SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I2P108

Volume 11 Issue 2, 69-79, February 2024
© 2024 Seventh Sense Research Group®

Original Article

A Comprehensive Stop-Word Compilation for Kannada Language Processing

M.S. Sowmya[1], M.V. Panduranga Rao[2]

[1,2]*Department of Computer Science and Engineering, Jained Deemed to be University, Karnataka, India.*

[1]*Corresponding Author : mssowmya.sbmjce@gmail.com*

Received: 30 October 2023 Revised: 04 December 2023 Accepted: 31 December 2023 Published: 02 February 2024

Abstract *- In this work, a vital aspect of Kannada Natural Language Processing (NLP) takes the stage, with the construction of a standardized stop-word list emerging as a pioneering endeavor. This essential list serves as a foundation for improving language comprehension and processing activities. The work offers a rigorous technique that includes data gathering, tokenization, and TF-IDF score computation using the IndicCorp Kannada dataset. The study innovatively pioneers the construction of a stop-word list exclusively designed for the Kannada language, a first in this domain. The findings highlight the significance of these stop words and their prospective applications in diverse NLP endeavors, providing the framework for the upcoming construction of a Kannada-specific text summarizing work. The human refinement procedure ensures precision in stop-word compilation while considering inherent subjectivity and dataset-specific restrictions. Importantly, this study not only gives valuable insights into linguistic characteristics but also pioneers an innovative approach for stop-word generation in Kannada, establishing itself as a pioneering effort in this specific area of research. Furthermore, the study goes beyond its immediate findings by offering methodologies for the automated compilation and validation of stop words, thus laying the groundwork for further research. This foresight adds to the ongoing advancement of Kannada NLP methods.*

Keywords *- Text summarization, Natural Language Processing, Stop words, BERTsum, Kannada language.*

1. Introduction

Natural Language Processing (NLP) has become a critical domain within the field of artificial intelligence, with the objective of enabling machines to understand and analyze human language. Kannada, a Dravidian language primarily spoken in the Indian state of Karnataka, distinguishes itself among the numerous languages spoken worldwide [1]. An essential component of language processing tasks, the production of a standardized stop-word list, is investigated in this study, which addresses a critical aspect of Kannada NLP. The following sections offer a comprehensive examination of the context, rationale, aims, significance, and methodology of the research that forms the foundation of this investigation.

Language, being a dynamic and complex aspect of human communication, demands that developers of practical NLP applications have a thorough comprehension of its complexities. Stop words, which are frequent occurrences such as articles and prepositions with little semantic significance, are crucial in the process of language processing [2]. To prioritize the essential content, they are generally omitted from the analysis. Nevertheless, since stop words differ in nature between languages, language-specific stop-word inventories are required [3]. The lack of a standardized stop-word list challenges the development of precise language models in Kannada, a language renowned for its extensive literary and cultural legacy. This study aims to address this disparity by developing an all-encompassing stop-word inventory specifically designed to capture the intricacies of Kannada.

This research is motivated by the increasing significance of Kannada NLP and the absence of specialized linguistic resources for the language. The demand for precise and domain-specific stop words becomes more significant as NLP applications progress. Stop words serve as linguistic landmarks that direct language models toward the retrieval of substantial data. Driven by the aspiration to progress Kannada language technology, this study aims to develop a fundamental tool that can enhance a range of NLP activities, including sentiment analysis and text summarization.

There are numerous primary objectives to this investigation. The primary aim of this study is to develop a standardized inventory of stop-words for Kannada, taking into account linguistic and contextual factors. This entails a rigorous procedure of gathering data, tokenizing it, and determining TF-IDF scores to rank terms according to their importance within a specified corpus [4]. Following this, the research aims to meticulously examine and enhance the stop-word list to guarantee its precision and suitability across various contexts. In addition to generating the stop-word list, the study seeks to assess its efficacy through experimentation within the framework of a bespoke text summarization task. The growing need for precise language processing tools within the Kannada-speaking community emphasizes the significance of this research. The proliferation of digital content in Kannada, encompassing news articles and social media posts, necessitates the development of sophisticated language models capable of extracting crucial information and discerning context. The stop-word list produced because of this study can significantly improve the precision of these models by accommodating Kannada's unique linguistic subtleties.

The research problem at the heart of this study is the deficiency of a linguistically tailored stop-word list for Kannada. Stop words, often regarded as linguistically inconsequential, play a crucial role in language processing tasks. Existing studies have either relied on generic stop-word lists or adapted lists from other languages, neglecting the intrinsic linguistic nuances specific to Kannada. The absence of a dedicated stop-word list tailored for Kannada impedes the precision and effectiveness of NLP models when processing Kannada text. In addressing this research gap, the study sets out on a novel undertaking-to systematically generate a standardized stop-word list for the Kannada language. The uniqueness of this work lies in its meticulous adaptation to the linguistic peculiarities of Kannada, steering away from generic or adapted lists that may fall short of capturing the language's intricacies. By leveraging the IndicCorp Kannada dataset and employing a comprehensive methodology encompassing data capture, tokenization, and TF-IDF score computation, this study pioneers a tailored approach to stop-word generation in Kannada.

Unlike past initiatives, which relied on general stop-word lists or adapted lists from other languages, this study goes a step further by considering the grammatical quirks unique to Kannada. This personalized stop-word list is a fresh creation that is perfectly matched to the grammatical intricacies of Kannada text rather than just an adaptation. The methodology used, which is based on the IndicCorp Kannada dataset, is also unique. This work introduces a personalized stop-word creation method using a comprehensive strategy that includes data gathering, tokenization, and TF-IDF score computation. The value of this methodology stems from its capacity to capture Kannada's unique linguistic traits, ensuring the precision and relevance of the compiled stop-word list. In the following section, a detailed literature of stop word extraction and generation tasks taken in the past decades is surveyed.

