National College of Ireland

Enhancing Natural Language Processing Models for Contextual Understanding in Low-Resource Languages+

MSc Research Project
Masters in Data Analytics

Pesaru Abhijith Reddy
Student ID: X22157131

School of Computing
National College of Ireland

Supervisor: Mr. Arjun Chikkankod

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Pesaru Abhijith Reddy
Student ID:	X22157131
Programme:	Masters in Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Mr. Arjun Chikkankod
Submission Due Date:	14/12/2023
Project Title:	Enhancing Natural Language Processing Models for Contextual Understanding in Low-Resource Languages
Word Count:	6661
Page Count:	17

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Pesaru Abhijit Reddy
Date:	31st January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Natural Language Processing Models for Contextual Understanding in Low-Resource Languages

Pesaru Abhijith Reddy
X22157131

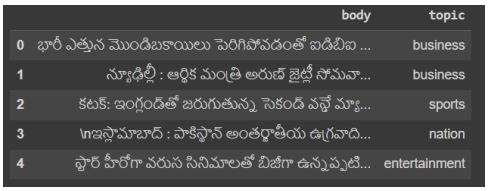
Abstract

This paper is focused on the improvement of language models that have limited linguistic resources, often referred to as "low-resource languages". The goal is to make Natural Language Processing models more effective and accurate in understanding and processing these languages. Natural language processing (NLP) is one of the most advanced fields in the area of Machine Learning (ML) which will help people interact with technology in their own language through various sources like chatbots which will empower people to take suggestions right from day to day to life routines to making future plans in their own language from a pre-trained model. But when it comes to low-resource languages like Telugu which doesn't have many linguistic online resources it makes NLP challenging for such kinds of languages. This Project aims at bridging this gap by using algorithms like Bert, Robert, LSTM along with stacking ensemble models for enhancing Telugu NLP. Through a comprehensive analysis of these techniques, we aim to improve the contextual understanding and overall performance of NLP models for the Telugu language.

1 Introduction

Low-resource language refers to as "less studied, resource-scarce, less computerized, less privileged, less commonly taught, or low density, among other denominations" Singh (2008). In the modern world, technology has become part and parcel of life in our day to day life activities right from smart watch which will not just helps us to keep track of time but also helps in keeping track of how many calories intake we have has through food and how many calories we have burnt through out the day which helps in taking right amount of food intake, this also helps to take answer calls while on the go without the need of mobile phone to mobile apps that helps in going to a newplace by giving directions and reviews and rating of that place. While the technology is progressing tremendously in every aspect and helping people in making their life's easier and better in the same way there is one more field which has an noble aim of helping people communicate with with technology Natural language processing (NLP) which will help people take suggestions from a large corpus of data and also helps in machine translation, sentimental analysis, chatbot and recommendation systems. When the world is seeing remarkable advancements in area of NLP but in the case of low -resource languages like Telugu the story is different, there is not much linguistic data, text corpa or training data available

for training the models. Telugu is one of the Dravidian languages which is mostly spoken in states of Telangana and Andhra Pradesh and this is the 3rd most spoken language in India and fastest growing languages in USA. This paper focuses on challenges and opportunities of enhancing the NLP for Telugu for which we use some pre-trained NLP models such as Bert, Robert, LSTM and stacking ensemble model using all these models as base models. We are using a dataset from Kaggle which has been labeled by Telugu native speaker. It has Telugu news classification dataset which has 5 unique labels which are Business, Editorial, Entertainment, Nation, Sport this has 5 columns and 17312 rows.



	body	topic
0	భారీ ఎత్తున మొండిబకాయిలు పెరిగిపోవడంతో ఐడిబిఐ ...	business
1	న్యూఢిల్లీ : ఆర్థిక మంత్రి అరుణ్ జైట్లీ సోమవా...	business
2	కటక్: ఇంగ్లండితో జరుగుతున్న సెకండ్ వన్డే మ్యా...	sports
3	\nఇస్లామాబాద్ : పాకిస్థాన్ అంతర్జాతీయ ఉగ్రవాది...	nation
4	స్టార్ హీరోగా వరుస సినిమాలతో బిజీగా ఉన్నప్పటి...	entertainment

Figure 1: Telugu dataset sample data

Figure 1 is a sample of the data that we are using for model building where the body column has news in Telugu language and the topic column has the category that the news belongs to. Before proceeding to the models we shall see the some previous work on these topics and then we shall briefly look at the methodologies used and their implementations after this we shall see the results before coming to a conclusion. We shall end this paper by presenting some future work.

1.1 Motivation

The reason choose this topic for research is that the AI and ML are going to an extent where it is becoming part and parcel all the technical devices and applications that we use in our day-to-day life but only few people are able to get the fruits of this advancements because of the language barrier. In this research I would like to extend one of the NLP methods to one of the low resource language Telugu which happens to be my mother tongue as well. Though there are 96 million people who speak and is one of the fastest growing language in USA this has barely has any resources for training in this research I will create any new resources but will apply NLP techniques that will help built or improve NLP technologies in Telugu. Though this paper majorly talks about Telugu and NLP techniques applied for it this can also be extended to other low resource language in the future work.

