Goldsmiths Research Online

Goldsmiths Research Online (GRO)
is the institutional research repository for
Goldsmiths, University of London

Citation

Kumar, Akshi; Mallik, Abhishek and Kumar, Sanjay. 2024. HumourHindiNet: Humour detection in Hindi web series using word embedding and convolutional neural network. ACM Transactions on Asian and Low-Resource Language Information Processing, 23(7), 98. ISSN 2375-4699 [Article]

Persistent URL

https://research.gold.ac.uk/id/eprint/36169/

Versions

The version presented here may differ from the published, performed or presented work. Please go to the persistent GRO record above for more information.

If you believe that any material held in the repository infringes copyright law, please contact the Repository Team at Goldsmiths, University of London via the following email address: gro@gold.ac.uk.

The item will be removed from the repository while any claim is being investigated. For more information, please contact the GRO team: gro@gold.ac.uk

Goldsmiths
UNIVERSITY OF LONDON

HumourHindiNet: Humour detection in Hindi web series using word embedding and convolutional neural network

AKSHI KUMAR, Department of Computing, Goldsmiths, University of London, United Kingdom
ABHISHEK MALLIK, Department of Computer Science and Engineering, Delhi Technological University, India
SANJAY KUMAR*, Department of Computer Science and Engineering, Delhi Technological University, India

Humour is a crucial aspect of human speech, and it is, therefore, imperative to create a system that can offer such detection. While data regarding humour in English speech is plentiful, the same cannot be said for a low-resource language like Hindi. Through this paper, we introduce two multimodal datasets for humour detection in the Hindi web series. The dataset was collected from over 500 minutes of conversations amongst the characters of the Hindi web series $Kota-Factory$ and $Panchayat$. Each dialogue is manually annotated as Humour or Non-Humour. Along with presenting a new Hindi language-based Humour detection dataset, we propose an improved framework for detecting humour in Hindi conversations. We start by preprocessing both datasets to obtain uniformity across the dialogues and datasets. The processed dialogues are then passed through the Skip-gram model for generating Hindi word embedding. The generated Hindi word embedding is then passed onto three convolutional neural network (CNN) architectures simultaneously, each having a different filter size for feature extraction. The extracted features are then passed through stacked Long Short-Term Memory (LSTM) layers for further processing and finally classifying the dialogues as Humour or Non-Humour. We conduct intensive experiments on both proposed Hindi datasets and evaluate several standard performance metrics. The performance of our proposed framework was also compared with several baselines and contemporary algorithms for Humour detection. The results demonstrate the effectiveness of our dataset to be used as a standard dataset for Humour detection in the Hindi web series. The proposed model yields an accuracy of 91.79 and 87.32 while an F1 score of 91.64 and 87.04 in percentage for the $Kota-Factory$ and $Panchayat$ datasets, respectively.

CCS Concepts: • **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

Additional Key Words and Phrases: Convolutional Neural Network (CNN), Hindi Web Series, Humour Detection, Long Short-Term Memory (LSTM), Low-Resource Languages, Social networks, Skip-gram Hindi Word Embedding

ACM Reference Format:
Akshi Kumar, Abhishek Mallik, and Sanjay Kumar. 2023. HumourHindiNet: Humour detection in Hindi web series using word embedding and convolutional neural network. In . ACM, New York, NY, USA, 21 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In recent years, low-resource languages have been the epicenter of numerous research problems for Natural Language Processing (NLP) tasks such as the detection of fake news, hate speech, offensive language, sentiment analysis, emotion classification, and many others [9, 13, 15, 16, 27]. Moreover, the recent developments in language analysis have helped detection and mitigation of humour in conversations [8, 29]. Humour can be characterized as the attribute of

*corresponding Author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
Manuscript submitted to ACM

being entertaining or comical, particularly as manifested in writing or speech. We frequently integrate humor into our discussions to elevate the conversation's quality or convey points in a more lighthearted manner. Since humor is typically conveyed through speech, it becomes imperative to devise programs capable of identifying humor in spoken language. Spoken humour is dependent on a multitude of factors, including but not limited to context, timing, and delivery [26]. Humour helps in effectively establishing communication channels and social relationships [7]. Humour detection refers to the task of detecting humour in various forms of conversation. The task of recognizing humour in conversations or day-to-day exchanges is called humour detection. Over the years, humour detection has gained great popularity in industry and academia. However, detecting humour is not a simple task due to several reasons the idiosyncrasies of humour are opposed to the dynamics of regular language. Moreover, humour is often expressed through subtle facial expressions body movements, or hand gestures that aren't easily captured. The use of sarcasm in between lines also adds to the difficulty of detecting humour in conversations. humour detection also varies as per age, gender, or ethnic background [32]. Humour detection frameworks can be used in sentiment analysis and inference [12]. With thorough analysis, they're able to interpret the conversations as humorous or not [6].

Humour detection in low-resource languages like Hindi, Urdu, Bengali, Telugu, and others is challenging due to limited data, cultural specificity, and the lack of pre-trained models. The scarcity of diverse and annotated data, coupled with the difficulty in capturing cultural nuances, poses obstacles to developing accurate and robust humour detection models for languages with fewer resources. Even though extensive work has been done in the field of humour detection. However, its application to the Indian subcontinent has still been very restricted. Amongst the various languages being spoken in India, Hindi is the most popular one, with about half a billion Hindi speakers in India alone. While there is sufficient data in the field of Humour detection for English, support for Indian languages is still required. The conventional methods used for the English language can't be used for Hindi as they differ significantly from each other across the dimensions. The methodology needed for the Hindi language is very different from the English language, from preprocessing to feature extraction to classification owing to the grammatical, syntactical, and semantical features of both languages.

The lack of availability of humour detection datasets in the Hindi language has motivated us to create a multi-modal dataset for humour detection in the Hindi web series. The datasets were collected from two popular Hindi web series. The entire scene was split into various dialogues, and then each dialogue was manually annotated as Humour or Non-Humour. The entire dataset spanned through the 1st season of both the web series and over 500 minutes of run-time. The exhaustive Hindi dataset proposed by us can be used as a benchmark dataset for Humour detection in the Hindi language or for other text classification or NLP based tasks related to the Hindi language.

In our pursuit of humour detection in the Hindi web series, we confront not only the scarcity of dedicated datasets but also the paucity of tailored word embedding techniques for the Hindi language. This research endeavors to bridge these critical gaps by proposing an innovative architecture that seamlessly integrates word embedding and convolutional neural networks to decode humour in the context of Hindi entertainment. The proposed work starts with data preprocessing, where we meticulously cleanse the raw dialogue excerpts harvested from web series transcripts. This preparatory phase meticulously rids the data of superfluous noise and extraneous special characters, establishing a foundation of data uniformity. Next, we introduce a pioneering Hindi-centric word embedding framework, cultivated through the training of a Skip-gram model on a corpus of Hindi text. This novel word embedding schema discerningly captures the nuances of Hindi language semantics and the intricate contextual intricacies embedded within the dialogues. Subsequently, we employ a trio of distinct convolutional neural network architectures, each wielding a unique filter size, to meticulously sift through the word embeddings. These parallel networks collectively unearth a rich tapestry of features, each tailored to a specific facet of the data, be it structural, semantic, or context-driven. These extracted features then embark on a transformative journey through stacked LSTM layers, where they undergo deep processing to uncover latent interdependencies among the words, a crucial step in deciphering humour's intricate timing and subtleties. These LSTM layers, serving as the final arbiter, adeptly categorize the dialogues into their respective humour or non-humour domains, encapsulating the essence of our pioneering HumourHindiNet architecture. We run intensive simulations on both datasets and evaluate several performance metrics. We compare the performance of our proposed framework with various baseline algorithms and several recently proposed Humour detection techniques. The experimental analysis reveals the utility of our proposed dataset to be used as a benchmark dataset for Humour detection in the Hindi web series. The experimental results also reveal the superiority of our proposed framework. The major contributions of this work can be summarised below.