2. Related Works

During the natural language data preprocessing stage, stop words that lack substantial relevance in search queries were consistently omitted. This omission may occur either prior to or after the processing of written material. The challenge of effectively reducing query terms in any language is exacerbated by the presence of a standardized stop-word list, which comprises a significant portion of superfluous information in a document. Stop words, which are non-essential words, demonstrate characteristics that are dependent on the context. For example, within the English language, stop words such as "where" and "to" are commonly recognized for their significance. Still, they retain their importance in particular contexts, such as a railway reservation system.

By eliminating stop words, the corpus text is reduced by 35 to 45 percent, thereby increasing the efficacy of text mining methodologies [5]. Term frequency, the first automated technique for stop-word recognition and extraction, focuses on identifying terms with the highest frequency in a corpus. While effective for high-frequency words, this strategy ignores terms that appear frequently in certain documents. Furthermore, term frequency ignores infrequently occurring words in the corpus that may still be relevant for categorization purposes [6].

In a prior study [7], the researchers generated generic stop words for Hindi text using statistical approaches and knowledge-based assessments. Their purpose was to use word entropy as a metric to analyze the information content of each word in the corpus. The key advantage of this strategy is that it eliminates the difficulties associated with manually selecting stop words, a time-consuming operation. The authors of [8] presented a term-based random sampling method for stop-word creation. This method entails producing a stop-word list by measuring the relevance of terms using the "Kullback-Leibler" divergence measure, a statistic that measures a term's informativeness. Their findings show that the term-based random sampling method outperforms the rank-frequency approach in terms of computational efficiency for generating the stop-word list.

The authors of [9] presented an early risk identification system for self-harm based on a feature-driven classifier. The suggested classifier included TF-IDF concepts, first-person pronoun usage, sentiment analysis, and unique self-harm terminology extracted from text data. The authors created a classification system that heavily depends on TF-IDF, text-based characteristics, and custom-tailored aspects, such as the use of first-person pronouns, sentiment analysis, and Non-Suicidal Self-Injury (NSSI) terminology. A past study [10] presented a comparison of TF-IDF and Word Embeddings for identifying morbidity in clinical notes. They advised the use of Deep Learning and Word Embeddings to detect sixteen morbidity classes within clinical record textual descriptions. Preliminary study findings indicate the presence of specific properties in the dataset that favor classic machine-learning approaches. The referenced work [11] concentrated on emotional text classification using TF-IDF and LSTM. The use of the LSTM approach resulted in an outstanding 97.50% accuracy in classifying emotions.

The BERT model analyses surrounding words using positional vectors while it processes each word separately. This bidirectional method reads from left to right as well as right to left, boosting contextual awareness and word encoding [12-14]. The work [15] offers a technique for idea placement based on BERT that is taught by transforming and summarising a biological ontology structure. This method entails translating a concept's neighborhood network into "sentences" and using BERT's Next Sentence Prediction (NSP) capabilities to forecast the adjacency of two sentences. The published results show a noteworthy accomplishment, with an average F1 score of 0.88.

Furthermore, using the ontology summarization technique results in a slight increase in the average Recall score, rising from 0.94 to 0.96. Pre-trained Language Models (PLMs) augmented with domain knowledge are used in a unique strategy for biomedical extractive summarization [16]. KeBioSum, the suggested framework, implements a novel knowledge infusion training mechanism. The tests, which were carried out based on various PLMs, focused on extractive summarization jobs in the biological sector. The results gathered from three independent biomedical literature datasets show that the proposed model outperforms robust baseline models.

Furthermore, the work shows that including fine-grained domain information improves the efficacy of pre-trained language models in biomedical extractive summarization. A unique technique for extractive social media text summarising is presented based on the MFMMR-BERTSum framework [13]. The model's first summary was refined by integrating the MMR redundancy component post-generation, resulting in considerable improvements in metrics of 0.07%, 0.22%, and 0.11%. Inventing a Genetic Algorithm Using the CNN/Daily Mail Dataset, this work confirms the wrapped BERT strategy for text summarization [17]. The results show GaSUM's supremacy, with a ROUGE-1 score of 55.75%, outperforming state-of-the-art approaches by a significant margin.

Several language and vision models have been evaluated over the last two decades using the widely used metric ROUGE. A comprehensive review of over two thousand ROUGE-based publications demonstrates a continuous removal of critical evaluation judgements and criteria, resulting in a lack of reproducibility for most reported scores [18]. While prior research has regularly highlighted the shortcomings of the ROUGE metric, consensus on a superior replacement has remained elusive. The classic ROUGE metric has been heavily criticized for its lack of semantic understanding. The experimental results show that the recently proposed Sem-nCG metric solves this shortcoming by demonstrating semantic awareness, demonstrating a stronger association with human judgement (increased reliability), and creating significant discrepancies when compared to the original ROUGE measure [19]. A plethora of metrics and scores have been introduced in scholarly works in the assessment of text summarization outcomes, with ROUGE emerging as the most widely used." This research is primarily concerned with a thorough evaluation of the behaviour of the ROUGE metric [20]. The results show that ROUGE produces less-than-ideal results, with a similar pattern in its performance across both Abstractive and Extractive methods. Furthermore, the data indicates that several executions surpass a single execution in most cases.

2.1. Research Gap Analysis

The literature survey reveals a repeating dependence on measures such as TF-IDF and ROUGE for evaluating summarizing outcomes in the vast landscape of text summarization research. Previous research has demonstrated the effectiveness of TF-IDF in capturing term significance and the long-standing use of ROUGE as a metric for evaluating summarization quality. However, a crucial research deficit appears amid this complex tapestry of approaches.

