Clickbait Detection in Telugu: Overcoming NLP Challenges in Resource-Poor Languages using Benchmarked Techniques

by

Mounika Marreddy, Subba Reddy Oota, Lakshmi Sireesha Vakada, Venkata Charan Chinni,
Radhika Mamidi

Report No: IIIT/TR/2021/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2021

Clickbait Detection in Telugu: Overcoming NLP Challenges in Resource-Poor Languages using Benchmarked Techniques

Mounika Marreddy [1]
mounika.marreddy@research.iiit.ac.in

Subba Reddy Oota[1]
oota.subba@students.iiit.ac.in

Lakshmi Sireesha Vakada[1]
lakshmi.sireesha@research.iiit.ac.in

Venkata Charan Chinni[1]
venkata.charan@students.iiit.ac.in

Radhika Mamidi[1]
radhika.mamidi@iiit.ac.in

[1] IIIT Hyderabad

***Abstract*—Clickbait headlines have become a nudge in social media and news websites. The methods to identify clickbaits are largely being developed for English. There is a need for the same in other languages as well with the increase in the usage of social media platforms in different languages. In this work, we present an annotated clickbait dataset of 112,657 headlines that can be used for building an automated clickbait detection system for Telugu, a resource-poor language. Our contribution in this paper includes (i) generation of the latest pre-trained language models, including RoBERTa, ALBERT, and ELECTRA trained on a large Telugu corpora of 8,015,588 sentences that we had collected, (ii) data analysis and benchmarking the performance of different approaches ranging from hand-crafted features to state-of-the-art models.**

We show that the pre-trained language models trained on Telugu outperform the existing pre-trained models viz. BERT-Mulingual-Case [1], XLM-MLM [2], and XLM-R [3] on clickbait task. On a large Telugu clickbait dataset of 112,657 samples, the Light Gradient Boosted Machines (LGBM) model achieves an F1-score of 0.94 for clickbait headlines. For Non-Clickbait headlines, F1-score of 0.93 is obtained which is similar to that of Clickbait class. We open-source our dataset, pre-trained models, and code[1]

***Index Terms*—clickbait, deep learning, Telugu annotated dataset**

I. Introduction

Clickbait detection aims to identify the popular headlines that catch the user's attention to read and click on the weblink. Also, clickbait headlines typically provide sufficient information to make readers curious, but insufficient to fulfill their curiosity which makes them click the linked content.

With the development of online social media applications and its increasingly dominant role as the content provider, it can attract and engage people to read or share the content with others. Moreover, online platforms such as news, social, and web media are depending on headlines to draw attention. In this way, headlines have become an essential way to increase viewership in the native language of the readers. One of the reasons to attract user's attention is to generate revenue from the subscriptions, clicks and views made by their readers [4].

Furthermore, in order to succeed in this competition, it becomes necessary for social media platforms to provide the headlines in the native language of the users to increase the readership and thereby, the revenue.

The impact of clickbait detection and generation has been drawing the attention of researchers also. During the current COVID-19 pandemic situation, many news websites and individuals saw this crisis as an opportunity to capitalize for increasing the count of viewers, people in the need of some vaccine, quarantine, or precaution related answers in online media[2,3]. For example, after reading the headline "తెలంగాణ పోలీస్ అకాడమీలో కరోనా కలకలం "*(Corona chaos in police academy)*[4], users may get an impression that probably an unexpected event has occurred, which motivates the user to know more [5].

A decent amount of research work is done to tackle clickbait, bizarre, or fake news for the resource-rich languages like English [6], [7]. This research is mainly due to the availability of resources, tools, and efficient feature representation methods. However, Indian languages are resource-poor due to the dearth of various qualitative tools and scarcity of annotated data.

Also, most of the work being done in Indian languages is from machine translation perspective and the resources created are also keeping the machine translation task in mind. But in case of Clickbait task, the meaning of clickbait content in source language may change, and the curiosity element of a clickbait will be lost if translated. Examples of such Telugu language clickbait headlines along with the WX notation (a standard notation used by all NLP practitioners in Indian languages)[5], and English

[1] https://github.com/subbareddy248/Clickbait-Resources

[2] https://rb.gy/y3uhuq

[3] https://rb.gy/f050qa

[4] https://rb.gy/90j4l3

[5] https://en.wikipedia.org/wiki/WX_notation

translation are reported in Table I. So, extending all the latest models in NLP to Indian languages becomes a challenge because of lack of appropriate data. The data we created for Telugu language will be a good resource for those working in Telugu NLP areas such as Text Classification, Question Answering and Semantic Role Labelling.

This is the main reason why creating abundant data, tools, machine learning and deep learning models for Indian languages is needed today as it will help increase better communication and understand different types of contextual difficulties. However, to the best of our knowledge, there is no resource or computational approach for tackling the clickbait headlines written in Telugu, which can negatively affect this large population.

