ABSTRACT

Sentiment analysis involves identifying and classifying the emotions conveyed within a text source. Due to the drastic usage of the internet, people are sharing their feelings on social media in their regional languages other than English. Sentiment Analysis of Indian languages is difficult due to their complex morphology and the scarcity of available annotated datasets for languages like Hindi, Bengali, Malayalam Telugu, Tamil, etc. There has been little published microscopic research on Telugu languages due to a lack of labelled datasets. People are usually interested in reading both positive and negative reviews, which include what customers don't like and what they do like about a service or product. In opinion mining, the aspects or features of the item or service are very important. In this paper, we created a Telugu corpus covering a wide range of topics. We focus on studying feature extractors because they are so important to how well classification works. The most important part of opinion classification is getting the right traits, because if they aren't, classification won't work as well. In our survey of the literature, we discovered that few pieces cover more than two feature extractors at most. To that end, we've included detailed discussions of, and FastText, Glove, Word2Vec and TF-IDF vectorization, four of the most widely used feature extractors, in the present work. This study will survey the landscape of opinion mining feature extractors currently available. The objective of this study is to ascertain the Bidirectional Long Short-Term Memory (BiLSTM) configuration that maximizes both accuracy and f1 score across all extractors. At long last, the BiLSTM+ FasTest extractor proved superior.

Keywords: BiLSTM Algorithm, FastTest, Sentiment analysis, Telugu Language, Word Embedding's.

1. Introduction

Sentiment analysis, often known as "opinion extraction," is the research into and interpretation of people's subjective sentiments[1]. Despite the fact that SA is mostly investigated in the context of NLP, It has a number of applications for data mining, including the extraction of information, data summarization, question and answer systems, and recommendation systems[2]. The quick expansion of this market may be linked back to the popularity of online social media such as product evaluations, blogs, online assessments, and other similar activities [3]. The advent of social media platforms has facilitated the exchange of knowledge and personal experiences among users. The proliferation of social networks has led to a significant increase in the production and dissemination of digital material. The utilization of thoughts, articles, and blogs shared on online platforms, such as tweets, has emerged as a valuable resource for firms seeking to extract pertinent information from the data and subsequently enhance their business operations. Commercial businesses must analyses and interpret these sentiments in order to acquire data and make money. Time-consuming and requiring a lot of human effort is the extraction of complicated features, selection of the essential features, and pattern extraction from enormous amounts of data. Deep Learning models exhibit remarkable performance when employed in conjunction with Natural Language Processing techniques for the aim of doing sentiment analysis on a vast dataset. The objective is to classify the concepts and sentiments conveyed by users. Currently, there exist several methodologies for sentiment analysis and ensemble models that can be utilized to aggregate data from multiple features.

The sentiment analysis methodology is employed to classify the testimonies into two distinct categories: positive and negative [4Sentiment analysis (SA) comprises the capability to analyze the gathered information, which may encompass subjective viewpoints, and has the capacity to discern certain fundamental attributes. This, in turn, facilitates decision-making for other users [5].

Sentiment analysis can be conducted using three distinct approaches: sentence-level analysis, document-level analysis, and aspect-level analysis. The objective of sentence-level assessment is to analyze the polarity value of an individual sentence within a given source text. Document-level analysis involves evaluating the overall polarity value of the entire document. Aspect-level analysis is a methodology employed to ascertain the polarity of individual aspects inside a certain textual examination.

Reviews are submitted by individuals on social media platforms in both the English language and their respective native languages. As a result, sentiment analysis was conducted on both English and other language textual data. The complexity of texts written in Indian languages necessitates the development of a dependable approach for conducting

sentiment analysis [6]. The classification of the positive and negative emotions connected to various emotions found in texts, such as joy, rage, and grief, has been done using a number of techniques [7]. India is a multilingual country; users always post their comments in their regional language. Sentiment classification for regional languages is difficult task. Here we have chosen. Telugu language for sentiment classification. The Indian states of Andhra Pradesh and Telangana give Telugu a unique status as their standard language. The mother languages of Telugu speakers include a wide range of dialects. Because of its perceived standardization and accessibility to the rest of the Telugu speaking population, the Krishna and Godavari dialects are adopted by the majority of Telugu print, journalism, and electronic media. (Krishnamurthi, 1961)

Telugu is ranked as the sixteenth most widely spoken language globally, as indicated by the Ethnologue list [8]. Telugu, a regional language spoken in India, boasts a substantial number of native speakers, believed to be at 95.7 million. Among Indian languages, Telugu ranks second in terms of regional data provided by users who communicate in that language, with Hindi being the only language that surpasses it. The determination of feelings in the Telugu language presents a significant challenge, primarily due to lack of annotated Telugu materials and the inherent morphological complexity of the language. Compared to the English language, there has been relatively less research conducted on the Telugu language, primarily due to the scarcity of available resources and techniques. Currently, deep learning methodologies such as Convolutional Neural Networks and Long Short-Term Memory have replaced simple machine learning models and dictionary-based methods due to their superior effectiveness. Here we planned to use BiLSTM with different word embedding's, word embedding's play a crucial role in sentiment analysis by representing words as dense vectors in a continuous space. The vector representations presented herein encapsulate both semantic and syntactic links among words, so enabling machine learning models to enhance their comprehension and analysis of sentiment.

In the present study, we used different word embedding's like Tf-Idf, wordtovec, glove and Fastest. FastText is an refinement of Word2Vec that also considers character n-grams. It is known for its speed and efficiency in both training and inference. FastText embedding's capture sub word information, enabling enhanced handling of out-of-vocabulary words compared to conventional word embedding's. [9].

The Long Short-Term Memory model is an enhanced iteration of the conventional recurrent neural network [5]. The LSTM architecture was developed with the objective of enhancing its performance by evaluating and determining the relevant information to retain while discarding unnecessary elements [6]. The memory capacity of recurrent neural networks is enhanced in order to construct LSTM models [7]. RNNs encounter challenges related to the amplification or vanishing of gradients, which can impede their training. LSTM networks address this issue by offering a solution. The LSTM model retains the error signal to facilitate its propagation upwards through the hierarchical structure. This enables the error to be utilized for enhancing learning in subsequent stages of the training process. The LSTM technique has been created to acquire knowledge on long-distance correlations within sequential data. Special cells are responsible for storing long-term semantic information regarding dependencies. The process of integrating and determining the retention or elimination of information is a crucial aspect of the LSTM unit, which comprises the output, forget gate, and input gates. The LSTM unit regulates the flow of provided data and determines the allocation of read, write, and delete privileges. In the context of employing bi-directional LSTM, the forward operation network The data is processed in a sequential manner, with a forward direction being followed, as well as a backward direction. operating network processes data in a backward direction. The user's text is already academic and does not require any rewriting [8].

To yet, there is a lack of documented microscopic investigations on Indian languages due to the absence of an annotated dataset that would substantiate such assertions. The significance of computerized sentiment analysis lies in its ability to enhance the importance of natural language processing by capturing the writer's intentions and the emotional content of a message within its contextual framework. Non-English texts encounter additional difficulties due to the absence of a labelled corpus in the Telugu language.

The following is a summary of this paper's contributions:

- A Telugu language corpus called "SentiKanna" is being developed using tweets from multiple categories, including movies, recipes, and tourism.
- Discussed the pre-processing stages in great detail.
- In this study, we examine the impact of FastText embedding on the efficacy of the BiLSTM model in the context of sentiment categorization.
- We present the BiLSTM model for the categorization of sentiments utilizing a number of different word embedding's, including TF/IDF, WordToVec, Glove, and FastText embedding.
- We examine how FastText embedding affects the BiLSTM model's ability to classify sentiment.

Throughout this article, we'll discuss about the approaches based on the principle of deep Learning that could be deployed on the dataset and to analyze the category of the sentiment. To initiate with, preprocessing of the text is done followed by the classification models. In the next paragraphs, we'll go over each section individually. Section 2 focuses on a literature review. Section 3 details the suggested approach. Experiment setup and procedures are covered in Section 4. Section 5 concludes with a few suggestions for further investigation.

