	UNIGRAM-1000	UNIGRAM-2000	BPE-1000	BPE-2000	mBERT - 1000	mBERT - 2000	Indic-BERT - 1000	Indic-BERT -2000	WHITE SPACE
PRECISION	0.05862068965517241	0.07979274611398963	0.05382674516400336	0.06965174129353234	0.05067567567567568	0.05067567567567568	0.013171225937183385	0.013171225937183385	0.13971742543171115
RECALL	0.2518518518518518	0.2851851851851852	0.2379182156133829	0.26022304832713755	0.2247191011235955	0.2247191011235955	0.04961832061068702	0.04961832061068702	0.3308550185873606
F-SCORE	0.0951048951048951	0.12469635627530366	0.0877914951989026	0.10989010989010989	0.08270158511371468	0.08270158511371468	0.02081665332265812 7	0.02081665332265812 7	0.1964679911699779 3

Unigram Tokenization:

Vocabulary Size 1000: Shows moderate precision (0.0586), relatively high recall (0.2519), and low F-score (0.0951).

Vocabulary Size 2000: Exhibits improved precision (0.0798), higher recall (0.2852), and a slightly better F-score (0.1247) compared to the 1000 vocabulary size.

Analysis: Unigram tokenization captures individual words as tokens, which results in a relatively high recall but lower precision due to tokenizing infrequent or rare words. Increasing the vocabulary size enhances the representation of the language, leading to better performance metrics.

BPE (Byte Pair Encoding) Tokenization:

Vocabulary Size 1000: Shows comparable precision (0.0538) and recall (0.2379) to Unigram but slightly lower F-score (0.0878).

Vocabulary Size 2000: Exhibits improved precision (0.0697), recall (0.2602), and F-score (0.1099) compared to the 1000 vocabulary size.

Analysis: BPE tokenization merges frequent character pairs iteratively to build a vocabulary, resulting in better tokenization of rare words and improved performance metrics compared to Unigram tokenization, especially with a larger vocabulary size.

mBERT (Multilingual BERT) Tokenization:

Max Length 1000: Shows moderate precision (0.0507), recall (0.2247), and F-score (0.0827) consistently for both vocabulary sizes (1000 and 2000).

Analysis: mBERT tokenization utilizes a pre-trained multilingual BERT model to tokenize text, providing contextual embeddings. However, in this comparison, it demonstrates lower precision, recall, and F-score compared to other methods, indicating potential limitations in capturing token boundaries effectively.

Indic-BERT Tokenization:

Max Length 1000: Exhibits the lowest precision (0.0132), recall (0.0496), and F-score (0.0208) among all methods and models.

Analysis: Indic-BERT is specifically designed for tokenizing Indic languages, including Hindi. However, in this comparison, it shows significantly lower performance metrics compared to other methods, indicating potential challenges in effectively tokenizing Hindi text.

White Space Tokenization:

No Vocabulary Size: Shows the highest precision (0.1397), recall (0.3309), and F-score (0.1965) among all methods and models.

Analysis: White space tokenization simply splits text based on whitespace characters, resulting in tokens that correspond to words. While it achieves high precision, recall, and F-score, it may struggle with tokenizing complex linguistic structures and word boundaries effectively.

Conclusion:

Increasing the vocabulary size generally leads to improvements in precision, recall, and F-score across different tokenization methods. This suggests that a larger vocabulary size allows for better representation of the language's complexities and nuances**.**

There are trade-offs between precision, recall, and F-score. While some methods/models may excel in one metric, they may lag in others**.**

- ➔ **White space tokenizer is giving best result since it is breaking word by word, whereas other tokenizers are also breaking words from between**