(i) Two datasets for humour detection in the Hindi language are introduced. The dataset was collected from two popular Hindi web series, and each dialogue was manually annotated.
(ii) We perform meticulous data preprocessing techniques, involving noise reduction and special character removal, laying a strong foundation for robust analysis. These preprocessing methods are valuable not only for humour detection but also for enhancing the overall quality of other related tasks involving Hindi text.
(iii) We introduce a novel Hindi word embedding model, a custom Skip-gram model trained on a Hindi text corpus. This equips the model to capture the subtle nuances and contextual intricacies specific to the Hindi language.
(iv) The concurrent use of multiple CNNs, each with distinct filter sizes, exemplifies an innovative approach to feature extraction. This technique allows the model to extract a diverse range of features from the input data, enhancing its ability to discern humour through structural, semantic, and contextual cues.
(v) We leverage stacked LSTM layers for sequential dependency analysis, which enables our model to delve deeper into the linguistic intricacies of humour, capturing nuances such as timing and word interdependencies.

The rest of the paper is organized as follows. Section 2 discusses some recent work done in the field of Humour detection. Section 3 describes the various details of the Hindi Humour detection datasets created by us. Section 4 illustrates the various phases of our proposed framework. The experimental results obtained by us are discussed in Section 6. Section 7 presents the concluding remarks and the scope for future improvement for this work.

2 RELATED WORK

In this section, we discuss some of the recent work done in the field of Humour detection. Modeling humour is fairly difficult as compared to other text classification tasks due to several factors like idiosyncrasy and contextual dependencies [10]. Idiosyncrasy refers to the use of twisted words, hand or body gestures, or misalignment of words to get across the humour. On the other hand, contextual dependencies refer to the fact that most of the time, a punchline is based on the conversations leading up to that point. Some of the English datasets that have been proposed in the recent study are: Big Bang Theory [1], Ted Laughter [4], PTT Jokes [5], Pun of the Day [30], and 16000 One-Liners [21]. Recently, the field of humour detection has drawn a lot of attention from both the industry and the research community. Consequently, a lot of work has been done to improve the performance of the humour detection frameworks. The existing works can be broadly classified into three main categories, namely, Language analysis, Feature engineering, and machine learning-based approaches.

Language analysis-based approaches refer to the analysis of the multi-modalities of the text, audio, and video to categorize the text as humourous or non-humourous [31]. Wang et al. [28] considered not only the literal meanings of the words but also the context in which they were spoken. They proposed a framework to model the visual and acoustic patterns that appear in the spoken word segments. Moreover, they also capture the dynamic nature of humour by modifying word representations based on shifting non-verbal behaviors. Pham et al. [24] propose a method to learn joint representations for source and target modalities by using the source modality as input and injecting the target modality at runtime. This reduces disturbance when making transitions and makes the model robust. A conversational memory network to detect emotions in video conversations was proposed by Hazarika et al. [11]. They leverage historical contextual information using a deep neural network architecture. They present a multimodal approach by combining textual, audio, and video features with the vanilla-gated recurrent neural network. It helped them in modeling the past utterances of each speaker into memories. The created memories are then merged using an attention framework to capture the inter-speaker dependencies. Liang et al. [18], proposed a Recurrent Multistage Fusion Network (RMFN) model. It solves the problem of fusing the various modalities by decomposing them into various stages, with each of them focused on a subset of multimodal signals. This helps in the particular and efficient fusion of the modalities. The multi-stage fusion approach builds upon the intermediate representations of the previous stages to model the cross-modal interactions. They finally used a recurrent neural network along with the fusion approach to integrate the temporal and cross-modal interactions. Chauhan et al. [3] proposed a Multimodal Multiparty Hindi Dataset For humour recognition in conversations. It has 6,191 utterances from 13 episodes of a top-rated TV series "Shrimaan Shrimati Phir Se". The various utterances of the episodes are labeled as humour and non-humour, containing acoustic, visual, and textual modalities. They also propose several multimodal baseline algorithms to show contextual and multimodal information for humour recognition in conversations.

Feature engineering-based approaches refer to explicitly modeling features based on the task and data at hand. It usually requires domain expertise and proficiency. Various works have been done to explore the feature engineering-based approach by developing features along the lines of effective dimensions [20], distribution of part of speech [14], stylistic [23], etc. Mihalcea and Strapparava [21] defined stylistic features by working on adult slang, antonyms, and alliteration. Yang et al. [30], engineered four structural features for humour and created characteristic sets for each of the structural features. Based on experiments they concluded that incongruity and ambiguity are the best performers as compared to the other latent semantic structural features. Even though feature engineering-based approaches give good results they have a major limitation in terms of generalization. This is due to the fact that feature engineering has to be done individually for each dataset instead of the task in general.

The popularity of deep learning methods for NLP based tasks has paved the way for exploring the applications of deep learning methods in the field of humour detection. Oliveira and Rodrigo [6] presented a study for detecting humour in Yelp reviews. They used bag-of-words and word vectors as the features for the dialogues. They started the classification study with shallow methods like Random forest and linear discriminant. Then they moved on to more complex methods like deep feed-forward neural networks, recurrent neural networks, and convolutional neural networks. Bertero and Fung [2] presented the first-ever application of using long short-term memory for humour detection in dialogues of popular sitcoms. They used the canned laughter in the audio files as annotations for categorizing the dialogues as humourous or non-humourous. They use long short-term memory for modeling the punchline relation of the conversational humour while the dialogue encodings were obtained from convolutional neural networks. Bertero and Fung [1] proposed a deep learning-based approach by combining the word-level and audio-level features using a linear-chain Conditional Random Field over the convolutional neural network and recurrent neural network. They also generate a new humour detection dataset using a very popular sitcom "*The Big Bang Theory*". Chen and Soo [5], implemented a CNN based approach with extensive filter sizes, numbers, and highway networks to enhance the depth of the deep learning architecture. They also collected and constructed four datasets for conducting experiments on humour detection, with distinct joke types in both English and Chinese. Kumar et al. [17], proposed DeepHumor, an automatic humour detection model. It is an amalgamation of CNN, LSTM, and highway networks. They used CNN layers for feature extraction, and the LSTM layers were used for sequence learning. The performance of the model is enhanced using the highway network. To overcome the overfitting problem, they also add dropout layers to their architectures. They compare the performance of their proposed model with several recently developed techniques over the Yelp user review dataset.

Table 1. Statistical details about the Hindi dataset for Humour detection created by us

Description	$Kota - Factory$	$Panchayat$
Number of Episodes	5	8
Total number of instances	2986	4145
Total number of "$Humour$" instances	1344	1713
Total number of "$Non - Humour$" instances	1642	2432

3 CREATED DATASET

In this section, we illustrate the Hindi dataset for humour detection created by us. For this purpose, we have gathered data from two popular Hindi web series, namely, $Kota - Factory$ and $Panchayat$. The data is collected from the first season of both web series. $Kota-Factory$ has 5 episodes while $Panchayat$ has 8 episodes. We have extracted utterances from different scenes from each episode and manually annotated them as "$Humour$" or "$Non - Humour$"—the various differentiators and context help in classifying the dialogues or utterances into their respective categories. The statistical details of both datasets are presented in Table 1. It gives us a clearer depiction of the empirical view of the data. Table 2 and Table 3 show the sample dialogues along with their respective categories for $Kota - Factory$ and $Panchayat$ datasets, respectively. For instance, "He founded this Jurassic park" dialogue in the "$Kota - Factory$" example refers to the coaching institute in the context of being referred to as a Jurassic park, hence it is a humour instance. While the "I'll let you know when we get there. We will not kidnap you" example from the "$Panchayat$" dataset, refers to the bus conductor saying that he'll drop the passenger at his destination and won't kidnap him. Hence, this can be considered as humour too. The exhaustive Hindi dataset proposed by us can be used as a benchmark dataset for Humour detection in the Hindi language or for other text classification or NLP-based tasks related to the Hindi language. The dataset can be accessed from here when the paper gets published HumourHindiNet dataset. The proposed datasets might be subject to some biases such as selection bias, as the choice of specific web series may limit the diversity of humour styles represented; subjectivity bias, as manual annotation of utterances, may lead to varied interpretations of comedic elements; and sampling bias, as extracting utterances from specific scenes may overlook nuanced comedic moments elsewhere in the series. By acknowledging and addressing these biases, researchers can enhance the validity and generalizability of their findings in humour detection research.

4 PROPOSED WORK

In this section, we illustrate our approach to the detection of humour in the Hindi web series by leveraging the power of word embedding and convolutional neural networks. The proposed model utilizes word embedding techniques that transform individual Hindi words into dense vectors, preserving semantic relationships and contextual nuances crucial

Table 2. Sample dialogues along with their categories for *Kota − Factory* Dataset.

