Cross-Domain Document-Level Sentiment Analysis for Telugu Language Data

Mrs.V.Tejaswi
Assistant Professor, Department of CSE,
Malla Reddy College of Engineering for Women.,
Maisammaguda., Medchal., TS, India

Abstract— Finding common threads of optimism or negative in text is the goal of sentiment analysis. In the business world, it is used to learn about a product's reception, a customer's identity, and their expectations of a firm by monitoring the tone of online conversations on forums like Reedit and Twitter. Sentiment analysis in various languages has arisen as a separate topic of NLP as businesses seek for client feedback in previously untapped markets. The stakes could not be greater, since Telugu is a Dravidian language spoken by almost 82 million people. Little effort, such as annotated data and software, is put towards supporting Telugu; hence the language is often overlooked. To better understand the sentiment of the data, we conducted our research using the "Sentirama" dataset and, as you'll see in the paper, we used a variety of Machine Learning models (including SVM-linear, SVM-quadratic, SVM-polynomial, Random Forest, Naive Bayes, and KNN) and Featured concepts (including word2vec+ (CBOW or skip-gram), TF-IDF, and Fastest). To find the best Deep-learning model for Telugu sentiment analysis, we also tried out LSTM, Bidirectional-LSTM, and 1D-CNN.

Sentiment analysis, emotional analysis, machine learning, deep learning, support vector machines (linear, quadratic, polynomial), kernel normalization (KNN), word embeddings (gram, word2vec, CBOW, skip-gram, TF-IDF, Fastest), recurrent neural networks (LSTM, Bidirectional LSTM), convolution neural networks (1D), and Sentirama.

I. INTRODUCTION

The ability to comprehend human feelings and situations is a great asset to the species. How close can we get AI to mimicking human behavior? This situation requires immediate attention. The goal of sentiment analysis is to reveal the inner workings of human thought and behavior. Give some thought to the link that exists between you, the customer, and me, the producer. With the proliferation of online reviews and ratings, sentiment analysis has emerged as a powerful method for gleaning insights from consumer and expert feedback. Consumers may benefit from gaining an appreciation of the tone of user-generated content like online reviews and forum threads. This summary may be useful to manufacturers since it offers a glimpse of consumer feedback patterns that may be utilized to fine-tune future production runs. The groundwork for doing sentiment analysis in English has been laid. Recent years have seen a precipitous surge in Telugu-language data due to the use of the language as an Internet interface. Yet, there is less material available for studying Telugu than English.

Use in the field of emotion analysis. That's why I'd want to earn a livelihood conducting sentiment analysis in Telugu.

Many fields have discovered applications for sentiment analysis, which aims to reveal the hidden subjectivity in text.

A. Recommender Systems

Nowadays, music recommendation algorithms are widely used in popular applications like savan and Wynk. Amazon and Flipkart are just two examples of the many e-commerce sites that have recently included contextual product suggestions to their user interfaces. Using sentiment analysis to sift through user evaluations and provide suggestions based on a person's stated tastes in music or merchandise is one use. The approach has the potential to improve the performance of recommender systems whether employed alone or in combination with a collaborative filtering process.

B. Assist in Decision-Making

If we want to know the answer to a query like "which gadget should I get," for instance, we may look at the existing evaluations left by past customers. Like the phrase "that cinema to assault," where our preparations are steered by a study of the hearing of the one defending it, etc., this line of thought emotionally aids in providing more responsibility.

C. Question and Answering

Questions soliciting an individual's opinion need special handling. It's possible that sentiment analysis might play a major role in answering such inquiries with relevant facts.

D. Business Intelligence

Customer feedback plays a significant role in informing product and service development decisions. As it would be challenging to offer the product or application directly to all of the people and there are also large risks that may not impress many people, it is usual practice for firms to test the waters with a smaller group of individuals. Businesses might benefit from doing sentiment analysis since it could help them better cater to their clients' wants and demands.

