Expert Systems With Applications

Hierarchical Attention-enhanced Contextual CapsuleNet for Multilingual Hope Speech Detection

--Manuscript Draft--

Manuscript Number:	ESWA-D-24-04872
Article Type:	Full length article
Keywords:	Hope speech detection; Code-mixed text; Multilingual low-resource languages; Capsule networks; Deep learning; Indic languages
Abstract:	Social media was initially intended for creative purposes, but a notable dissemination of offensive material adversely affects users of these platforms. It is imperative to spotlight and endorse positive and uplifting instances, often overshadowed by the massive influx of user-generated content. To address the challenge of detecting hopeful messages in languages such as Tamil, Malayalam, and Kannada that have limited resources, we propose HopeCap, a novel multilingual hope speech detection framework. HopeCap employs a hierarchical attention-enhanced novel capsule network. The proposed CapsuleNet leverages the integrated representation of the prediction vector from the child capsule and the final vector obtained through dynamic routing. This allows CapsuleNet to capture spatial information more effectively. The hierarchical attention module captures the word-level and sentence-level attention features that are integrated with capsule features and auxiliary features. The proposed method computes three classification probabilities that are computed for translated, transliterated Indic, and transliterated Roman versions of the comment. The mean of three probabilities provides enhanced classification performance. Through rigorous analysis of HopeCap on three low-resource Indic languages, the study sheds light on the effectiveness of the proposed approach. HopeCap outperforms the existing state-of-the-art methods with an average increase of 6.13%, 6.58%, and 4.26% in terms of weighted-F1 for Tamil, Malayalam, and Kannada languages, respectively.

To, Date: 2024-04-05

Professor Dr. Binshan Lin, Ph.D.,
Louisiana State University in Shreveport,
Shreveport, Louisiana, United States of America.

Dear Editor-in-Chief,

We are pleased to submit our manuscript entitled “Hierarchical Attention-enhanced Contextual CapsuleNet for Multilingual Hope Speech Detection” for publication in Expert Systems with Applications. This work is focused on identifying and categorizing multilingual content that conveys hopeful, optimistic, encouraging, or positive sentiments in low-resource Indic languages. The goal is to automatically recognize instances where individuals express hope or inspire positivity in their communication. Hopeful and optimistic content detection in low-resource languages is emerging as a critical area of research and development due to its societal impact and the growing demand for technological solutions to address online harms influenced by hateful content.

The existing works focus on capturing the context through sequential information while neglecting the spatial and hierarchical relationship within the text. Our system employs a novel capsule network with capsule-level attention to extract spatial relations within the text. The CapsuleNet proposed in this study utilizes the combined representation of the prediction vector from the child capsule and the final vector obtained via dynamic routing. A hierarchical attention network is employed to focus on hierarchical information within text features. Additionally, to improve the generalization of predictions, we use three versions for a single comment: translated, transliterated Indic, and transliterated Roman script versions. To the best of our knowledge, none of the existing works make use of the collective benefits of these techniques for hopeful content detection.

Identifying hopeful content can automatically benefit users by categorizing and recommending positive material. This contributes to the responsible and ethical use of AI techniques, making social media a safer and more positive environment. Leveraging Artificial Intelligence, hopeful content detection enhances the social media experience for users.

We have prior research experience in the domain of natural language processing and low-resource languages. One of our works on low-resource languages, entitled “User-aware Multilingual Abusive Content Detection in Social Media” has been published in Information Processing & Management, 2023. In the text-based question-answering domain, one of our works, entitled “KisanQRS: A Deep Learning-based Automated Query-Response System for Agricultural Decision-Making,” has been published in Computers and Electronics in Agriculture, 2023. Another work, entitled “Multilingual Personalized Hashtag Recommendation for Low Resource Indic Languages using Graph-based Deep Neural Network” has been published in Expert Systems with Applications, 2023.

We have neither submitted nor published this research work in any other journal. We have presented our original work in this research paper.

Thank you for your consideration.