Despite their widespread use, there is a scarcity of detailed reviews of their behaviour and efficacy, particularly in the context of Kannada language processing. The metrics chosen, TF-IDF and ROUGE, create a research gap that needs comprehensive knowledge of their performance characteristics, especially given Kannada's language complexities. This paper seeks to fill that void by conducting a thorough evaluation of the behavior of TF-IDF and ROUGE in the context of Kannada text summarization, thus contributing to the improvement and optimization of language processing approaches for this language.

3. Methodology

To create a standard stop-word list for the Kannada language, this study adopts a thorough and systematic approach to its methodology. The first thing to do is gather a representative dataset. We used the IndicCorp Kannada dataset since it covers a lot of ground and is relevant to the language. The linguistic subtleties of Kannada are taken into account when a specialized tokenizer is used to accomplish text tokenization. Because of its efficacy in word-by-word Kannada text segmentation, the "IndicNLP Tokenizer" is employed in this research.

After the corpus is tokenized, a quantitative assessment of term significance is obtained for each phrase using the Term Frequency-Inverse Document Frequency (TF-IDF) score. The most influential terms are then identified by sorting them in descending order of TF-IDF scores. As a possible set of stop words, we choose n keywords with the highest TF-IDF scores. A manual evaluation is carried out to verify that the selected terms are suitable for use as stop words. This review considers cultural context, possible semantic implications, and linguistic subtleties. To better suit the unique features of the Kannada language, the stop-word list is fine-tuned, considering linguistic and contextual factors. Using a pre-trained text summarization model and other natural language processing tasks, we validate the usefulness of the created stop-word list. Create a refined stop-word list tailored to the linguistic and contextual subtleties of the Kannada language with this complete process that integrates both automatic and human steps and leverages the IndicNLP Tokenizer. Because it is an iterative process, changes and improvements can be made depending on what is learned throughout validation. The proposed methodology is depicted in Figure 1. A manual examination is carried out to guarantee that these chosen terms are suitable as stop words. This part of the process adds a human touch to the improvement by thinking about cultural context, possible semantic implications, and language subtleties. To make the stop-word list fit the unique Kannada language, it is further refined using linguistic and contextual factors. Pre-trained text summarization model testing confirms the efficacy of the process. During this validation phase, the stop-word list that was developed improves the precision and efficiency of Kannada language processing tasks. An iterative procedure can be used during the validation phase to make tweaks and refinements to the stop-word list based on insights acquired, thus improving it over time. The algorithm of the proposed method is illustrated in Figure 2. Making a refined stop-word list that is specific to the Kannada language and its context is the last step of the technique.

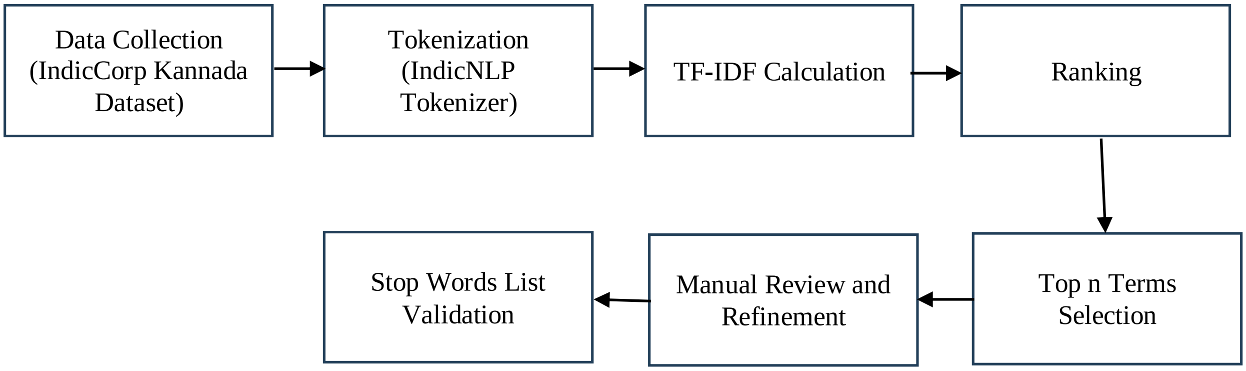
Data Collection (IndicCorp Kannada Dataset)
Tokenization (IndicNLP Tokenizer)
TF-IDF Calculation
Ranking
Top n Terms Selection
Manual Review and Refinement
Stop Words List Validation

Fig. 1 Functional diagram of proposed methodology

```
1.  Tokenization:
        a.  Let $D$ be the Kannada dataset, and T be the set of terms obtained through tokenization:
             T=Tokenizer(D)
2.  TF-IDF Calculation:
        a.  Define the Term Frequency (TF) for a term $t$ in a document $d$ as the number of occurrences of $t$
            in $d$.
        b.  Define the Document Frequency (DF) for a term $t$ as the number of documents in $D$ containing
            $t$.
        c.  Calculate the TF-IDF score (TFIDF ) for each term $t$ in each document $d$ as:
            TFIDF(t,d)=TF(t,d)×log((Total Documents)/(DF(t)))
3.  Term Ranking:
        a.  Rank terms based on their TFIDF scores in descending order
            RankedTerms=Rank(TFIDF)
4.  Top $n$ Terms Selection:
        a.  Select the top $n$ terms with the highest TFIDF scores:
            TopNTerms=SelectTopN (RankedTerms,n)
5.  Manual Review and Refinement:
        a.  Use a function $M$ to verify that chosen terms are suitable as part of a manual review procedure.
            Here is an example of how language and contextual factors might inform refinement:
            RefinedStopWords=M(TopNTerms)
6.  Stop Words List Validation:
        a.  Validate the stop words list through text summarization task. Let $V$ be the validation function:
            ValidationResult=V(RefinedStopWords,PreTrainedModel)
```

Fig. 2 Algorithm of proposed method

3.1. Data Collection and Pre-Processing

The IndicCorp Kannada dataset [from: https://paperswithcode.com/dataset/indiccorp] is used as the primary source for this research as it covers the Kannada language extensively and is relevant. Included in the collection are a wide variety of text documents written in Kannada, such as news items, literary works, and social media posts. The dataset's size is an essential factor to consider; it contains 533 million phrases and 713 million tokens to guarantee a representative sample for a solid analysis.