II. Literature Review

It is possible to enhance the accuracy of deep learning sentiment analysis by using ensemble techniques, as detailed in [1,]. These methods enabled options in representation, improved speed, and automated feature extraction. Several models are offered in this research, many of which include both manually-created and automatically-collected variables from big datasets to provide superior results. In [2], the feelings of native Telugu speakers were studied utilizing a Hybrid method of analysis. This method included the best features of both lexical and machine learning methodologies. The Na ve Bayes classifier achieved the highest accuracy (85%) in binary Sentiment Analysis. The specifics of TF-IDF and how it operates are described in [3]. A procedure's efficacy may be gauged by looking at its output. The preprocessing step of removing the many redundant phrases has the potential to improve precision and accuracy. By using n-gram features and pre-trained word embeddings, the best F1 score that can be obtained by [4] is 82.05%. Many machine learning and deep learning models are used for this purpose. Due to the pre-trained word embeddings made available by Divyanshu Kakwani et al., Carrie was compelled to look into the Telugu language. By our study, we proved that a KNN model can be built with a text classification framework and the tf-idf algorithm. Inspecting the output of the model revealed the need for pre-processing, since the presence of undesired words in the text hampered the precision of the classification. To solve this problem, they fused tf-idf with KNN with a few tweaks, and the resulting model was more effective.

The findings of a study titled "Effect of Translation on Sentiment Analysis" were made available. Gangula Rama Rohit Reddy wrote this study. English-language product and book reviews are included in this publication. As the tone of a work may be lost in translation, checking the original against the translated version is often necessary. Evidently, there are situations when this kind of automatically translated material might mislead, by eliciting a different emotional reaction than what was originally intended. Machine translation has the potential to distort the meaning of the original text; hence it is best to have it done by a human translator. Using Telugu Senti Word Net, a sentiment analysis was conducted as described in [6]. Few studies have been conducted in Indian languages because of the difficulty in gaining access to annotated data. Data is being subjectively categorized for this purpose. It is said that a statement is objective when it does not include any kind of opinion and subjective when it does. When you want to associate feelings with words or phrases, SentiWordNet is your tool. Specifically, we use the doc2vec program for this goal. A SentiWordNet is essentially a WordNet with the addition of sentiment analysis. For subjective classification, the suggested technique has 74% accuracy, while for sentiment classification, it gets 81% accuracy.

An 86% accuracy rate was achieved when a bidirectional LSTM was applied to tweets in Tamil in [7].

By 2020, LAL KHAN et al. will have completed their all-encompassing examination of opinion mining using sentiment analysis algorithms and published their findings. Then, we do some basic preprocessing on the data before feeding it into the opinion mining method and identifying its polarity. The cleansed data was fed into several algorithms that fall within the supervised learning paradigm.

At first, a decision tree was used, and then a neural network, and lastly, a support vector machine were chosen as the most accurate classifier. In a first step, we used probabilistic classifiers like maximum entropy, naive Bayes, and Bayesian networks; later, we moved on to rule-based classifiers. A set of calculations was used to examine and compare the accuracy. In [9], a bidirectional LSTM was used to analyze Tamil tweets, leading to an 86% success rate. It will be published in 2020 by LAL KHAN and company. Using n-gram features and pre-trained word embeddings, the ML and DL models in [10] in this paper they compared three algorithms such as Naive Bayes, Decision Tree, KNN on dataset and for both text classification and for opinion mining Naives Bayes gets better performance amongst them for its probabilistic technique [11] conducted a binary sentiment analysis on the aforementioned corpus, with the latter achieving an accuracy of 76%. In the words of S. Tamminen and company. [14]5-fold cross validation is used and they used 200-dimension feature sentence vector ,Doc2Vec provided by genesis on top of it they used Naive Bayes, Logistic Regression, SVM, MLP,Decision Trees, Random Forest [15] in this paper they used both tf-idf and Doc2vec for text pre-processing they used ACTSA dataset in this paper

III. Methodology

A. DATA

Sentiraama" [12] was created by G. Rama Rohit Reddy of the Linguistic Technologies Research Centre (KTRC), KCIS, IIIT Hyderabad. Each document in the four data sets that make up the corpus has been assigned a simple two-value positive/negative sentiment score. Book, product, film, and music reviews are only some of the forms of media critiques included in the corpus. All of them followed the same protocol when it came to annotation. Just 3 books, 1 product, and 1 film were reviewed for the final report. Over 267 reviews cover a wide variety of Telugu films and may be found in the "Movie Reviews" section of the corpus. A total of 136 people gave it a positive rating, while 131 people gave it a negative rating. There are a grand total of 20,000 separate phrases and 165,049 words. There are 200 unique reviews of different goods in the Telugu "Product Reviews" area of the corpus. There are exactly equal numbers of favorable and negative comments, at 100 each. Sum of all words: 259189 (or