1.2 Research Question and Objective

Q. Considering the lack of annotated data and linguistic resources, how can we improve the performance of NLP models for contextual understanding in low-resource languages? In order to solve this question there are some objectives that need to be fulfilled.

1) To find the data resources in Telugu languages for training the models it would be good if the dataset is not too small that it is not enough to train the model and it should not be too large where the required computational power is too high. Achieving both of these objectives and finding a quality dataset for a low resource language will be challenging and this would be our first objective.

2) Then we need to fix on some models which would likely help us build a LRL model that is efficient and can give good results on Telugu dataset. All the models should not have same characteristics. This wil be our second objective.

3) Our third objective is to find ways to enhance the model in order to achieve better results.

1.3 Contribution

The main contribution of this work is to provide techniques and models that will help build a efficient model for a low resource language which will enhance the performance of the NLP in that LRL.

1.4 Structure of Paper

The starts by briefly introducing the topic on which this paper is focused(section 1) then it will talk some of the previous works which have done research in these areas and their outcomes (section 2) then we shall talk about the methodology wherein all the information about the data retrieval, its pre-processing, and the models used in this work will be discussed (section 3) then it talks about design specifications of the projects where it will discuss about the architecture that has been used during the implementation of research work (Section 4). Now it will talk about the implementation part wherein how this data is used inorder to build the model (section 5) Then it talks about the outcomes of these implemtaion works which is (section 6) discussed under Results and discussion section

2 Related Work

2.1 A Review of Past and Future Challenges

This majorly talks about the groundbreaking achievements made in are of Low resource language problems in NLP and also gives some suggests about the research required in the future in this field. Though there are 7000 languages around the world NLP research is focused on only 20 languages which leaves many languages as understudied. LRL's can be understood or generally addressed as less studied, resource scarce, less computerized, less privileged, less commonly taught, or low density, among other denominations(4,5,6). One of the main reason for the need of LRLs is that countries like India and Africa hosts about 2.5 billion people developing NLP for such countries will open a gate for economic perspective. Another major reason is having an NLP in any particular language will prevent it from extension and will open the knowledge contained in the original works of that language to everyone. First technique that is talked about in this paper is the projection technique this is highly used in High resource language but when it comes to low resource languages it is tough to apply because of the less annotated data. Then it talks about the data creation for LRLs there are two ways one is creating new dataset by labelling existing raw text and the other one is gathering raw text and aligning it with HRL. In next section it talks about Parts of speech tagging where it mentions about hidden Markov model it suggests using this method for future projects where HRLs can be used for tagging a LRL and also suggests that it would be better if two LRLs with similarity could be used for tagging. Then it talks about speech recognition

and about the techniques like Multilayer perceptron's and hidden Markov models the future challenges that would be faced is that the lack of homogeneity among speech training data. Then it talks about embeddings such as data augmentation and multilevel augmentation the major problem that needs to be addressed is how to handle LRL correctly including structure of sentence, grammar and word formation. Next it talks about machine translation which includes methods like transfer learning where some of the future challenges will include that the parent language should have the same lexicon features as of LRLs and it is challenging but will help in better results. Most of the future work in the area of LRL will take place in creation of new data using embeddind, transfer learning and etc,.

2.2 Selection Criteria for Low Resource Language

This paper describes the criteria required to choose a language for Low Resource Language study with labels given or similar one. This talks about the information provided by authors that have done research or development in the area of LRL in order to select a language for LRL but this paper doesn't give the actual reasons for the selection of language for an LRL. This paper often uses a word Human language technology (HLT) which basically refers to NLP. Past decade has seen significant growth in the area of HLT with the low resource languages. In the United States alone has 4 programs namely TIDES, REFLEX LCTL, Babel and LORELEI which are primarily focused on developing resources and technologies for low-resource languages. When we see outside US there is one more program funded by EU which is METANET started in 2010, as the majority of EU languages are resourced this program helps in bringing collective efforts in order to create technologies that are missing for LRLs and transfer them to the languages that are the edge of digital extinction. The United States-funded National Science Foundation's Document Endangered Language Program (2014) states different agenda which says "Most of what is known about human communication and cognition is based on less than 10 percent of the world's 7,000 languages. We must do our best to document living endangered languages and their associated cultural and scientific information before they disappear." When there are communities with such different thoughts this paper attempts to address reasons for selecting a language in order to do research as these kinds of studies require lot of resources it is worth understanding and studying these reasons. There are some low resource languages which are endangered. Endangered in this case refers to losing of languages because of lack of native speakers or due to shift in native speakers to different languages. When we look at the languages the scholarships for critical languages given by US include Urdu, Turkish, Swahili, Russian, Punjabi, Persian, Korean, Japanese, Indonesian, Hindi, Chinese, Bangla, Azerbaijani, Arabic surprisingly none of them are endangered languages and most of the research in this has selected a language which is not endangered, these researches have mostly benefited standard Arabic and Chinese in increasing those language resources.