2. Literature survey

2.1. Sentiment Analysis in Telugu Language

Researchers from every corner of the world are developing sentiment analysis technologies in their native tongues. However, almost all of the research was done only in English. Few resources or methods are available for rapidly building robust sentiment classifiers in other languages. The obvious question is whether or if existing tools and methods for English sentiment analysis, along with automatic machine translation, can be used to construct sentiment classification systems in other languages. Sentiment classification in Telugu social media posts poses several challenges due to the language's complex morphology and limited research in this area. Unlike English, Telugu sentiment analysis has received relatively less attention, primarily due to the scarcity of resources and tools.

Naidu et al. [10] conducted a study in 2017, using Telugu SentiWordNet for subjectivity classification and achieved an accuracy of 70%. In this article, it is discovered that certain unigram phrases in the already existing Telugu SWNet contain ambiguous meanings and do not provide enough information for sentiment analysis [11]. In situations like these, trigram and bigram words have been utilized and thus the ambiguity issue that arised when attempting to predict sentiment has been rectified. As a result, the authors suggested a model for developing Telugu SentiPhraseNet (SPNet), which is used for Telugu sentiment analysis. SPNet, which fixed the issues with the current SWNet, was built using data gathered from a variety of channels, which include NLTK Indian Telugu, Telugu e-newspapers and Twitter. Naive Bayes (NB), SVM, Multilayer Perceptron Neural Networks (MPNs) and Logistic Regression, were used to aquire insights and compared to the SPNet proposed by the researchers. In 2018, Charan and Radhika [12] built a sentiment analysis database for Telugu using annotated Telugu news sentences and established standard guidelines for review annotation. They applied word2vec and machine learning models to achieve a 76% accuracy in binary sentiment analysis.

A further investigation endeavors to utilize word-level sentiment annotations in order to construct a meticulously annotated corpus, with the objective of enhancing sentiment analysis work that can be done in Telugu [13]. The methodology for polarity annotation is thoroughly examined by the authors, who validate the resource by implementing several models including Linear SVM, KNN, and RF. This research was designed to produce a benchmark corpus with high precision, serving as an expansion to SentiWordNet. Additionally, it aims to establish a baseline accuracy for a model [13] that predicts sentiment by utilizing lexicon annotations. The main emphasis of this work revolves around machine learning techniques and the utilization of annotations of word-level emotion for the purpose of in the field of emotion recognition automation. Moreover, the precision of the target corpus can be enhanced by the process of annotating its bi-grams.

The hybrid methodology utilized by the author [14] involved the integration of a lexical analysis with machine learning techniques, achieving an 85% accuracy using the Naive Bayes model. A rule-based approach was employed by Pravarsha Jonnalagadda [15] to build a system for evaluating sentiment in Telugu language using Telugu SentiWordNet. At the outset, annotated corpora were utilized to generate datasets for the objective of doing sentiment categorization in the Telugu language. Subsequently, a Parts of Speech tagger is employed to partition the phrases into their constituent PoS can remove any redundant stop words. In the study, the researchers employed sentiwordnet to classify emotions, while textblob was utilized to perform the same task for words that were not before seen. The utilisation of a rule-based approach has boosted Productivity in a Huge Way of natural language processing in the Telugu language. Nevertheless, this approach has not yielded comprehensive accuracy in determining the emotion of entire sentences.

The author [16] introduced the sentiphrasenet approach for analysing the emotional tone of Telugu text. They collected annotated corpus-based datasets, employed a PoS tagger, eliminated stop words that aren't needed, and used rules for bigrams and trigrams to find sentences. Textblob was utilized for emotion categorization. In [17], the objective of the work is to classify the sentiments into either positive or negative. When we feed a trained model an input, it categorizes it into one of these groups. TF-IDF and DOC2VEC are two methods for converting text into numerical vectors for text pre-processing. Tf–Idf is used for the preliminary processing of text and doc2vec are implemented separately, and advanced ensemble techniques are compared before proposing an improved model that combines different techniques. The RF, stacking and ADA-boosting algorithms have been used, and good results are obtained. Authors [18] used the Telugu language for testing the model. Root words were manually tagged to the polarity in BootCat, which was used to create the corpus for this study. Several key rules were developed by the researchers in order to accurately determine a sentence's mood. The researches [19] developed an aspect-based sentiment evolution for Telugu movie reviews. They collected data from online movie review websites, annotated the corpus with useful information for phrase detection and polarity labeling, and employed deep learning techniques. Authors [20] proposed a model which is working based on Bi-directional Recurrent Neural Networks for analyzing audience reactions to Telugu films. They gathered tweets on Telugu films, fixed grammatical errors, and utilized a morphological analyzer for vector representations. The primary objective of the author [21] was the identification of sarcastic expressions within Telugu conversational sentences. The researchers employed hyperbolic features, including interjections, intensifiers, questions, and exclamation marks, in order to classify the tagged data into several sarcastic phrase patterns.This research study demonstrates how to do opinion research using Telugu [22]. The researchers were able to identify subjective statements in the Telugu corpus by employing a technique known as Telugu SentiWordNet, which is built on a Lexicon. Second, they employed a number of machine learning techniques, including SVMs, NNs, and RFs to classify the feelings conveyed in the corpus. A maximum accuracy of 85 percent has been obtained.

2.2. Here we can see the some of the papers published with following word embedding's

Word representation is a crucial aspect in the process of text classification using deep learning models. Word embedding is a technique applied to the purpose of expressing textual data. In contrast to TF-IDF, word embedding uses a dense vector to represent the word. According to [23], a machine learning system is more likely to pick a dense vector as features than a sparse vector. Further benefit highlighted by the author [23] is that the compact vector representation mitigates the issue of overfitting and effectively encompasses synonymous terms. The authors conducted an analysis on the impact of TF/IDF feature level on the SS-Tweet dataset in order to extract opinions [24]. The researchers discovered that utilizing the extraction of TF-IDF features resulted in a sentiment evolution performance that was 3-4% superior compared to utilizing the N-gram feature. In their publication, the writers (25) provide a novel Word2vec model that incorporates further linguistic attributes in order to effectively handle limited Chinese datasets. The extensive dataset is being compared to the pattern derived from Internet content. The available empirical evidence suggests that the approach we have developed holds promise in improving the accuracy of opinion

classification in six distinct categories on the Weibo platform. The word2vec pipeline was enhanced by the modifications proposed in Reference [26], resulting in an improvement in the quality of the generated word vectors. A large dataset is used in the work [27] for crucial sentence retrieval and categorization using a hybrid approach that combines the mixture of contextual analysis, string similarity measurements, and embedding glove words. The empirical results show that when it comes to measuring the similarity of key sentences and strings in large datasets, the Glove extraction pattern outperforms the current measures.The investigation additionally conducted a comparative analysis of the performance of Glove embedding and FastText embedding on the Word Analogy, Rare Words, and Squad datasets.In [28], the writers analyze the FastText extractor, and their experiments reveal that it can get an area under the ROC curve of 0.97, an F-measure of 94.2 percent, and a CPU inference time of 74.8 milliseconds The FastText embedding is extensively utilised as a word embedding in several applications.

2.2.3 We take a look at a selection of the studies that have been recently published in the field of deep learning that make use of word embedding's in several foreign languages.

