Enhancing Telugu News Understanding: Comparative Study of ML Algorithms for Category Prediction

Manish Rama Gopal Nadella, Venkata Krishna Rayalu Garapati, Eswar Sudhan S.K., Gouthami Jangala, Soman K P, and **Sachin Kumar S**
Amrita School of Artificial Intelligence, Coimbatore,
Amrita Vishwa Vidyapeetham, India.
s_sachinkumar@cb.amrita.edu, manishnadella03@gmail.com

Abstract

Telugu, a widely used language in India, boasts a substantial audience and an extensive repository of news content. Predicting the categories of Telugu news articles not only streamlines organization but also facilitates trend analysis, targeted advertising, and personalized recommendations. This study endeavors to identify the optimal approach for precise Telugu news category prediction, by contrasting diverse machine learning (ML) methods including support vector machines (SVM), random forests, and naive Bayes. Performance metrics like accuracy, precision, recall, and F1-score are employed to gauge algorithm efficacy. This comparative exploration addresses the intricacies of the Telugu language, contributing insights to the field of news category prediction. The study's implications extend to enhancing news organization and recommendation systems for Telugu-speaking consumers, delivering tailored and pertinent news experiences. Our findings underscore that, while other models warrant further investigation, the combination of W2Vec-skip gram and polynomial SVM emerges as the most proficient choice.

1 Introduction

News is the latest information about recent developments and events that are relevant to the general audience (Sundarababu et al. (2020)). It is distributed using a variety of media and covers a broad range of themes. News serves to inform the public, encourage openness, and facilitate informed decision making. The categorization and prediction of news articles have become crucial in the quickly changing environment of information transmission for effective organization and improved user experience. The challenge of predicting news category has significantly advanced with the introduction of machine learning (ML) techniques. We compare many machines learning (ML) techniques for predicting Telugu news category in this research (Sultana et al. (2021)).

One of the most widely used languages in India is Telugu (Sultana et al. (2021)), which has a large audience and an extensive library of news stories. Predicting the categories of Telugu news articles (Boddupalli et al. (2019)) not only allows for effective organization but also makes it possible for trend research, advertising that is specifically targeted, and personalized suggestions. We seek to determine the most efficient strategy for precise Telugu news category prediction by examining and contrasting various ML algorithms.

The comparative study will include numerous kinds of machine learning (ML) techniques, such as support vector machines (SVM), random forests, and naive Bayes (Sheth et al. (2022)). The effectiveness of these algorithms in correctly classifying Telugu news articles will be evaluated based on their performance indicators, such as accuracy, precision, recall, and F1- score.

By performing this comparative analysis, we aim to add to the body of knowledge already available on news category prediction while taking into account the special difficulties and complexities of the Telugu language. The results of this study could improve how Telugu news articles are stored and found, giving users access to more relevant and individualized news consumption experiences (Kumar et al. (2022)).

With the ultimate goal of increasing the effectiveness and efficiency of news organization and recommendation systems for Telugu-speaking users, this study intends to shed light on the comparative analysis of various ML algorithms for Telugu news category prediction.

2 Related Works

In Sheth et al. (2022) a thorough comparative analysis was conducted to assess the effectiveness of various data mining classification techniques. The major goal was to evaluate the performance of the Naive Bayes, Support Vector Machines (SVM), De-

Third Workshop on Speech and Language Technologies for Dravidian Languages,
pages 108–115, Varna, Bulgaria, Sep 7, 2023.
https://doi.org/10.26615/978-954-452-085-4_014

cision Trees, and K-Nearest Neighbor classifiers. Accuracy, recall, precision, and F1 score were the evaluation criteria, and several datasets were used for the evaluation. The study's results consistently showed that, in terms of these evaluation measures, the Naive Bayes method performed better than the other classifiers. In comparison to the other algorithms, it regularly shows greater accuracy, recall, precision, and F1 score values. In the comparison, SVM took second place, K-Nearest Neighbor came in third, and Decision Trees came in fourth as the top classifier. These findings emphasize the importance of carefully choosing the right classifier based on the unique properties of the dataset and the surrounding circumstances. It highlights the significance of avoiding a one-size-fits-all strategy and instead taking into account the particular requirements and subtleties of the current challenge. In data mining jobs, selecting the most appropriate classifier based on the particular dataset and context can produce more accurate and dependable results.