In this paper, we aim at bridging the gap by creating resources for detecting clickbait headlines in Telugu. The task of clickbait detection can be extended to other Indian languages that are closer to Telugu culturally and linguistically by translating this resource without losing the curiosity aspect. This is the first work that employs neural methods to Telugu language – a language that does not have good tools like NER, parsers and embeddings. Our work is the first attempt in this direction to provide good models in Telugu language by exploring different methods with available resources. Our contributions can be summarized as follows:

- We publicly release an annotated dataset of 112,657 Telugu clickbait and non-clickbait headlines that can be a key resource for building automated clickbait detection systems in Telugu.
- We generate the three latest pre-trained language models, including RoBERTa [8], ALBERT [9], and ELECTRA [10] for Telugu language, trained on large Telugu corpus (8,015,588 sentences) from scratch.
- We develop a benchmark system for detecting clickbait headlines written in Telugu by investigating a wide range of features from traditional to state-of-the-art representations.
- We explore the feasibility of different neural architectures and pre-trained transformer models in this problem.
- We present a detailed analysis of the methods, results, and compare the existing pre-trained embeddings for detecting clickbait headlines in Telugu.

II. Related Work

With the impact of clickbait headlines in social media platforms, research on clickbait detection has addressed the issues of data creation, feature representation, and model building to automate the solution. The earlier works in the literature relied on creating annotated datasets and a rich set of hand-crafted features to categorize headlines as clickbait or non-clickbait [11]. Similar to the work mentioned above, there are a publicly available clickbait dataset developed on Twitter by considering the news genre, tweets from Twitter, and importance-based acquisition in [12], the headline from tweets, and gatekeeper-based acquisition in [13]. Here, [12] uses common words and extracted some other tweet specific features to classify tweets as clickbait or not. Furthermore, in [14], the authors described and constructed a new clickbait dataset by choosing the eight types of clickbait headlines. The main limitation of these works are that they focus on using hand-crafted features rather than automated ones to categorize the clickbait news headlines.

TABLE I
Example sentences of clickbait headlines in Telugu, WX notation and English.

Sentence
వింత వాయిద్యం.. వినూత్న సంగీతం. viMwa vAyixyaM.. vinUwna saMgIwaM. Strange instrument .. innovative music.
నైన్టీస్‌లో టీన్!. nEntIslo tIn. Teen in the Nineties.
మనిషి చర్మంతో పుస్తకాలు. maniRi carmaMwo puswakAlu. Books with human skin.

Deep Learning based Word Embeddings: Recently, the efficient feature representation methods using deep learning yielded fruitful results in NLP. In [6], authors developed a deep learning model called Bidirectional RNN (Recurrent Neural Networks) with character and word embeddings as the features to classify the online content as clickbait or not. Also, many researchers use deep learning models to recognize the clickbaits in [15]–[17]. Focusing on hand-crafted and word-embeddings features, a clickbait detection model was developed in [18] and a bizarre news identification model in [7]. Unlike earlier methods, the conducted clickbait challenge requires users to calculate a clickbait score of a tweet post in [19].

Low-Resource Language Clickbait Detection: Though studies achieved significant advancement in the resource-rich English language, very few works are available for other languages. Existing works on non-English resource-poor languages focus on dataset creation from various news articles to build the hoax detection model. For example, for Indonesian language in [20], and a real-time news certificate system for the Chinese language in [21]. There is not much work done in Indian languages.

To the best of our knowledge, we are the first to leverage the creation of a clickbait dataset and considering the task of detection for Telugu language belonging to Dravidian language family spoken in the southern part of India. Along with the three pre-trained embeddings (RoBERTa, ALBERT, and ELECTRA), we use the pre-trained embeddings such as Word2Vec, GloVe, FastText, Meta-Embeddings, Skip-Thought, ELMo, and BERT created for Telugu language[6].

[6] https://github.com/subbareddy248/Clickbait-Resources

III. A New Dataset for Detecting Clickbait in Telugu

We followed certain guidelines for collecting the data extensively for clickbait and non-clickbait headlines.

Inter-Annotator Agreement The annotation process on sentences for clickbait is a subjective task. In the guidelines to annotators, we defined the attractiveness of clickbait headlines as something that generates interest, arouses curiosity by using suspenseful language there, and motivates the user to click on the link. Moreover, the attractiveness of the sentences is related to one's personal interests and curiosity. To perform the annotation, we made the five annotators work on a smaller dataset for verification as a first step. The Fleiss' kappa score[7] was 0.95. Then the annotation on the whole dataset was done for which the inter-annotator agreement was 0.91.

Clickbait vs Non-Clickbait: To collect the clickbait and Non-clickbait headlines, we manually selected four Telugu websites including, Prajasakti[8], Gulte[9], Andhrabhoomi[10], Manatelangana[11] which publish gossips, trends, and latest buzzes. While collecting misleading headlines from these sites, we found that most of the sites have the same headlines. To avoid the redundant text, we considered the Jaccard similarity for discarding duplicate sentences (above 90% match) and reported a total of 112,657 headlines. To filter the non-clickbait (i.e., the headlines in these domains which are not attractive to users), we provided the data and a web-based framework to an *Elancer IT Solutions Private Limited*[12] annotation company for labeling the headlines. We choose five native Telugu language speakers from *Elancer IT Solutions Private Limited* company to annotate headlines as clickbait and non-clickbait. Since five annotators label each headline, we obtained a 'substantial' inter-annotator agreement with a Fleiss' kappa score of 0.91.