Technology advancements have led to extensive utilization of deep learning methodologies in order to achieve sentiment classification. With the rise of more complicated deep learning methodologies came the suggestion of a simpler BiLSTM model for sentiment categorization [29]. These models' performance is evaluated next to that of more intricately designed rivals. CNN and RNN were integrated in the work cited in [30], which successfully captured persistent reliance and yielded favorable two dataset's worth of results. In order to conduct sentiment analysis, the authors [31] integrated word2vec with CNN. Word2vec feeds text to CNN after it has been transformed to a vector. This research proves that CNN plus word2vec outperforms the RNN technique. To get samples from the convolutional matrix, the authors of [31] use three convolutional layers in their model design. Since CNN can gather data from around the world, the author [32] relied on it for his Twitter sentiment analysis. This study demonstrates that when it comes to the classification of Twitter sentiments, CNN can achieve better results than SVM and Naive Bayes. In order to classify data, the model [33] employs n-gram feature extraction and a BiLSTM. This research shows that our suggested model outperforms both LSTM and BiLSTM. An enhanced BiLSTM-CNN model for emotion recognition was proposed in reference [34]. By taking into account the connection between features, this model differs from the standard BiLSTM-CNN. Subsequently, the performance of the models was compared to that of BiLSTM and Bi-LSTM-CNN, where it was shown to be superior. In [35], they estimate the Arabic tweet sentiment estimation using a CNN and LSTM models ensembling. Their findings demonstrate a notable improvement in model performance, with an F1-score of 64.46 percent being attained. In [36], The analysis of Arabic tweets encompassed five distinct designs, including CNN, RNN, LSTM, stacking LSTM, and combination LSTM. To facilitate the training of the models, a combination of dynamic and static Continuous CBOW and SG word embedding's were employed. Based on the experimental findings, the mixed LSTM model trained with dynamic CBOW performed the best. A DL model [37] for analyzing the sentiment that was suggested that this be conveyed at the aspect level. Text embedding's and aspect embeddings are both used as inputs to the model, which is built on an LSTM network. The suggested model makes use of both types of embedding's. The FastText + CNN model achieves the best accuracy [38], using the MR dataset with 80% accuracy and the SST2 dataset with 84% accuracy. This research disproves the hypothesis that CNN cannot compete with more advanced models like BiLSTM and BiGRU when it comes to sentiment categorization. This research also shows that the efficiency of the basic BiLSTM model can be enhanced by employing FastText embedding. This [39] document discusses various widely used feature extractors, including Bag-of-Words (BOW), Glove, FastText, N-grams, TF/IDF, and Word2Vec. Present study, we survey the landscape of opinion mining feature extractors and examine their strengths and weaknesses. Additionally, both the benefits and drawbacks of each extractor will be detailed. Furthermore, a comparative analysis is performed to determine the optimal combination of CNN/extractor using criteria such F1 Score, precision, recall, and accuracy.

3. The proposed model

In this Article, we used deep learning methods to suggest a new approach to sentiment analysis in Telugu. Based on the available information at now, the lack of a suitable dataset has resulted in a paucity of deep learning approaches being applied to Telugu. Telugu is a morphological language and more datasets are not readily available for researchers to employ in their work. Consequently, most authors have resorted to using machine learning algorithms instead. In this contest, we developed a dataset in Telugu language corpus called "sentikanna" and collected tweets from multiple categories, including movies, recipes, and tourism. We present the BiLSTM model for the categorization of sentiments utilizing a number of different word embedding's, including TF/IDF, Word to Vec, Glove, and FastText embedding. We examine how FastText embedding affects the BiLSTM model's ability to classify sentiment. the researchers determine the sentence level sentiment polarity, such as positive or negative, using these deep learning models. A few example sentences that describe the polarity of a given sentence has been shown in table 3.

3.1. Dataset Creation

3.1.1. Data gathering and annotation procedure

The main sources of information utilized in this study were the online platforms Youtube.com, Makemytrip.com, and Facebook.com. These sources were accessed through web scraping and data harvesting techniques. The examination of sentiment analysis within the contexts of films, recipes, and tourism yields distinct and valuable observations pertaining to the interests, preferences, and cultural characteristics of those who speak the Telugu language. This facilitates a more focused and pertinent comprehension of sentiment within various domains, empowering stakeholders to make well-informed choices, better their products, and elevate client experiences. Table 1 shows summary statistics about our data set.

Table 1 Statistical information of our Dataset

S. No	Dataset	Positive Reviews	Negative Reviews	Total Reviews
1	Movie	550	459	1009
2	Tourism	581	504	1085
3	Recipe	576	445	1021
Total Reviews				3115

Social media platforms have facilitated the gathering of diverse user perspectives from many regions, enabling a full analysis of sentiments. Telugu language speakers from various civilizations and geographical locations convey their emotions. A comprehensive evaluation was conducted to ascertain the relevance and use of various information sources for our project. We collected reviews from many domains, namely Movies, Tourism, and Recipes, correspondingly. The utilization of sentiment analysis in the field of movie criticism holds significant importance as it unveils the collective perception of the audience towards the picture. But a similar concern exists, specifically the absence of reviews for Telugu films. The Python Scraper facilitates the retrieval of reviews and enables the user to respond to them. Subsequently, the data is exported into an Excel file. The Excel file comprises many data fields, including the reviewer's name, comment, timestamp, date, number of likes, and reply count. Subsequently, the superfluous columns are eliminated from the Excel file. The Excel file ultimately comprises three columns, namely S.no., Review, and Polarity. An attempt was made to extract a significant portion of the available data from the website YouTube.com. The aggregate number of movie reviews amounts to 1009, comprising 550 positive reviews and 459 negative reviews. The reviews within the dataset were manually tagged using the following methodology: The reviews

consist of favorable assessments of the movie, which are categorized as positive reviews, while unfavorable evaluations are classified as negative reviews. We excluded evaluations that exhibited uncertainty, neutrality, etc. A comprehensive examination of the full review and its accompanying annotations is conducted in order to do a recheck. Remarkably, it was discovered that no errors were detected in the annotation conducted using this particular methodology. Ultimately, the Excel file undergoes conversion into a CSV file, followed by the application of preprocessing processes to the provided file.

Our dataset was produced by utilizing data mining techniques on recipe reviews from YouTube and Facebook. The recipe reviews were extracted and afterwards purged of any HTML coding and other content. A dataset with 1021 Telugu reviews was generated. As a result of limited resources and a scarcity of unfavorable evaluations in the Telugu language, our ability to extract a substantial number of reviews was significantly constrained. Subsequently, in relation to each review encompassed inside our dataset, it is imperative to eliminate any extraneous reviews and accompanying tags. Consequently, the resultant file will exclusively comprise of reviews and their corresponding sentiment.

Table 2 Sample Positive and Negative sentences from the dataset

S.NO	Dataset	Positive Reviews	Negative Reviews
1	Movie	గొప్ప నటులు నటించారు, రాజమౌళి సార్ చాలా బాగుంది.నేను ఇలా నవ్వి చాలా కాలమైంది. 4వ సారి చూస్తున్నాను...చాలా బాగుందిసినిమా.	నేను చూసిన చెత్త సినిమాల్లో ఇదొకటి. అది ఎంత ఘోరంగా ఉందో నేను ఇంకా అధిగమించడానికి ప్రయత్నిస్తున్నాను.
2	Tourism	చేసిన సర్వీస్ మరియు సూచనలు చాలా బాగున్నాయి మరియు రూమ్‌లు చాలా శుభ్రంగా ఉన్నాయి మరియు చాలా మంచి సేవలందిస్తున్నారు ధన్యవాదాలు సర్	సిగరేట్లు మరియు మద్యం వాసనతో లాబీ స్పష్టంగా దుర్వాసన వెదజల్లుతోంది.తువ్వాళ్లు కూడా చాలా పాతవి
3	Recipe	నేను మీరు చెప్పినట్టే సిద్ధం చేసాను...అద్భుతంగా వచ్చింది..రుచి కూడా సూపర్బ్.మీ చక్కని వీడియోలకు ధన్యవాదాలు.	వాంతి,దయచేసి చికెన్‌ను సరిగ్గా కడగాలి.వండడానికి ముందు.చెత్త బిర్యానీ

The process of making and annotating both movie reviews and recipe reviews is the same. We scraped the information from MakeMyTrip.com using Python Scraper. All the data in the collection was annotated with positive and negative ratings on a 2-point scale. In all of these circumstances, a review is considered favorable if the person or user who provided the review had a positive experience, and negative otherwise. The dataset has a total of 1085 reviews, with 581 positive and 504 negative. If you save the original file and then delete the extra columns, you'll be left with ratings and comments. The prepressing procedures were performed on the dataset later. Table 2 presents an illustrative instance of a positive and negative sentence derived from a provided dataset.

3.2 Pre-processing

In the first stage of our methodology, called "sentence pre-processing," we convert Telugu sentences into a format that can be read by a sentiment analysis programme. Punctuation, English letters, stop words, numbers, tokenization, normalization, and light stemming are just some of the pre-processing tasks we can handle. Word ambiguity is reduced using these linguistic methods, making the overall method more precise and efficient.