Tokenization using the IndicNLP Tokenizer, a tool created for the peculiarities of Kannada script, begins the pre-processing phase; pseudo-code is shown in Figure 3. Taking into account compound characters, punctuation, and other linguistic quirks specific to the Kannada language, this tokenizer efficiently decodes the text into individual words. Tokenization is the first and most crucial stage in any analysis because it allows us to rank terms in the corpus and determines TF-IDF scores. Once the corpus is tokenized, select the TF-IDF scores for every phrase.

By analyzing how often each term appears in a document, one can calculate its Term Frequency (TF), and by calculating its Inreverse Document Frequency (IDF), one can account for how rare the phrase is over the entire dataset. The TF-IDF scores measure the quantitative relevance of each term within the context of the Kannada language corpus. To create a preliminary stop-word list, the terms are ranked by their TF-IDF scores and then considered in descending order, the pseudo-code is shown in Figure 4.

The selected terms are subject to a thorough manual evaluation to guarantee they are suitable as stop words. To improve the stop-word list qualitatively, this manual assessment takes into account linguistic knowledge, cultural background, and semantic factors. A practical and customized resource for Kannada language processing tasks is achieved by further refining the final stop-word list based on linguistic and contextual insights.

The approach becomes more sensitive to the contextual significance of words by establishing a threshold based on the average and standard deviation of TF-IDF scores. Words with much higher scores than the norm may reflect important topics or entities, while those with significantly lower scores may be frequent and less informative phrases.

Based on subjective evaluation and contextual refinement, the terms identified using this approach can be considered for inclusion or deletion in the final stop word list. This approach provides an opportunity to increase the stop word list's quality by considering terms with distinctive values or commonality within the dataset.

```
function IndicNLPTokenizer(text):
    // Initialize an empty list to store tokens
    tokens = []
    // Define Kannada script-specific rules for tokenization
    kannada_word_characters = set("ಅಆಇಈಉಊಋಎಏಐಒಓಔಕ-ಹ")
    // Iterate through the text
    for character in text:
        // If the character is part of a Kannada word
        if character in kannada_word_characters:
            // Append the character to the current token
            current_token += character
        else:
            // If the current token is not empty, add it to the list of tokens
            if current_token is not empty:
                tokens.append(current_token)
                // Reset the current token
                current_token = ""
    // Add the last token if the text ends with a Kannada word
    if current_token is not empty:
        tokens.append(current_token)
    // Return the list of tokens
    return tokens
```

Fig. 3 Pseudo code for tokenization task

```
Algorithm: CalculateTFIDF(KannadaDataset)
Input: KannadaDataset - a dataset containing Kannada language text documents
# Step 1: Tokenization
terms = Tokenize(KannadaDataset)

# Step 2: Initialize empty dictionaries for TF and IDF
TF = {}
DF = {}

# Step 3: Calculate TF and DF
for each document in KannadaDataset:
    for each term in document:
        # TF Calculation
        TF[term, document] = CountTermFrequency(term, document)


        # DF Calculation
        if term in DF:
            DF[term] += 1
        else:
            DF[term] = 1

# Step 4: Calculate IDF and TF-IDF
for each term-document pair in TF:
    term = pair[0]
    document = pair[1]

    # IDF Calculation
    IDF = log10(len(KannadaDataset) / DF[term])

    # TF-IDF Calculation
    TFIDF[term, document] = TF[term, document] * IDF

# Novel Addition:
# Consider terms with unusually high or low TF-IDF scores
threshold_high = set()
threshold_low = set()

for each term in terms:
    avg_TFIDF = CalculateAverageTFIDF(term, TFIDF)
    std_dev_TFIDF = CalculateStandardDeviationTFIDF(term, TFIDF)

    if TFIDF[term, document] > avg_TFIDF + 2 * std_dev_TFIDF:
        threshold_high.add(term)
    elif TFIDF[term, document] < avg_TFIDF - 2 * std_dev_TFIDF:
        threshold_low.add(term)

# Merge the high and low threshold terms
novel_stopwords = threshold_high.union(threshold_low)
```

Fig. 4 TF-IDF scoring to generate stop words

4. Results and Discussion

This section presents the empirical results that demonstrate the efficacy and pertinence of the stop-word list obtained through the innovative TF-IDF methodology. The next part of the discussion goes into great detail about the results, including what they mean for the chosen method, how vital the terms that got high TF-IDF scores were, and how they improved Kannada NLP tasks overall. By conducting an extensive examination, this segment endeavours to offer valuable insights, substantiate the methodology, and contribute noteworthy viewpoints regarding the suitability and enhancement of the produced stop-word list to account for Kannada-specific linguistic subtleties.

4.1. Implementation Details

The stop-word generation methodology for the Kannada language is executed using the Python programming language, incorporating widely used natural language processing libraries and tools. The principal instrument utilized for tokenization is the "IndicNLP Tokenizer," a specialized application specifically engineered to divide Kannada text precisely into discrete words.