Now, this paper goes through some programs starting with DARPA TIDES (Translingual Information Detection, Extraction and Summarization) which has majorly focused on English and (Mandarin) Chinese and Modern Standard Arabic which was actually planned for a surprise language exercise. Here surprise language refers to specific term which defines the time required to port a HLT technique and the extent to which they are portable in case of any events such as natural calamities. But we should also remember that the Chinese and Arabian language did not have many resources until the TIDES

program has concentrated in increased the number, size, and quality. The next program is REFLEX (Research on English and Foreign Language Exploitation) LCTL this is aimed at creating technologies required for resource-low languages such as Thai, Urdu, Amazigh, Guarani, Maguindanao, Bengali, Punjabi many experiments have been on these languages including collection of quality raw text, applying bootstrapping systems from material in related languages. The next program is NIST (National Institute of Standards and Technologies) LRE (Language Recognition) which is also a US-based program for which LDC has provided data since 1996. LRE especially doesn't work on LRLs but works on developing robust technologies in such a way that they can work on variety of languages since the well resourced languages are very less it indirectly says that it works on LRLs, data used in this is are typically parts of broadcast and telephone conversations with linguistic variety spoken, speaker number, sex and sound quality. Thus, the success of these are dependent on the availability, desired data types and the capability of native speaker on labelling. This program has conducted a campaign in 2011 which includes languages such as Iraqi, Levantine, Maghreb and Modern Standard Arabic; American and Indian English; Czech, Polish, Russian, Slovak and Ukrainian; Dari and Persian; Bengali, Hindi, Punjabi and Urdu; and Thai and Lao plus Mandarin, Pashto, Spanish, Tamil, Turkish. Same selection process was again implemented in 2015 the only change was it added Egyptian to Arabic cluster and British was added to English cluster and thereby creating 3 new clusters which are Chinese (Mandarin, Cantonese, Min Nan, Wu); SpanishPortuguese (Brazilian Portuguese, Caribbean Spanish, European Spanish, Latin American Spanish) and French (Haitian Creole, West African French).

This paper now talks about the selection criteria for language to be considered for these programs which is defined by Simpson and colleagues in 2008 as "All meet the basic criteria of being significant in terms of the number of native speakers but poorly represented in terms of available language resources." It uses the number of native speakers as one of the speakers where Mandarin stands first then comes Spanish and then comes English in 3rd place. Hammarstorm's GLP is one of the ranking system where it considers two criteria one is population and other is economic power, in other words it focuses on languages with many native speakers and less economic power. To conclude this paper has considered resources, demography and linguistic varieties are considered while selecting an LRL.

2.3 Low-Resource Language Modelling of South African Languages

This paper works on evaluating the performance of open vocabulary language models on low resource South African languages using byte pair encoding to handle the rich morphology of these languages. They evaluated different types of n-gram models, feed-forward neural networks, and recurrent neural networks. Most of the advantage of NLP are mainly enjoyed by high resource languages due to large neural models such as GPT2, BERT and XLNet (Radford et al.; 2019; Devlin et al.; 2019; Yang et al.; 2019). Generally traditional models estimates next words using probability that Is spread as a distribution on fixed vocabulary and all words outside vocabulary is given an Unknown token but in case of African language Benue-Congo this character level tokenization is not possible as the language is highly agglutinative so they use a model names byte-pair encoding (Gage; 1994) which will break words into sub words based upon their frequency. In this they spoke about two evaluations one is intrinsic which uses statistical evaluation to asses a

model's quality and the other one is extrinsic which evaluates models usefulness in some applied tasks such as speech recognition or machine translation. They have used an evaluation method called bits per character (BPC) which is a measure of cross-entropy normalized by character, hence it is independent of the tokenization. Then this paper talks about the models it has used. First model that it talks about is n-grams which will predict the next word in the series based on Markov assumption, these n-grams generally work based on many smoothening methods, of which modified KnesserNey method which has been seen providing best performance (Kneser and Ney; 1995; Chen and Goodman; 1999). Then the next model which is talked about is Feedforward neural network these were one of the first neural network language models which also based on Markov assumption. One of the major advantages of these neural networks over n-grams is that the word embeddings will allow them to generalize better as words with similar meaning or grammatical functions will have similar embeddings (Mikolov et al.; 2013). Next model discussed is LSTM which is one of the RNN variants that allows better modelling by using gates along with a memory vector in recurrent cell (Hochreiter and Schmidhuber; 1997), this model is regularized using dropouts, but for RNN model it can not be applied between steps so standard approach is to apply dropouts only on input and output connections Zaremba et al. (2015). Other models that are discussed here are AWD-LSTM, Quasi-Recurrent neural network and Transformers. In the results they mentioned that n-grams and feed forward neural networks performed fairly similar and AWD-LSTM and QRNN have perfoemed very closely. To conclude, AWD-LSTM and QRNN performed better than n-grams and other models. n-grams, FFNNs and basic LSTM have underperformed when compared to advanced models which however are expensive in terms of computational power.