The following procedures are engaged in the preliminary processing of Telugu sentences:

3.2.1 Tokenization

Tokenization is a crucial step in NLP since it separates a string of text into smaller pieces that are easier to manage, such as words and phrases. Common punctuation marks for separating words include commas, semicolons, quotation marks, and spaces. Tokens are examples of single words or phrases that have been altered without regard to their context or meaning.

3.2.2 Stop word Elimination

Stop word removal refers to the process of omitting words from a sentence that are not essential to understanding the meaning of the phrase. Words like "the," "a," and "an" in English and in Telugu “చేత”,” వలన”, “గూర్చి”, “కొరకు” are examples of insignificant words.

3.2.3 Erasure of Punctuations

The purpose of the Punctuation Removal tool is to clean up documents by removing inefficient punctuation symbols such as (#, -,",").

3.2.4 Latin characters and digits removing

Any characters or numerals from the Latin alphabet that might appear in the statement will have no relevance and cannot be identified. As a result, these characters have been taken out of the game.

After annotation and preprocessing the given data, the dataset is converted into a csv file, which contains three columns: s. no., tweet, and sentiment. Sentences with a positive feeling are given a sentiment polarity of 1, while those with a negative mood are given a sentiment polarity of 0. Table 3 displays the sentences exhibiting sentiment polarity.

Table 3 Sample tweets with sentiment polarity

S. No	Dataset	Tweet	Sentiment
1	**Movie**	సినిమా చాలా బోరింగ్‌గా ఉంది. ఒక వాస్తవాన్ని భర్తీ చేయలేదువిపరీతంగా బోరింగ్ సినిమా.	0
2	**Tourism**	నా బంధువులతో నా బసను నేను నిజంగా ఆనందించాను మరియు సిబ్బంది కూడా చాలా బాగున్నారు. ఫ్రంట్ ఆఫీస్ నుండి ప్రత్యేకంగా సుమన్ మరియు భువన మాకు చాలా మధురంగా ఉన్నారు.	1
3	**Recipe**	హాయ్ అక్కా..అదే నువ్వు చెప్పినట్టే నేను కూడా ట్రై చేసాను చాలా బాగా వచ్చింది ధన్యవాదాలు	1

3.3. *Feature Extraction*

Feature extraction is very important in sentiment analysis, which seeks to identify the underlying emotional tone of a piece of text. The goal of feature extraction is to provide a numerical representation of the input for use by machine learning algorithms. These numerical features capture the relevant information from the text that helps the model learn to predict sentiment accurately. Word Embedding is the one of common techniques for feature extraction in sentiment analysis.

3.3.1. Word Embedding’s:

In word embeddings, each word is mapped onto a dense vector in a smooth space. These vectors capture semantic relationships between words. FastTest, Glove, and Word2Vec are examples of pre-trained word embedding’s that may be used.These embedding’s are able to be utilized as input features for machine learning methods or may undergo

fine-tuning in the training process. Here we used four methods to extract the features: TF/IDF vectorization, Word2Vec, Glove, and FastText.

A brief description of the mentioned feature extractors is explained below.

3.3.2. TF-IDF Vectorization

There is a well-known weighting mechanism called TF-IDF, but its accuracy is still pretty comparable with newer approaches. The (TF-IDF) is a statistical metric utilized to assess the significance of phrases inside a document, serving as the foundational principle for a machine learning algorithm. The text can be presented either as a singular document or as a group of articles, which is often called a corpus. The Frequency of Use of a Term and the Frequency of Use of an Inverse Document are being combined.

The TF score is calculated by considering the relative frequency of words. Initially, the frequency at which terms occur in the papers is assessed. The process of determining word frequency entails dividing the frequency of a given word (i) inside a document by the total number (N) of words present in such document (j).

$$TF(i,j) = log(frequency(i,j)/log\ (N(j)) \quad (1)$$

The determination of a word's scarcity is based on its IDF score. The importance of term frequency (is significant in this context since it assigns higher importance to words that occur more frequently. Nevertheless, the data included inside words that are rarely utilized in the corpus can be of great worth. The Israeli Defence Forces (IDF) collects this information. The formula utilized for its calculation is as follows: The ratio of the total number of documents (N) to the number of documents (d) that contain the specific terms (i) is worked out.

$$IDF(i,j) = \log\left(\frac{N(d)}{frequency(d,i)}\right) \quad (2)$$

In order to mitigate the impact of excessively high TF and IDF frequency scores, the logarithm function is included in the aforementioned equations. The final TF-IDF result is calculated by multiplying the scores received from the TF and IDF components.

$$TF\text{-}\ IDF(ij) = TF(ij) * IDF(ij) \quad (3)$$

3.3.3. Word2Vec extractor

The word embedding's, that are generated using the Word2Vec method are dense vector representations of words in a continuous vector space. comparable words can have comparable vector representations since these embedding's capture the semantic relationships among them. The fundamental concept underlying Word2Vec is acquiring this embedding's through the training of a neural network on an extensive collection of textual data. Word2Vec utilizes two primary architectures: Continuous Bag of Words (CBOW) and Skip-gramWord2Vec embedding's possess several significant advantages, including their capacity to effectively capture semantic links, enhance performance on subsequent tasks, and decrease the dimensionality of textual input. Pre-trained Word2Vec embedding's are readily accessible for various languages and domains, offering the advantage of reducing the duration and allocation of training efforts and resources. Some common libraries used for working with Word2Vec embedding's include `gensim` in Python, It offers resources for the purpose of training and use Word2Vec embedding's, and libraries like `spaCy` that incorporate pre-trained Word2Vec embedding's.

3.3.4. Glove Extractor

The Glove Extractor aims to construct a vector space model of a phrase by leveraging the constancy of term similarities. The GloVe is a hybrid pattern that takes advantages of both the Continuous Bag of Words pattern and the Skip-gram pattern. The computational time of the previous pattern is very efficient, its evolution rate is low, the latter pattern's rate of classification is high, but its computational time is inefficient. The Glove is an attempt to combine the methods developed by two patterns; it has proven to be more effective and accurate than either of the original patterns. GloVe, also known as Global Vectors for Word Representation, is a widely used methodology for producing word embedding's that is comparable to Word2Vec. It was developed by researchers at Stanford University and aims to create word embedding's by considering global statistics of word co-occurrence frequencies in a large corpus.

The main idea behind GloVe is to factorize the word co-occurrence matrix to learn word embeddings. The co-occurrence matrix is a representation of the frequency with which words occur together in a certain context, often within a predetermined frame of words. The GloVe model seeks to find word embeddings that are optimized. In order to depict the associations of words on the basis of their connections on these co-occurrence frequencies. One of the advantages of GloVe is its ability to incorporate both global word co-occurrence statistics and local context windows, which can lead to embedding's that capture both syntactic and semantic information. Pre-existing GloVe embedding's are accessible for multiple languages and can be employed in diverse natural language processing endeavours. These embeddings can be loaded and used with libraries like `gensim` or directly integrated into neural network architectures for downstream tasks. GloVe embedding's, like Word2Vec embedding's, have been widely adopted and have proven effective in improving the performance of a variety range of NLP tasks, Made up of word analogy tasks among others,text categorization, sentiment classification, and computer translation.

3.3.5. FastText extractor

FastText is an innovative word embedding system that has been developed by researchers at Facebook. This system offers a rapid and efficient method for representing individual terms within a vector space and for classifying sentiments based on textual data the main objective of FastText word embedding is to investigate the internal structure of words rather than to acquire knowledge about term representations. FastText operates by enabling the user to manually select a window and move it across the input text, at which point the programme will either use the BOW method to learn the key phrase from the surrounding text or the Skip-gram method to learn all the surrounding terms from the key term. The FastText method is quite analogous toWord2Vec method; the main distinction is that FastText is trained to learn the vector representation of individual letter n-grams that make up a term.