In (Sundarababu et al. (2020)), the authors address challenges in mining large electronic data. They focus on accuracy and the Zero Frequency Problem. They propose using Multinomial Naive Bayes Algorithm to forecast online story popularity. Python is highlighted for AI due to adaptability. While Naive Bayes is used in various areas, its assumption of independence is a drawback. Yet, it handles many features, is simple, and offers quick training. Their aim is to enhance news popularity prediction using Multinomial Naive Bayes and discuss its pros and cons in electronic data analysis, noting Python's suitability for AI.

In (Jang et al., 2019) Beakcheol Jang et al., The goal of the study is to assess word2vec Convolutional Neural Networks' (CNNs') performance in classifying news articles and tweets as related or unrelated. In particular, the study looks into how well the word embedding techniques CBOW (Continuous Bag-of-Words) and Skip-gram perform while creating CNN models for classification. The study's experimental analysis shows that the use of word2vec considerably improves the classification models' accuracy. Interesting results are found when the performance of the CBOW and Skip-gram models are compared. When applied to tweets, the Skip-gram model performs better, whereas the CBOW model performs better and more consistently when applied to news items. This performance disparity shows that the word embedding approach selected should be adapted to the unique characteristics of the text under study. Word2vec- enabled CNN models perform better than models without word embedding. These findings help us comprehend how the choice of word embedding models affects CNN-based classification for news articles and tweets. The study emphasizes the potential of utilizing cutting-edge neural network approaches for efficient text categorization in the context of news and social media analysis by highlighting the benefits of word2vec in enhancing classification accuracy.

In Sultana et al. (2021), the authors explore Telugu news sentiment categorization using machine learning. They classify news into categories and sentiment (positive, negative, neutral). Various models are compared based on accuracy, precision, recall, and F1-score. Sentiment analysis's importance for Telugu news, addressing regional languages like Telugu, is highlighted. Techniques like Naive Bayes, Random Forest, SVM, among others, are used. A framework with feature selection, training, testing, and performance evaluation is presented. Passive Aggressive Classifier stands out with 80

In most of the researches, the focus is much tilted towards the various algorithms rather than the multiple features that are associated with natural language processing, this leaves a significant gap for us to make this research a vital part.

3 Feature Extraction and Classification Algorithms

3.1 N_gram

Natural language processing relies heavily on N-grams (Cavnar and Trenkle (2001)). They are groups of (n) items retrieved from text, such as words or characters. N-grams expose word relationships, aid in word prediction, discover common patterns, and make realistic writing. They evaluate the likelihood of word sequences in language modelling to ensure coherent and fluent output.

3.2 Tf_idf

TF-IDF (Sammut and Webb (2010)) is vital in text mining and retrieval. It combines term frequency (TF) and inverse document frequency (IDF) to gauge phrase importance across documents. TF measures word frequency in a doc, IDF gauges term rarity in the collection. TF-IDF shows term

relevance compared to full set. It's used for ranking, categorization, keyword extraction, and info retrieval, aiding term identification, text categorization, and info extraction.

3.3 Fasttext

FastText (Bojanowski et al. (2017)) is a notable NLP tool for text tasks and word representation. It creates word vectors using character n-grams, aiding with rare words and semantics. It's efficient in training and inference, supports various loss functions, excelling in tasks like sentiment analysis. FastText offers a Python API and CLI, easy to integrate. Being open source, it's customizable for experimentation by academics and professionals.