43199 phrases). There are 200 reviews of different books written in Telugu script included in the "Book Reviews" section. It's possible that both raving and damning reviews may appear. The whole work consists of 6808 sentences and 33179 words. Sentiraama is a software application designed to make Telugu sentiment analysis easier.

B. PREPROCESSING

- **Removing stop word**
  Stop words are words that are not taken into account because of their low importance. If a sentence doesn't add anything to the whole, there's no need to maintain it.
- **Special characters removal**
  In this Data there will saw few special character such as (#,!!!," ") that is not required so we removed it.
- **Tokenization**
  At this stage, we break down the longer strings into individual words, or tokens.

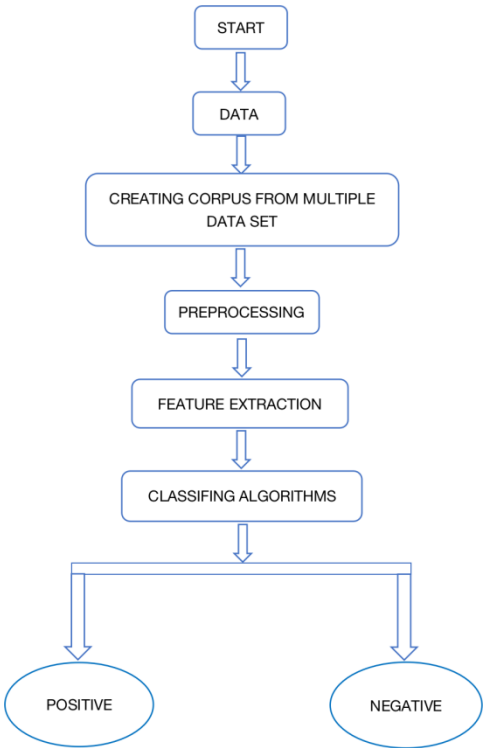

Fig. 1. ML Flow Chart

C. FEATURE EXTRACTION

- **TF-IDF**
  Inconsistency in Official Documents during Terms the TF-IDF algorithm examines how often certain terms and phrases appear in a set of texts. Determining the importance of a word or word sequence within a corpus or series of words in relation to a given text. The textual frequency effect, which states that the more often a word occurs in the text, the more meaning is obtained from its usage, is mitigated by the corpus frequency effect (data set).
- **Word2Vec**
  Word2Vec is a neural network that uses just two layers to do its task of reconstructing sentences from their component words. Inputting a huge body of text into the system results in the generation of a vector space with hundreds of dimensions, where each word is represented by a separate vector. In the corpus, words that have similar contexts are grouped together in the vector space. Word2Vec is an effective prediction model for generating word embeddings from unprocessed text. Both the Skip-Gram and Continuous Bag-of-Words (CBOW) paradigms are easily accessible.
  - **Continuous Bag-of-Words (CBOW)**
    In order to determine the intended meaning of a target word, CBOW analyzes the surrounding words. Since CBOW treats the whole context as a single observation, it tends to smooth over a lot of the distributional information. For more manageable datasets, this tool appears to be beneficial.
  - **Skip-Gram**
    The opposite of CBOW, skip-gram uses the target words to create predictions about the surrounding context words. The statistical performance of skip-gram increases with the amount of available data since it considers each context-target combination to be a new observation.
- **Fast text**
  Fast text is simply an extension of the word2vec model, which decomposes each word into a series of character n-grams. To put it another way, the vector representation of any given word is the same as the sum of its character n-gram components. In contrast to Fast Text, Word2vec's embeddings may still be excellent for uncommon words that appear seldom in the training corpus since their character n-grams are shared with other words.
- **N-Grams**
  Sentences with precisely N words are called N-Grams. Being a statement broken down into its simplest pieces, a Unigram is one possible definition. A "bigram" is a device that may be used to break down a sentence into its individual words. A trigram is a set of three words that may be used to make a whole sentence.
- **RELU**
  All digits after the decimal point are set to zero at this level. Any input to the node less than the threshold value will cause the function to have no impact. If the input is negative, then there will be no output. Beyond a particular threshold, the exogenous variable begins to linearly correlate with the dependent variable. By avoiding a zero sum, this activation function may hasten the training of a deep neural network's data set in a way that other activation functions cannot.
- **SELU**
  As its name suggests, SELU is a neural network that automatically adjusts its activation function to achieve