Dialogue in Hindi	Dialogue in English	Class
इस जुरासिक पार्क की स्थापना उन्होंने की थी।	He founded this Jurassic Park.	Humour
महिला होगी तेरी माँ, लड़की कहते हैं	woman will be your mother, people say girl	Humour
फ्रिज सार्वजनिक है, मैं नहीं।	The fridge is public, I'm not.	Humour
क्या तुम पीयूष से बात कर सकते हो?	Can you talk to Piyush?	Non-Humour
हमें तब ही एडमिशन लेना चाहिए था।	We should have taken admission only then.	Non-Humour
और आईआईटी में सीट कितनी हैं ?	And how many seats are there in IIT?	Non-Humour

Table 3. Sample dialogues along with their categories for *Panchayat* Dataset.

Dialogue in Hindi	Dialogue in English	Class
जब हम वहां पहुंचेंगे तो आपको बता दूंगा। हम तुम्हारा अपहरण नहीं करेंगे।	I'll let you know when we get there. We will not kidnap you.	Humour
और आप मिट्टी के पुत्र होने का मौका छोड़ना चाहते हैं	And you wanna miss the chance to be the son of the soil	Humour
अगर आपको मेरा काम इतना पसंद है तो आप भी अप्लाई क्यों नहीं करते।	If you like my work so much why don't you apply too.	Humour
झे फुलेरा नामक गांव में तैनात किया गया है।	I have been posted in a village called Phulera.	Non-Humour
मुझे कॉलेज में बहुत मेहनत करनी चाहिए थी।	I should have worked very hard in college.	Non-Humour
आप केवल नकारात्मक को उजागर कर रहे हैं। सकारात्मक पर भी ध्यान दें।	You are only exposing the negative. Focus on the positive too.	Non-Humour

for understanding humour. These embeddings serve as the initial input to a CNN, specifically tailored to capture complex linguistic structures inherent in comedic content. The CNN acts as a feature extractor, identifying subtle cues such as wordplay, sarcasm, and comedic timing by analyzing the input text at different levels of abstraction. Importantly, the incorporation of stacked LSTM layers further enhances the model's capability by capturing sequential dependencies and temporal context within the text. The stacked LSTM layers further enable the model to capture sequential dependencies, vital in discerning comedic timing and punchlines. This allows HumourHindiNet to recognize the sequential flow of jokes, discern comedic timing, and accurately identify punchlines, contributing to a more nuanced understanding of comedic content. By amalgamating these techniques, HumourHindiNet offers a sophisticated and effective solution to the challenging task of humour detection in Hindi web series. Furthermore, its contributions extend beyond natural language processing, as it facilitates deeper insights into entertainment analytics, enabling better analysis and understanding of comedic elements within digital content produced in the Hindi language.

The various stages of the proposed framework are described below.

4.1 Data Preprocessing

In the data preprocessing phase, we diligently address the challenges presented by the raw transcripts extracted from the Hindi web series. These transcripts often contain a myriad of special characters, URLs, and other extraneous elements that introduce non-pertinent noise into the dialogues, potentially obstructing accurate humour detection. To mitigate this, we employ a range of text-cleaning techniques, a standard practice in Natural Language Processing, to systematically rid the data of such distractions. Given the linguistic complexities of the Hindi language, we harness the capabilities of the iNLTK library, a publicly available resource equipped with fundamental functions tailored for Natural Language Processing in Indian languages. This library aids in refining our approach to processing the obtained Hindi dialogues, ensuring linguistic subtleties are preserved. Furthermore, recognizing the pivotal role of class labels in our model's performance, we encode the dialogue labels as 2-dimensional binary arrays. In this encoding scheme, dialogues categorized as humour are represented as [1, 0], while those classified as non-humour are denoted as [0, 1]. This encoding enhances our model's processing capabilities, facilitating more efficient and precise classification of humour and non-humour dialogues in the subsequent stages of our analysis.

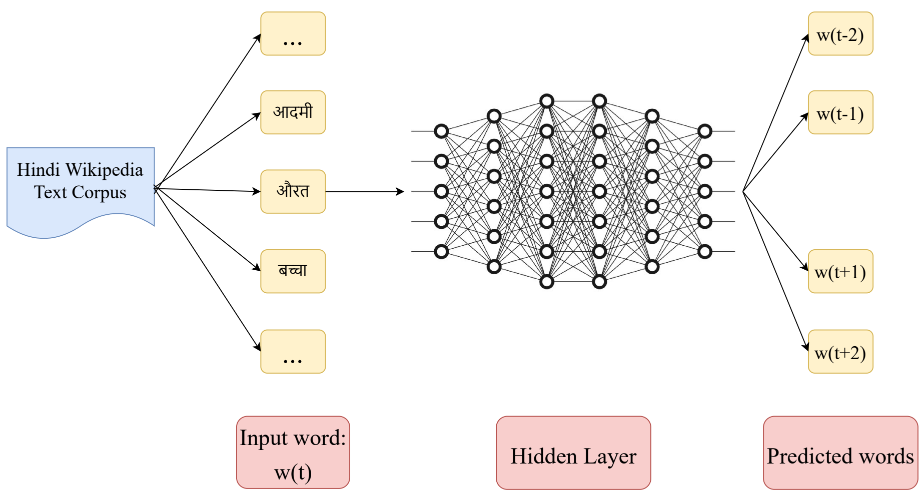


Fig. 1. Process of generating Hindi Word Embeddings using Skip-gram architecture.

4.2 Skip-gram model based Hindi Word Embeddings generation

In this section, we address the need to transform the processed Hindi dialogues, obtained in the previous stage, into numerical vectors suitable for machine learning and deep learning models. Since these dialogues remain in their original Hindi form, they can't be directly utilized with such models. To bridge this gap, we employ a method commonly employed in NLP: word embeddings. While Word2Vec and Global Vectors (GloVe) are popular choices for generating word embeddings, they are primarily tailored for the English language, rendering them less effective for Hindi text.

To overcome this limitation, we embark on the creation of our Hindi-specific word embeddings. To achieve this, we employ a Skip-gram model, a technique pioneered by Mikolov et al. [22]. The Skip-gram model stands out for its ability to predict context based on a target word, a departure from the Continuous Bag of Words (CBOW) model, which predicts the target word based on context [19]. This predictive capability aligns seamlessly with our goal of humour

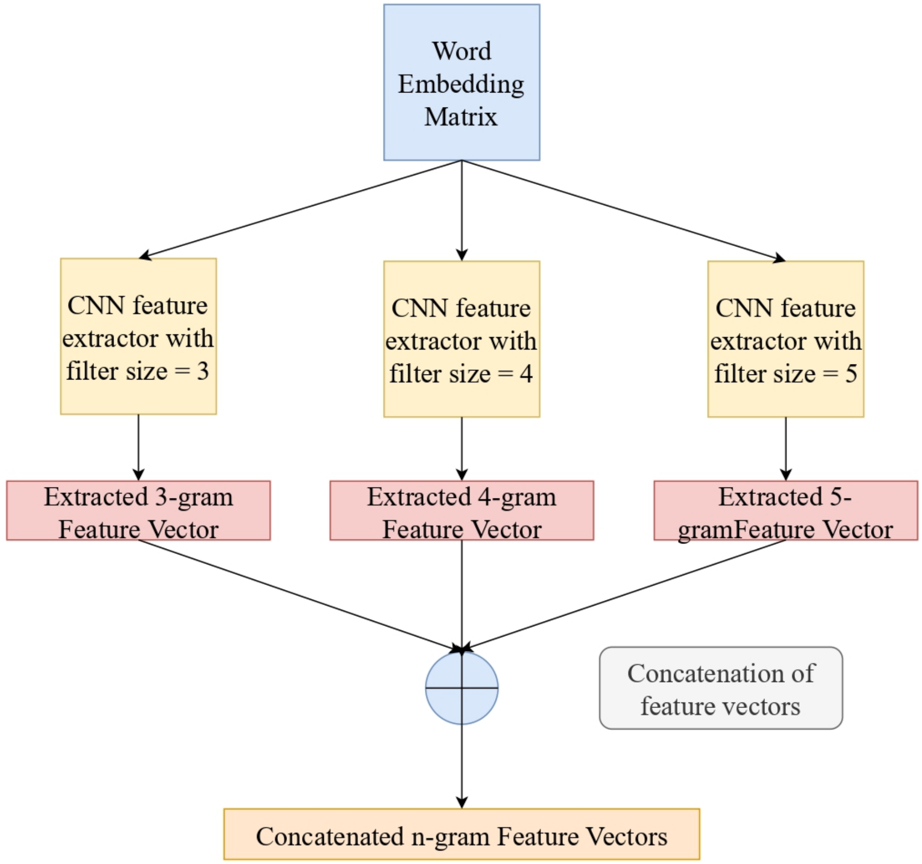


Fig. 2. Process of extracting n-gram features using CNN architecture.