Dataset Statistics Taking the majority vote as ground truth, a total of 79,190 headlines were marked as clickbait and 33,467 as non-clickbait.

IV. Feature Representation Methods

This section discusses the detailed analysis of various feature representation methods ranging from hand-crafted to automated features to develop the models. To perform better training, we use the romanized text format, i.e., whole Telugu corpus and annotated data converted into WX notation[13]. In this paper, we use Polyglot tokenizer[14] for tokenizing the sentences.

TABLE II
Analysis of structural features.

Average occurences	**Clickbait**	**Non-Clickbait**
Average length	4.98	12.25
Average stopwords	0.51	2.28
Average Hapax Legomena Ratio	0.13	0.10
Average Dis Legomena Ratio	0.03	0.01
Ratio of sentences with word length<4	0.17	0.16
Ratio of sentences with word length>6	0.02	0.02
Ratio of sentences with Exclamation	0.15	0.01
Ratio of Sentences with Question mark	0.12	0.02

TABLE III
Analysis of POS Tag features.

Pos Tag Features	**Clickbait**	**Non-Clickbait**
Average nouns per sentence	2.60	5.62
Average verbs per sentence	0.61	2.14
Average adjectives per sentence	0.07	0.25
Average adverbs per sentence	0.08	0.35
Average pronouns per sentence	0.09	0.34
Average postpositions per sentence	0.06	0.22
Average punctuations per sentence	0.94	1.29

A. Traditional Features

To extract the traditional features like Bag-Of-Words (BoW) and Term frequency-Inverse document frequency (TF-IDF), we build a feature set that resulted in a dataset size of 112,657 (samples) x 2000 (features).

B. Hand-Crafted Features

Structural Features Here, we will discuss how the sentence structure and punctuation features can help discriminate between clickbait and non-clickbait headlines. From the Table II, we observe that frequency of the presence of exclamation marks (!), question marks (?), word length < 4, and Hapax Legomena Ratio ($\frac{\text{number of unique words occurred only once}}{\text{total number of words}}$) are high in clickbait headlines. However, the average length of the sentence and the number of stop words display high values for non-clickbait.

Part-of-Speech (POS) Features The POS tagging features are often useful to understand the particular patterns in morphologically complex languages like Telugu. We use a feature set that includes the number of nouns (NN), number of verb mains (VM), number of symbols (SYM), number of adverbs (RB), number of prepositions (PSP), number of adjectives (JJ), and number of pronouns (PRP) from each sentence using part-of-speech (POS) tagging[15]. Table III(b) report the average number of times each POS tag occurs in the two kinds of headlines.

C. Distributed Word Representations

Distributed word representations capture a large number of precise syntactic and semantic word relationships in text classification problems. This subsection describes the generation of different word embeddings such as Word2Vec, GloVe, FastText, and Meta-Embeddings on large Telugu corpora (Wikipedia + Crawled corpus) consists of 8,015,588 sentences.

[7] https://en.wikipedia.org/wiki/Fleiss%27_kappa

[8] http://www.prajasakti.com

[9] https://telugu.gulte.com

[10] http://www.andhrabhoomi.net

[11] https://www.manatelangana.news/

[12] http://elancerits.com/

[13] https://github.com/irshadbhat/indic-wx-converter

[14] https://github.com/ltrc/polyglot-tokenizer

[15] https://bitbucket.org/sivareddyg/telugu-part-of-speech-tagger/src

Word2Vec Embeddings Word2Vec model provides a non-deterministic way to determine the word representations [22]. It can learn similar word vectors for words in a similar context. We used the Skip-Gram (SG) Word2vec [22] model with a large vocabulary size of 460,000 words where each word frequency is at least 5. The number of linear neurons in the hidden layer is 300, and the context window size is of 5 as the hyper-parameters used for the model training.

GloVe Embeddings The input used in the GloVe model is a non-zero word-word co-occurrence matrix [23], which adds the global context information by default, unlike the use of local context in Word2Vec [22]. We use the parameters with a learning rate of 0.05, the context window size is 10, and the word vector dimension is 50.

FastText Embeddings Since the FastText model considers the bag of character n-grams to represent each word [24], it allows us to compute rare word representations. We considered the words with a frequency of at least three in the vocabulary and obtained a vocabulary size of 462,232 in the training process. We use a successful Skip-Gram model with a context window size of 5, and the word vector dimension is 200.

Meta-Embeddings Meta-embeddings are shown to be successful for resource-rich language English, and it has two benefits compared to individual embedding sets: (i) enhancement of performance and (ii) improved coverage of the vocabulary [25]. These prior studies motivate us to create Meta-Embeddings for the Telugu language by using an ensemble of Telugu Word2Vec, GloVe, and FastText embeddings. In this work, we created the Meta-Embeddings using the average of encoded source embeddings that yields each word vector dimension of 300 used for extracting the text features.

D. Skip-Thought Sentence Vectors

The Skip-Thought model is a sentence encoding-decoding model that produces a fixed-length vector for every given sentence [26]. Inspired by the Word2Vec approach, the skip-thought model extends the Skip-Gram model to generate sentence embeddings from word embeddings. Rather than predicting the context given the input word, the skip-thought model predicts both the next and previous sentences given the target sentence.