2.4 Evaluating Language Model Finetuning Techniques

In this paper they have chosen Filipino as low resource language and they have created a dataset which they are calling as "WikiTEXT-TL-39". They have used techniques like BERT and ULMFiT to train robut classifiers while changing the hyperparameters and decreasing dataset from 10k to 1k with utmost validation error of 0.0782. While neural networks are very effective in NLP they do not work well with less resource data, generally, these languages may not have resources with pre-trained word embeddings and expert annotated data which is commonly available for high-resource languages (Adams et al.; 2017). Such data resource problems can be solved by making a data corpus with annotation which is costly and time consuming process. In this paper they have mentioned about two things that will build a large unlabeled dataset to train pretrained languages and next thing is they will evaluate learning performance on a privately held sentiment dataset. They have used ULMFiT for this which is transfer learning method. ULMFiT uses AWD-LSTM as its base model which is then finetuned to a downstream task in 2 steps. First step is finetuning the model to adapt syntactically and second step is to apply a classification layer to model and then finetuned for classification task. Then they have also used BERT which is a transformer-based model. To pre-train a BERT model they need Word-Piece vocabulary for which they opted for Byte Pair Encoding (BPE). They have pretrained a BERT base model with 12 attention heads, 768 neurons per hidden layer and 12 layers and for ULMFiT they have applied some pre processing steps such as converting all words to lower case and giving special token to all unknown words and limited vocabulary to max of 30k. With results they have achieved from

Pretraining and fine-tuning they have concluded that finetuning methods will help in achieving good results when the availability of labelled data is less. They have proposed to use ULMFiT for for fine tuning base as its pertaining is less expensive when compared to BERT.

3 Methodology

In order to achieve the objectives of this projects there are some specific stages that this project has to go through which are mentioned in below diagram.

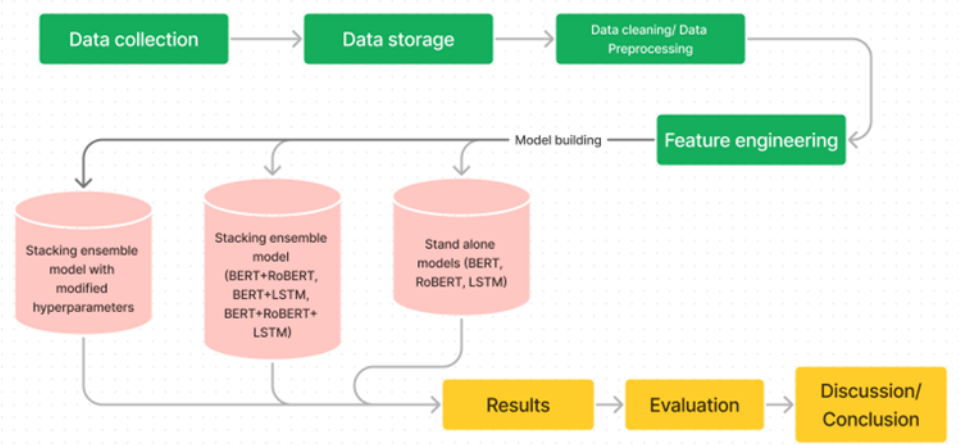


Figure 2: methodology of the project

Figure 2 is a representation of the methodology in detail, In the First step, Data collection, we sourced data, then we have stored the data in a local machine and cloud, then we have applied some data cleaning and data preprocessing steps, then we have applied feature engineering we have used methods like label encoding in this step. All the steps that have followed till now are to manipulate the dataset in order to have better results while training a model. Then we have built the models of which there are standalone models, stacking ensemble models, and stacking ensemble models with modified hyperparameters. Then we have results from these models which are evaluated using parameters such as accuracy and f1-score. Based on these outcomes we have made a conclusion on which model and method is best for low-resource language models.

3.1 Data Loading and Preprocessing

For this project, a Telugu news dataset has been used wherein it has five columns of which only 2 columns body and topic are being used for the project. This dataset is in CSV format which is uploaded to the Colab environment through google cloud and then it is imported into notebook through pandas library in python. Then I have checked for null values in dataset. There was only 1 null value which was deleted in order to have quality data. Then I have defined a function named “clean_text” using some regular expressions in order to remove punctuation marks in the Telugu text and same function has been used for cleansing data for every model.

3.2 Label encoding

The target column "topic" has 5 unique labels which are Business, Editorial, Entertainment, Nation, Sport each label has been replaced with a interger label using a dictionary named "topic_dic" this was used for BERT model but for rest of the models Label_encoder() was used to encode labels.

3.3 Tokenization

The process of converting a long text into smaller pieces is known as Tokenization. For tokenization we have tokenization methods for each model for instance BERT has a bert tokenizer likewise RoBert has a Robert tokenizer but for LSTM we need to a usual tokenizer as it doesn't have any special tokenizers like other two models.

3.4 Data Splitting

We split the dataset into train and test with 80 and 20 percentage respectively using random_split function from PyTorch library. This split will help train the model on larger dataset and also will ensure that there is an unseen data inorder to evaluate the model performance. After splitting for both train and test, dataloaders have been created for each of them with names "train_dataloader" and "test_dataloader". These dataloaders will help in loading data in batched for training and testing very efficiently which optimizes the training process. This way we can ensure that the models are adaptive to new and unseen data which is very crucial part of building and NLP model.