optimal performance. It's an offshoot of the European Language Union with the same goal of expanding language learning throughout the continent. As SELU is capable of self-normalization, we may rest certain that the outputs will be consistent regardless of the parameters. As a result, further levels of Batch Normalization are pointless.

- **Exponential Linear Unit (ELU)**
  The Exponential Linear Unit (ELU) is an activation function that shortens the time it takes to train a model while simultaneously increasing its accuracy.
- **Support Vector Machine (SVM)**
  The SVM classifier uses hyper planes in a high-dimensional space to do classification based on supervised learning. The SVM is a non-probabilistic linear classifier. The SVM and NN both have many similar features. SVM analyzes the input data to forecast to which class each row most closely belongs.
- **Naive Bayes**
  Classifiers of the Naive Bayes (NB) kind base their decisions on the Bayes Theorem. By comparing the likelihood of one occurrence to that of another, this classifier arrives at a conclusion on the former's likeliness. For issues that lend themselves to simple categorization along linear dimensions, the Naive Bayes classifier shines. In addition, it works well for problems that cannot be partitioned along a linear axis.
- **KNN**
  By comparing a new data point to its neighbors, K-Nearest Neighbors (KNN) may automatically place it into one of many predefined categories.
- **CNN**
  You should put out your best effort to get an accurate numerical representation of the text. One would think that in a perfect world, when words are transformed into vectors, man minus woman would equal monarch and queen. Word2vec algorithms make an effort in this direction, albeit they aren't always successful. Following that, a vector sequence is used to teach the Network how to detect pictures rather than words. Convolution neural networks (CNNs) are often used in this setting (spacy uses CNNs for sentiment analysis). Contrary to popular belief, CNNs aren't solely useful for removing image-based characteristics. When dealing with lengthy texts or huge embedding vectors, its compression skills come in quite handy. The performance of CNNs tends to be higher than that of RNNs when it comes to linguistic NLP tasks. This is due to the fact that CNNs are superior to RNNs at interpreting single words.
- **LSTM**
  Yet, neural network-based algorithms can learn to recognize phrases based on their meaning, whereas word-based systems can only learn to categorize based on statistical averages. Each and every one of those words has its own subset of classifications.
  The usage of LSTM represents a novel strategy. Because of the LSTM approach, we can now label a dataset with a combination of terms rather than a single one. Useful for computer programs trying to understand human speech. The best possible output class may be generated if an LSTM model is developed with appropriate embedding and encoding layers, which in turn enables the model to find the meaning of the input text.

D. CLASSIFING ALGORITHMS

- **Random Forest**
  Several decision trees may be combined into one "Random Forest" (RF). Random Forests are prediction ensembles made up of several individual decision trees. Decision trees do well on the training set but badly on the test set because they over fit the data used to create them. Due to the usage of several decision trees, Random Forests effectively reduce over-fitting.

IV.RESULTS

A. Precision (positive predictive value)

A preference for the value of 1 (high). When both the numerator and denominator are 0, the precision equals 1 (TP = TP + FP). Since the denominator becomes greater than the numerator as FP rises, precision decreases.

$$Precision = \frac{TP}{TP + FP}$$

B. Recall

In most cases, a good classifier will have a recall value very near to 1. (High). When the numerator and denominator are the same in TP = TP + FN, the recall is 1 and the FN is 0. Since the denominator is greater as FN increases, memory recall suffers.

$$Recall = \frac{TP}{TP + FN}$$

C. F1 Score

If accuracy and memory are both perfect, the F1 score will be 1. Having high F1 requires stellar performance in both accuracy and recall. The F1 score is more advantageous than accuracy since it reflects a harmonic mean of the two metrics (precision and recall).