3.4 Word2Vec-CBOW

Word2Vec (Jang et al. (2019)) is a popular word embedding technique in NLP. Using dense vectors, it represents words. Continuous Bag of Words (CBOW) predicts a word from its context, adjusting embeddings for semantic links. It's efficient for local context-reliant tasks like sentiment analysis. CBOW is used in sentiment analysis, text categorization, and info retrieval. Compared to Skip-gram (better with rare words but slower), CBOW might struggle with uncommon words.

3.5 Word2Vec-skip gram

Word2Vec (Jang et al. (2019)) is neural network-based for dense word embeddings. Skip-gram, a variant, predicts context words from a target. It learns from large text data, producing embeddings for semantics. Used in tasks like similarity and classification. Skip-gram excels with rare words and links but needs more data due to computational intensity.

3.6 Support Vector Machines

Support Vector Machine (Cortes and Vapnik (2009)) classifies by finding a hyperplane in the feature space for linear separation, maximizing margin between classes. Linear SVM suits linearly separable data but struggles with complex cases. Polynomial and quadratic SVMs tackle this using kernel functions, capturing nonlinear patterns. Polynomial kernel involves raising dot product, quadratic squaring it, enabling complex interactions. Polynomial and quadratic SVMs offer flexible decision boundaries but demand careful kernel choice and regularization to avoid overfitting. Quadratic SVMs handle intricate patterns but are computationally expensive, needing regularization for better generalization.

3.7 KNN

K-Nearest Neighbors (KNN) is for classification and regression. It uses similar data points, based on distance, to predict outcomes. KNN retains the training dataset, finds k closest neighbors for a new point. Majority class among neighbors is used for classification, average/median for regression. Picking k, distance metric (like Euclidean), scaling, handling imbalanced data, and addressing dimensionality through selection/reduction are key KNN considerations.

3.8 Multinomial Naive Bayes

Multinomial Naive Bayes is a text classification method assuming feature independence within a class. It's effective for discrete features like word frequencies. It models class probabilities from features and predicts based on highest probability. It's commonly used in text categorization with representations like bag-of-words or TF-IDF. Despite assuming feature independence and sensitivity to imbalanced data, it's popular due to simplicity and low computational needs, delivering competitive results.

3.9 Random Forest

Random Forest (Louppe (2015)) is an ensemble technique for classification and regression. It uses multiple decision trees that vote or average predictions for better accuracy.The final prediction is determined by majority voting. Trees are trained on different data subsets with random feature selection. It boosts performance using bootstrap sampling. Gini or entropy measures guide node splitting. It's popular for complex data due to robustness.

4 Dataset and Preprocessing

We used an in-house dataset for performing the experiments in this paper. The dataset was prepared by scrapping (Bhardwaj et al. (2021)) an online Telugu news website, gulte.com. This dataset consists of 6 classes with a total of 38637 news articles (Sachin Kumar et al. (2020)). Figure 1 shows the category distribution in the dataset. Figure 2 shows the samples for each category from the dataset.

The data which has been scrapped from the websites consists of several unwanted characters, white