Sincerely,
Corresponding author
Dr Nagendra Kumar
Assistant Professor
Indian Institute of Technology (IIT) Indore
Khandwa Road, Simrol, Madhya Pradesh 453552, India
Email: nagendra@iiti.ac.in

Hierarchical Attention-enhanced Contextual CapsuleNet for Multilingual Hope Speech Detection

Mohammad Zia Ur Rehman[a] (phd2101201005@iiti.ac.in)
Devraj Raghuvanshi[b] (gs2019042@sgsitsindore.in)
Harshit Pachar[a] (cse200001027@iiti.ac.in)
Chandravardhan Singh Raghaw[a] (phd2201101016@iiti.ac.in)
Nagendra Kumar[a] (nagendra@iiti.ac.in)

[a] Department of Computer Science and Engineering, Indian Institute of Technology Indore, Khandwa Road, Simrol, Indore, 453552, Madhya Pradesh, India
[b] Department of Information Technology, Shri Govindram Seksaria Institute of Technology and Science, Sir M. Visvesvaraya Marg, Indore, 452003, Madhya Pradesh, India

Corresponding Author:
Nagendra Kumar
Department of Computer Science and Engineering,
Indian Institute of Technology Indore, Indore 453552, India
Tel: +91-7316603225
Email: nagendra@iiti.ac.in

Hierarchical Attention-enhanced Contextual CapsuleNet for Multilingual Hope Speech Detection

Mohammad Zia Ur Rehman[a], Devraj Raghuvanshi[b], Harshit Pachar[a], Chandravardhan Singh Raghaw[a] and Nagendra Kumar[a,*]

[a] *Department of Computer Science and Engineering, Indian Institute of Technology Indore, India*
[b] *Department of Information Technology, Shri Govindram Seksaria Institute of Technology and Science, Indore, India*

ARTICLE INFO

Keywords:
Hope speech detection
Code-mixed text
Multilingual low-resource languages
Capsule networks
Deep learning
Indic languages

Abstract

Social media was initially intended for creative purposes, but a notable dissemination of offensive material adversely affects users of these platforms. It is imperative to spotlight and endorse positive and uplifting instances, often overshadowed by the massive influx of user-generated content. To address the challenge of detecting hopeful messages in languages such as Tamil, Malayalam, and Kannada that have limited resources, we propose `HopeCap`, a novel multilingual hope speech detection framework. `HopeCap` employs a hierarchical attention-enhanced novel capsule network. The proposed CapsuleNet leverages the integrated representation of the prediction vector from the child capsule and the final vector obtained through dynamic routing. This allows CapsuleNet to capture spatial information more effectively. The hierarchical attention module captures the word-level and sentence-level attention features that are integrated with capsule features and auxiliary features. The proposed method computes three classification probabilities that are computed for translated, transliterated Indic, and transliterated Roman versions of the comment. The mean of three probabilities provides enhanced classification performance. Through rigorous analysis of `HopeCap` on three low-resource Indic languages, the study sheds light on the effectiveness of the proposed approach. `HopeCap` outperforms the existing state-of-the-art methods with an average increase of 6.13%, 6.58%, and 4.26% in terms of weighted-F1 for Tamil, Malayalam, and Kannada languages, respectively.

1. Introduction

Language serves as a conduit for human expression, and social media platforms have exponentially increased the volume of online communication in different languages (Kazienko, Bielaniewicz, Gruza, Kanclerz, Karanowski, Miłkowski and Kocoń, 2023). There is an observable trend of increasing negative and hostile content on social media platforms. This surge poses severe threats to social cohesion, mental well-being, and even public safety (Min, Lin, Li, Zhao, Lu, Yang and Xu, 2023; Noorian, Ghenai, Moradisani, Zarrinkalam and Alavijeh, 2024). These issues can be tackled through a two-fold process. The first aspect is the identification and removal of offensive content (Liu, Xu, Zhao, Zeng, Hu, Zhang, Luo and Cao, 2023). Another aspect involves the detection of hopeful and inspiring content that can be recommended to users. Chakravarthi *et al.* (Chakravarthi, 2020; Chakravarthi and Muralidaran, 2021; Chakravarthi, Muralidaran, Priyadharshini, Cn, McCrae, García, Jiménez-Zafra, Valencia-García, Kumaresan, Ponnusamy et al., 2022) define Hopeful content as comments or posts that provide support, reassurance, suggestions, and inspiration or the comments that promote happiness, satisfaction, equality and fairness for minority groups. Existing works have extensively explored the problem of offensive content detection. However, the identification of hopeful content in social media has been relatively under-explored (Balouchzahi, Sidorov and Gelbukh, 2023). Detecting hope speech allows for the identification of supportive and uplifting content (Chakravarthi, 2022a), which can offer valuable resources to people dealing with anxiety, depression, and other mental health challenges. (Anshul, Pranav, Rehman and Kumar, 2023). A few examples of hope and non-hope speech