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

D. Accuracy

The accuracy measures how many data examples were successfully categorized out of a total of data instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

V. Conclusion

- By using parameters like Quick text embedding with 200 dimensions and Windows capacity 5, the machine learning method Naive Bayes is able to get an accuracy of 74%. When given parameters like a 2-layer LSTM with Selu activation in the first 2 layers and softmax activation in the final layer, LSTM achieves the highest accuracy (78%) among the deep learning algorithms.
- These results during training may benefit from further information. This may serve as an example in a number of ways.

VI. Future Work

- It will be required in the future (1) to collect larger datasets that include information from a variety of fields in order to enhance the outcomes from training these models.
- The second recommendation is that ensemble techniques be investigated in the future since they have the potential to provide even better outcomes than those shown here.
- Lastly, in the future, it will be crucial to deal with pre-trained models for classifications; thus, Transformers are something we'll need to work with if we want to improve our outcomes.

TABLE I
SVM Linear results

Feature	Precision	Recall	F-measure	Acc
ngram, n=3	0.714	0.7	0.696	0.7
ngram, n=4	0.688	0.628	0.6	0.628
TF-IDF	0.742	0.742	0.742	0.742
3-gram+TF-IDF	0.739	0.73	0.727	0.73
4-gram+TF-IDF	0.701	0.634	0.597	0.634
Word2Vec-,CBOW, 100dim, win-3	0.591	0.526	0.408	0.526
Word2Vec-,CBOW, 100dim, win-5	0.609	0.532	0.419	0.532
Word2Vec-,skip-gram, 100dim, win-3	0.694	0.688	0.685	0.688
Word2Vec-,skip-gram, 100dim, win-5	0.708	0.706	0.705	0.706
Word2Vec-,CBOW, 300dim, win-3	0.259	0.508	0.343	0.508
Word2Vec-,CBOW, 300dim, win-5	0.751	0.514	0.356	0.514
Word2Vec-,skip-gram, 300dim, win-3	0.642	0.622	0.612	0.622
Word2Vec-,skip-gram, 300dim, win-5	0.701	0.7	0.699	0.7
Fasttext, 100dim, win-3	0.711	0.7	0.695	0.7
Fasttext, 200dim, win-3	0.732	0.724	0.721	0.724
Fasttext, 300dim, win-3	0.71	0.688	0.678	0.688
Fasttext, 100dim, win-5	0.727	0.724	0.723	0.724
Fasttext, 200dim, win-5	0.735	0.73	0.728	0.73
Fasttext, 300dim, win-5	0.73	0.724	0.722	0.724

TABLE II
SVM Quadratic results

Feature	Precision	Recall	F-measure	Acc
ngram, n=3	0.241	0.491	0.323	0.491
ngram, n=4	0.241	0.491	0.323	0.491
TF-IDF	0.732	0.73	0.73	0.73
3-gram+TF-IDF	0.758	0.538	0.406	0.538
4-gram+TF-IDF	0.758	0.538	0.406	0.538
Word2Vec-,CBOW, 100dim, window-3	0.751	0.514	0.356	0.514
Word2Vec-,CBOW, 100dim, window-5	0.751	0.514	0.356	0.514
Word2Vec-,skip-gram, 100dim, window-3	0.691	0.688	0.686	0.688
Word2Vec-,skip-gram, 100dim, window-5	0.7	0.7	0.7	0.7
Word2Vec-,CBOW, 300dim, window-3	0.614	0.526	0.4	0.526
Word2Vec-,CBOW, 300dim, window-5	0.751	0.514	0.356	0.514
Word2Vec-,skip-gram, 300dim, window-3	0.598	0.598	0.598	0.598
Word2Vec-,skip-gram, 300dim, window-5	0.679	0.676	0.675	0.676
Fasttext, 100dim, win-3	0.649	0.58	0.519	0.58
Fasttext, 200dim, win-3	0.649	0.58	0.519	0.58
Fasttext, 300dim, win-3	0.649	0.58	0.519	0.58
Fasttext, 100dim, win-5	0.67	0.67	0.67	0.67
Fasttext, 200dim, win-5	0.696	0.694	0.693	0.694
Fasttext, 300dim, win-5	0.683	0.682	0.682	0.682