detection, as it aids our framework in better comprehending the contextual nuances of humour or non-humour in dialogues. Our Skip-gram model is meticulously trained on the Hindi Wikipedia Corpus, a publicly accessible resource here. During training, we iteratively update the feature vectors to ensure that target words and their corresponding contexts are positioned closer to each other in the feature space. As a result, we generate feature vectors of dimension 300 for each word within the dialogues. This dimensionality strikes a balance between capturing a rich information set from the dialogue corpus while maintaining computational scalability, ultimately bolstering the effectiveness of our humour detection framework. The process of generating the Hindi word embedding using the Skip-gram architecture is shown in Figure 1

4.3 N-gram feature extraction using CNN

In this section of our model, we delve into the process of extracting vital features from the feature vectors acquired in the preceding stages. These feature vectors are transformed into a 2-dimensional matrix, a pivotal step in preparing them for efficient processing by a CNN. This matrix representation mimics the format of an image, aligning well with the CNN architecture, which inherently operates on 2-dimensional data.

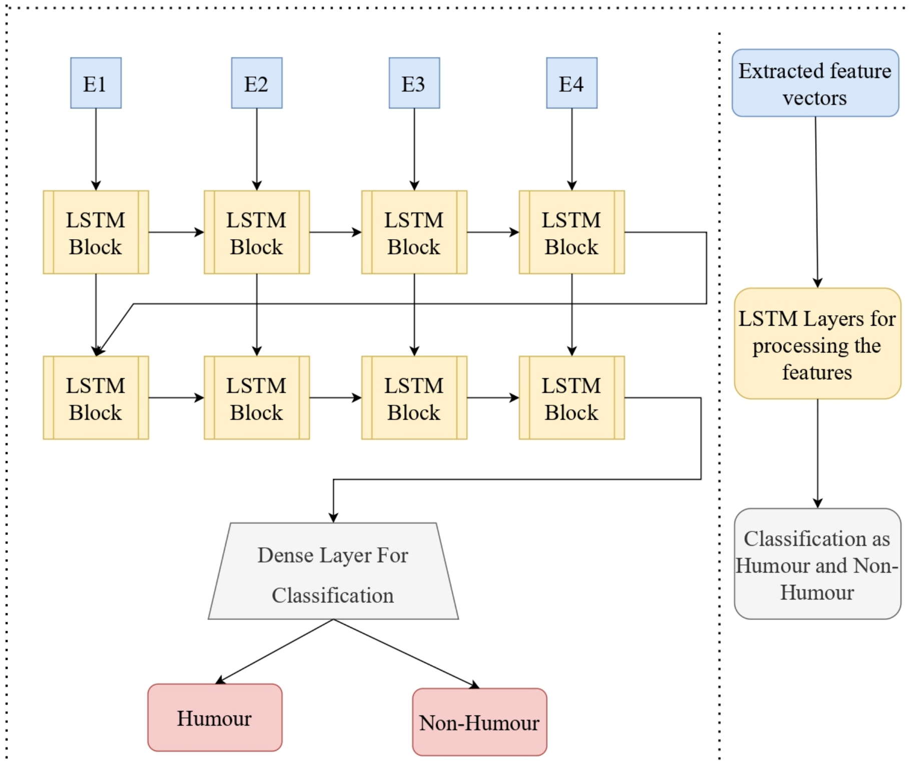


Fig. 3. Process of classification of dialogues using stacked LSTM layers

Within our framework, we concurrently feed these 2-dimensional dialogue matrices into three distinct CNN architectures, each employing filter sizes of 3, 4, and 5. This simultaneous processing enables the extraction of n-gram features, encompassing 3-grams, 4-grams, and 5-grams, dispersed throughout the dialogues. This approach synergizes with the context-capturing prowess of the Skip-gram-based word embeddings, as the CNN-based n-gram feature extraction probes various word prefixes and suffixes to unearth intricate contextual nuances. To further harness the information gleaned from these n-gram features, we consolidate them through a concatenation layer. This amalgamation enhances feature utilization and preserves the sentence's semantic structure, judiciously situating context and target words to facilitate more precise humour detection. The process of extracting the n-gram features using parallel CNN architectures is shown in Figure 2

4.4 Humour Classification using Stacked LSTM Layers

In this section of our model, we take the extracted features from the previous stages and subject them to the scrutiny of stacked LSTM layers. This critical phase serves as the decision-making core of our humour detection framework, where dialogues are definitively classified as either Humour or Non-Humour. LSTMs were chosen due to their well-established effectiveness in handling data holistically and their exceptional aptitude for processing sequential data while retaining contextual nuances. The recurrent nature of LSTMs, coupled with their memory cells, equips them to capture long-term dependencies without falling prey to the vanishing gradient problem. Consequently, the stacked LSTM layers represent an optimal choice for processing the extracted features. They adeptly explore these features, ferreting out hidden long-term contextual dependencies that are often pivotal in humour comprehension The processed features are then passed into a deep neural network to ultimately assign the appropriate labels of Humour or Non-Humour to the dialogues, constituting the final classification step in our approach. The process of classifying the dialogues as Humour or Non-Humour is shown in Figure 3.

5 EVALUATION METRICS

In this section, we explain the various evaluation metrics used for estimating the performance measure of our proposed algorithm are described as follows.

(i) Accuracy: Accuracy is an indicator of the share of correctly made classifications to the total number of classifications. Mathematically, it can be expressed as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

(ii) Precision: Precision is an indicator of the share of positive classifications made that were actually true. Mathematically it can be expressed as follows.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

(iii) Recall: It is a performance metric, which attempts to implicate what share of actual positives were identified correctly. For a binary classification task, recall can be expressed mathematically as follows.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

(iv) F1 score: F1 score indicates the balance between precision and recall. Formally speaking, it is the harmonic mean of both precision and recall. Mathematically it is expressed as follows.

$$F1\ score = \frac{2 * (Precision) * (Recall)}{Precision + Recall} \tag{4}$$

Here, True positive (TP) - Correctly identified, True Negative (TN) - Correctly rejected, False positive (FP) - Incorrectly identified, and False negative (FN) - Incorrectly rejected.

6 EXPERIMENTAL RESULTS

In this section, we discuss the experimental analysis of the proposed work. All the experiments are conducted on both the datasets proposed in Section 3, and several standard performance metrics like Accuracy, Precision, Recall, and F1 Score are evaluated for the same as mentioned in Section 5. For conducting the experiments, we split the entire dataset into 80:20 ratios by keeping 80% of the dataset for training and 20% of the dataset for testing the performance. We evaluate the performance of several baseline machine learning algorithms on both the proposed datasets and compare it with our proposed framework. Moreover, we also compare the performance of our proposed framework with several recent humour detection frameworks.