[*] Corresponding author

✉ phd2101201005@iiti.ac.in (M.Z.U. Rehman); gs2019042@sgsitsindore.in (D. Raghuvanshi); cse200001027@iiti.ac.in (H. Pachar); phd2201101016@iiti.ac.in (C.S. Raghaw); nagendra@iiti.ac.in (N. Kumar)

ORCID(s):

Table 1
Examples of Comments

S.No	Comment	Translation	Label
1.	Both side poga venom.namma vali thani valinu Iruka venditha	Lets go to both sides Our pain should not be a separate pain.	Hope-speech
2.	വളരെനല്ല അറിവ് സാർ. താങ്കളെ പോലുള്ള ആളുകൾ സാധാരണക്കാർക്ക് സമാധാനം നന്ദി	Not much knowledge sir. Thanks to people like you and peace to the common people.	Hope-speech
3.	ಕಾಲಮಾನಕ್ಕೆ ತಕ್ಕಂತೆ ಜೀವಿಗಳ ಜೀವಿತಾವಧಿ ಕೂಡ ಬದಲಾಗುತ್ತದೆ	The lifespan of living things also changes with time.	Not Hope-speech
4.	ಇದು ಹಾಡು ಅಂದ್ರೆ ಥಿಯೇಟರ್ ಲಿ ನೋಡಿದ ಮೇಲಂತೂ ಇನ್ನೂ ಕಿಕ್ ಕೊಡ್ತ	This song gave a kick even after seeing it in the theater.	Not Hope-speech

comments are shown in Table 1. The linguistic diversity of India (Roy, 2024) provides a compelling backdrop for this exploration. Hence, through this study, we aim to make a distinctive contribution by focusing on the detection of hopeful expressions in three low-resource Indic languages, namely Tamil, Malayalam, and Kannada.

1.1. The Linguistic Diversity in India

India, with its rich linguistic landscape, is home to numerous languages spoken by millions. Examining the linguistic demographics[1], Tamil is spoken by 5.70% people across India, Kannada by 3.61%, and finally, with 2.88%, Malayalam comes last in the top ten languages by number of speakers in India. To understand the magnitude of the number of speakers, even Malayalam, which is tenth in the list, has more than 34 million speakers. For a comprehensive understanding of the effects of hope speech across linguistic communities, it is imperative to analyze low-resource languages that are spoken by substantial demographic segments (Mahmud, Ptaszynski, Eronen and Masui, 2023). The selected languages, Tamil, Kannada, and Malayalam, represent significant user bases where hopeful content detection can potentially foster positive engagement.

1.2. Existing Approaches for Hope Speech Detection

Chakravarthi *et al.* (Chakravarthi and Muralidaran, 2021) explore hope speech detection across multiple languages, including Tamil, English, and Malayalam (Chakravarthi, 2020). The released dataset in their work is part of the EACL 2021 event, specifically within the framework of the Language Technology for Equality, Diversity, and Inclusion. Another study on hope speech detection in the Kannada language is introduced by Hande *et al.* (Hande, Priyadharshini, Sampath, Thamburaj, Chandran and Chakravarthi, 2021). Among the solutions given for the multilingual hope speech detection, some of the highlighted approaches are: the finetuning of transformer-based models (Mahajan, Al-Hossami and Shaikh, 2021), Machine Learning methods (Hossain, Sharif and Hoque, 2021), a combination of transformer-based methods with classical methods such as TF-IDF (Huang and Bai, 2021) and context-aware string embeddings with RNN (Junaida and Ajees, 2021). A few approaches, such as MUCS (Balouchzahi, Aparna and Shashirekha, 2021), use an ensemble of classifiers approach. Most of the top-performing solutions obtain similar results with negligible margins. Among transformer-based methods, XLM-R performs relatively better for code-mixed multilingual comments.