Here's a brief overview of how FastText works:

- **Sub word Representations:** The FastText algorithm decomposes words into smaller subword components known as "n-grams" or character n-grams. As an illustration, the term "apple" can be deconstructed into various subunits such as "<ap", "app", "ppl", "ple", "le>", and so on, wherein the symbols "<" and ">" serve as distinctive delimiters. FastText is able to effectively capture significant information, even for terms that are infrequently used or contain spelling errors.
- **Learning Word Vectors:** The FastText model utilises a neural network to acquire vector representations for both individual words and subwords. The representation of each word vector is obtained by summing its individual sub word vectors.
- **Skipgram Training:** FastText, like Word2Vec, employs a skipgram architecture for the purpose of training word vectors. When fed a target word, the model will try to guess what other words might be in the immediate vicinity.
- **Hierarchical Softmax:** In order to enhance the efficiency of training, FastText utilises a hierarchical softmax technique that arranges words into an ordered binary structure. This approach decreases the number of nodes that produce data that require updating throughout the training process.
- **Applications:** FastText embeddings have gained significant popularity in the field of NLP due to its extensive use in many tasks, including but not limited to text categorization, sentiment classification, and computer translation. These models demonstrate notable efficacy when applied to languages that include intricate morphological structures and extensive lexicons.

In order to utilize FastText embeddings, it is customary to initially train the model on a substantial text corpus and subsequently employ the acquired vectors as input characteristics for subsequent tasks. Pretrained embeddings possess the ability to capture both semantic and syntactic links between words, hence enhancing the efficacy of many NLP tasks.

3.4. Bi-LSTM: (Bi-directional long short term memory)

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to handle sequence data and capture long-range dependencies. LSTMs are particularly effective at learning patterns in sequential data, making them popular for tasks such as time series forecasting, Natural Language Processing, speech recognition, and more.

The BiLSTM model is a composite model consisting of both a forward LSTM and a backward LSTM. The LSTM utilized in this model is a specific variation of the recurrent neural network (RNN). To combat the gradient disappearance issue plaguing conventional RNNs, we augment the LSTM with a gating unit, giving it enhanced capacity for capturing long-term dependencies and allowing the RNN to more effectively detect and the interdependencies inherent in long-distance data.

Every LSTM unit has a novel structure made up of four primary components: an input gate (i_t), an output gate (o_t), a forget gate (f_t), and a storage unit (c_t). Figure 4 depicts the internal anatomy of a single LSTM module cell.

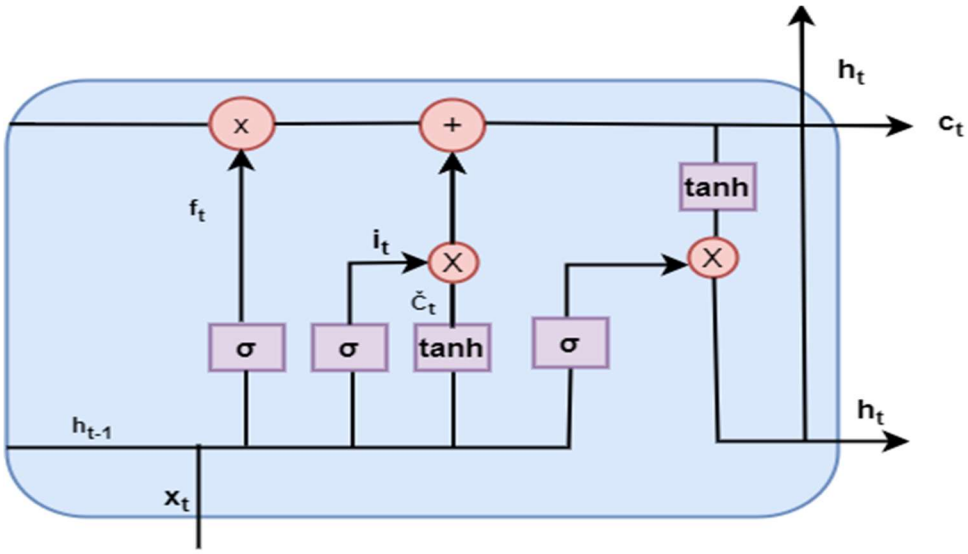

Fig. 1. Architecture of Single LSTM Unit.

The LSTM model is composed of various gates and operates through a specific process

- **Forget gate mechanism:**

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (4)$$

Resetting a memory device is done using the forget gate f_t. The Sigmoid activation function is represented by σ , The weight matrix, denoted as W_f.The variable $x_t$represents the input at a certain time point *t*. The hidden layer's output possesses a weight matrix, denoted by U, hidden state, denoted by the notation h_{t-1}, stores information temporarily. The offset vector is denoted by b_f .

- **Input gate mechanism:**

$$i_t = \sigma(W_i . x_t + U_i . h_{t-1} + b_i) \quad (5)$$

The input gate controls how much data enters the memory cell. The Sigmoid activation function is represented by σ , The weight matrix, denoted as W_i.The variable $x_t$represents the input at a certain time point t . The hidden layer's output possesses a weight matrix, denoted by U, hidden state, denoted by the notation h_{t-1}, stores information temporarily. The offset vector is denoted by b_i .

- **Update unit status:**

$$c_t = f_t \odot c_{t-1} + i_t \, tanh(W_c . x_t + U_c . h_{t-1} + b_c \quad (6)$$

c_t is the state of the cell and stands for long-term memory.The f_t is forget gate is a memory reset device. c_{t-1} is the state of the cell at time h_{t-1} ,this represents the cell's long-term memory. The input gates that manage the memory cell's input are denoted by i_t. The Weight Matrix is denoted by W_c. The time t, the input is denoted by x_t . The hidden layer's output possesses weight matrix, denoted by U, hidden state, denoted by the notation h_{t-1}, x_t stores information temporarily. The offset vector is denoted by b_c .

- **Output gate mechanism:**

$$o_t = \sigma(W_o x_t + U_0 h_{t-1} + b_o) \quad (7)$$

The output gate o_t denotes the output gates that regulate the functioning of the memory cell. The Sigmoid activation function is represented by σ . The weight matrix, denoted as W_o , $x_t$representsthe input at time *t*. And x_t denotes the input at a certain time point *t*. The output of the hidden layer has a weight matrix, denoted by U, hidden state, denoted by the notation h_{t-1}, stores information temporarily. The offset vector is denoted by b_o.

- **The present condition of the hidden layer:**

$$h_t = o_t \odot \, tanh(c_t) \quad (8)$$

Here hidden state is represented by h_t . The tanh-designated Sigmoid activation function. The cell state is defined as c_t , and it can defines the long-term memory. The output gates are represented as o_t and that can control the output of the memory cell.

In BiLSTM approach, one of the LSTMs will read the statement forward, while the other will read it backward. At this point, after each LSTM has processed its last word, you will go on to the next phase, which is the aggregation of its hidden states. In the BiLSTM architecture, the forward and backward movements are each managed by their own independent LSTM units. $\boldsymbol{LSTM_f}$ is a term that stands for the forward layer of the hidden state $\overrightarrow{\boldsymbol{h_t}}$, T_n is a term that stands for the number of tokens from 1 to n. Similarly, $\boldsymbol{LSTM_b}$ is the reverse layer of the hidden state state $\overleftarrow{\boldsymbol{h_t}}$, and $\boldsymbol{T_n}$ is the token count decreasing from n to 1. In the provided LSTM Model, the concatenation operator $\oplus$ is used to combine the two hidden states, states $\overrightarrow{\boldsymbol{h_t}}$ and $\overleftarrow{\boldsymbol{h_t}}$ into a single hidden state, $\overleftrightarrow{\boldsymbol{h_t}}$.

$$\overrightarrow{\boldsymbol{h_t}} = \boldsymbol{LSTM_f}(\boldsymbol{T_n})\boldsymbol{\varepsilon}(\boldsymbol{1}, \ldots \boldsymbol{n}) \quad (9)$$

$$\overleftarrow{\boldsymbol{h_t}} = \boldsymbol{LSTM_b}(\boldsymbol{T_n})\boldsymbol{\varepsilon}(\boldsymbol{n}, \ldots \boldsymbol{1}) \quad (10)$$

$$\overleftrightarrow{\boldsymbol{h_t}} = \overrightarrow{\boldsymbol{h_t}} \oplus \overleftarrow{\boldsymbol{h_t}} \quad (11)$$