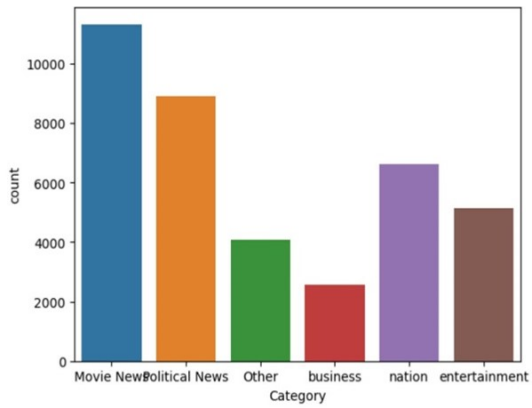

Figure 1: Category Distribution

Category	Content	Translation
Movie	నేడు పుట్టినరోజు జరుపుకుంటున్న రామ్ చరణ్	Ram Charan celebrates his birthday today
Political	ఈ నెల 31 వరకు టోటల్ లాక్ డౌన్ ప్రకటించింది ప్రభుత్వం	The government has announced a total lockdown till The 31st of this month
Other	కారణం ఎదైనా కావొచ్చు..స్మోక్ చేసే అలవాటు ఉందా?	Whatever the reason may be.. Do you have the habit of smoking?
Business	భారీగా తగ్గుముఖం పట్టిన ధర	Crude oil prices fall sharply
Nation	హిమాచల్ సీఎం రేసులో నడ్డా	Nadda in race for Himachal CM's post
Entertainment	ఇష్టపడిన అమ్మాయి వెంటపడి, తన ఇష్టాయిష్టాలను తెలుసు	Chased by a girl she liked and knew her likes and dislikes

Figure 2: Sample news for each category from dataset

spaces, html tags or some numbers. These kinds of characters have been removed from the entire dataset (Varshini et al. (2023)), furthermore we have also removed stop words, belonging to Telugu language and also perform stemming.

5 Experimental Setup

The pre-processed dataset has been used to perform all the experiments in this paper. We trained Word2vec and fasttext on the pre-processed dataset to get the word embeddings instead of using the pre-trained models. Linear SVM, Quadratic SVM, Polynomial SVM, Random Forest, KNN and Multinomial Naive Bayes models are trained for each of the feature extraction methods.

For feature extraction we have used W2Vec-SG, W2Vec-CBOW, n-gram, Fast Text and TF-IDF. The dimension of feature vectors for n-gram and TF-IDF, is 10,000 whereas for other features W2Vec-SG, W2Vec-CBOW, and Fast Text it is 100.The feature visualization for vectors of such large dimensions can be done through t-SNE. t-SNE is a technique for revealing patterns and correlations in word vectors by visualizing them in a lower-dimensional environment. It entails gathering word vectors, using t-SNE to reduce dimensionality, then presenting the results on a scatter plot. Figure 3 shows t-SNE visualization for TF-IDF word vectors taken for 500 samples where each color represents a category. To understand the word vectors in a better way, we have also used an online embeddings projector for which the result is shown in Figure 4 where each circle represents a word vector and words which are similar are grouped together.

Raw Words	35881750
Effective Words	354422
Vocab	477946
Vector size	100
alpha	0.025
window	3
epoch	4

Table 1: Parameters for Word2Vec CBOW-model

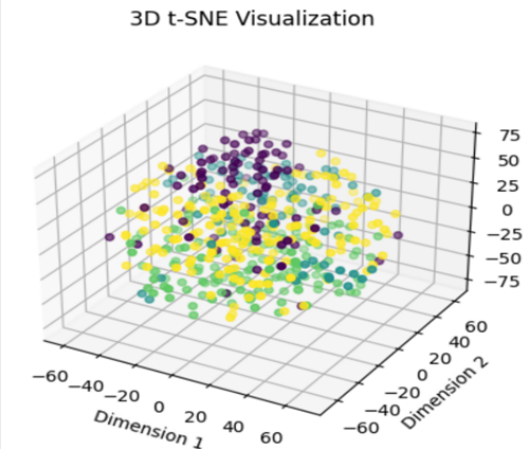

Figure 3: Example t-SNE visualization

All the models initially take a set of parameters and train on these set of parameters to find the best fit, then the model is evaluated on the best fit parameters. At N-fold cross validation has also been performed on each of the machine learning algorithms to evaluate the algorithm's performance for unseen data. Table 1 shows the inputs and parameters given to train the Word2Vec model.

6 Results and Discussion

Refer the Tables 2 to 8 for the performance metrics and cross Validation score for various algorithms against popular features. After Studying the performance metrics of all the algorithms, we have observed that quadratic SVM for n-gram is not performing up to the mark. Figure 5 shows the confusion matrix accordingly.