1.3. Research Gaps and Contributions

Existing approaches for hope speech detection mostly focus only on the original or single version of the comment, which may not capture the overall context of the comment; on the other hand, the transliterated and translated versions of comments can be explored, which may provide different contextual aspects for the comments. Further, Capsule Networks (CapsuleNet) (Sabour, Frosst and Hinton, 2017) were initially developed for computer vision tasks (Rajasegaran, Jayasundara, Jayasekara, Jayasekara, Seneviratne and

[1] https://censusindia.gov.in/census.website/data/census-tables/

Rodrigo, 2019; Mittal and Verma, 2023); there have been a few studies that explore their potential application in natural language processing (Kim, Jang, Park and Choi, 2020; Kamal, Anwar, Sejwal and Fazil, 2023), including understanding spatial relationships in text. However, existing studies on CapsuleNet consider the features obtained through the routings only. To obtain a high degree of spatial relationships within the text, initial predictions from child capsules have not been explored. Additionally, advanced attention mechanisms such as hierarchical attention have not been explored for code-mixed texts. Hierarchical attention allows models to focus on relevant information across multiple levels of abstraction and enhances the model's ability to capture important contextual information and dependencies within hierarchical structures, leading to improved performance in various natural language processing tasks. Understanding and effectively utilizing code-mix datasets for hope speech analysis becomes crucial for developing methods that can navigate the intricacies of languages spoken by millions in India.

In the light of the aforementioned research gaps, the key contributions of this work are as follows:

1. We present a novel multilingual framework to address hope speech detection for three low-resource Indic languages by leveraging a hierarchical attention-enhanced capsule network framework.

2. To extract enough context for a comment, we generate three versions of the comment, translated, transliterated in Indic script, and transliterated in Roman script. We also extract two sets of auxiliary features belonging to emotion and sentiment to gain additional insights for a comment.

3. We employ a novel capsule network enhanced with Capsule-level Attention (CLA). The proposed CapsuleNet integrates the prediction vector from the child capsule and the vector obtained through dynamic routing. The integration provides additional information on spatial relationships within words, whereas CLA further highlights the pertinent capsule features.

4. We employ a dedicated hierarchical attention module that focuses on individual words in a sentence that are further enhanced by a sentence-level feature. The framework leverages the integrated features from the capsule network, hierarchical attention module, and auxiliary feature module for final classification.

5. We perform extensive experiments on low-resource Indic language datasets. Both qualitative and quantitative evaluations demonstrate the efficacy of the proposed `HopeCap` approach.

2. Related Work

This section presents a review of existing works that provide valuable insights into the methodologies that researchers have employed for hope speech detection. At the end of this section, we present a brief description of limitations in existing works.

Chakravarthi *et al.*(Chakravarthi, 2020) focus on hope speech detection in low-resource Indian languages. The study employs a combination of pre-trained language models and transfer learning. By leveraging these techniques, the study achieves promising results in accurately detecting hope speech in languages with limited linguistic resources. Balouchzahi *et al.* (Balouchzahi et al., 2023) introduce a dataset for hope speech analysis, categorizing tweets into Hope and Not Hope, and further into three nuanced hope categories: Generalized Hope, Realistic Hope, and Unrealistic Hope. Baselines using traditional machine learning, deep learning, and transformers are evaluated. Transformer models consistently attained high F1-scores, surpassing other learning models, with BERT, Roberta, and XLNet models yielding the highest results for binary classification tasks. For multiclass tasks, BERT outperformed other transformers, highlighting the significant performance differences between transformers and other learning approaches. In the other work, Chakravarthi *et al.*(Chakravarthi, 2022b) introduce a BERT-based approach for multilingual hope speech detection. The study aims to develop a model that can effectively capture the nuances of hope speech across diverse linguistic structures. The BERT-based approach proved to be a versatile model for multilingual hope speech detection. Addressing the challenges of code-mixing in social media, Malik *et al.* (Malik, Nazarova, Jamjoom and Ignatov, 2023) propose a model for hope speech detection. By incorporating code-mixing awareness into the model architecture, the work provides a nuanced understanding of hope speech within the context of multilingual code-mixed data. The study by Saumya *et al.* (Saumya and Mishra,