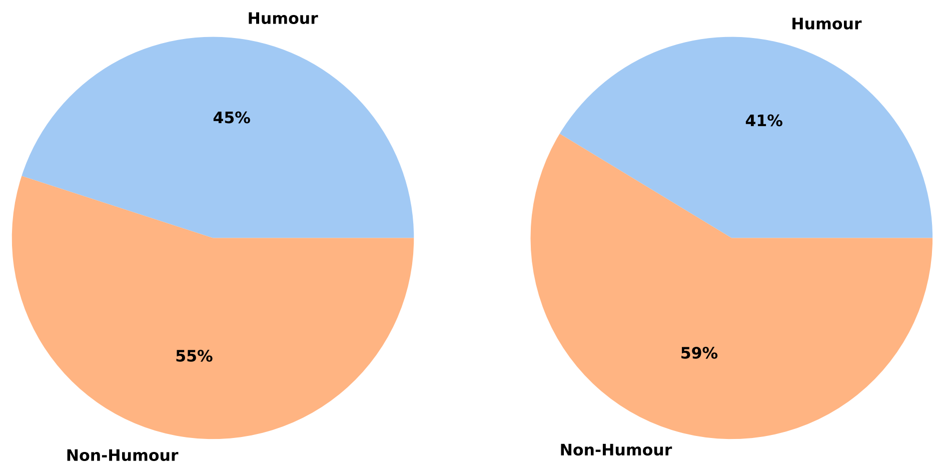
We also evaluate the AUC values of our algorithm to understand the bias of our model. Time and space complexity analysis are also presented for our algorithm. Figure 4, shows the percentage of humour and non-humour dialogues or utterances in both the proposed datasets. From Figure 4, we see that 45% of the dialogues of the **Kota-Factory** dataset are humour while 41% of the dialogues of the *Panchayat* dataset are humour. This shows that both datasets are well-balanced in terms of the number of samples in both classes. Table 4 lists the various hyperparameters for our framework. We employ two LSTM layers and two Dense layers. We use three CNN layers having filter sizes of 3, 4, and 5. The dropouts and recurrent dropouts values used for the two LSTM layers are 0.25 and 0.20 respectively. The optimizer used to compile the model is Adam. The activation function used in the first dense layer is ReLu while that used for the second layer is Sigmoid. The loss function used to monitor the loss is Binary-crossentropy. We used 50 epochs as around 50 epochs the training loss stabilised. The batch size used is 64. We also used callbacks to train our model. The callback method used is ReduceLROnPlateau, which reduces the learning rate when the monitored metric, which in this case is validation accuracy, has stopped improving. We also use 5-fold cross-validation and L1 regularization to improve the model performance. The obtained experimental results are as follows.

Table 4. Hyperparameters for SENet

Hyperparameter	**Description or Value**
Number of LSTM layers	2
Number of CNN layers	3
Filter Sizes	3, 4, 5
Number of Dense layers	2
Dropout rate	0.25
Recurrent dropout rate	0.20
Optimizer	Adam
Activation function	ReLu and Sigmoid
Loss function	Binary-crossentropy
Regularization	L1 Regularization
Resampling method	Cross-Validation
Type of Cross-Validation	5-fold
Number of epochs	50
Batch size	64
Callbacks	ReduceLROnPlateau

6.1 Comparison with Baseline algorithms

In this section, we compare the performance of our proposed HumourHindiNet framework with several baseline machine learning and deep learning-based approaches for humour detection on both of our proposed datasets. We also use a good number of supervised and unsupervised algorithms, along with several language-based models. The various machine learning algorithms used are K-Means Clustering, K Nearest Neighbors, Logistic Regression, Gaussian Naive Bayes, Decision Trees, and Random Forests. We use Artificial Neural Networks (ANNs), CNNs, and Recurrent Neural Networks (RNNs) as the deep learning methods. The GloVe embedding method and the Bidirectional Encoder Representations from Transformers (BERT) as the language-based models. The obtained results are presented in Table 5 for the $Kota - Factory$ dataset and Table 6 for the *Panchayat* dataset. For the $Kota - Factory$ dataset, we see that the unsupervised clustering algorithms are the weakest performers. However, the performance improves as we go to the supervised algorithms as can be seen by the good results achieved by Random Forest, Decision Trees, etc. Deep learning-based models like ANN and RNN are able to appropriately model the non-linear multi-dimensional relationships of the dialogues to generate even better results. Amongst all the baseline models used for the $Kota - Factory$



(a) $Kota - Factory$ (b) $Panchayat$

Fig. 4. Representation of the percentage of humour and non-humour samples in both the proposed datasets

Table 5. Performance comparison of our proposed HumourHindiNet framework with several baseline machine learning and deep learning algorithms on $Kota - Factory$ dataset.

Methods	Accuracy	Precision	Recall	F1 Score
K-Means Clustering	67.61	64.12	76.68	69.84
KNN	75.91	78.32	72.38	75.23
Logistic Regression	78.92	75.92	85.38	80.37
Gaussian Naive Bayes	77.28	75.8	79.03	77.38
Decision Trees	80.47	78.58	83.49	80.96
Random Forest	85.22	83.3	86.77	85.0
ANN	88.59	87.34	**90.07**	88.69
CNN	73.08	73.9	71.06	72.45
RNN	83.94	87.4	79.86	83.46
GloVe	89.69	95.37	83.27	88.91
BERT	89.87	92.86	86.19	89.4
HumourHindiNet	**91.79**	**96.29**	87.41	**91.64**

dataset, the language-based models are the best performers. This can be attributed to the optimal language dependency modeling done by the language-based models. The best performer for the *Kota – Factory* dataset is our proposed HumourHindiNet model. For the *Panchayat* dataset, all the unsupervised and supervised machine learning models along with the deep learning models give sub-optimal results which are in close agreement with each other. The language-based models are the only models amongst the baseline algorithms to cross the 80% accuracy mark. However, they're still second to our proposed HumourHindiNet. The optimal performance of our proposed HumourHindiNet model can be attributed to the ability of our model to appropriately model the latent language dependencies along with using proper deep learning architectures for feature extraction.

Table 6. Performance comparison of the proposed HumourHindiNet framework with several baseline machine learning and deep learning algorithms on *Panchayat* dataset.

Methods	Accuracy	Precision	Recall	F1 Score
K-Means Clustering	76.37	73.45	81.23	77.14
KNN	75.18	80.0	69.26	74.24
Logistic Regression	75.91	77.6	71.85	74.61
Gaussian Naive Bayes	75.55	81.48	68.99	74.72
Decision Trees	78.83	81.92	75.53	78.6
Random Forest	78.01	82.4	70.72	76.11
ANN	78.56	84.03	70.33	76.57
CNN	74.09	70.95	79.44	74.95
RNN	78.92	79.52	78.22	78.86
GloVe	84.76	83.0	86.28	84.61
BERT	85.77	84.11	**87.55**	85.79
HumourHindiNet	**87.32**	**90.15**	84.14	**87.04**

Table 7. Performance comparison of our proposed HumourHindiNet framework with several recently proposed humour detection algorithms on $Kota - Factory$ dataset.

Methods	Accuracy	Precision	Recall	F1 Score
Bertero and Fung [2]	72.84	68.36	85.5	75.97
Chen and Lee [4]	78.91	76.37	82.3	79.22
Chen and Soo [5]	84.22	84.34	84.34	84.34
Yang et al. [30]	88.62	88.43	**89.22**	88.82
Oliveira and Rodrigo [6]	84.98	87.19	82.79	84.93
Kumar et al. [17]	85.76	84.10	87.54	85.79
HumourHindiNet	**91.79**	**96.29**	87.41	**91.64**

6.2 Comparison with contemporary models

In this section, we compare the performance of our proposed framework with several recently proposed humour detection models. Most of the recently proposed models can be broadly categorized as follow: Language-model based, Feature engineering-based, and Deep learning based. To obtain a good sense of comparative study, we have selected a balance of algorithms from all these categories as follows. We use the deep-learning architectures of the various contemporary as it is. However, we convert the Hindi dialogues from our datasets to be compatible with the input of these models. We use our skip-gram-based model to tokenize the Hindi dialogues. For feature-engineering-based models, we generate the humour anchors for the Hindi language similar to the original work.

The obtained results are presented in Table 7 for the $Kota - Factory$ dataset and Table 8 for the *Panchayat* dataset. For both the datasets, we observe that the language-model based Chen and Lee [4] model is the second worst performer after the deep learning-based Bertero and Fung [2]. On the other hand, the deep learning-based models, namely, Chen and Soo [5], and Rodrigo and Rodrigo [6], are the best performers in terms of the recently proposed algorithms due to their ability to uncover and train on the various intricacies of the language and working to explore a deep understanding of the problem-specific details. The feature-engineering based Yang et al. [30], gives optimal and stable performance for both the datasets, due to its appropriate task-specific feature engineering, making it suitable for humour detection.

Table 8. Performance comparison of our proposed HumourHindiNet framework with several recently proposed humour detection algorithms on *Panchayat* dataset.