We can observe that there is a huge misclassification for category consisting of news articles that are contained in the 'Nation' class. Furthermore, the number of true classifications or correct

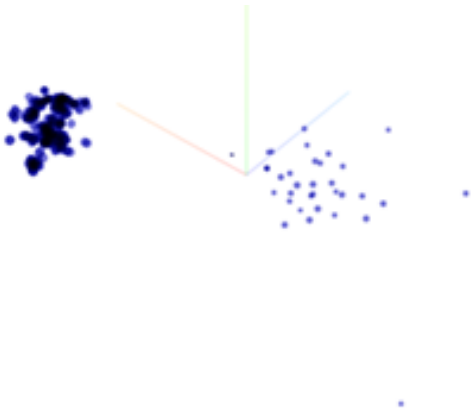

Figure 4: Smoothened Visualization

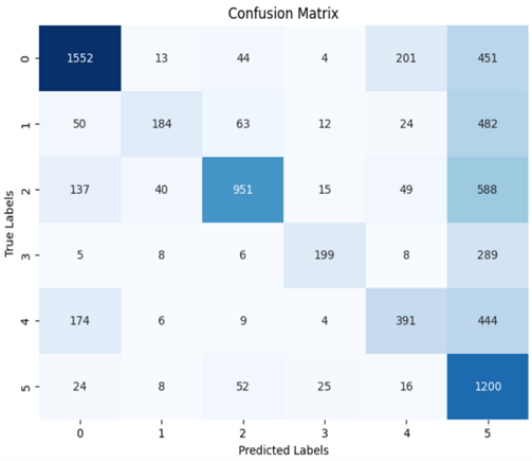

Figure 5: Confusion Matrix for Quadratic SVM using n-gram

classifications for the categories Other, Business, Entertainment are relatively less. This can be because of two factors, one being the huge imbalance in the dataset and the other can be because of use of similar words in the articles, which leads to misclassification.

Figure 6 Confusion Matrix for Word2Vec-CBOW for same method, Quadratic SVM. It is clearly observed that the model performed much better and the number of misclassifications for the category nation are very less compared to that of n-gram, we can also see that the number of true classifications for the categories Others, Business, Entertainment is relatively more compared to n-gram. This can be because Word2Vec model was able to generate better word vectors for the given corpus.

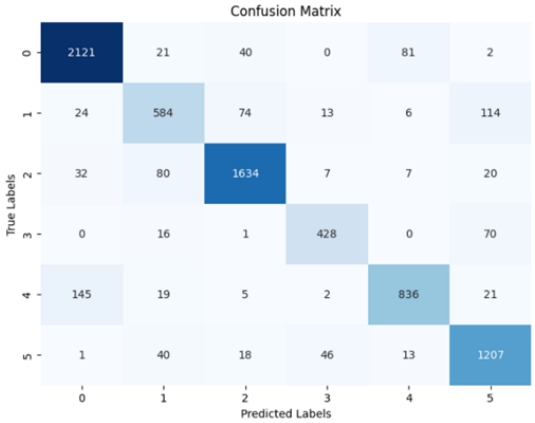

Figure 6: Confusion Matrix for Quadratic SVM using Word2Vec-CBOW

7 Conclusion

Word-2-vec CBOW and Skip gram are the two models which showed constant and reasonable performance for all the algorithms except Multinomial Naive Bayes. The performance of Fast-Text was also consistent except for Multinomial Naive Bayes but it under performed when compared to word-2-vec. N-gram showed the third best performance among the feature extraction methods, its performance was better even for Multinomial Naive Bayes, but it showed poor performance for KNN.

Finally, Tf-IDF showed reasonable performance except for Polynomial SVM and KNN. Further it has been observed that changing the parameter value, 'C' in all three types of SVM, resulted in minute increase of model's accuracy. On setting the 'C' value to 10,000 the models showed up to 5%-10% increase in accuracy. In general, a higher 'C' value in SVM results in higher penalty and smaller margin. It also reduces the regularization strength which may lead to overfitting, these can be some of the reasons for improved performance.

