**Project Report**

**Title: A Comparative Study on analysing the performance of Tokenizers on various langauges**

**Introduction (3-4 Paras)**

~~Brief overview of the problem (1)~~

~~What is NLP (1)~~

~~Brief overview of the problem (2)~~

~~About the project – Intro, Purpose (1)(3)~~

**Methodology (7-8 Paras)**

~~About the targeted languages (n) (Two for each lang)~~

Dataset used (CC100) – About it (Along with each lang’s description)

Tools used (1)(End)

Extraction/ tokenization pipeline –

**data flow, parameters, cleaning process (3)**

**Results (Paras and graphs)**

Benchmark analysis

Discuss the results

**Conclusion (Depends on the results)**

What are the conclusions drawn from the results?

**Future Scope (5-6 bullet points)**

What can be done to improve this?

**References**

Papers

List of dependencies

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| S.No | Section Name | Page Number |
| **1.** | **Introduction** |  |
| **2.** | **Methodology** |  |
| **3.** | **Results** |  |
| **4.** | **Conclusion** |  |
| **5.** | **Future Scope** |  |
| **6.** | **References** |  |

**1. Introduction (Draft)**

There are thousands of languages that have been adopted by various corners and cultures of our world. Since the dawn of mankind, languages lay the foundation for oral and written communication between humans from different cultures and backgrounds. Natural Language Processing attempts to bridge the gap between human language and computer language by providing well established and tested procedures for converting human language into an intermediate format that can be interpreted by our computers. The boom of state of the art (SOTA) Large Language Models with the introduction of the transformer architecture bought this domain to the spotlight. Research is being carried out with the focus on making this technology as accessible as possible to even the most remote parts of the world.

As stated earlier, Natural Language Processing (NLP for short) is a discipline in computer science (AI to be more specific) that focuses on developing tools and techniques (or leveraging existing ones) to process natural language. The purpose of this field is

to provide a computer with the necessary tools to translate human readable language (Ex: English) into the machine’s native readable format, by storing text in an encoded form (ASCII, utf-8) which can be converted. Some of the core objectives of NLP is to enable word segmentation, speech recognition, text to speech, Named entity recognition, Tagging parts of speech in a sentencen etc. Some of the methodologies adopted over the years to tackle such tasks is through using Rule based, Statistical methods and Deep Learning models for mapping text patterns.

The major focus of this study is on **tokenization,** a technique that lays the foundation for all Natural Language Processing tasks.. Tokenization is regarded as the process of segmenting/splitting a given corpus of text into it’s constituent buidling blocks: **tokens**.The precise definition of a token is dependent on the tokenization process. There are three main types of tokenization, thus producing three different types of tokens: **Word tokenization (tokens=words), subword tokenization(tokens=subwords) and character tokenization (tokens=characters).** A tokenizer is a program that performs the tokenization process on a corpus of text and produce the respective list of tokens. There are many approaches for building tokenizers, such as:

1. **1Rule-Based:** Tokenization using predefined rules involves segmenting text based on explicit, manually crafted patterns such as punctuation marks or whitespace. This method is straightforward and effective for languages with clear delimiters but can be inflexible and struggle with ambiguous or complex text. It requires continual updating and fine-tuning as language use evolves.
2. **Knowledge-Based:** This approach utilizes linguistic resources like dictionaries, lexicons, and morphological rules to identify tokens. By leveraging pre-existing linguistic knowledge, it can effectively handle domain-specific terms and proper nouns. However, it may require extensive and specialized resources to cover all variations in text.
3. **Statistical:** Statistical tokenization relies on probabilistic models trained on large corpora to learn typical patterns and boundaries for tokens. These models can adapt to different contexts and handle variations in text effectively. They may require substantial computational resources and large datasets to achieve high accuracy.
4. **Neural-Based:** Neural-based tokenizers use deep learning techniques and neural networks to learn tokenization patterns from large training datasets. These models can adapt to various languages and contexts and often handle complex and ambiguous text more robustly. They require significant computational power and large annotated corpora for training.
5. **Hybrid Approaches:** Hybrid tokenizers combine elements of rule-based, knowledge-based, statistical, and neural methods to leverage the strengths of each approach. This method can achieve higher accuracy and robustness by addressing the limitations of individual techniques. It aims to balance precision, adaptability, and resource requirements for optimal performance.

