers, granting a broader spectrum of individuals the opportunity to explore, implement, and advance state-of-the-art NLP capabilities. This democratization of cutting-edge NLP technology has fostered greater experimentation, practical applications, and innovation in the field.

Torch

PyTorch, often referred to simply as "Torch," stands as an open-source machine learning framework meticulously crafted with a primary focus on deep learning applications. It enjoys extensive adoption among a diverse community of researchers, data scientists, and developers who employ it for the creation and training of neural networks. PyTorch has earned its reputation for its dynamic computational graph, a feature that sets it apart by providing a more intuitive and adaptable framework for model construction in contrast to traditional static graph-based frameworks. The torch package implements mathematical operations on multi-dimensional tensors and provides data structures for working with them. It also includes a wide variety of utilities, such as those for the efficient serialization of Tensors and other kinds. It also offers a CUDA version, which lets you do tensor calculations on NVIDIA graphics processing units (GPUs) with compute capabilities more than or equal to 3.0. Torch allows us to utilize the raw power that is within the CUDA GPU cores and allows us to perform multi-dimensional tensor matrix multiplication which is the core to any machine learning language model, this fast processing on GPU makes it very efficient for the task.

Experiment Tools

The UI of this project has been developed using streamlit. Streamlit is a free and open-source Python library that facilitates the development and distribution of high-quality, data-science and machine learning-specific web applications. Python tools and modules like Hugging face transformers, NumPy, and Pandas make it possible to create and release robust data applications in a matter of minutes.

III. ANALYSIS STEPS

The analysis of this project can be broken down into multiple steps ranging from data collection to deployment. Depending upon the complexity of the language model the steps vary widely. The following steps are involved in the proposed system.

Data Collection

The initial phase of the current system approach entails gathering the necessary data for the project. This entails pinpointing data sources suitable for training the machine learning model. These sources encompass a range of possibilities including social media platforms, news articles, blogs, and other pertinent outlets. Subsequent to source identification, the data is amassed and organized into a suitable format for subsequent processing.

Language Detection

The subsequent phase involves identifying the language present within the input text. This holds significance due to the language-specific nature of subsequent translation and summarization models. It is imperative for the language detection model to possess the capability to precisely ascertain the language contained within the input text. In this phase, our objective is to identify specific languages within the provided input text. This task gains prominence as the subsequent utilization of translation and summarization models is contingent upon the detected language. Specifically, our language identification technology is optimized to correctly identify a wide range of languages, including Arabic, French, Japanese, Hindi, Russian, Spanish, and others. The success of the following natural language processing activities depends on the accuracy with which the language of the input text is identified.

Translation

Translating the source material into the target language is the next step. A machine translation model that has already been trained is used to carry out this task. This model is fed the input text and its translation in both the source and destination languages. Afterwards, the model produces the translated text as its output. Data translation is an important problem in the field of natural language processing. You may use this method to translate text from one language to another, facilitating mutual comprehension and communication across linguistic boundaries. It's important to note that a number of factors—including the quality of the pre-trained module, the quantity and diversity of the training data, and the complexity of the source and destination languages—determine how well the translation process works. Furthermore, it's important to recognize that diverse languages might demand differing extents of post-processing and human verification to attain optimal results. In recent times, notable progress within the realm of Natural Language Processing (NLP), particularly propelled by transformer-based models like BERT, GPT, & counterparts, has led to remarkable enhancements in the precision and fluency of machine translation systems. This evolution has contributed to remarkably smoother cross-lingual communication experiences, marking a significant advancement in this domain.

Text Summarization

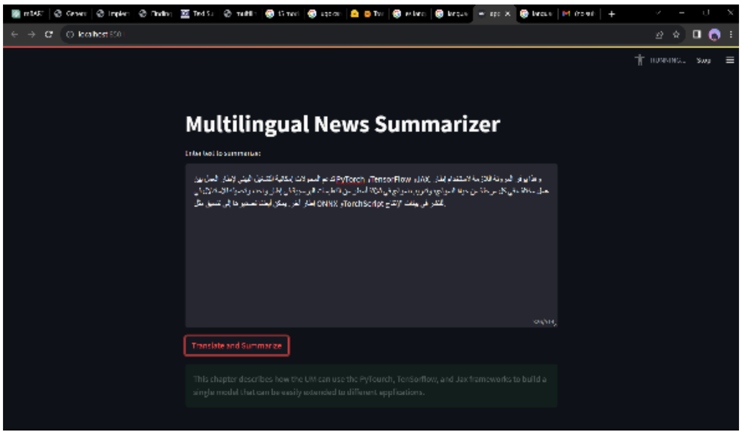
Moving to the subsequent phase, our focus shifts to condensing the translated text. This is accomplished through the utilization of pre-trained text summarization model. In this step, translated text serves as the input to the model, which in turn generates a succinct and condensed rendition of the original content. To address text summarization tasks that encompass multilingual translation and summarization, the "mT5" (multilingual Text-To-Text Transfer Transformer) model emerges as a compelling option. An extension of the T5 model, mT5 is expertly crafted to manage a multitude of languages, effectively catering to translation and summarization endeavors. This model operates within the text-to-text framework, granting it the capacity to excel in a broad spectrum of text generation tasks, encompassing both translation and summarization. The foundation of mT5's versatility lies in its comprehensive training across numerous languages. This extensive training equips mT5 to seamlessly process input in diverse languages and produce concise, coherent summaries. By finetuning the mT5 model according to the specifics of your task and dataset, you can unlock its potential for generating summaries that are not only multilingually astute but also aligned with your quality benchmarks.