3.5 Model initialization

3.5.1 BERT

BERT (Bidirectional Encoder Representative from Transformers) is transformer model developed by Google. This has revolutionized NLP in the area of contextual understanding approach towards a language which means that it will understand the meaning of the of words in the context of surrounding words. In contrast to the traditional models BERT is bidirectional which means it will consider both left and right side text while processing a word. Because this ability it will improve quality of understanding a language. Bert is a transformer model which is basically a kind of neural network architecture introduced in the paper "Attention is all you need" (Vaswani et al.; 2017). Wherein this will allow leveraging self-attention mechanisms thereby allowing different weight to different parts of the input sequence. Since the BERT has been trained on large amounts of unlabeled data as result of this it has a remarkable capability of understanding grammar and structure of the language. BERT also has the ability to understand the a word with different meaning in different sceniarios because the its contextual based understanding. BERTfirst learns from a huge data while pretraining and then fine-tunes on a specific task with labeled data this will enable BERT to applicable for some downstream tasks such as NER, classification and etc,. BERT also uses a technique while training called maskded language model in which some words in the given text are hidden and model will trying to predict these masked words which will enable BERT to understand the relation between words even when some of them are hidden during the training phase.

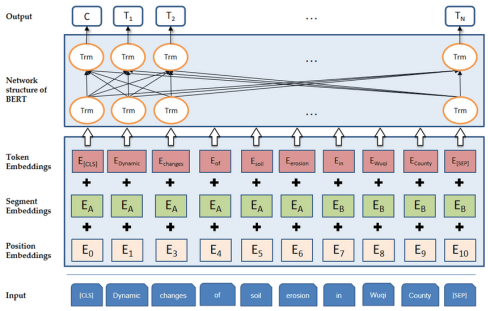


Figure 3: BERT model

Figure 3 Above is the representational image of BERT model
Source: ResearchGate (Sun et al.; 2022)

3.5.2 RoBERT

RoBERT (Robustly optimized Bidirectional Encoder Representative from Transformers) as name says it is the robustly optimized BERT which is optimized while pretraining by focusing on certain weaknesses that are identified in case of BERT. These modifications will help in getting better model by generalization and robustness, making it one of the best choices for downstream applications. RoBERT unlike the BERT doesn't include Next Sentence Prediction (NSP) task during pretraining which will allow RoBERT to better capture the contextual information within the sentences thus resulting in more effective language understanding. RoBERT has capability of training on longer sequences than in BERT which enable to understand more relationships and dependencies in the sentences contributing to the better understandingof overall language. And RoBERT deploys batches in smaller scale which will help in increased efficiency and scalability. RoBERT has dynamic masking method during training which will changes masked token for every batch which results in more diverse learning experience for model thereby helping it generalize better across different linguistic context. Because of all these characteristics like optimized pretraining, improved contextual understanding, scalability and dynamic masking system makes RoBERT to stand as a testimony for the latest research works by creating avenues for more nuanced and comprehensive language understanding.

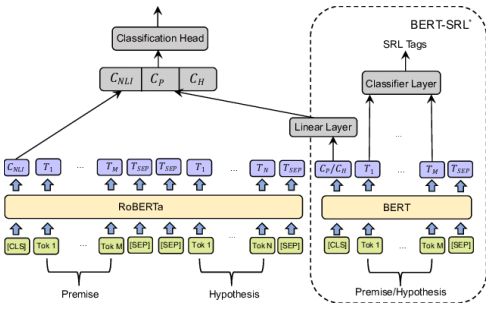


Figure 4: Robert model

Figure 4 is the representational image of RoBERT model
Source : ResearchGate (Saha et al.; 2020)

3.5.3 LSTM

LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) which is designed to address the challenges associated with understanding the long-range dependencies in sequential data. This is build inorder to overcome vanishing gradient problem in RNNs. LSTM uses memory cells for storage and retrieval of information over long sequences which allows the model to understand the dependencies over extended periods which makes them a choice for tasks that require an understanding of context beyond immediate neighbors. Another significance of LSTM is "Forget Gates" which controls the flow of information over time helping the model's ability to capture and remember relevant information from the memory which addresses the problem of vanishing gradient problem. Then it has input and output gates which governs the addition and extraction of relevant information from memory cells. Because of these gates LSTM also supports parallelization which will allow model to process multiple sequences simultaneously making the training process efficient when compared to traditional RNNS.

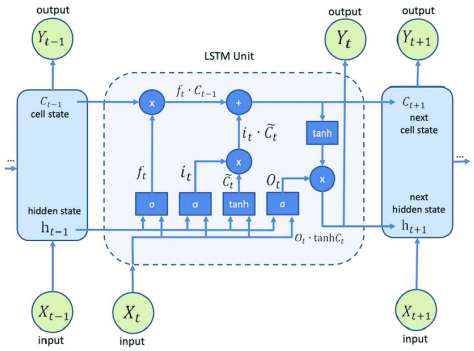


Figure 5: LSTM model

Figure 5 is the representational image of LSTM model
Source : ResearchGate (Zhou et al.; 2022)

3.5.4 Stacking ensemble

Stacking Ensemble learning is a powerful method in Machine Learning where multiple models are combined in order to improve prediction performance. In Stacking various base models are trained independently then a meta model is made which is a combination of base models where it takes output of base models as input and makes final prediction. So, this is basically a two stage learning as one has to train base model and then train meta-model wherein the base model will capture patterns in data while the meta-model learns to combine their predictions optimally. And also it gives the flexibility of choosing base models and meta-models which will help build a stacking ensemble model where one model can compensate for the weakness of the another.