TABLE III
SVM polynomial results

Feature	Precision	Recall	F-measure	Acc
ngram, n=3	0.241	0.491	0.323	0.491
ngram, n=4	0.241	0.491	0.323	0.491
TF-IDF	0.766	0.766	0.766	0.766
3-gram+TF-IDF	0.688	0.538	0.415	0.538
4-gram+TF-IDF	0.758	0.538	0.406	0.538
Word2Vec-,CBOW, 100dim, window-3	0.754	0.526	0.382	0.526
Word2Vec-,CBOW, 100dim, window-5	0.754	0.526	0.382	0.526
Word2Vec-,skip-gram, 100dim, window-3	0.671	0.67	0.67	0.67
Word2Vec-,skip-gram, 100dim, window-5	0.694	0.694	0.694	0.694
Word2Vec-,CBOW, 300dim, window-3	0.488	0.502	0.408	0.502
Word2Vec-,CBOW, 300dim, window-5	0.753	0.52	0.369	0.52
Word2Vec-,skip-gram, 300dim, window-3	0.581	0.58	0.58	0.58
Word2Vec-,skip-gram, 300dim, window-5	0.677	0.676	0.676	0.676
Fasttext, 100dim, win-3	0.662	0.592	0.537	0.592
Fasttext, 200dim, win-3	0.651	0.61	0.577	0.61
Fasttext, 300dim, win-3	0.656	0.586	0.528	0.586
Fasttext, 100dim, win-5	0.696	0.688	0.684	0.688
Fasttext, 200dim, win-5	0.679	0.676	0.674	0.676
Fasttext, 300dim, win-5	0.649	0.64	0.633	0.64

TABLE IV
Random forest results

Feature	Precision	Recall	F-measure	Acc
ngram, n=3	0.731	0.652	0.623	0.652
ngram, n=4	0.694	0.658	0.644	0.658
TF-IDF	0.748	0.748	0.748	0.748
3-gram+TF-IDF	0.768	0.562	0.465	0.562
4-gram+TF-IDF	0.527	0.526	0.505	0.526
Word2Vec-,CBOW, 100dim, window-3	0.653	0.652	0.652	0.652
Word2Vec-,CBOW, 100dim, window-5	0.665	0.664	0.664	0.664
Word2Vec-,skip-gram, 100dim, window-3	0.727	0.724	0.723	0.724
Word2Vec-,skip-gram, 100dim, window-5	0.689	0.688	0.687	0.688
Word2Vec-,CBOW, 300dim, window-3	0.656	0.526	0.391	0.526
Word2Vec-,CBOW, 300dim, window-5	0.658	0.658	0.658	0.658
Word2Vec-,skip-gram, 300dim, window-3	0.464	0.467	0.461	0.467
Word2Vec-,skip-gram, 300dim, window-5	0.7	0.7	0.7	0.7
Fasttext, 100dim, win-3	0.712	0.712	0.712	0.712
Fasttext, 200dim, win-3	0.76	0.76	0.76	0.76
Fasttext, 300dim, win-3	0.724	0.724	0.724	0.724
Fasttext, 100dim, win-5	0.73	0.73	0.73	0.73
Fasttext, 200dim, win-5	0.724	0.724	0.724	0.724
Fasttext, 300dim, win-5	0.724	0.724	0.724	0.724