**Note:** Other approaches includel: Byte Pair Encoding, SentencePiece, WordPiece, n-gram language models

A Language is said to be a high resource language (Ex: English) when there are sufficient amounts of resources openly available in the text format in the internet, else it is called a low resource language (Ex: Urdu). Research focusing on these low resourceful languages is sparse, due to the lack of text data availability to perform experiments. Each language has it’s own diverse set of rules, making it very difficult to develop tools to efficiently process a large variety of language texts, and the various semantic and syntactic interpretations only adds to the difficulty to convert spoken data into text, and following NLP’s procedures for that language. Such challenges decrease the feasibility and scope of experimentation, making the results less reliable.

**1.1 Overview**

The purpose of this project is investigate the performance of readily available tokenizers, and compare it versus tokenizers that were built by keeping the specifically rules of the target language in mind. The study has been carried out for 2 langauges namely: Urdu from the parso arabic family, Chinese (simplified) from the logographic family. This report presents all of the findings and is split in the following manner:

1. **Section-2: Methodology** covers all the technical details of the project, including the datasets used, employed tools and techniques to extract a list of unique tokens.
2. **Section-3: Results** presents all the metrics through which each tokenizer’s performance has been measured on a given corpus of text through plots.
3. **Section-4: Conclusion** focuses on providing concrete suggestions based on the previous two sections.

**2. Methodology (Draft)**

**2.1 About the Unicode Standard**

Simply put, the unicode standard is an encoding format that is used for computers to better represent text that isn’t just english. Formally, the unicode standard is a text encoding standard designed to support the use of text in the digital format. The formulation of this standard was to solve the compatibility problem between encoding formats, where often the encoding representation of one format is interpreted as garbage by the other. Many normal characters have been unified under its umbrella, making it easier to eliminate language specific digital representation systems. The underlying principle is to encode every grapheme/grapheme-like units by assigning each grapheme with a **code point,** which enables it to support a wide variety of high resource and low resourceful langauges.

The default encoding form of any language for the entire internet is the **Unicode Transformation Format (utf-8),** a lightweight unicode format. It is optimized for byte-oriented systems where backward compatibility with ASCIIis important, and is easier to parse compared t o legacy encoding schemes. This is importnant because the datasets that have been obtained during this study are from an internet repository, and each file is stored in the **utf-8** format.

**2.2 Target Languages**

**1. About urdu**

The Urdu language alphabet is a modified form of the Perso-Arabic script, which itself is derived from the Arabic script. It consists of 39 primary characters, including 27 consonants and 12 vowels. Urdu is written from right to left, and like Arabic, it is an abjad script, meaning it primarily represents consonants, with vowel sounds indicated by diacritics or implied through context. The script has adapted additional letters to accommodate phonetic sounds specific to Urdu that are not present in Arabic. For instance, Urdu includes characters such as ٹ, ڈ, ڑ, and others, which cater to the unique phonetic needs of the language.

A key feature of Urdu script is its use of ligatures, where characters combine into complex forms depending on their position in a word (initial, medial, final, or isolated). This script's cursive nature requires that letters be connected, which can make reading and writing more fluid but also requires more intricate training. Furthermore, Urdu integrates a significant number of loanwords from Persian and Arabic, contributing to a rich vocabulary and diverse orthographic influences. This script's adaptability and depth reflect Urdu's historical and cultural interactions with various languages and civilizations.