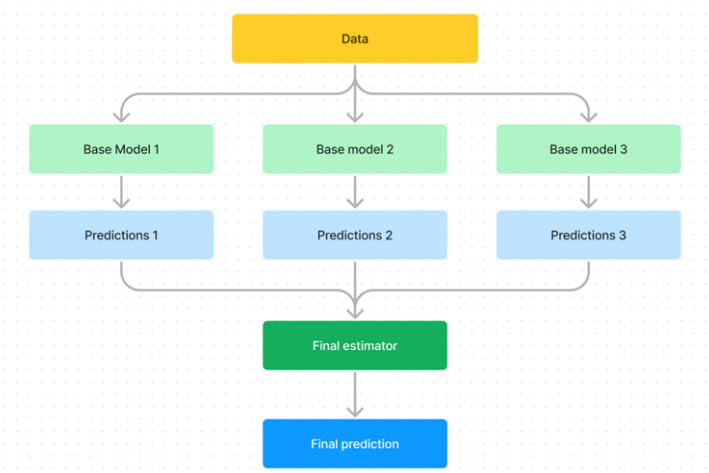


Figure 6: This is a caption

Figure 6 is the representational image of the stacking ensemble model
Source : ResearchGate (Saha et al. 2020)

4 Design Specification

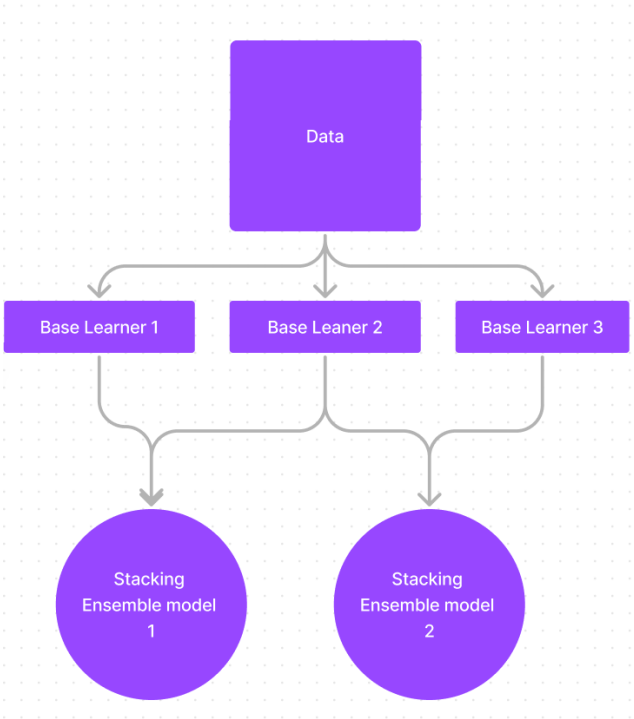


Figure 7: Specification

Figure 7 is the representational image of stacking ensemble model

The first step of the design specification is to identify the NLP modals that have different characteristics and build standalone models which has done using BERT, RoBERTa and LSTM model. Then these models are used to build stacking ensemble models in our case we have built four stacking models which are

i) BERT+LSTM

ii) BERT+RoBERT

iii) BERT+RoBERT+LSTM

iv) BERT+LSTM (Hyperparameter optimized)

All the above stacking ensemble model takes about 24 hours to run on any average environment in order to have a faster runtime Colab pro has been used which run the same code in 6 to 7 hours because the runtime restriction all the models have been evaluated based on classification report.

5 Implementation

In this project we have used stand alone models and their stacking ensemble models since most of the models are transformer models we need to follow before giving data for training for transformer models we need follow some steps such as using dataloaders which will help to handle data in batches. In case of BERT we have a transformer library from Hugging face then we have also used AdamW optimizer with a learning rate of 2e-5. We have given epoch as 3 which will iterate data as batches of training data for each batch the model computes loss, performance backpropagation and updates the parameters using gradient descent. Incase of RoBERTa it is almost same as BERT wherein we will use transformers library from hugging face and also AdamW optimizer is also used and all the other parameters were same as BERT. In LSTM will have embedding layer and also we have used Adam optimizer with learning rate of 0.001. We have 10 epochs in case of LSTM standalone model and 3 epochs in case of stacking ensemble model. BERT has masked training which RoBERT doesn't have so But RoBERT is more robust then BERT so BERT+RoBERT stacking model will compensate for each other negative and add their positives to give best results.