TABLE V KNN
(K=15) RESULTS

Feature	Precision	Recall	F-measure	Acc
ngram, n=3	0.241	0.491	0.323	0.491
ngram, n=4	0.241	0.491	0.323	0.491
TF-IDF	0.725	0.706	0.699	0.706
3-gram+TF-IDF	0.705	0.7	0.699	0.7
4-gram+TF-IDF	0.7	0.634	0.605	0.634
Word2Vec-,CBOW, 100dim, window-3	0.527	0.526	0.526	0.526
Word2Vec-,CBOW, 100dim, window-5	0.509	0.508	0.508	0.508
Word2Vec-,skip-gram, 100dim, window-3	0.656	0.652	0.649	0.652
Word2Vec-,skip-gram, 100dim, window-5	0.693	0.688	0.685	0.688
Word2Vec-,CBOW, 300dim, window-3	0.467	0.467	0.464	0.467
Word2Vec-,CBOW, 300dim, window-5	0.484	0.485	0.484	0.485
Word2Vec-,skip-gram, 300dim, window-3	0.628	0.592	0.556	0.592
Word2Vec-,skip-gram, 300dim, window-5	0.672	0.67	0.668	0.67
Fasttext, 100dim, win-3	0.688	0.688	0.688	0.688
Fasttext, 200dim, win-3	0.701	0.7	0.7	0.7
Fasttext, 300dim, win-3	0.719	0.718	0.717	0.718
Fasttext, 100dim, win-5	0.701	0.7	0.7	0.7
Fasttext, 200dim, win-5	0.706	0.706	0.706	0.706
Fasttext, 300dim, win-5	0.73	0.73	0.73	0.73

TABLE VI
NAIVE BAYES RESULTS

Feature	Precision	Recall	F-measure	Acc
ngram, n=3	0.704	0.7	0.698	0.7
ngram, n=4	0.689	0.67	0.664	0.67
TF-IDF	0.641	0.64	0.64	0.64
3-gram+TF-IDF	0.705	0.7	0.698	0.7
4-gram+TF-IDF	0.681	0.664	0.658	0.664
Word2Vec-,CBOW, 100dim, window-3	0.505	0.508	0.469	0.508
Word2Vec-,CBOW, 100dim, window-5	0.505	0.508	0.469	0.508
Word2Vec-,skip-gram, 100dim, window-3	0.717	0.712	0.71	0.712
Word2Vec-,skip-gram, 100dim, window-5	0.76	0.76	0.76	0.76
Word2Vec-,CBOW, 300dim, window-3	0.241	0.491	0.323	0.491
Word2Vec-,CBOW, 300dim, window-5	0.498	0.502	0.471	0.502
Word2Vec-,skip-gram, 300dim, window-3	0.569	0.568	0.565	0.568
Word2Vec-,skip-gram, 300dim, window-5	0.726	0.724	0.724	0.724
Fasttext, 100dim, win-3	0.724	0.724	0.724	0.724
Fasttext, 200dim, win-3	0.76	0.76	0.76	0.76
Fasttext, 300dim, win-3	0.724	0.724	0.724	0.724
Fasttext, 100dim, win-5	0.79	0.79	0.79	0.79
Fasttext, 200dim, win-5	0.748	0.748	0.748	0.748
Fasttext, 300dim, win-5	0.719	0.718	0.718	0.718

TABLE VII
ACTIVATION RELU RESULTS

Feature	Precision	Recall	F-measure	Acc
LSTM- 2 layer	0.223	0.5	0.309	0.44
LSTM- 3 layer	0.223	0.5	0.309	0.44
1D-CNN- 1 layer	0.662	0.624	0.581	0.597
1D-CNN- 2 layer	0.665	0.636	0.601	0.61
BiLSTM- 2 layer	0.223	0.5	0.309	0.447
BiLSTM- 3 layer	0.223	0.5	0.309	0.447
GRU- 2 layer	0.768	0.771	0.768	0.768
GRU- 3 layer	0.728	0.728	0.728	0.731

TABLE VIII
ACTIVATION ELU RESULTS

Feature	Precision	Recall	F-measure	Acc
LSTM- 2 layer	0.223	0.5	0.309	0.447
LSTM- 3 layer	0.223	0.5	0.309	0.447
1D-CNN- 1 layer	0.628	0.605	0.57	0.582
1D-CNN- 2 layer	0.742	0.741	0.731	0.731
BiLSTM- 2 layer	0.223	0.5	0.309	0.44
BiLSTM- 3 layer	0.223	0.5	0.309	0.44
GRU- 2 layer	0.773	0.775	0.774	0.776
GRU- 3 layer	0.736	0.738	0.737	0.738