The primary goal of this research has been to choose the best classifier and feature extraction pair from the most popular techniques. We have observed that W2Vec-skip gram and polynomial SVM is the best pair for this task. However, other models may be considered in the future work for comparison and selection.

References

Bhavya Bhardwaj, Syed Ishtiyaq Ahmed, J Jaiharie, R Sorabh Dadhich, and M Ganesan. 2021. Web scraping using summarization and named entity

	Performance metrics				Cross-Validation score			
	Precision	**Recall**	**Accuracy**	**F1-Score**	**Mean-Precision**	**Mean-Recall**	**Mean-Accuracy**	**Mean F1-Score**
W2Vec-SG	0.86	0.85	0.87	0.85	0.85	0.80	0.85	0.81
W2Vec-CBOW	0.81	0.72	0.79	0.77	0.84	0.77	0.83	0.79
n-gram	0.57	0.58	0.62	0.57	0.55	0.55	0.60	0.55
Fast Text	0.68	0.60	0.66	0.61	0.70	0.65	0.71	0.66
TF-IDF	0.57	0.57	0.62	0.57	0.56	0.56	0.61	0.56

Table 2: Performance of Linear SVM

	Performance metrics				Cross-Validation score			
	Precision	**Recall**	**Accuracy**	**F1-Score**	**Mean-Precision**	**Mean-Recall**	**Mean-Accuracy**	**Mean F1-Score**
W2Vec-SG	0.89	0.88	0.90	0.88	0.88	0.87	0.89	0.88
W2Vec-CBOW	0.88	0.87	0.89	0.88	0.87	0.85	0.88	0.86
n-gram	0.70	0.32	0.63	0.30	0.66	0.50	0.56	0.51
Fast Text	0.74	0.71	0.76	0.71	0.74	0.70	0.75	0.71
TF-IDF	0.74	0.62	0.71	0.65	0.67	0.56	0.61	0.57

Table 3: Performance of Quadratic SVM

recognition (ner). In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 261–265.

Srikiran Boddupalli, Anitha Sai Saranya, Usha Mundra, Pratyusha Dasam, and Padmamala Sriram. 2019. Sentiment analysis of telugu data and comparing advanced ensemble techniques using different text processing methods. In *2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*, pages 1–6.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information.

William Cavnar and John Trenkle. 2001. N-gram-based text categorization. *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*.

Corinna Cortes and Vladimir Vapnik. 2009. Support-vector networks. *Chem. Biol. Drug Des.*, 297:273–297.

Beakcheol Jang, Inhwan Kim, and Jong Kim. 2019. Word2vec convolutional neural networks for classification of news articles and tweets. *PLOS ONE*, 14:e0220976.

Sruthi S. Kumar, S. Sachin Kumar, and K. P. Soman. 2022. Deep learning-based emotion classification of hindi text from social media. In *Advanced Machine Intelligence and Signal Processing*, pages 535–543, Singapore. Springer Nature Singapore.

Gilles Louppe. 2015. Understanding random forests: From theory to practice.

S. Sachin Kumar, M. Anand Kumar, K. P. Soman, and Prabaharan Poornachandran. 2020. Dynamic mode-based feature with random mapping for sentiment analysis. In *Intelligent Systems, Technologies and Applications*, pages 1–15, Singapore. Springer Singapore.

	Performance metrics				Cross-Validation score			
	Precision	**Recall**	**Accuracy**	**F1-Score**	**Mean-Precision**	**Mean-Recall**	**Mean-Accuracy**	**Mean F1-Score**
W2Vec-SG	0.90	0.89	0.91	0.89	0.90	0.89	0.91	0.89
W2Vec-CBOW	0.89	0.88	0.90	0.88	0.88	0.87	0.89	0.88
n-gram	0.70	0.32	0.36	0.30	0.68	0.30	0.33	0.26
Fast Text	0.75	0.72	0.76	0.73	0.75	0.71	0.76	0.72
TF-IDF	0.44	0.21	0.32	0.17	0.60	0.22	0.20	0.15