Methods	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Bertero and Fung [2]	72.38	66.43	87.89	75.67
Chen and Lee [4]	84.98	82.96	88.66	85.71
Chen and Soo [5]	86.8	87.87	86.59	87.22
Yang et al. [30]	84.37	84.01	85.76	84.88
Oliveira and Rodrigo [6]	85.73	84.68	**87.72**	86.18
Kumar et al. [17]	84.76	83.01	86.27	84.68
HumourHindiNet	**87.32**	**90.15**	84.14	**87.04**

The proposed HumourHindiNet's superior performance can be primarily attributed to its ability to capture contextual dependencies within the dialogues, facilitated by the LSTM section of our model. Unlike traditional algorithms that may struggle to grasp the sequential nature of humour, the proposed HumourHindiNet' effectively leverages stacked LSTM layers to capture the intricate temporal relationships inherent in comedic content. This enables the model to discern not only individual humorous elements but also the overarching comedic structure within the dialogues, leading to more accurate predictions. Additionally, the utilization of appropriate preprocessing techniques ensures uniformity and well-structured dialogue across the datasets, reducing noise and enhancing the model's ability to extract relevant features. Furthermore, the CNN-based n-gram feature extraction mechanism employed by the proposed HumourHindiNet enables a deeper understanding of latent language dependencies, allowing the model to uncover subtle linguistic nuances that contribute to the comedic effect. By combining these advanced techniques, the proposed HumourHindiNet achieves superior performance across all performance metrics, establishing it as the top performer in humour detection for Hindi web series.

Table 9. Accuracy Comparisons with Contemporary models on Non-Hindi (English) Datasets.

Methods	**16000 One-Liners**	**Pun of the Day**	**Short Jokes**
Bertero and Fung [2]	79.6	83.6	84.3
Chen and Lee [4]	84.8	86.1	87.6
Chen and Soo [5]	89.7	89.4	90.6
Yang et al. [30]	79.7	85.4	87.6
Oliveira and Rodrigo [6]	87.5	88.7	88.1
Kumar et al. [17]	86.6	86.6	83.2
HumourHindiNet	90.2	90.7	91.1

6.3 Comparison with Non-Hindi Datasets

In this section, we compare the performance of different contemporary models with our proposed model on non-Hindi specifically English datasets. We utilize several benchmark humour English datasets, namely, 16000 One-Liners [21], Pun of the Day [30], and Short Jokes Kaggle. We modify the proposed approach by replacing skip-gram based Hindi embedding with skip-gram-based English embedding using an English text corpus in Section 4.2. This helps the proposed model to generate word embedding for the English words in these datasets. The generated embedding is then fed to the "N-gram based feature extraction using CNN" module as mentioned in Section 4.3. This enables our model to adapt to non-Hindi Humour datasets. Table 9 lists the results of the proposed model along with some contemporary

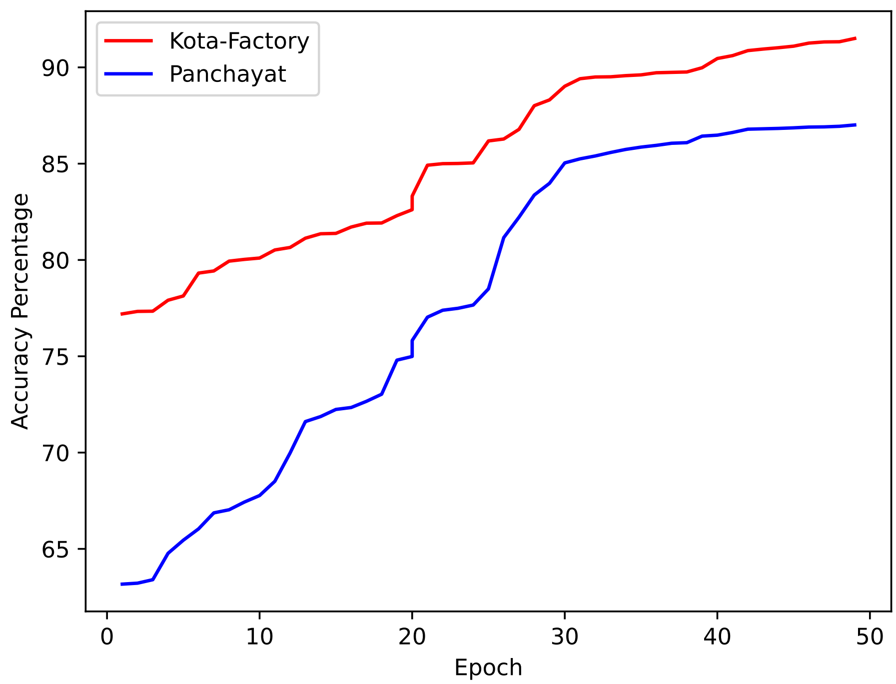


Fig. 5. The Accuracy values obtained by the proposed model for the two datasets across all the epochs

models on above mentioned English datasets. The obtained results highlight HumourHindiNet's superior accuracy for these datasets, underscoring its advanced approach that combines word embeddings, CNNs, and stacked LSTM layers. This blend allows for a nuanced understanding of humor's complex linguistic and sequential nature, evident in its leading accuracy figures: 90.2% on 16000 One-Liners, 90.7% on Pun of the Day, and 91.1% on Short Jokes. Such performance not only demonstrates HumourHindiNet's adeptness at humor detection, particularly in the nuanced and culturally rich context of Hindi web series but also for English datasets.

6.4 Accuracy and Loss vs Epoch Analysis

In this section, we perform the Accuracy vs Epoch and Loss vs Epoch analysis. Figure 5 shows the Accuracy results obtained by our proposed framework for different epochs. For the $Kota - Factory$ dataset, the accuracy significantly improved from 77.2% to a remarkable peak of 91.79% throughout 50 epochs. This ascent demonstrates the model's ability to learn and capture humour patterns within the Hindi web series content. While the accuracy exhibits impressive growth early in training, it begins to plateau towards the latter epochs, suggesting that the model has captured a substantial portion of the humour-related features present in the dataset. In the case of the *Panchayat* dataset, a similar trend is observed. The accuracy starts at 63.17% and gradually climbs to a peak of 87.32% after 50 epochs. This growth indicates the model's effectiveness in discerning humour in this specific dataset. Like the $Kota - Factory$ dataset, the accuracy curve levels off in the later epochs, signaling a diminishing rate of improvement as the model has likely captured the predominant humour-related patterns within the data.

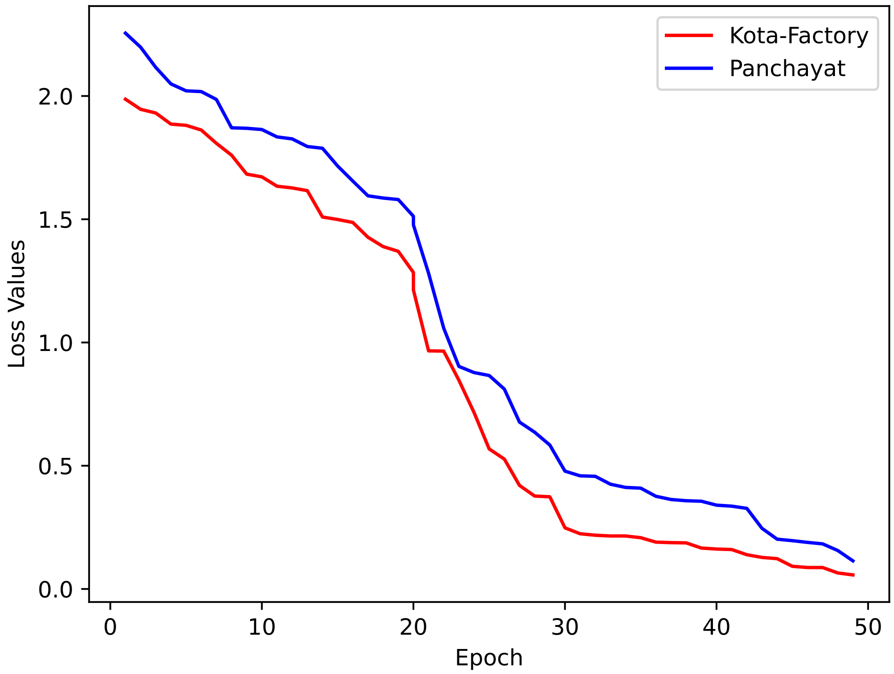


Fig. 6. The Loss values obtained by the proposed model for the two datasets across all the epochs.

Figure 6 shows the accuracy results obtained by our proposed framework for different epochs. In terms of loss, the $Kota-Factory$ dataset exhibited a substantial decrease during training. The loss started at 1.987 and dropped consistently as the model learned. This decline reflects the model's ability to reduce the errors in its predictions. Similar to the accuracy trend, the loss curve begins to plateau towards the later epochs, indicating that the model's learning rate has slowed, and is approaching convergence. The $Panchayat$ dataset showcases a comparable loss reduction over the 50 epochs. Starting at 2.255, the loss steadily diminishes as the model refines its understanding of humour within this dataset. However, similar to the $Kota-Factory$ dataset, the loss reduction rate decelerates in the later epochs, signifying that the model has captured the primary humour-related patterns in the data, and further significant loss reduction becomes challenging. In summary, both datasets exhibit notable improvements in accuracy and reductions in loss over the training period, indicating that the model is effective at humour detection. The plateauing of these trends in the later epochs suggests that the model's learning rate slows as it approaches convergence, having already captured a substantial portion of the underlying humour patterns within the respective datasets.