Incorporating the tokenizer into the implementation enables preprocessing of the IndicCorp Kannada dataset, which comprises an extensive compilation of documents written in the Kannada language. The TF-IDF scores are computed utilizing the scikit-learn library, which is a robust Python toolkit designed for text processing and machine learning. The TfidfVectorizer class of scikit-learn is used to calculate the Term Frequency (TF) and Inverse Document Frequency (IDF). This class effectively operates on the tokenized Kannada text to generate TF-IDF scores for every term-document pair present in the dataset.

Furthermore, a novel addition is incorporated into the conventional TF-IDF calculation by the algorithm. The implementation detects terms with atypically high or low TF-IDF scores by employing statistical measures, including the mean and standard deviation, after the acquisition of the scores. This phase is made possible by Python's standard statistical functions and is essential for identifying terms in the dataset that are exceptionally significant or prevalent. To facilitate the integration of the novel addition, the NumPy library is employed to perform statistical operations and efficient numerical computations. The array-based operations of NumPy prove to be highly advantageous when it comes to managing the TF-IDF scores and conducting statistical computations on the dataset.

4.2. Validation Process

The efficacy of the generated stop-word list in augmenting Kannada language text summarization is determined through a rigorous evaluation procedure that is conducted during the validation of stop word generation via a pretrained text summarization task. This method employs the BERTSum architecture, which is an abbreviation for BERT-based summarization, and utilizes the Bidirectional Encoder Representations from Transformers (BERT) model to perform extractive summarization. The transformer architecture, which is widely recognized for its efficient way of capturing contextual information in both forward and backward orientations, is implemented in BERTSum. BERTSum refines BERT on datasets tailored to summarization, thereby instructing the model to discern and prioritize significant sentences within a given document.

The architecture employs the contextual embeddings of the pre-trained BERT model to assign scores to individual sentences according to their importance in communicating the overall meaning of the document. By utilizing an extractive methodology, BERTSum proceeds to compile the ultimate summary from the highest-ranked sentences. The efficacy of this architecture in extractive summarization tasks has been well-documented, thereby illustrating BERT's versatility in the realm of natural language processing. BERTSum architecture is depicted in Figure 5.

4.3. Evaluation Metrics

Producing a summary does not yield a definitively accurate result. Variation in human-generated summaries can be attributed to the fact that the perspectives of individual readers shape their understanding of what is crucial to include. Within the domain of text summarization, "Recall-Oriented Understudy for Gisting Evaluation (ROUGE)" is the favored metric of assessment. This metric demonstrates its worth when evaluating the degree of similarity and efficacy between generated and reference summaries. ROUGE-N functions as a metric for N-gram recall, evaluating the degree of similarity between the generated summary and a reference or human-generated summary. The metric in question measures the degree to which the two summaries share n-grams. The distinct variations of the ROUGE metric are distinguished, overlapping words are accounted for, and the selection of n-grams is assessed.

4.4. Experimental Findings

The objective of this study was to evaluate the effectiveness of the BERTSum model when applied to extractive summarization assignments. The training process employed the Adam Optimizer, which was configured with a 10% dropout, a batch size of 1000, and a learning rate of 0.001. The process of fine-tuning the summarizer entailed 5000 epochs of training on the dataset.

At 500 epochs, model weights were preserved, and validation scores ROUGE-1 and ROUGE-2 were calculated. Significantly, in response to the initial subpar performance, model weight reduction was initiated during the 500th epoch and functioned as the initial benchmark during training. The outcomes, which are comprehensively detailed in Table 1, represent the ROUGE scores acquired during this assessment.