**2. About chinese**

Simplified Chinese characters are part of a writing system that was developed to increase literacy and simplify the traditional Chinese script. Introduced officially in the 1950s and 1960s, Simplified Chinese reduces the number of strokes in many characters, making them easier to learn and write. The script is used predominantly in Mainland China, Singapore, and Malaysia. It comprises fewer radicals and simplified forms of the more complex traditional characters, which can greatly aid in speed and efficiency for both learners and writers.

One key feature of Simplified Chinese is its character reduction and streamlining. While Simplified characters maintain the core semantic elements of their traditional counterparts, they often have fewer strokes and less complex forms. This simplification is intended to make reading and writing more accessible, particularly for those who may not have had extensive formal education. Despite the simplifications, Simplified Chinese retains a deep connection to its traditional roots, preserving much of the language's rich historical and cultural context.

**2.2 Dataset info**

The CC100 dataset is a large-scale collection of text corpora gathered in multiple languages. The primary aim of this dataset is to enhance the accessibility of natural language processing (NLP) tasks across a broad range of languages. The CC100 dataset comprises texts collected through web scraping from diverse sources, including books, articles, and websites scattered across the internet. This comprehensive collection ensures a wide representation of topics and styles, making it highly valuable for training and evaluating language models. The creation of CC100 reflects a concerted effort to diversify and enrich the corpus of high-quality text available for training artificial intelligence models.

For the purposes of this study, the following language datasets have been extracted from the CC100 collection:

1. CC100 - Urdu: Contains up to 15 GB of Urdu text.
2. CC100 - Chinese (Simplified): Contains up to 60 GB of Simplified Chinese text.
3. CC100 - Arabic: Contains up to 30 GB of Arabic text.

These subsets of the CC100 dataset provide a significant amount of text in their respective languages, facilitating various NLP tasks such as language modeling, translation, and text classification. By offering extensive and diverse text data, CC100 supports advancements in multilingual NLP research and helps build more robust and inclusive language models.

**2.4 Phases**

**Phase-1: Extraction**

- \*\*Loading and Cleaning the Text\*\*:

- \*\*Loading\*\*: The raw text file is read into memory to prepare the content for processing. This step ensures that the entire text data is accessible for subsequent operations.

- \*\*Cleaning\*\*: Apply language-specific cleaning rules to normalize the text. This includes removing unnecessary characters, standardizing formatting, and addressing any language-specific issues to ensure consistency and readability.

- \*\*Creating the Intermediate File\*\*:

- \*\*Saving Cleaned Text\*\*: Store the cleaned text in a new file, formatted with each line representing a distinct segment of text. This intermediate file serves as a structured, ready-to-process source for the next stage of tokenization.

- \*\*Tokenization\*\*:

- \*\*Applying Tokenizers\*\*: Utilize various tokenization methods suited to the target language to process the intermediate file:

- \*\*For Urdu\*\*: Apply NLTK’s Treebank tokenizer for rule-based segmentation, SpaCy’s tokenizer for deep learning-based processing, and Stanza’s neural-based tokenizer for advanced accuracy.

- \*\*For Simplified Chinese\*\*: Use Jieba’s tokenizer for a combination of dictionary and statistical methods, SpaCy’s tokenizer for deep learning approaches, and Stanza’s neural-based tokenizer for high precision.

- \*\*Generating Tokens\*\*: Each tokenizer produces a set of tokens from the text, which are analyzed for effectiveness and accuracy in representing the text content.

- \*\*Filtering and Generating Unique Tokens\*\*:

- \*\*Filtering\*\*: Clean the tokenized data by removing duplicates and irrelevant tokens to refine the list. This step ensures that the tokens are meaningful and relevant for analysis.

- \*\*Generating Unique Tokens\*\*: Compile a final list of unique tokens from the filtered data. This list is essential for further analysis, including frequency counts and model development, and provides a clear representation of the distinct elements in the text.

**3. Results**

**3.1 Phase-2: Analysis**

**3.2 Graphs**

For urdu:

For chinese:

**4. Conclusions**

**5. Future scope**

**6. References**