6 Result and Discussion

Bert model alone takes about 24 hours environment and about 4 hours in Colab pro environment. Initially, we have started using Classification report which has accuracy, f1-score and precision and recall score which were basis for conclusion which will be discussed under discussion section.

6.1 BERT Model Evaluation

Figure 8 Classification report for BERT model

By looking at the score that BERT model has obtained says that it is has performed well on the Telugu text dataset where it has achieved 88% percent accuracy and also the weighted average of precision and f1-score is also good with 89 and 88 percentage respectively.

	precision	recall	f1-score	support
0	0.89	0.84	0.86	523
1	0.96	0.87	0.91	394
2	0.82	0.96	0.89	1305
3	0.95	0.94	0.94	1046
4	0.80	0.24	0.37	195
accuracy			0.88	3463
macro avg	0.88	0.77	0.79	3463
weighted avg	0.89	0.88	0.87	3463

Figure 8: BERT score

6.2 RoBERT Model Evaluation

	precision	recall	f1-score	support
0	0.67	0.55	0.60	530
1	0.71	0.54	0.61	354
2	0.72	0.80	0.76	1384
3	0.72	0.91	0.80	974
4	1.00	0.01	0.02	221
accuracy			0.71	3463
macro avg	0.77	0.56	0.56	3463
weighted avg	0.73	0.71	0.69	3463

Figure 9: RoBERT score

Figure 9 Classification report for RoBERT model

When compared to BERT, RoBERT scores are not very encouraging but the thing to remember here is that RoBERT has better understanding capacity towards language than the BERT model because RoBERT need not to deal with masking task. Hence it is quite possible that RoBERT has found out more relations than BERT But when we talk about statistics the accuracy of the model is 71% and the weighted average of precision and recall are 73% and 71% respectively.

6.3 LSTM Model

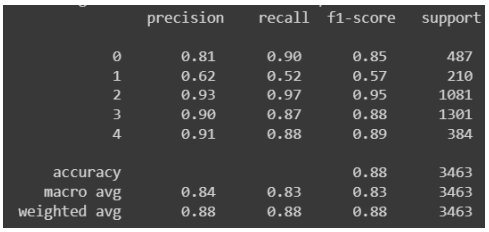
	precision	recall	f1-score	support
business	0.44	0.71	0.54	487
editorial	0.00	0.00	0.00	210
entertainment	0.91	0.92	0.91	1081
nation	0.78	0.91	0.84	1301
sports	0.28	0.05	0.08	384
accuracy			0.73	3463
macro avg	0.48	0.52	0.48	3463
weighted avg	0.67	0.73	0.68	3463

Figure 10: LSTM score

Figure 10 Classification report for LSTM model

LSTM has given results with accuracy of 73% and weighed average score of precision and recall are 67% and 73% percent respectively and also the computational power required is very less when compared to BERT and RoBERT.

6.4 BERT + LSTM ensemble Model



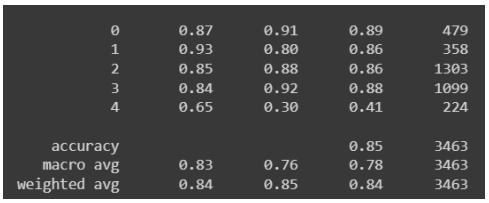
	precision	recall	f1-score	support
0	0.81	0.90	0.85	487
1	0.62	0.52	0.57	210
2	0.93	0.97	0.95	1081
3	0.90	0.87	0.88	1301
4	0.91	0.88	0.89	384
accuracy			0.88	3463
macro avg	0.84	0.83	0.83	3463
weighted avg	0.88	0.88	0.88	3463

Figure 11: BERT+LSTM score

Figure 11 classification report for BERT+ LSTM stacking ensemble model

BERT+LSTM stacking ensemble model is good but the stand alone model BERT itself has a better performance when we consider precision and recall along with accuracy as they have briefly little high value than this stacking ensemble model

6.5 BERT + RoBERT ensemble Model



0	0.87	0.91	0.89	479
1	0.93	0.80	0.86	358
2	0.85	0.88	0.86	1303
3	0.84	0.92	0.88	1099
4	0.65	0.30	0.41	224
accuracy			0.85	3463
macro avg	0.83	0.76	0.78	3463
weighted avg	0.84	0.85	0.84	3463

Figure 12: BERT+RoBERT model

Figure 12 classification report of BERT+RoBERT stacking ensemble model

The stacking ensemble model of BERT and RoBERT model is very promising though the BERT model alone has 88% accuracy and stacking ensemble model has 85% accuracy if look at only these parameters then it might be convincing to implement such models but when we consider RoBERT accuracy which was about 71% which has contributed in having an accuracy of 85% which is a drastic increase. Even though this model has less accuracy score than BERT model alone this model would be better because it has ability masked tasks due to BERT and long sequence understanding ability due to RoBERT.