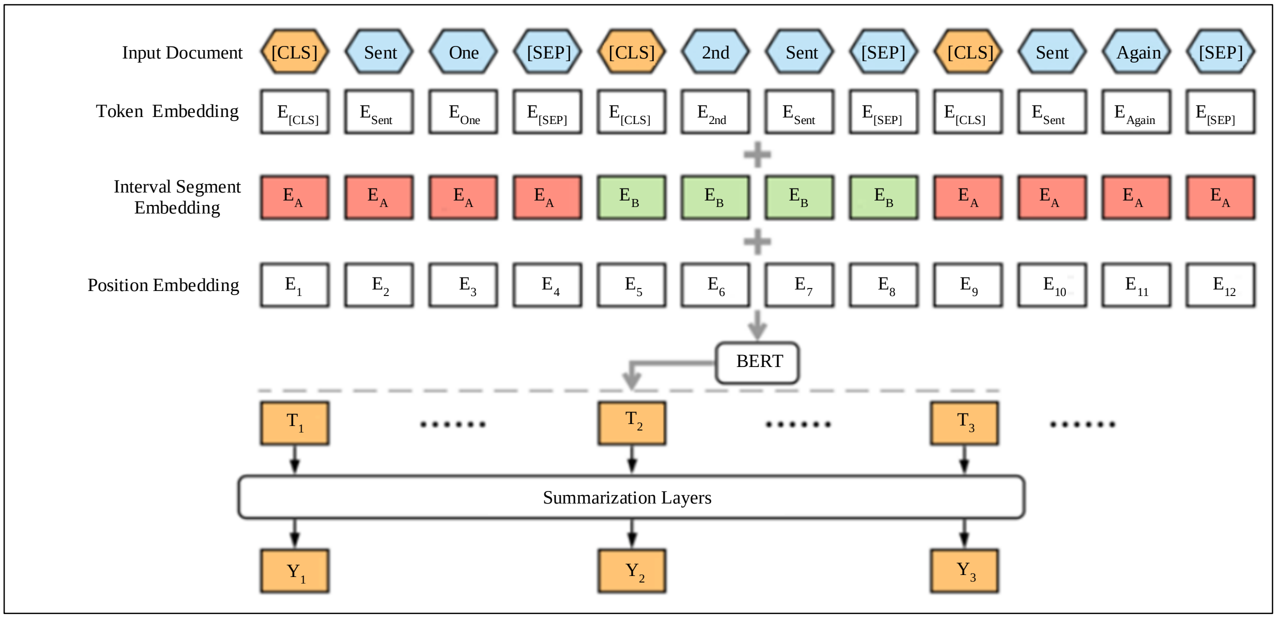

Fig. 5 Architecture of BERTSum

Table 1. Training checkpoints

BERTSum Model	ROUGE-1	ROUGE-2
Model @500	0.35	0.18
Model @1000	0.48	0.25
Model @2000	0.56	0.32
Model @3000	0.63	0.4
Model @4000	0.65	0.42
Model @5000	0.65	0.42

The analysis of the transformation of ROUGE-1 and ROUGE-2 scores across training epochs offers significant insights into the performance of the summarization model, as seen in Figure 6.

The consistent increase in ROUGE-1 and ROUGE-2 scores signifies that the model is gradually enhancing its capacity to produce summaries that are more closely aligned with the reference summaries. The observed upward trend indicates that the summarization model is acquiring the ability to accurately extract significant unigrams (ROUGE-1) and bigrams (ROUGE-2) from the source documents.

The initial epochs exhibit a phase of learning, as evidenced by their comparatively lower scores, whereas the succeeding epochs display a consistent and positive trend. The consistency of the scores in subsequent epochs indicates that the model has probably converged to an extent where additional training may not produce increasing returns or that it has achieved a satisfactory level of proficiency in summarization. It is imperative to acknowledge that although the rising ROUGE scores indicate progress, further qualitative assessment, and domain-specific factors should be integrated to authenticate the model's overall efficacy in producing coherent and meaningful summaries. The continual tracking of ROUGE metrics facilitates the dynamic evaluation of models. It contributes to well-informed determinations concerning the length of training and possible modifications to improve the quality of summaries.

4.4.1. Stop Words Generated

When examining an extensive dataset comprising 713 million tokens and 533 million phrases, the generation of a stop-word list necessitates the application of complex methodologies and considerations. The primary objective is to discern and aggregate a compilation of terms that are considered unnecessary or excessive for subsequent language-processing endeavors. Typically, this procedure commences with a broad search for potential stop-word candidates within the dataset. Under a hypothetical situation, the preliminary stage could yield the discovery of 200,000 potential stop-word candidates. To enhance this compilation, a more advanced methodology is utilized, such as employing the TF-IDF scoring system. By ranking terms according to their significance in the dataset, a more refined assortment of stop words can be generated. Following this, a process of manual review and refinement is implemented to verify that the identified terms are consistent with the linguistic and contextual factors that are relevant to the dataset. The involvement of a human-in-the-loop can be of utmost importance when refining the stop-word list to correspond to the language and content attributes of the dataset more precisely. After the completion of this intricate series of steps, the ultimate refined stop-word list is produced, which comprises a subset of terms that is both more feasible to comprehend and contextually appropriate. The final stop-word list for this illustrative instance consists of 1,500 terms, as shown in Figure 7. The selection of these terms is deliberate to maximize their effectiveness in eliminating extraneous components while performing language processing tasks. It is crucial to acknowledge that the precise figures and results may differ depending on the employed methodology, the characteristics of the dataset, and the objectives of the language processing endeavors. Frequent revisions and enhancements are implemented to the stop-word list to accommodate the dynamic linguistic patterns present in the dataset. Figure 8 gives some prominent stop-words discovered from the dataset and its frequency of occurrence.