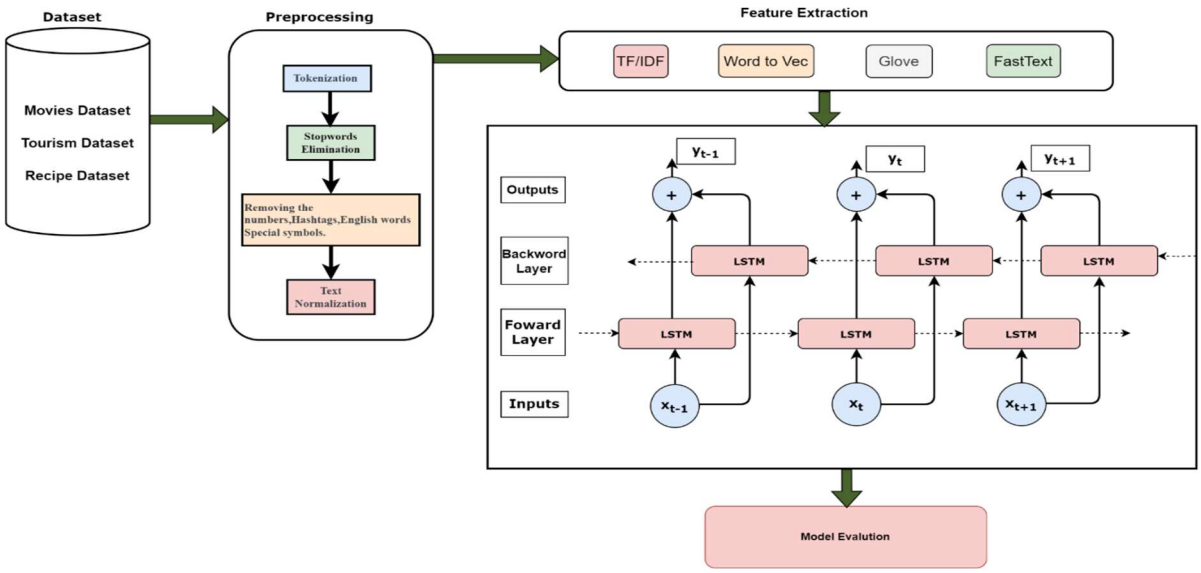

Fig. 2. The architecture of the proposed Bi-LSTM model

4. Results and discussion

Measurement metrics are referring to the various quantitative and qualitative criteria used to assess and evaluate the results, outcomes, or performance of a study, experiment, or project. These metrics are crucial for demonstrating the validity, reliability, and impact of the research. The selection of metrics is contingent upon the particular study aims, methodology, and disciplines at hand. Here are some measurement metrics used in this research papers:

The harmonic mean is used to determine the F1-score, taking into account both precision and recall. It gives a balanced measure that taking into consideration both positives and false negatives. The F1- score demonstrates its utility in scenarios when there exists an imbalanced distribution of classes. F1 score is derived in the Eq. (12).

$$\mathrm{F1-Score} = \frac{2 * (\mathrm{Precision} * \mathrm{Recall})}{(\mathrm{Precision} + \mathrm{Recall})} \tag{12}$$

Accuracy is a quantitative measure that evaluates the ratio of accurate predictions, encompassing both true positives and negatives, relative to the overall no. of predictions generated. it affords us a comprehensive evaluation of the effectiveness of the model. The derivation of the equation (13) is shown.

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})} \tag{13}$$

4.1 Parameter Setting

Several parameters can be adjusted to accomplish to optimize the model. The ideal parameters were identified through a series of experiments conducted in the course of this inquiry. Only the settings that yield the most favorable results are presented. The study used a training dataset, which constitutes 0.8% of the total dataset, and a test dataset, which constitutes 0.2% of the total dataset, in order to accomplish this objective. A preset epoch counts of 20 has been established for each experiment. Dropout is a regularization technique commonly employed in neural networks to mitigate the risk of overfitting. It involves randomly deactivating a fraction of the neurons during training, typically with a dropout rate of 0.3. Table 6 contains supplementary parameter configurations.

Table 4 Optimal parameter configurations for effective evolutionary processes

Word Embedding	Embedding Size	Dropout	Batch Size	Optimizer	Train max epoch
Pre-trained embedding's	128	0.3	16	Adam	20

Table 5 The outcomes of Accuracy and F1-Score for the models proposed on the Movies dataset are presented in a percentage format

MOVIES	Accuracy	F1 Score
TF/IDF+BILSTM	71.33	69.02
WORD TO VEC + BILSTM	68.26	72.34
GLOVE+BILSTM	79.56	79.23
FASTTEXT+BILSTM	**82.83**	**81.35**

Table 6 The outcomes of Accuracy and F1-Score for the models proposed on the Recipe dataset are presented in a percentage format.

RECIPE	Accuracy	F1 Score
TF/IDF+BILSTM	75.34	75.69
WORD TO VEC + BILSTM	81.46	76.71
GLOVE+BILSTM	80.41	84.21
FASTTEXT+BILSTM	**94.00**	92.57

Table 7 The outcomes of Accuracy and F1-Score for the models proposed on the Tourism dataset are presented in a percentage format.

TOURISM	Accuracy	F1 Score

TF/IDF+BILSTM	80.31	78.54
WORD TO VEC + BILSTM	83.24	79.76
GLOVE+BILSTM	81.34	80.26
FASTTEXT+BILSTM	**94.93**	93.44

Table 12 Comparison of Accuracy and F1-Score for our proposed models with multiple embedding's in percentage

Dataset	Performance Metrics	TF/IDF+BILSTM	WORD TO VEC + BILSTM	GLOVE+BILSTM	FASTTEXT+BILSTM
Movies	Accuracy	71.33	68.26	79.56	**82.83**
	F1 Score	69.02	72.34	79.23	**81.35**
Recipe	Accuracy	75.34	81.46	80.41	**94.00**
	F1 Score	75.69	76.71	84.21	**92.57**
Tourism	Accuracy	80.31	83.24	81.34	**94.93**
	F1 Score	78.54	79.76	80.26	**93.44**

4.2 Experimental Results

The implementation of the developed Telugu Sentiment Analysis methods stated in Part 3 is compared in this section 4. Furthermore, the present research compares the results of our deep learning model to different word embedding's are FastTest, Glove, TF/IDF and Word to vec. As of now, according to my knowledge, no one applies all these mentioned word embedding's in Telugu. We implemented the BiLSTM model with all these word embedding's and got good results. Compared to all methods, FatTest with BiLSTM yields good results. In the existing method, the authors did not analyze the accuracy of the models whereas the present work computes the accuracy and it is compared.

Tables 5, 6, and 7 explain the BiLSTM model performance with multiple word embedding's on the three multi-domain datasets. The accuracy and F1 score for each domain are compared in this section and shown in the tables. The results are compared with the BiLSTM model for all word embedding's.

Table 7 shows that the FastTest+BiLSTM model consistently outperforms when compared to other embedding methods. It has been observed that the accuracy and F1 score measures for all models have outperformed the other models. The suggested model demonstrates the capacity to accurately discern the pertinent samples. From the experimental results, it has been proven that the models understood the labels properly and were able to categories accordingly.