Deployment

The ultimate phase entails the deployment of the trained model within a production setting. This integration encompasses embedding the model into a software application or service, rendering it accessible to end-users. The operationalized model becomes adept at scrutinizing novel texts and furnishing real-time sentiment analysis outcomes. Finally, the translated and summarized text is displayed to the user through a user interface. In this project, Streamlit is used to build the user interface. Next, we employ the Streamlit's text area widget, facilitating user input entry. Upon the user's interaction with the "Translate and Summarize" button, we invoke the multilingual summarizer function, passing along the provided input text. The resulting summarized text is presented to the user via Streamlit's title widget, completing the interaction loop.

IV. RESULTS

Within the function, our initial step involves employing the Lang detect library to identify the language of the input text. In presence of supported languages, we harness pre-trained machine translation model sourced from the Hugging Face Transformers library for text translation. Subsequently, we tap into another pre-trained text summarization model from the same library for generating concise summary of translated text. The procedure culminates in furnishing the user with the summarized output. In scenarios where the language isn't supported, a user-friendly warning message is presented. Moving ahead, we harness Streamlit's text area widget, enabling users to input their desired text. Upon triggering the "Translate and Summarize" button, the multilingual summarizer function is invoked, with the input text as its parameter. The generated summarized output is then elegantly showcased to the user, facilitated by Streamlit's title widget.



Multilingual News Summarizer

Enter text to summarize:

[illegible] JAX، TensorFlow، PyTorch [illegible] TorchScript، ONNX [illegible]

This chapter describes how the UM can use the PyTourch, TensorFlow, and Jax frameworks to build a single model that can be easily extended to different applications.

Figure 3: User facing text input box, Provides users with translation and summary

V. CONCLUSION

Into our article we demonstrated use of langdetect by taking user provided input and detecting the language of the input automatically, this input is then converted into a common base representation of tokens which allows the machine learning model to understand the input text. The model doesn't really understand the text that is given but it understands the mathematical representation of the words in a multi-dimensional space, it constructs a multi-dimensional tensor and categorizes all the input tokens, we control the parameters and tweak it to correct the model. It is then able to take another stream of input and apply the same tokenization process to breakdown the input and perform the translation operation, this breaking down of text and finding identical words in another language from millions of words requires a lot of computational power, we utilize GPUs for this by making use of PyTorch. By utilizing Lang detect we were able to identify 8-10 different languages from across the world, then translate those to English and also provide a summary of it. The pre-trained large language models allowed the process of translation to be smooth. The language model utilizes statistical differences between the probabilities of occurrences of a word in question to all the pre computed statistical data from its learnings to determine if the word that the model is going to choose for the translation is a good candidate in the broader and current context of translation, this is done by statistical grouping of all the words during models training.

REFERENCES

1. Lloret, E., & Palomar, M. (2011). Text summarisation in progress: a literature review. Artificial Intelligence Review, 37(1), 1–41. https://doi.org/10.1007/s10462-011-9216-z
2. Godbole, S., Jadhav, V., & Birajdar, G. (2020). Indian Language Identification using Deep Learning. ITM Web of Conferences, 32, 01010. https://doi.org/10.1051/itmconf/20203201010
3. S. (2017). MULTILINGUAL TEXT SUMMARIZATION TECHNIQUES. International Journal of Research in Engineering and Technology, 06(07), 28–31. https://doi.org/10.15623/ijret.2017.0607005
4. Simon-Maeda, A. (2003). Resisting Linguistic Imperialism in English Teaching. English for Specific Purposes, 22(4), 428–431. https://doi.org/10.1016/s0889-4906(03)00004-8
5. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

6. Katsafados, A. G., Androutsopoulos, I., Chalkidis, I., Fergadiotis, E., Leledakis, G. N., & Pyrgiotakis, E. G. (2021). Using textual analysis to identify merger participants: Evidence from the U.S. banking industry. Finance Research Letters, 42, 101949. https://doi.org/10.1016/j.frl.2021.101949
7. Oshimo, J., & Kamiya, T. (2006). Japanese Sentence Patterns for Effective Communication: A Self-Study Course and Reference. Japanese Language and Literature, 40(1), 129. https://doi.org/10.2307/30198004
8. Gidiotis, A., & Tsoumakas, G. (2023). Bayesian active summarization. Computer Speech & Language, 83, 101553. https://doi.org/10.1016/j.csl.2023.101553
9. Preprint repository arXiv achieves milestone million uploads. (2014). Physics Today. https://doi.org/10.1063/pt.5.028530.
10. Machine Learning Based News Text Classification. (2020). Machine Learning Theory and Practice, 1(1). https://doi.org/10.38007/ml.2020.010103