6.6 BERT+RoBERT+ LSTM ensemble model

Figure 13 classification report of BERT+RoBERT+LSTM ensemble model

This model also gives better results but this is very expensive in terms of time and energy required for computational power. And the results that it achieved would not be worth the price as this results can be achieved in stand alone and other ensemble models with only 2 base learners.

	precision	recall	f1-score	support
0	0.81	0.90	0.85	487
1	0.63	0.52	0.57	210
2	0.93	0.97	0.95	1081
3	0.90	0.87	0.88	1301
4	0.91	0.88	0.89	384
accuracy			0.88	3463
macro avg	0.84	0.83	0.83	3463
weighted avg	0.88	0.88	0.88	3463

Figure 13: BERT+RoBERT+LSTM model

	precision	recall	f1-score	support
0	0.86	0.87	0.86	487
1	0.76	0.59	0.66	210
2	0.94	0.96	0.95	1081
3	0.88	0.92	0.90	1301
4	0.93	0.83	0.87	384
accuracy			0.90	3463
macro avg	0.87	0.83	0.85	3463
weighted avg	0.89	0.90	0.89	3463

Figure 14: BERT + LSTM (Modified Hyperparameters)

6.7 BERT + LSTM (Modified Hyperparameters)

Figure 14 classification report of BERT+LSTM (modified Hyperparameters) stacking ensemble model

The accuracy of this model is very convincing that the stacking model with the optimized hyper paameters will enhance the accuracy of the model. The accuracy of the model is 90% and also the weighted average of precision and recall is 89% and 90% respectively.

6.8 Discussion

This project aimed at enhancing NLP models for low resource languages for which it has incorporated stacking ensemble models where it has performed 4 stacking ensemble models of which three of them have used base model directly but one of the models have used model with modified hyperparameters which has given an accuracy of about 90%. But also not to ignore the BERT+RoBERT model because the stand alone RoBERT was giving only 71% whereas when it is used along with BERT it is giving 85% accuracy in stacking ensemble model it is also good to remember that BERT accuracy was 88% which 3% higher than the stacking model. But the stacking model of BERT and RoBERT have an advantage of both the models models where BERT will have masked token tasks while RoBERT will have the ability to understand the longer sequences in languages and the relationship between them.

7 Conclusion and Future Work

We can see clear dominance of BERT in Figure 15 when it comes to standalone models, combining base to get a stacking ensemble model has led to the balanced precision and recall score this can be due to the ensemble model which is helping individual models to compensate for their weaknesses. The stacking ensemble model BERT+LSTM with

	BERT	RoBERT	LSTM	BERT+RoBERT	BERT+LSTM	BERT+RoBERT+LSTM	BERT+LSTM (Modified Hyperparameters)
Accuracy	88%	71%	73%	85%	88%	88%	90%
Precision	89%	73%	67%	84%	88%	88%	89%
Recall	88%	71%	73%	85%	88%	88%	90%

Figure 15: Result of all models

modified hyperparameters has achieved the highest accuracy indicating that fine-tuning can significantly impact model performance. Our main goal of the project is to enhance the NLP models for a low-resource language which we have achieved through this paper by using stacking ensemble model and fine-tuning the base models, which gave 90% accuracy score. Yet there is lot of work left to do in future in this, one is trying other stand alone models which can give more than 80% accuracy and then using it with BERT in order to see how two well performing models will result in what kind of ensemble model and only one model was used with hyperparameters optimization, in future this should be extended to all models. So, by forming a ensemble model with two base models with accuracy over 80% and fine-tuning them with hyperparamets might result in very efficient model with promising results.

References

Adams, O., Makarucha, A., Neubig, G., Bird, S. and Cohn, T. (2017). Cross-lingual word embeddings for low-resource language modeling, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Association for Computational Linguistics, pp. 937–947.

Chen, S. F. and Goodman, J. (1999). An empirical study of smoothing techniques for language modeling, *Computer Speech & Language* **13**(4): 359–394.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, Association for Computational Linguistics, pp. 4171–4186.

Gage, P. (1994). A new algorithm for data compression, *C Users Journal* **12**(2): 23–38.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory, *Neural computation* **9**(8): 1735–1780.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling, *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Vol. 1, IEEE, pp. 181–184.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J. (2013). Distributed representations of words and phrases and their compositionality, *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., pp. 3111–3119.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I. (2019). Language models are unsupervised multitask learners, *Technical report*, OpenAI.

Saha, S., Nie, Y. and Bansal, M. (2020). Conjnli: Natural language inference over conjunctive sentences.

Singh, A. K. (2008). Natural language processing for less privileged languages: Where do we come from? where are we going?, *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*.

Sun, J., Liu, Y., Cui, J. and He, H. (2022). Deep learning-based methods for natural hazard named entity recognition, *Scientific Reports* **12**: 4598.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, and Polosukhin, I. (2017). Attention is all you need, *Advances in neural information processing systems*, Vol. 30.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R. and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding, *Advances in Neural Information Processing Systems*, Vol. 32, Curran Associates, Inc., pp. 5753–5763.

Zaremba, W., Sutskever, I. and Vinyals, O. (2015). Recurrent neural network regularization, *CoRR* .

Zhou, D., Zuo, X. and Zhao, Z. (2022). Constructing a large-scale urban land subsidence prediction method based on neural network algorithm from the perspective of multiple factors, *Remote Sensing* **14**: 1803.