TABLE IX
ACTIVATION SELU RESULTS

Feature	Precision	Recall	F-measure	A
LSTM- 2 layer	0.785	0.788	0.783	(
LSTM- 3 layer	0.636	0.634	0.635	(
1D-CNN- 1 layer	0.691	0.64	0.594	(
1D-CNN- 2 layer	0.832	0.789	0.794	(
BiLSTM- 2 layer	0.743	0.703	0.671	(
BiLSTM- 3 layer	0.223	0.5	0.309	(
GRU- 2 layer	0.775	0.778	0.775	(
GRU- 3 layer	0.781	0.783	0.782	(

REFERENCES

1. *O. Araque, I. Corcuera-Platas, J. F. Sanchez-Rada, and C. A. Igle- sias,"Enhancing deep learning sentiment analysis with ensemble tech- niques in social applications,"Expert Systems with Applications, vol. 77,pp.236-246, 2017.*
2. *Tammina, "A Hybrid Learning approach for entiment Classifica-tion in Telugu Language," 2020 International Conference on Artifi-cial Intelligence and Signal Processing (AISP), 2020, pp. 1-6, doi: 10.1109/AISP48273.2020.9073109*
3. *Li-Ping Jing, Hou-Kuan Huang, and Hong-Bo Shi, "Improved feature selection approach tfidf in text mining,"in Proceedings. International*
4. *Conference on Machine Learning and Cybernetics, vol. 2, Nov 2002,pp. 944-946 vol.2.*
5. *L. Khan, A. Amjad, N. Ashraf, H. -T. Chang and A. Gelbukh, "Urdu Sentiment Analysis With Deep Learning Methods," in IEEE Access, vol.9, pp. 97803-97812, 2021, doi: 10.1109/ACCESS.2021.3093078*
6. *B. Trstenjak, S. Mikac, and D. Donko, "Knn with tf-idf based framework for text categorization," Procedia Engineering, vol. 69, pp. 1356– 1364, 2014, 24th DAAAM International Symposium on IntelligentManufac- turing and Automation, 2013.*
7. *R. Naidu, S. K. Bharti, K. S. Babu, and R. K. Mohapatra, "Sentiment analysis using telugu sentiwordnet,"in 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSP NET), March 2017, pp. 666-670.*
8. *S. Anbukkarasi and S. Varadhaganapathy, "Analyzing Sentiment in Tamil Tweets using Deep Neural Network," 2020 Fourth International Conference on Computing Methodologies and Communication (IC-CMC), 2020, pp. 449-453, doi: 10.1109/ICCMC48092.2020.ICCMC- 00084.*
9. *M, J. Vala, and P. Balani, "A survey on sentiment analysis algorithmsfor opinion mining," International Journal of Computer Applications, vol. 133, pp. 7–11, 01 2016.*
10. *Mohana Pranadeep potti,ManneDineshKumar,Nagabhyrava Saswanth Ram,P.V.R.Sandeep,P.R.KRISHNAPRASAD, "Sentiment Analysis On Movie Reviews Using NAÏVE BAYES Classifier", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.6, Issue 1, pp.777-781, March 2018, Available at :http://www.ijcrt.org/papers/IJCRT1801641.pdf*
11. *L. Khan, A. Amjad, N. Ashraf, H. -T. Chang and A. Gelbukh, "Urdu Sentiment Analysis With Deep Learning Methods," in IEEE Access, vol.9, pp. 97803-97812, 2021, doi: 10.1109/ACCESS.2021.3093078*
12. *Mukku, Sandeep. (2017). Sentiment Analysis for Telugu Language. 10.13140/RG.2.2.22769.38243.Sentiraama Corpus by Gangula Rama Rohit Reddy, Radhika Mamidi. Language Technologies Research Centre, KCIS, IIIT Hyderabad. ltrc.iiit.ac.in/showfile.php?filename=downloads/sentiraama/*
13. *Mukku, S.S., Choudhary, N., Mamidi, R. (2016). Enhanced Sentiment Classification of Telugu Text using ML Techniques. SAAIP@IJCAI.*
14. *S. Boddupalli, A. S. Saranya, U. Mundra, P. Dasam and P. Sriram, "Sentiment Analysis of Telugu data and comparing advanced ensemble techniques using different text processing methods," 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, 2019, pp. 1-6, doi: 10.1109/ICCUBEA47591.2019.9128877.*