E. Context Level Features

Earlier word embeddings such as Word2Vec, GloVe, FastText, and Meta-Embeddings provides a unique word representation throughout the corpus. This unique representation is the main limitation of these methods, especially words which are having different contexts. To overcome these limitations, in this section, we describe the recent state-of-the-art pre-trained models, including ELMo, BERT, RoBERTa, ALBERT, and ELECTRA. The parameters used for pre-training the models on Telugu are reported here[16].

ELMo Embeddings for Telugu Embeddings from Language Models (ELMo) is a successful NLP framework developed by AllenNLP [27] group. Unlike earlier embeddings, the ELMo embeddings represent the words in a contextual fashion using a bidirectional LSTM model. We generate ELMo embeddings for Telugu language, trained on a large Telugu corpus with a total number of 2,086,488 train tokens, and a vocabulary size of 793,384.

BERT Embeddings for Telugu BERT model [28] provides a word contextual information by looking at previous and next words, which is one of the main limitations in earlier methods. To train the BERT model from scratch for Telugu language corpus, we build a vocabulary size of 832,000 using the BERTWordPiece tokenizer. We use the hidden representation associated with CLS token was considered as numeric representation of the given sentence.

RoBERTa Embeddings for Telugu The recent successful RoBERTa model [8] is an optimized pre-training language model that improves on BERT, the self-supervised method that achieves state-of-the-art results. The main objectives of RoBERTa model include (i) training the model longer, with bigger batches, over more data; (2) removing the next sentence prediction approach; (3) training on longer sequences; and (4) dynamically changing the masking pattern applied to the training data. Here, we use a vocabulary size of 200,000 and a hidden word dimension of 768 when training RoBERTa masked language model on Telugu corpus.

ALBERT Embeddings for Telugu ALBERT (A lite BERT) [9] model is a novel pre-training method that mainly contributed to three solutions to overcome limitations of the BERT model, including (i) parameter reduction using factorized embedding parameterization, (ii) Cross-layer parameter sharing, and (iii) Inter-sentence coherence loss. With the advantage of a minimal number of parameters and yet achieve state-of-the-art results than the BERT model [9]. To train ALBERT embeddings for Telugu, we use a subword-byte level tokenizer to build the vocabulary size of 105,686 with the hidden embedding dimension is of 768.

ELECTRA Embeddings for Telugu Recently developed language models such as BERT [28], ALBERT [9], and RoBERTa [8] fall under the category of masked language models, that predict words which have been masked out of the input. Thus, masked language models only predict a small subset — the 15% that was masked out, reducing the amount learned from each sentence.

The recent state-of-the-art model ELECTRA uses a new pre-training approach, called replaced token detection (RTD), that trains a bidirectional model (like a Masked Language Model(MLM)) while learning from all input positions (like an LM). In the training process, the

[16] https://github.com/subbareddy248/Clickbait-Resources

TABLE IV
Clickbait Detection Results: Different feature sets classification comparison for Logistic Regression, and LightGBM models using different sampling methods with No/Over/Under-Sampling.

	Logistic Regression									**LightGBM**								
Sampling Method→	NS			OS			US			NS			OS			US		
Feature set↓	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BoW	0.87	0.82	0.84	0.83	0.84	0.84	0.84	0.84	0.83	0.89	0.80	0.83	0.84	0.84	0.84	0.83	0.84	0.83
TF-IDF	0.86	0.82	0.83	0.81	0.83	0.82	0.81	0.83	0.82	0.90	0.81	0.84	0.82	0.83	0.83	0.81	0.83	0.82
Structure-Based	0.90	0.87	0.88	0.87	0.88	0.87	0.87	0.88	0.87	0.90	0.87	0.88	0.87	0.88	0.87	0.87	0.88	0.87
Pos Tagging Set	0.90	0.87	0.88	0.87	0.88	0.87	0.87	0.88	0.87	0.90	0.87	0.88	0.87	0.88	0.87	0.87	0.88	0.87
Word2Vec	0.75	0.73	0.74	0.74	0.77	0.75	0.74	0.77	0.75	0.90	0.88	0.89	0.89	0.89	0.89	0.86	0.88	0.87
GloVe	0.66	0.63	0.63	0.67	0.69	0.67	0.67	0.70	0.67	0.83	0.81	0.82	0.82	0.83	0.82	0.79	0.82	0.80
FastText	0.78	0.77	0.77	0.76	0.80	0.77	0.76	0.80	0.77	0.90	0.89	0.89	0.90	0.89	0.90	0.87	0.89	0.88
Meta-Embeddings	0.75	0.73	0.74	0.74	0.78	0.75	0.73	0.76	0.74	0.91	0.89	0.90	0.90	0.90	0.90	0.88	0.90	0.88
Skip-Thought	0.92	0.90	0.91	0.91	0.91	0.91	0.89	0.90	0.90	0.92	0.90	0.91	0.91	0.91	0.91	0.89	0.91	0.90
BERT	0.89	0.86	0.87	0.86	0.87	0.86	0.85	0.88	0.86	0.90	0.87	0.89	0.88	0.88	0.88	0.86	0.88	0.87
ALBERT	0.89	0.84	0.86	0.85	0.85	0.85	0.85	0.85	0.85	0.89	0.85	0.86	0.86	0.86	0.86	0.86	0.86	0.86
RoBERTa	0.93	0.91	0.92	0.91	0.92	0.91	0.90	0.91	0.90	0.93	0.91	0.92	0.92	0.91	0.92	0.90	0.92	0.91
ELECTRA	0.88	0.86	0.87	0.87	0.87	0.87	0.86	0.87	0.86	0.91	0.89	0.90	0.90	0.90	0.90	0.88	0.90	0.89
ELMo	0.95	0.94	0.94	**0.94**	**0.94**	**0.94**	0.93	0.94	0.93	0.95	0.93	0.94	**0.94**	**0.94**	**0.94**	0.93	0.94	0.94