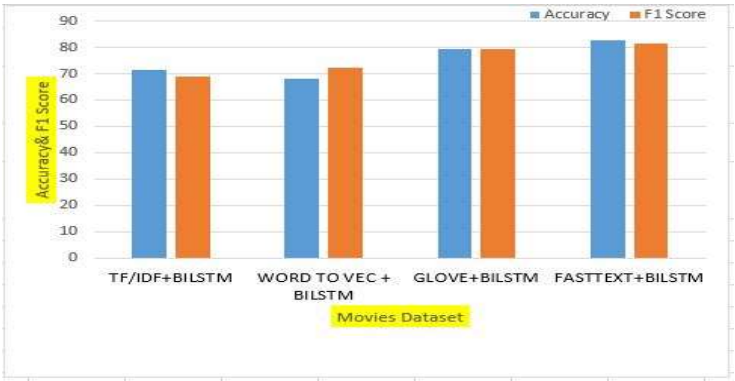

Fig. 4. The comparative analysis of accuracy and F1-Score of the BiLSTM model with all embedding's in the Movies Dataset

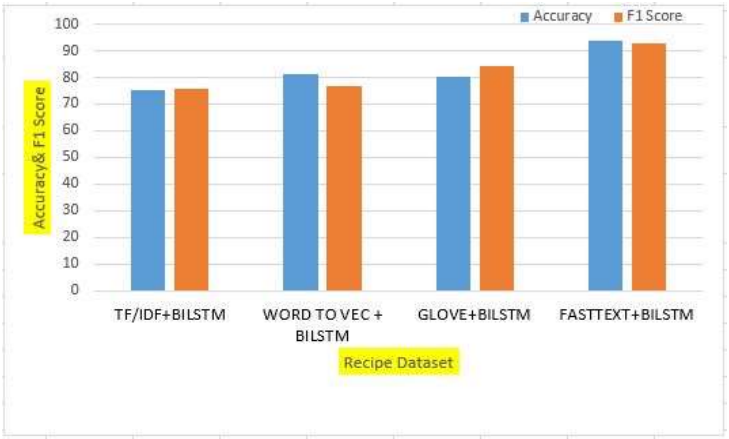

Fig. 5. The comparative analysis of accuracy and F1-Score of the BiLSTM model with all embeddings in the Recipe Dataset.

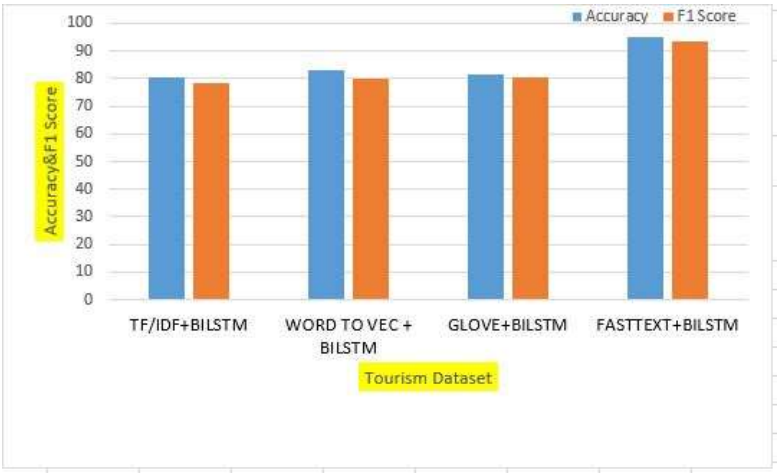

Fig. 6. The comparative analysis of accuracy and F1-Score of the BiLSTM model with all embeddings in the Tourism Dataset.

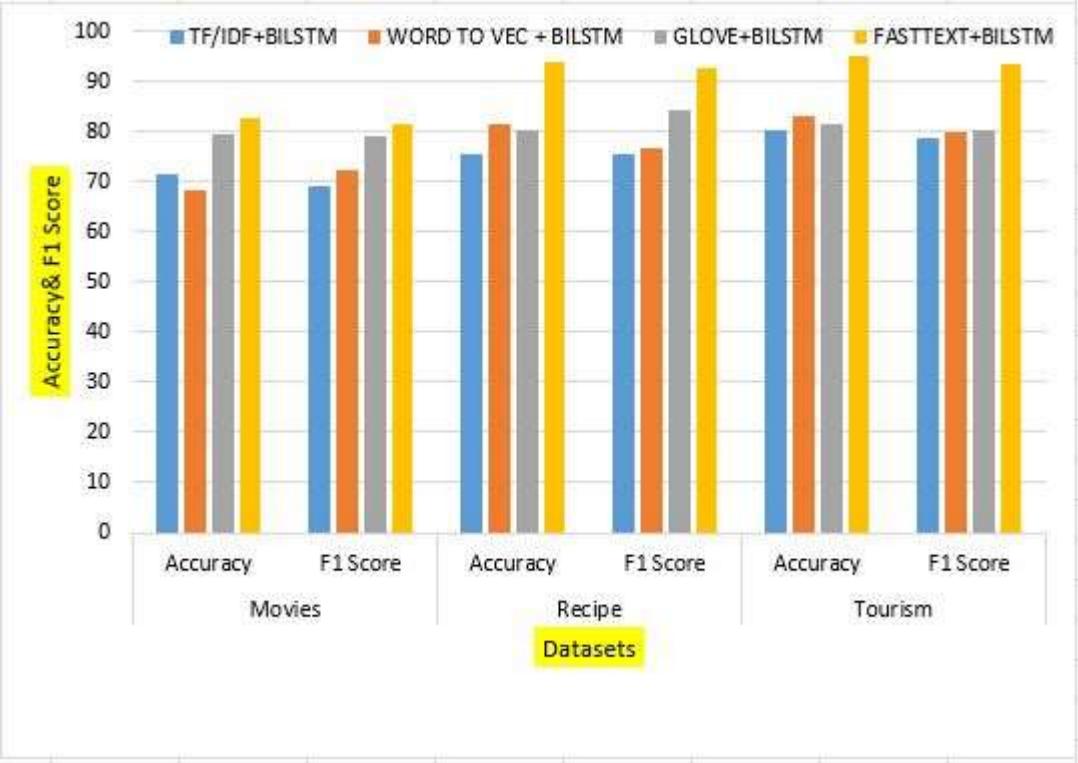

Fig .7. The comparative analysis of accuracy and F1-Score of the BiLSTM model with all embedding's and all datasets

5. Conclusion and future work

To achieve satisfactory results in sentiment classification, feature extraction is essential. The goal of feature extraction is to improve a classifier's performance by selecting the most informative and powerful set of features. In addition, feature extraction is the most crucial part of opinion classification since improper feature selection might have a detrimental impact on classification effectiveness. Because of its reputation as a morphologically weak resource language, Telugu necessitates careful attention to detail during the design phase. It is acknowledged that there is a wide range of needs when it comes to doing a sentiment analysis. This calls for the creation of a dataset with varying types of information. In this research, we presented an exploratory analysis of the state of the art among Telugu language feature extractors. In addition, we used a BILSTM model with multiple word embedding techniques, including TF/IDF, Word to vec, Glove, and FastTest, to find the best feature extractor for improving classifier performance. Experiments show that the effectiveness of the employed classifiers changes slightly depending on the characteristics of the word embedding sets. In terms of accuracy and F1 measure, we found that BILSTM+FastText is superior to any other combination. Because pre-trained FastText embedding's are trained using a combination of position-dependent features, phrase representation, and sub word information, they are able to handle out-of-vocabulary and unusual words with ease. This study demonstrates that alternative models can achieve comparable performance to the state-of-the-art BiLSTM model. The results of this study show that it is possible to develop a model with a minimal architecture for sentiment classification problems that achieves state-of-the-art results.

In future investigations, our objective is to evaluate the suggested methodology by employing optimization techniques. This has the potential to significantly improve the effectiveness of the BILSTM model in classifying both short and long texts. Furthermore, there is a desire to execute the aforementioned concept by employing a combination of multiple languages.

References

[1] Reddy, G.R.R., 2020. Enhancing Sentiment Prediction and Bias Detection for Telugu Language across Multiple Domains using ML and Deep Learning (Doctoral dissertation, International Institute of Information Technology Hyderabad).

[2] Yang, C., Zhang, H., Jiang, B. and Li, K., "Aspect-based sentiment analysis with alternating coattention networks". Information Processing & Management, 56(3), pp.463-478., 2019

[3] Ali, F., Kwak, D., Khan, P., El-Sappagh, S., Ali, A., Ullah, S., Kim, K.H. and Kwak, K.S., Transportation sentiment analysis using word embedding and ontology-based topic modeling. Knowledge-Based Systems, 174, pp.27-42.,2019.

[4] Rehman, A.U., Malik, A.K., Raza, B. and Ali, W., 2019. A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis. Multimedia Tools and Applications, 78(18), pp.26597-26613.

[5] Jagdale, R.S., Shirsat, V.S. and Deshmukh, S.N., 2019. Sentiment analysis on product reviews using machine learning techniques. In Cognitive Informatics and Soft Computing (pp. 639- 647). Springer, Singapore.