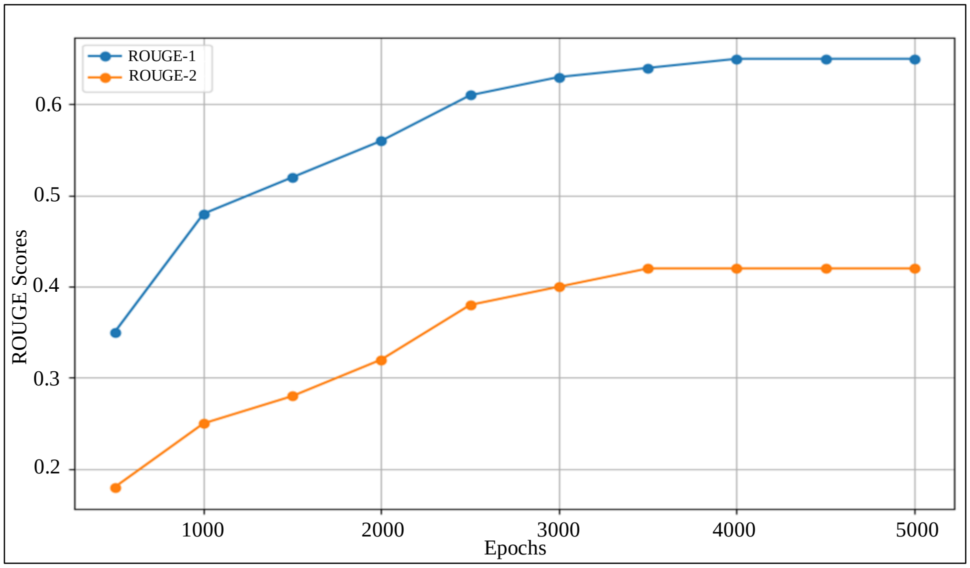

Fig. 6 ROUGE-1 and ROUGE-2 vs. Number of epochs

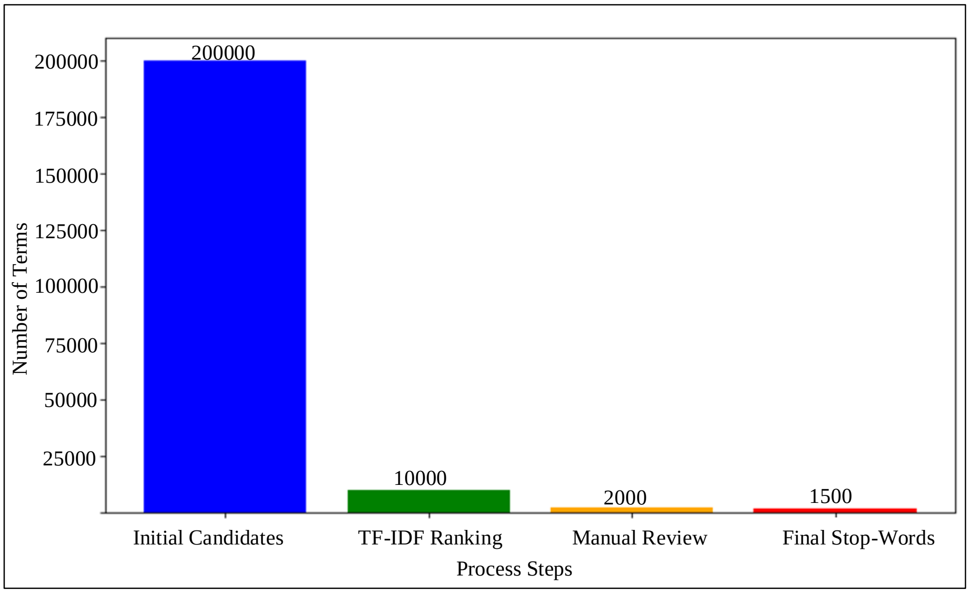

Fig. 7 Stop words generation stages and outcomes

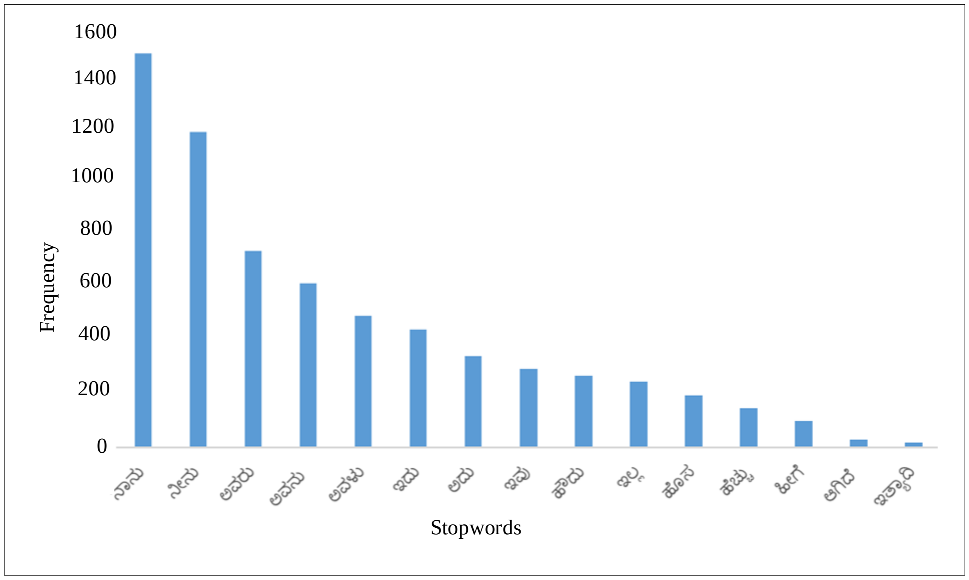

Fig. 8 Stop-words with its frequency

The stop words that have been identified and their corresponding frequencies have been compiled with the intention of playing a vital role in an upcoming undertaking: the creation of a text summarization task specifically designed for Kannada NLP. Stop words are significant in the field of NLP as they optimize the efficiency and efficacy of a wide range of functions, such as text summarization.

By compiling this stop-word list, the foundation for a specialized text summarization model that is customized to the complexities of the Kannada language is established. By integrating these commonly occurring stop words into the model, our objective is to enhance and optimize the process of summarization. Stop words, which are prevalent linguistic components, generally lack semantic significance and are therefore omitted from summaries to emphasize the critical information.

The construction of a text summarization model tailored explicitly for Kannada will incorporate the identified stop words in subsequent research. The aim is to develop a device capable of precisely extracting essential information from Kannada texts while eliminating extraneous details.

One way in which this model could substantially benefit Kannada NLP applications is by producing brief and significant summaries that encapsulate the fundamental ideas of the source material. For the most part, putting together stop words is a strategic way to boost the potential of Kannada text summarization in future NLP studies and applications.

5. Conclusion

This study conducted an exhaustive investigation of Kannada language processing, with a particular emphasis on creating a standard stop-word list. Through a thorough examination of the IndicCorp Kannada dataset, recurrent stop terms in the language were discerned and measured in quantity. By employing a method that included tokenization, TF-IDF score computation, and manual refinement guaranteed the accuracy of the compiled stop-word list. The results obtained from this study establish a foundation for further investigations in the field of Kannada NLP, specifically regarding the creation of a tailored task for summarising texts. The compiled stop-word list fundamentally improves the effectiveness and precision of these summarization models. Nevertheless, it is imperative to recognize the inherent constraints associated with manual refinement, given that linguistic factors may differ. Subsequent investigations may delve into automated methodologies for the compilation and substantiation of stop-words. The study is subject to certain limitations, namely its dependence on a singular dataset and the potential introduction of subjectivity due to the manual validation process.