NS = No-Sampling, OS = Over-Sampling, US = Under-Sampling, P = Precision, R = Recall, F1 = F1-score

ELECTRA model used the GANs approach to distinguish between "real" and "fake" input tokens. It corrupts the input by replacing some input tokens with incorrect, but somewhat plausible, fakes. We create ELECTRA-small embeddings for Telugu language, trained on a Telugu corpus with a vocabulary size of 388,292 and a hidden dimension of 256.

V. Methodologies

We experimented with the two off-the-shelf machine learning classifiers, including, logistic regression (baseline) [29], and LightGBM [30] to evaluate our clickbait dataset. We also experimented with a deep learning architecture LSTM [31] to perform clickbait classification.

Logistic Regression (LR) We train a Logistic Regression classifier on the training data with regularization by passing the parameters: {C = 20.0, dual = False, penalty = l2}. Formally, the input to the logistic regression model [29] is a text vector extracted from one of the feature representation method shown in Section IV.

LightGBM We choose the LightGBM model [30] as one of our training methods because of its high speed and consumption of less memory on large datasets. To train the LightGBM model, we used the same input feature representations as to the LR model. We pass the hyper-parameters: {max_depth = 1, min_child_samples = 13, number of decision trees = 100, boosting type = gradient boosting, learning_rate = 1.0} when training the model.

Long Short-Term Memory Networks (LSTM) In the literature, LSTMs [31] are successful in dealing with sequence-based problems in the field of NLP [32]. Using LSTM architecture, we tried the five different feature representations as input in the training process. We train each LSTM model separately for extracted token features and four-word embeddings.

Choice of Hyper-Parameters We used genetic algorithm for hyper-parameter tuning [33] to optimize both LR and LightGBM models. The hyper-parameters which resulted in best F1-score was considered.

Imbalanced Dataset Handling From section III, we observe that clickbait data has a class imbalance issue, as the majority class ("Yes") has ∼3x more instances compared to the "No" class. To circumvent this problem, we tried methods such as random under-sampling, and SMOTE (Synthetic Minority Oversampling Technique) [34] to generate synthetic samples for the minority class (non-clickbait dataset) in over-sampling method.

VI. Experiments & Results

Dataset Splitting We followed the stratified 5-fold cross-validation setup where 4-folds data is used for training, and one-fold is used for testing the model. We calculated the average of 5-folds and reported the results.

Evaluation Metrics We use classification metrics such as macro-average precision, recall, and F1-score to evaluate our methods. To understand how each class performs, we choose macro averaging that gives each class equal weight to evaluate systems performance across the two-classes.

A. Results and Analysis

The results for our 14-feature representations with two machine learning models are reported in Table IV. Impressively, the ELMo-Telugu features outperform all the feature representations, with the best model, LightGBM, in the over-sampling setting yield an F1-score of 0.94. We observed the following observations from the Table IV.

Baseline Results We considered the two traditional feature representations (BoW & TF-IDF) as our baseline features. Since the traditional features do not capture the semantic and syntactic information, the baseline system achieves an F1-measure of 0.84.

Hand-Crafted Feature Results Hand-crafted features such as Structure-based and POS-tagging outperforms the baseline features with an increasing F1-score of 0.04.

TABLE V
CLICKBAIT DETECTION RESULTS: THE TABLE DISPLAY THE FINE TUNING RESULTS OF THE EXISTING PRE-TRAINED LANGUAGE MODELS

	Fine-Tuning		
Feature set↓	Precision	Recall	F1-score
BERT	0.89	0.86	0.87
ALBERT	0.89	0.84	0.86
RoBERTa	0.91	0.87	0.89
ELECTRA	0.88	0.86	0.87
Bert-Multilingual-case	0.74	0.79	0.74
XLM-Roberta-base	0.78	0.83	0.79

Word-Vector Results The four rows in the third block of the Table IV employs the word-embedding results when passing the input features as Word2Vec, GloVe, FastText, and Meta-Embeddings. With the local and global context word-embeddings, the system achieves an F1-measure of 0.89 (Word2Vec), 0.82 (GloVe), using the LightGBM method in the over-sampling strategy. Like Word2Vec, GloVe, FastText and Meta-Embedding features as input yield an F1-score of 0.90 using the LightGBM method in the over-sampling strategy. Among the four word-embeddings, FastText, and Meta-Embedding features improve the F1-measure by 0.06 when compared with baseline. However, the performance of word embedding results when trained with Logistic Regression is lower than other methods. To interpret the results of LR and LightGBM models for Word2Vec, we use Local Interpretable Model-Agnostic Explanations (LIME) [35] tool to showcase the highlighted words causing the model towards clickbait or non-clickbait prediction in the sub-section VI-D.