[6] Kumar, S.S., Kumar, M.A., Soman, K.P. and Poornachandran, P., 2020. Dynamic mode-based feature with random mapping for sentiment analysis. In Intelligent systems, technologies and applications (pp. 1-15). Springer, Singapore.

[7] Hasan, A., Moin, S., Karim, A. and Shamshirband, S., 2018. Machine learning-based sentiment analysis for twitter accounts. Mathematical and Computational Applications, 23(1), p.11.

[8] Telugu Rank, Ethnologue list. [Online]. Available: https://www.ethnologue.com/statistics/size

[9] Fatima Es-sabery1 , Khadija Es-sabery1 , Hamid Garmani1 , Junaid Qadir2 , and Abdellatif Hair1 W ,2022 Evaluation of different extractors of features at the level of sentiment analysis, nfocommunications journal, doi: 10.36244/ICJ.2022.2.9

[10] Reddy Naidu, Santosh Kumar Bharti, Korra Sathya Babu, and Ramesh Kumar Mohapatra. 2017. Sentiment analysis using telugu sent wordnet.

[11] Reddy Naidu, 2018. Building SentiPhraseNet for Sentiment Analysis in Telugu, Council International Conference 15th Indin from IEEE Xplore.

[12] Sandeep Sricharan Mukku and Radhika Mamidi. 2017. ACTSA: Annotated corpus for Telugu sentiment analysis. In Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems, pages 54–58.

[13] Sreekavitha Parupalli, Vijjini Anvesh Rao, and Radhika Mamidi, 2018 ,Bcsat: A benchmark corpus for sentiment analysis in Telugu using word-level annotations, in Proceedings of ACL 2018, Student Research Workshop. pp. 99–104, Association for Computational Linguistics 3

[14] S. Tammina, "A Hybrid Learning approach for Sentiment Classification in Telugu Language," 2020 International Conference on Artificial Intelligence and Signal Processing (AISP), 2020, pp. 1-6, doi: 10.1109/AISP48273.2020.9073109

[15] Pravarsha Jonnalagadda, Krishna Pratheek Hari, Sandeep Batha, Hashresh Boyina, 2019. A rule based sentiment analysis in Telugu, International Journal of Advance Research, Ideas and Innovations in Technology

[16] Garapati, A., Bora, N., Balla, H. and Sai, M., 2019. SentiPhraseNet: An extended SentiWordNet approach for Telugu sentiment analysis

[17] Srikiran Boddupalli, Anitha Sai Saranya, Usha Mundra, Pratyusha Dasam,2019, Sentiment Analysis of Telugu data and comparing advanced ensemble techniques using different text processing methods, 5th International Conference on Computing Communication Control and Automation (ICCUBEA)-2019

[18] Manukonda, Durga & Kodali, Rohith & Guduri, Dheeraj. (2019). Phrase Based Heuristic Sentiment Analyzer for the Telugu Language. 6. 245- 251 13

[19] Regatte, Y.R., Gangula, R.R.R. and Mamidi, R., 2020, May. Dataset Creation and Evaluation of Aspect Based Sentiment Analysis in Telugu, a Low Resource Language. In Proceedings of The 12th Language Resources and Evaluation Conference (pp. 5017-5024).

[20] Kumar, R.G. and Shriram, R., Sentiment Analysis using Bi-directional Recurrent Neural Network for Telugu Movies.

[21] Bharti, S.K., Naidu, R. and Babu, K.S., 2020. Hyperbolic Feature-based Sarcasm Detection in Telugu Conversation Sentences. Journal of Intelligent Systems, 30(1), pp.73-89.

[22] Srikanth Tammina, 2020 , A Hybrid Learning approach for Sentiment Classification in Telugu Language, International Conference on Artificial Intelligence and Signal Processing (AISP).

[23] Jurafsky, D., Martin, J.H., 2008. Speech and language processing: An introduction to speech recognition. Computational Linguistics and Natural Language Processing.2nd Edn., Prentice Hall, ISBN 10, 794–800.

[24] [24] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, "The Impact of Features Extraction on the Sentiment Analysis," Procedia Computer Science, vol. 152, pp. 341–348, Jan. 2019, doi: 10.1016/j.procs.2019.05.008.

[25] [25] B. Shi, J. Zhao, and K. Xu, “A Word2vec Model for Sentiment Analysis of Weibo,” in 2019 16th International Conference on Service Systems and Service Management (ICSSSM), Jul. 2019, pp. 1–6. doi: 10.1109/ICSSSM.2019.8887652.

[26] [26]Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A., 2018. Advances in Pre-Training Distributed Word Representations. LREC 2018 - 11th International Conference on Language Resources and Evaluation arXiv:1712.09405.

[27] [27] S. Anjali Devi and S. Sivakumar, “An efficient contextual glove feature extraction model on large textual databases,” Int J Speech Technol, Sep. 2021, https://doi.org/10.1007/s10772-021-09884-2.

[28] [28] J. Kralicek and J. Matas, “Fast Text vs. Non-text Classification of Images,” in Document Analysis and Recognition – ICDAR 2021, Cham, 2021, pp. 18–32. doi: 10.1007/978-3-030-86337-1_2.

[29] [29] Hameed, Z., Garcia-zapirain, B., 2020. Sentiment Classification Using a Single-Layered BiLSTM Model. IEEE Access 8. doi:10.1109/ ACCESS.2020.2988550.

[30] [30] Hassan, A., Mahmood, A., 2018. Convolutional Recurrent Deep Learning Model for Sentence Classification. IEEE Access 6, 13949–13957. doi:10.1109/ACCESS.2018.2814818.

[31] [31] Ouyang, X., Zhou, P., Li, C.H., Liu, L., 2015. Sentiment analysis using convolutional neural network. Proceedings - 15th IEEE International Conference on Computer and Information Technology, CIT 2015, 14th IEEE International Conference on Ubiquitous Computing and Communications, IUCC 2015, 13th IEEE International Conference on Dependable, Autonomic and Se , 2359–2364doi:10.1109/CIT/IUCC/DASC/ PICOM.2015.349.

[32] [32] Liao, S., Wang, J., Yu, R., Sato, K., Cheng, Z., 2017. CNN for situations understanding based on sentiment analysis of twitter data. Procedia Computer Science 111, 376–381. URL: http://dx.doi.org/10.1016/j.procs.2017.06.037, doi:10.1016/j.procs.2017.06.037.

[33] [33] Zhang, Y., Rao, Z., 2020. N-BiLSTM: BiLSTM with n-gram Features for Text Classification. Proceedings of 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference, ITOEC 2020 , 1056–1059doi:10.1109/ITOEC49072.2020.9141692

[34] [34] Ma, R., Teragawa, S., Fu, Z., 2020. Text sentiment classification based on feature fusion. 2020 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC) doi:10.18280/ria.340418.

[35] [35] M. Heikal, M. Torki, and N. El-Makky, “Sentiment Analysis of Arabic Tweets using Deep Learning,” Procedia Comput. Sci., vol. 142, pp. 114–122, 2018, doi: 10.1016/j.procs.2018.10.466.

[36] [36] S. Al-Azani and E.-S. M. El-Alfy, “Hybrid deep learning for sentiment polarity determination of arabic microblogs,” in International Conference on Neural Information Processing, 2017, pp. 491–500.

[37] [37] M. Al-Smadi, B. Talafha, M. Al-Ayyoub, and Y. Jararweh, “Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews,” Int. J. Mach. Learn. Cybern., vol. 10, no. 8, pp. 2163–2175, 2019.

[38] [38]Isnaini Nurul Khasanah 2021Sentiment Classification Using fastText Embedding and Deep Learning Model ,5th International Conference on AI in Computational Linguistics

[39] [39]Fatima Es-sabery1 , Khadija Es-sabery1 , Hamid Garmani1 , Junaid Qadir2 , and Abdellatif Hair1 W 2022,Evaluation of different extractors of features at the level of sentiment analysis ,Infocommunications journal, DOI: 10.36244/ICJ.2022.2.9

sentiment classification in the Telugu language by employing several feature extractors.