6.5 AUC Analysis

In this section, we evaluate the AUC values for our algorithm. Since the number of data points in our datasets is extremely less as compared to other pre-trained models, hence, the high performance of our model could be due to the inherent bias of our model. To evaluate the bias of our model, we calculate the AUC values for our model. An AUC value near 0 shows that the model has the worst distinguishing capability and it is classifying positive samples as negative and vice-versa. An AUC value of 0.5 means that the model has no distinguishing capability while a value close to 1 represents that the model can appropriately distinguish amongst the classes. Figure 7 shows the AUC values obtained by our model for the two datasets. From the figure, we can see that the obtained AUC values are well over 0.5, hence it shows that our model has great distinguishing capability and doesn't suffer from bias.

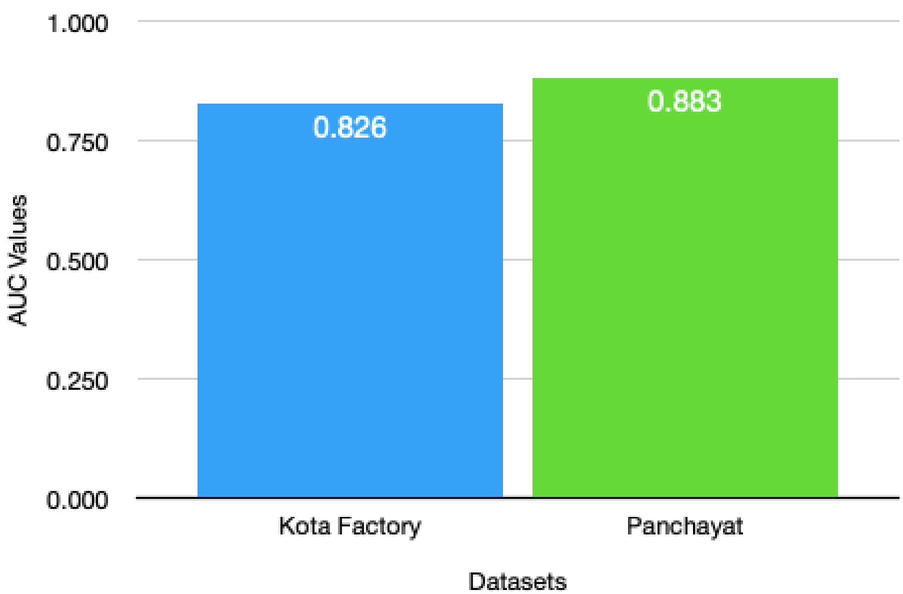


Fig. 7. The AUC values obtained by our model for the two datasets.

6.6 Time Complexity

In this section, we study the time complexity of the various components of our proposed model. Our model involves various steps, namely, (i) Skip-gram model based word embedding generation, (ii) feature extraction using CNN layers, (iii) stacked LSTM layers, and (iv) a deep neural network classifier. The time complexity of the various steps is given below.

(i) Skip-gram model-based word embedding generation: Let N be the size of the entire text corpus, D be the size of the desired embeddings, and the V be the vocabulary of the unique words in the text corpus. We perform a binary search on the unique words vocabulary and we generate a D dimensional embedding for all the N words in the corpus, hence the time complexity for the skip-gram model would be $O(N \times D \times log(V))$.

(ii) Feature extraction using CNN layers: Let D be the size of the word embedding extracted in the previous step. Let D be expressed as a 2D matrix of size $n \times d$ for convolutions. Let k be the kernel size and f be the filter size. Since, we evaluate convolutions for k kernels across the entire $n \times d$, hence the time complexity becomes $O(k \times n \times d)$. Since this is repeated for the entire filter size f, hence the final time complexity comes out to be $O(f \times k \times n \times d)$.

(iii) As per the works of Raja [25], let I be the number of inputs, K be the number of outputs and H be the number of cells in the hidden layers. The time complexity for the LSTM model can then be defined as $O(4IH + 4H2 + 3H + HK)$.

(iv) Deep neural network classifier: For a deep neural network classifier, the output of every node in the input layer is multiplied by the input of every node in the first hidden layer, and the output of every node of the last hidden layer is multiplied with the input of every node of the output layer. Moreover, a similar multiplication happens across the nodes of all the layers of the hidden layers. Hence, let P be the product of the number of layer-wise nodes in the input layer, all the hidden layers, and the output layer. Let E be the number of epochs and S be the total number of training samples. Hence, the time complexity of a deep neural network classifier is $O(E \times S \times P)$.

(v) Overall time-complexity: Combining the above four phases of our model, we arrive at an overall time complexity of $O(N \times D \times log(V) + f \times k \times n \times d + 4IH + 4H2 + 3H + HK + E \times S \times P)$

6.7 Space Complexity

In this section, we study the space complexity of the various components of our proposed model. Our model involves various steps, namely, (i) Skip-gram model based word embedding generation, (ii) feature extraction using CNN layers, (iii) stacked LSTM layers, and (iv) a deep neural network classifier. The time complexity of the various steps is given below.

(i) Skip-gram model-based word embedding generation: Let N be the size of the entire text corpus, D be the size of the desired embeddings. Since we calculate and store the embedding for every word in the text corpus, hence the space complexity for the skip-gram model would be $O(N \times D)$.

(ii) Feature extraction using CNN layers: For CNN layers, the output dimensions decrease after every layer. Hence, the largest space that we need is the output of the first layer. Let D be the size of the word embedding extracted in the previous step and let f be the filter size. The dimension of the output of the CNN layer is $((D - f)/2) + 1$. Hence the space complexity becomes $O(((D - f)/2) + 1)$. Since $f << D$, this can also be written as $O(D)$.

(iii) As per the works of Raja [25], let I be the number of inputs, and H be the number of cells in the hidden layers. The space complexity for the LSTM model can then be defined as $O(4 \times (I + H) \times H + K)$.

(iv) Deep neural network classifier: For a deep neural network classifier, the weights are stored for every neuron in every layer of the network. Let P be the product of the number of layer-wise nodes in the input layer, all the hidden layers, and the output layer. Hence, the space complexity of a deep neural network classifier is $O(P)$.

(v) Overall space-complexity: Combining the above four phases of our model, we arrive at an overall time complexity of $O(N \times D + D + 4 \times (I + H) \times H + K + P)$. Since, $D << N \times D$ and $K << 4 \times (I + H) \times H$, hence the overall space complexity can be written as $O(N \times D + 4 \times (I + H) \times H + P)$

6.8 Advantages and Limitations

The proposed approach offers several notable advantages. Firstly, it leverages word embedding techniques to enhance the contextual understanding of Hindi dialogues. This is particularly crucial for humour detection, as humour often relies on subtle contextual cues. By representing words in a dense vector space, semantic relationships are preserved, significantly improving the model's grasp of contextual humour. Secondly, the integration of convolutional neural networks, such as proposed HumourHindiNet, provides an automatic feature learning mechanism. This reduces the need for labor-intensive handcrafted feature engineering and potentially results in more robust and adaptable models. Moreover, the inclusion of stacked LSTM layers allows the model to capture sequential dependencies within dialogues, an essential aspect of humour recognition. Lastly, the research addresses a timely need in the digital age, where web series and digital content are gaining immense popularity. It contributes to the field of entertainment analytics, offering practical applications and insights into humour detection.

Despite its merits, the proposed approach does come with certain challenges. Firstly, its success hinges on the availability of suitable datasets for training and evaluation. If there's a scarcity of diverse and high-quality datasets specific to humour in Hindi web series, the model's performance could be constrained. Secondly, the computational complexity associated with deep learning models must be considered. These models demand substantial computational resources for training and inference, potentially limiting their accessibility to researchers with constrained computing power. Additionally, there's a risk of overfitting, especially if the dataset is limited or not well-regularized, which could affect the model's ability to generalize to new data. Lastly, deep learning models are often criticized for their lack of interpretability, making it challenging to understand the rationale behind specific humour detection decisions. This limitation might hinder their real-world applicability in certain contexts.