Skip-Thought & Contextual Embeddings Results Using sentence embedding method Skip-Thought that performed the best with an F1 score of 0.91, higher than the methods mentioned above. Overall, LGBM provides a better results across BERT based models with similar in performance. The best F1-score is obtained when RoBERTa (0.92) and ELECTRA (0.90) feature sets are used. With contextual embedding features, ELMo increases the system's performance to 0.94 F1-score higher than all feature methods.

LSTM Results Figure 1 showcase the clickbait detection results of LSTM trained on (i) sequence of tokens, and (ii) with the four-word embeddings. From Figure 1, we observe that LSTM with the FastText features as input yields a better F1-score of 0.89 similar to the Table IV. Moreover, GloVe word vectors as input to LSTM report 0.88 F1-score higher than the F1-score displayed in Table IV.

Overall, except for distributed word-embedding results, the remaining pre-trained models show similar performance with Logistic Regression and LightGBM models.

Fine-Tuning Results We evaluate whether fine-tuning the existing pre-trained langugae models (BERT-Multilingual-case, XLM-Roberta-base), and the pre-trained models built on Telugu is useful for adapting the click-bait task. From the Table V, we find that fine-tuning results shows better performance than earlier word-embeddings and baselines. However, extracting the features from pre-trained language models have increase in performance as it may allow us to adapt a general-purpose representations, reported in Table IV.

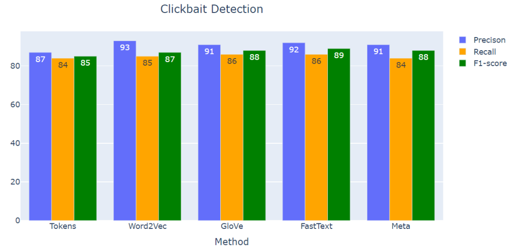

Fig. 1. Comparison of F1-score performance of input word representations i) Sequence of Tokens, ii) Word2Vec, iii) GloVe, iv) FastText, and v) Meta-Embedding. The LSTM model is used as a trained model for each of the feature representations in the 2-class setups are shown here

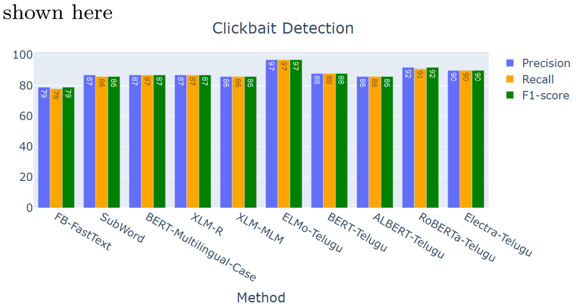

Fig. 2. Comparison of F1-score performance of i) Facebook FastText, ii) Subword Byte Pair, iii) BERT-Multilingual-Case, iv) XLM-R, v) XLM-MLM, vi) ELMo-Telugu, vii) BERT-Telugu, viii) ALBERT-Telugu, ix) RoBERTa-Telugu, and x) ELECTRA-Telugu embeddings. The LightGBM classifier is used to train each of the feature representations in the clickbait task setup is shown here.

B. Comparative Analysis

Figure 2 showcase the comparative results on clickbait-detection task, where the features extracted from existing pre-trained embeddings trained on the Telugu language, such as Bert-Multilingual-Case [1] (768-dimension), Cross-Lingual Language Model with RoBERTa (XLM-R) [3] (1024-dimension), Cross-Lingual Masked Language Model (XLM-MLM) [2] (1280-dimension), Sub-word Byte Pair embeddings [36] (300-dimension), and Facebook FastText word-embeddings [37] (200-dimension). We compare the embeddings BERT, RoBERTa, ALBERT, ELECTRA, and ELMo trained on Telugu corpus (Wikipedia + Crawled Data) with the above-mentioned pre-trained embeddings. We observe that our generated pre-trained embeddings for the Telugu language outperform the existing pre-trained embeddings in the two-class clickbait-detection task. Here, our best model's performance is from the ELMo feature representation with an F1-score of 0.94, an improvement of 0.05 than existing pre-trained methods. These results indicate that the performance of the generated pre-trained embeddings BERT, ELMo, RoBERTa, & ELECTRA on Telugu corpus would have been much better than existing pre-trained embeddings.