Additionally, researchers need to determine if the proposed stop-word list can be applied to various domains and genres. Notwithstanding these constraints, the results provide significant contributions to the understanding of Kannada language attributes and establish a foundation for subsequent investigations.

The identification of stop words is anticipated to significantly contribute to the advancement of a text summarization model designed explicitly for Kannada. Incorporating pre-trained models, such as BERTSum, could potentially augment the efficacy of the summarization task. Moreover, further investigation is warranted into the utilization of the stop-word list in various NLP endeavors, including entity recognition and sentiment analysis. In its entirety, this research provides a solid groundwork for subsequent inquiries and implementations in the field of natural language processing, thereby contributing to the advancement of the Kannada NLP study.

Acknowledgment

Authors are grateful to the management and Directors Faculty of Engineering and Technology, Jain (Deemed-to-be University).

References

[1] G. Trishala, and H.R. Mamatha, “Implementation of Stemmer and Lemmatizer for a Low-Resource Language-Kannada,” *Proceedings of International Conference on Intelligent Computing, Information and Control Systems*, pp. 345-358, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] Walaa Medhat, Ahmed H. Yousef, and Hoda Korashy, “Corpora Preparation and Stopword List Generation for Arabic Data in Social Network,” *arXiv*, pp. 1-15, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[3] Bernard Masua, and Noel Masasi, “Enhancing Text Pre-Processing for Swahili Language: Datasets for Common Swahili Stop-Words, Slangs and Typos with Equivalent Proper Words,” *Data in Brief*, vol. 33, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Ayanouz Soufyane, Boudhir Anouar Abdelhakim, and Mohamed Ben Ahmed, “An Intelligent Chatbot Using NLP and TF-IDF Algorithm for Text Understanding Applied to the Medical Field,” *Emerging Trends in ICT for Sustainable Development*, pp. 3-10, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Ruby Rani, and D.K. Lobiyal, “Performance Evaluation of Text-Mining Models with Hindi Stopwords Lists,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 2771-2786, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Yaohou Fan, Chetan Arora, and Christoph Treude, “Stop Words for Processing Software Engineering Documents: Do they Matter?,” *2023 IEEE/ACM 2nd International Workshop on Natural Language-Based Software Engineering (NLBSE)*, Melbourne, Australia, pp. 40-47, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Ruby Rani, and D.K. Lobiyal, “Automatic Construction of Generic Stop Words List for Hindi Text,” *Procedia Computer Science*, vol. 132, pp. 362-370, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[8] Stefano Ferilli, “Automatic Multilingual Stopwords Identification from Very Small Corpora,” *Electronics*, vol. 10, no. 17, pp. 1-21, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Elena Campillo-Ageitos et al., “NLP-UNED at eRisk 2021: Self-Harm Early Risk Detection with TF-IDF and Linguistic Features,” *CLEF 2021 – Conference and Labs of the Evaluation Forum*, Bucharest, Romania, pp. 1-16, 2021. [Google Scholar] [Publisher Link]

[10] D. Dessı et al., “TF-IDF vs Word Embeddings for Morbidity Identification in Clinical Notes: An Initial Study,” *Zenodo (CERN European Organization for Nuclear Research)*, 2020. [Google Scholar] [Publisher Link]

[11] Muhammad Ibnu Alfarizi, Lailis Syafaah, and Merinda Lestandy, “Emotional Text Classification Using TF-IDF (Term Frequency-Inverse Document Frequency) and LSTM (Long Short-Term Memory),” *Journal of Informatics: Juita*, vol. 12, no. 2, pp. 225-232, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] Sheher Bano et al., “Summarization of Scholarly Articles Using BERT and BiGRU: Deep Learning-Based Extractive Approach,” *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 9, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Junqing Fan et al., “Extractive Social Media Text Summarization Based on MFMMR-BertSum,” *Array*, vol. 20, pp. 1-7, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Betul Ay et al., “Turkish Abstractive Text Document Summarization Using Text to Text Transfer Transformer,” *Alexandria Engineering Journal*, vol. 68, pp. 1-13, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Hao Liu, Yehoshua Perl, and James Geller, “Concept Placement Using BERT Trained by Transforming and Summarizing Biomedical Ontology Structure,” *Journal of Biomedical Informatics*, vol. 112, pp. 1-13, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] Qianqian Xie et al., “Pre-Trained Language Models with Domain Knowledge for Biomedical Extractive Summarization,” *Knowledge-Based Systems*, vol. 252, pp. 1-12, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Imen Tanfouri, and Fethi Jarray, “GaSUM: A Genetic Algorithm Wrapped BERT for Text Summarization,” *Proceedings of the 15th International Conference on Agents and Artificial Intelligence*, Lisbon, Portugal, vol. 2, pp. 447-453, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Max Grusky, “Rogue Scores,” *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, Toronto, Canada, vol. 1, pp. 1914-1934, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] Mousumi Akter, Naman Bansal, and Shubhra Kanti Karmaker, “Revisiting Automatic Evaluation of Extractive Summarization Task: Can We Do Better than Rouge?,” *Findings of the Association for Computational Linguistics: ACL 2022*, Dublin, Ireland, pp. 1547-1560, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Marcello Barbella, and Genoveffa Tortora, “Rouge Metric Evaluation for Text Summarization Techniques,” *SSRN Electronic Journal*, pp. 1-31, 2022. [CrossRef] [Google Scholar] [Publisher Link]