In summary, while the proposed approach holds promise for advancing humour detection in Hindi web series, it's essential to carefully address challenges related to data availability, computational resources, model complexity, and interpretability to maximize its potential impact in the field of natural language processing and entertainment analytics.

7 CONCLUSION

Humour detection is an important task in the field of NLP. However, relatively it is a difficult task as humour is often contorted in between conversations or expressed through feeble expressions or gestures that it isn't easily detectable. Moreover, it also requires good language-specific knowledge to detect humour appropriately. There is also a huge shortage of humour detection datasets in the Hindi language making it increasingly difficult to obtain good performance for detecting humour in the Hindi language. Following along these lines, in this paper, we proposed two Hindi datasets that are extracted from two recent popular web series, namely, $Kota - Factory$ and $Panchayat$. We also proposed Hindi word embedding and convolutional network-based framework, named HumourHindiNet, for humour detection in Hindi web series. We started by preprocessing the dataset to remove irrelevant data and obtain uniformity across datasets. The processed dataset is then used by our proposed Hindi language-based word embedding generation model to create word embeddings for all the dialogues in the introduced datasets. The generated word embeddings are then passed through parallelly stacked CNN layers to extract n-gram features based on varying filter sizes which range from 3 to 5. The extracted features are then passed through stacked LSTM for further processing and exploring the latent language and contextual dependencies. The final extracted features are then passed to a deep neural network architecture to make the final classifications as humour or non-humour. We conducted intensive experiments on both the proposed datasets and evaluate several standard performance metrics. We compared the performance of our proposed HumourHindiNet model with several baseline unsupervised and supervised machine learning algorithms, deep learning algorithms, and language-based models. We also compared our performance with several recently proposed language-based algorithms, feature-engineering based algorithms, and deep learning-based algorithms for humour detection. The obtained results demonstrated that our proposed HumourHindiNet model gives the best performance and proves its efficiency and efficacy for humour detection. This work can further be extended by using multi-language or multi-modal datasets. We can also work on creating humour detection datasets for other low-resource languages.

REFERENCES

[1] Dario Bertero and Pascale Fung. 2016. Deep learning of audio and language features for humor prediction. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. 496–501.

[2] Dario Bertero and Pascale Fung. 2016. A long short-term memory framework for predicting humor in dialogues. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 130–135.

[3] Dushyant Singh Chauhan, Gopendra Vikram Singh, Navonil Majumder, Amir Zadeh, Asif Ekbal, Pushpak Bhattacharyya, Louis-philippe Morency, and Soujanya Poria. 2021. M2H2: A Multimodal Multiparty Hindi Dataset For Humor Recognition in Conversations. In *Proceedings of the 2021 International Conference on Multimodal Interaction*. 773–777.

[4] Lei Chen and Chungmin Lee. 2017. Predicting audience's laughter during presentations using convolutional neural network. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. 86–90.

[5] Peng-Yu Chen and Von-Wun Soo. 2018. Humor recognition using deep learning. In *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 2 (short papers)*. 113–117.

[6] Luke De Oliveira and Alfredo L Rodrigo. 2015. Humor detection in yelp reviews. *Retrieved on December* 15 (2015), 2019.

[7] J Devlin, MW Chang, K Lee, and KB Toutanova. 2019. Pre-training of deep bidirectional transformers for language understanding In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). *Minneapolis, MN: Association for Computational Linguistics* (2019), 4171–86.

[8] Xiaochao Fan, Hongfei Lin, Liang Yang, Yufeng Diao, Chen Shen, Yonghe Chu, and Yanbo Zou. 2020. Humor detection via an internal and external neural network. *Neurocomputing* 394 (2020), 105–111.

[9] Sayani Ghosal and Amita Jain. 2023. HateCircle and Unsupervised Hate Speech Detection Incorporating Emotion and Contextual Semantics. *ACM Transactions on Asian and Low-Resource Language Information Processing* 22, 4 (2023), 1–28.

[10] Md Kamrul Hasan, Wasifur Rahman, Amir Zadeh, Jianyuan Zhong, Md Iftekhar Tanveer, Louis-Philippe Morency, et al. 2019. UR-FUNNY: A multimodal language dataset for understanding humor. *arXiv preprint arXiv:1904.06618* (2019).

[11] Devamanyu Hazarika, Soujanya Poria, Amir Zadeh, Erik Cambria, Louis-Philippe Morency, and Roger Zimmermann. 2018. Conversational memory network for emotion recognition in dyadic dialogue videos. In *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting*, Vol. 2018. NIH Public Access, 2122.

[12] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (2016).

[13] Sulaiman Khan, Shah Nazir, and Habib Ullah Khan. 2023. Analysis of Cursive Text Recognition Systems: A Systematic Literature Review. *ACM Transactions on Asian and Low-Resource Language Information Processing* (2023).

[14] Chloe Kiddon and Yuriy Brun. 2011. That's what she said: double entendre identification. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. 89–94.

[15] Dhanashree S Kulkarni and Sunil S Rodd. 2021. Sentiment Analysis in Hindi—A Survey on the State-of-the-art Techniques. *Transactions on Asian and Low-Resource Language Information Processing* 21, 1 (2021), 1–46.

[16] Sanjay Kumar. [n. d.]. Negative Stances Detection from Multilingual Data Streams in Low-resource Languages on Social Media using BERT and CNN based Transfer Learning Model. *ACM Transactions on Asian and Low-Resource Language Information Processing* ([n. d.]).

[17] Vijay Kumar, Ranjeet Walia, and Shivam Sharma. 2022. DeepHumor: a novel deep learning framework for humor detection. *Multimedia Tools and Applications* 81, 12 (2022), 16797–16812.

[18] Paul Pu Liang, Ziyin Liu, Amir Zadeh, and Louis-Philippe Morency. 2018. Multimodal language analysis with recurrent multistage fusion. *arXiv preprint arXiv:1808.03920* (2018).

[19] Bing Liu. 2020. Text sentiment analysis based on CBOW model and deep learning in big data environment. *Journal of ambient intelligence and humanized computing* 11, 2 (2020), 451–458.

[20] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. 2014. Exploiting geographical neighborhood characteristics for location recommendation. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. 739–748.

[21] Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: Investigations in automatic humor recognition. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. 531–538.

[22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013).

[23] Reynier Ortega-Bueno, Carlos E Muniz-Cuza, José E Medina Pagola, and Paolo Rosso. 2018. UO UPV: Deep linguistic humor detection in Spanish social media. In *Proceedings of the third workshop on evaluation of human language technologies for Iberian languages (IberEval 2018) co-located with 34th conference of the Spanish society for natural language processing (SEPLN 2018)*. 204–213.

[24] Hai Pham, Paul Pu Liang, Thomas Manzini, Louis-Philippe Morency, and Barnabás Póczos. 2019. Found in translation: Learning robust joint representations by cyclic translations between modalities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6892–6899.

[25] Sudeep Raja. 2021. FNNs, RNNs, LSTM and BLSTM.

[26] Alison Ross. 2005. *The language of humour*. Routledge.

[27] Ramsha Saeed, Hammad Afzal, Sadaf Abdul Rauf, and Naima Iltaf. 2023. Detection of Offensive Language and ITS Severity for Low Resource Language. *ACM Transactions on Asian and Low-Resource Language Information Processing* 22, 6 (2023), 1–27.

[28] Yansen Wang, Ying Shen, Zhun Liu, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. 2019. Words can shift: Dynamically adjusting word representations using nonverbal behaviors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7216–7223.

[29] Haojie Xu, Weifeng Liu, Jiangwei Liu, Mingzheng Li, Yu Feng, Yasi Peng, Yunwei Shi, Xiao Sun, and Meng Wang. 2022. Hybrid Multimodal Fusion for Humor Detection. In *Proceedings of the 3rd International on Multimodal Sentiment Analysis Workshop and Challenge*. 15–21.

[30] Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. 2015. Humor recognition and humor anchor extraction. In *Proceedings of the 2015 conference on empirical methods in natural language processing*. 2367–2376.

[31] Amir Zadeh, Paul Pu Liang, Navonil Mazumder, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2018. Memory fusion network for multi-view sequential learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[32] Yftah Ziser, Elad Kravi, and David Carmel. 2020. Humor detection in product question answering systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 519–528.

Received 01 January 2024; revised ; accepted