C. Error Analysis

We analyzed the error cases in detail for both the clickbait and non-clickbait samples. We observed from

TABLE VI
ELMo: Confusion matrix for clickbait classification

		Predicted	
		Non-Clickbait	Clickbait
Actual	Non-Clickbait	6092	545
	Clickbait	604	15291

TABLE VII
ELMo: Wrong predictions by the model

Headline	Act	Pred	Reason
వృద్ధాప్య లక్షణాలను దూరం చేయడానికి ఇవి ఫాలో అవ్వండి.... vqxXApya lakRaNAlanu xUraM ceyadAniki.... Follow these to get rid of the signs of aging ...	yes	no	Model could not understand the implicature
వాటిలో తెలంగాణ పోలీసులు ముందున్నారు... vAtilo weVlaMgANa polIsulu muMxunnAru... Among them, the Telangana police are leading ...	yes	no	Model could not understand the deictic information
భిన్న కోణాల్లో దర్యాప్తు చేస్తున్న పోలీసు బృందం... Binna koNAllo xaryApwu ceswunna polIsu bqMxa... Police team investigating from different angles ...	no	yes	Model guessed Noun ellipsis as clickbait

Table VI that (i) 3.8% cases clickbait headlines predicting as non-clickbait, and (ii) 8.2% cases non-clickbait headlines predicting as clickbait. Our system makes incorrect predictions where the sentences contain the implicatures. Table VII display the failure cases where the model got confused due to the pragmatic notations like "Implicatures", "Deictics", and "Noun Ellipsis". We report the detailed confusion matrices for contextualized word embeddings (BERT, ALBERT, RoBERTa, and ELECTRA) in Tables VIII, IX, X,and XI respectively. Observations from the Tables VIII, IX, X,and XI that both ELECTRA and RoBERTa models have low false positives compared to BERT and ALBERT.

TABLE VIII
BERT: Confusion matrix for clickbait classification

		Predicted	
		Non-Clickbait	Clickbait
Actual	Non-Clickbait	5510	1184
	Clickbait	1027	14811

TABLE IX
ALBERT: Confusion matrix for clickbait classification

		Predicted	
		Non-Clickbait	Clickbait
Actual	Non-Clickbait	5367	1327
	Clickbait	1375	14463

TABLE X
RoBERTa: Confusion matrix for clickbait classification

		Predicted	
		Non-Clickbait	Clickbait
Actual	Non-Clickbait	5776	918
	Clickbait	604	15234

TABLE XI
ELECTRA: Confusion matrix for clickbait classification

		Predicted	
		Non-Clickbait	Clickbait
Actual	Non-Clickbait	5412	1282
	Clickbait	565	15273

D. LIME Results

Figure 3 showcase the words highlighted for both LR and LightGBM models using LIME. We can observe from the Figure 3 that the LightGBM model capture attractive words when compared with the LR model. Although the input sentence in Table XII is a clickbait, the LR model is predicted as non-clickbait whereas the LightGBM model is predicted correctly.

TABLE XII
Lime: Input sentence in Telugu, WX, and English.

Headline
సూర్య 24.. ఏంటి కథ?. sUrya 24.. eMti kaWa? Surya 24 .. What story ?.

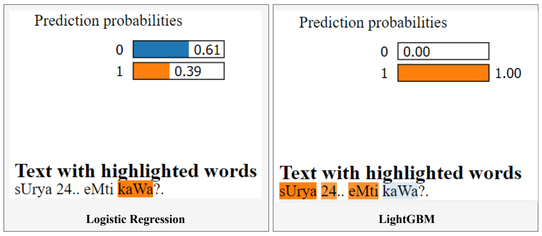

Fig. 3. Local Interpretable Model-Agnostic Explanations: showcase the words highlighted for both LR and LightGBM models.

TABLE XIII
Post-hoc Tukey Test Results

Feature Pairs	P-value	Statistically Significant
(ALBERT vs. GloVe)	0.001	Yes
(BERT vs. GloVe)	0.001	Yes
(BOW vs. ELMo)	0.001	Yes
(BOW vs. GloVe)	0.0081	Yes
(ELECTRA vs. GloVe)	0.001	Yes
(ELMo vs. FastText)	0.001	Yes
(ELMo vs. Meta-Embeddings)	0.001	Yes
(ELMo vs. TF-IDF)	0.001	Yes
(GloVe vs. POS Tags)	0.001	Yes
(GloVe vs. Structure Based)	0.001	Yes
(ALBERT vs. Meta-Embeddings)	0.9	No
(Structure Based vs. Pos Tags)	0.9	No
(BoW vs. ELECTRA)	0.792	No
(BoW vs. TF-IDF)	0.9	No
(TF-IDF vs. Word2vec)	0.9	No
(RoBERTa vs. Word2vec)	0.0101	No
(ELECTRA vs. TF-IDF)	0.5856	No
(FastText vs. Skip-Thought)	0.1215	No
(Skip-Thought vs. Word2vec)	0.0232	No

E. Statistical Analysis

Statistical significance using one-Anova test [38] showcase the significant difference between the 14 feature representations used in the above experiments. Here, we perform the statistical significance test on a clickbait-detection task with a two-class setup. To perform the Anova test, we use the three-sampling results (F1-measure) of two training classifiers LR, LightGBM. The one-way Anova test provides an F-statistic [F(13,70) = 9.8035, P = 2.9e^{-11}] concludes 14 feature representations are significantly different. We use post-hoc Tukey-HSD test [39] to obtain the results between different pairs reported in Table XIII.