Table 4: Performance of Polynomial SVM

	Performance metrics				Cross-Validation score			
	Precision	**Recall**	**Accuracy**	**F1-Score**	**Mean-Precision**	**Mean-Recall**	**Mean-Accuracy**	**Mean F1-Score**
W2Vec-SG	0.87	0.86	0.88	0.86	0.87	0.85	0.88	0.86
W2Vec-CBOW	0.86	0.84	0.87	0.85	0.86	0.84	0.87	0.85
n-gram	0.68	0.64	0.70	0.65	0.60	0.59	0.63	0.59
Fast Text	0.71	0.56	0.67	0.58	0.71	0.56	0.67	0.58
TF-IDF	0.83	0.61	0.73	0.64	0.64	0.60	0.66	0.61

Table 5: Performance of Random Forest Classifier

Claude Sammut and Geoffrey I. Webb, editors. 2010. *TF–IDF*, pages 986–987. Springer US, Boston, MA.

Vraj Sheth, Urvashi Tripathi, and Ankit Sharma. 2022. A comparative analysis of machine learning algorithms for classification purpose. *Procedia Computer Science*, 215:422–431. 4th International Conference on Innovative Data Communication Technology and Application.

J. Sultana, Usha Rani Macigi, and G. Priya. 2021. *Telugu News Data Classification using Machine Learning Approach*, pages 181–194.

Mr Sundarababu, Ch Chandramohan, Mahendra Suthar, Ch Harsha, Lubna Juveria, B Blessy, and Sameer Mohammad. 2020. News classification using machine learning. *SSRN Electronic Journal*, 7:657–660.

Surisetty Hima Varshini, Gottimukkala Sarayu Varma, and Supriya M. 2023. A recognizer and parser for basic sentences in telugu using cyk algorithm. In *2023 3rd International Conference on Intelligent Technologies (CONIT)*, pages 1–5.

	Performance metrics				Cross-Validation score			
	Precision	**Recall**	**Accuracy**	**F1-Score**	**Mean-Precision**	**Mean-Recall**	**Mean-Accuracy**	**Mean F1-Score**
W2Vec-SG	0.88	0.87	0.89	0.87	0.87	0.86	0.89	0.87
W2Vec-CBOW	0.86	0.85	0.87	0.85	0.85	0.84	0.87	0.85
n-gram	0.43	0.43	0.41	0.39	0.42	0.41	0.39	0.37
Fast Text	0.66	0.60	0.68	0.62	0.65	0.59	0.66	0.61
TF-IDF	0.46	0.33	0.28	0.30	0.46	0.31	0.27	0.29

Table 6: Performance of KNN

	Performance metrics				Cross-Validation score			
	Precision	**Recall**	**Accuracy**	**F1-Score**	**Mean-Precision**	**Mean-Recall**	**Mean-Accuracy**	**Mean F1-Score**
W2Vec-SG	0.61	0.58	0.67	0.58	0.62	0.58	0.67	0.58
W2Vec-CBOW	0.64	0.62	0.69	0.62	0.64	0.61	0.68	0.62
n-gram	0.67	0.65	0.70	0.66	0.67	0.65	0.69	0.66
Fast Text	0.46	0.46	0.52	0.45	0.46	0.46	0.52	0.45
TF-IDF	0.67	0.65	0.70	0.66	0.67	0.65	0.69	0.66

Table 7: Performance of Multinomial Naive Bayes

	Single Layer		Two Layers	
	Accuracy	**Validation Accuracy**	**Accuracy**	**Validation Accuracy**
W2Vec-SG	0.86	0.855	0.874	0.876
W2Vec-CBOW	0.846	0.842	0.861	0.858
n-gram	0.925	0.666	0.925	0.667
Fast Text	0.736	0.710	0.745	0.73
TF-IDF	0.965	0.734	0.986	0.783

Table 8: Performance of 1D-CNN