VII. Conclusion

In this work, we present a new annotated dataset of approximately 113k headlines that can be used for building an automated clickbait detection system for a resource-poor language Telugu. Here the evaluation of two machine learning classifiers and LSTM based models suggest that simple machine learning classifiers with ELMo features perform better than the baseline and LSTM based models. As a future task, we aim to look at the strength of the clickbait (going beyond the binary class) and explore related tasks like fake news identification or sarcasm detection.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Multilingual bert -r," *https://github.com/google-research/bert/blob/master/multilingual.md*, 2018.

[2] G. Lample and A. Conneau, "Cross-lingual language model pretraining," *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[3] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, É. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8440–8451.

[4] J. C. S. Dos Rieis, F. B. de Souza, P. O. S. V. de Melo, R. O. Prates, H. Kwak, and J. An, "Breaking the news: First impressions matter on online news," in *Ninth International AAAI conference on web and social media*, 2015.

[5] G. Loewenstein, "The psychology of curiosity: A review and reinterpretation." *Psychological bulletin*, vol. 116, no. 1, p. 75, 1994.

[6] A. Anand, T. Chakraborty, and N. Park, "We used neural networks to detect clickbaits: You won't believe what happened next!" in *European Conference on Information Retrieval*. Springer, 2017, pp. 541–547.

[7] V. Indurthi, S. R. Oota, M. Gupta, and V. Varma, in *Proceedings of the ACM India joint international conference on data science and management of data*, 2018, pp. 257–264.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[9] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[10] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," *arXiv preprint arXiv:2003.10555*, 2020.

[11] A. Chakraborty, B. Paranjape, S. Kakarla, and N. Ganguly, "Stop clickbait: Detecting and preventing clickbaits in online news media," in *2016 ieee/acm international conference on advances in social networks analysis and mining (asonam)*. IEEE, 2016, pp. 9–16.

[12] M. Potthast, S. Köpsel, B. Stein, and M. Hagen, "Clickbait detection," in *European Conference on Information Retrieval*. Springer, 2016, pp. 810–817.

[13] A. Agrawal, "Clickbait detection using deep learning," in *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*. IEEE, 2016, pp. 268–272.

[14] P. Biyani, K. Tsioutsiouliklis, and J. Blackmer, "" 8 amazing secrets for getting more clicks": Detecting clickbaits in news streams using article informality," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[15] S. Gairola, Y. K. Lal, V. Kumar, and D. Khattar, "A neural clickbait detection engine," *arXiv preprint arXiv:1710.01507*, 2017.

[16] M. M. U. Rony, N. Hassan, and M. Yousuf, "Diving deep into clickbaits: Who use them to what extents in which topics with what effects?" in *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017*, 2017, pp. 232–239.

[17] S. Kaur, P. Kumar, and P. Kumaraguru, "Detecting clickbaits using two-phase hybrid cnn-lstm biterm model," *Expert Systems with Applications*, p. 113350, 2020.

[18] V. Indurthi and S. R. Oota, "Clickbait detection using word embeddings," *arXiv preprint arXiv:1710.02861*, 2017.

[19] M. Potthast, T. Gollub, M. Hagen, and B. Stein, "The clickbait challenge 2017: Towards a regression model for clickbait strength," *arXiv preprint arXiv:1812.10847*, 2018.

[20] I. Y. R. Pratiwi, R. A. Asmara, and F. Rahutomo, "Study of hoax news detection using naïve bayes classifier in indonesian language," in *2017 11th International Conference on Information & Communication Technology and System (ICTS)*. IEEE, 2017, pp. 73–78.

[21] X. Zhou, J. Cao, Z. Jin, F. Xie, Y. Su, D. Chu, X. Cao, and J. Zhang, "Real-time news cer tification system on sina weibo," in *Proceedings of the 24th International Conference on World Wide Web*, 2015, pp. 983–988.

[22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[23] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[24] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.

[25] W. Yin and H. Schütze, "Learning word meta-embeddings," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1351–1360.

[26] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Advances in neural information processing systems*, 2015, pp. 3294–3302.

[27] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[29] H.-F. Yu, F.-L. Huang, and C.-J. Lin, "Dual coordinate descent methods for logistic regression and maximum entropy models," *Machine Learning*, vol. 85, no. 1-2, pp. 41–75, 2011.

[30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, 2017, pp. 3146–3154.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[32] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu, "Text classification improved by integrating bidirectional lstm with two-dimensional max pooling," *arXiv preprint arXiv:1611.06639*, 2016.

[33] T. T. Le, W. Fu, and J. H. Moore, "Scaling tree-based automated machine learning to biomedical big data with a feature set selector," *Bioinformatics*, vol. 36, no. 1, pp. 250–256, 2020.

[34] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[35] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[36] B. Heinzerling and M. Strube, "BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 7-12, 2018 2018.

[37] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[38] H.-Y. Kim, "Analysis of variance (anova) comparing means of more than two groups," *Restorative dentistry & endodontics*, vol. 39, no. 1, pp. 74–77, 2014.

[39] G. D. Ruxton and G. Beauchamp, "Time for some a priori thinking about post hoc testing," *Behavioral ecology*, vol. 19, no. 3, pp. 690–693, 2008.