2021) explore hope speech detection in low-resource languages through a multi-modal learning approach. Integrating textual and visual features, the model showcases the potential of combining multiple modalities for improved accuracy, especially in linguistic landscapes with limited resources.

Junaida *et al.* (Junaida and Ajees, 2021) delve into cross-lingual hope speech detection by employing multi-modal transfer learning. By transferring knowledge across languages, the model demonstrates the ability to detect hope speech effectively, even in linguistic contexts different from those in which it was trained. Hossain *et al.* (Hossain et al., 2021) introduce a transformer-based approach for cross-lingual hope speech detection. The model's architecture allows it to adapt to various languages, showcasing the versatility of transformer models in capturing linguistic nuances across different language structures. Focusing on South Asian languages, Arunima *et al.* (Arunima, Ramakrishnan, Balaji, Thenmozhi et al., 2021) present a multilingual analysis of hope speech detection. This work investigates linguistic commonalities and variations, contributing to a deeper understanding of hope speech across diverse linguistic landscapes in the South Asian region. Hande *et al.* (Hande et al., 2021) explore the application of transfer learning in hope speech detection for low-resource languages. By leveraging pre-trained models and adapting them to the specific linguistic characteristics of low-resource languages, the study offers a practical approach to address resource limitations. Focusing on code-mixed data, Dowlagar *et al.* (Dowlagar and Mamidi, 2021) present a BERT-based approach for hope speech detection. The model is designed to handle the challenges posed by code-mixing, providing insights into the complexities of detecting hope speech in linguistically diverse and dynamic environments. Pan *et al.* (Pan, Alcaraz-Mármol and Garcıa-Sánchez, 2023) delve into the detection of both hopeful and discouraging messages in social media. By adopting a broader perspective, the research contributes to a nuanced understanding of sentiment analysis, exploring the dynamic interplay between hopeful and discouraging expressions in online discourse. Balaji *et al.* (Balaji, Kannan, Balaji and Singh, 2023) focus on the role of contextual embeddings. This research investigates their effectiveness in hope speech detection. By capturing the contextual nuances of language, the model enhances the precision and accuracy of hope speech detection, providing a deeper understanding of the contextual factors influencing sentiment.

In examining these diverse existing approaches, it is evident that the researchers have used diverse approaches for hope speech detection, from classical Machine Learning methods to advanced transformer-based methods, to address the challenges posed by linguistic diversity and resource limitations. However, most of the works revolve around either using the pre-trained transformer-based methods or fine-tuning these methods. Existing works do not consider different scripts for the text, such as transliterated to Indic or Roman script

3. Problem Statement and Objectives

Let $\mathcal{D} = \{(s_i, Y_i)\}_{i=1}^{N}$ represent a dataset containing N samples of textual comments in l language where s_i represents the i-th comment and $y_i \in \{0, 1\}$ denotes its corresponding label. The comments may exist in either Indic language or code-mixed with English. Given a comment (s_i) and its corresponding label (y_i), our task is to classify s_i into either Hope Speech (HS) or Not Hope Speech (NHS). To achieve this objective, we aim to maximize the function ϕ, as defined in Equation 1.

$$\phi = \prod_{i=1}^{N} p(y_i | s_i; \theta) \tag{1}$$

Here, ϕ represents the product of the conditional probabilities of determining the correct label given the comment, and θ denotes the model parameters.

The following are the research objectives for the aforementioned task:

- Exploring the influence of novel CapsuleNet by integrating two-level prediction vectors from capsule layer.
- To explore the influence of capsule-level attention and hierarchical attention for the given task and provide a comparison with other state-of-the-art methods.

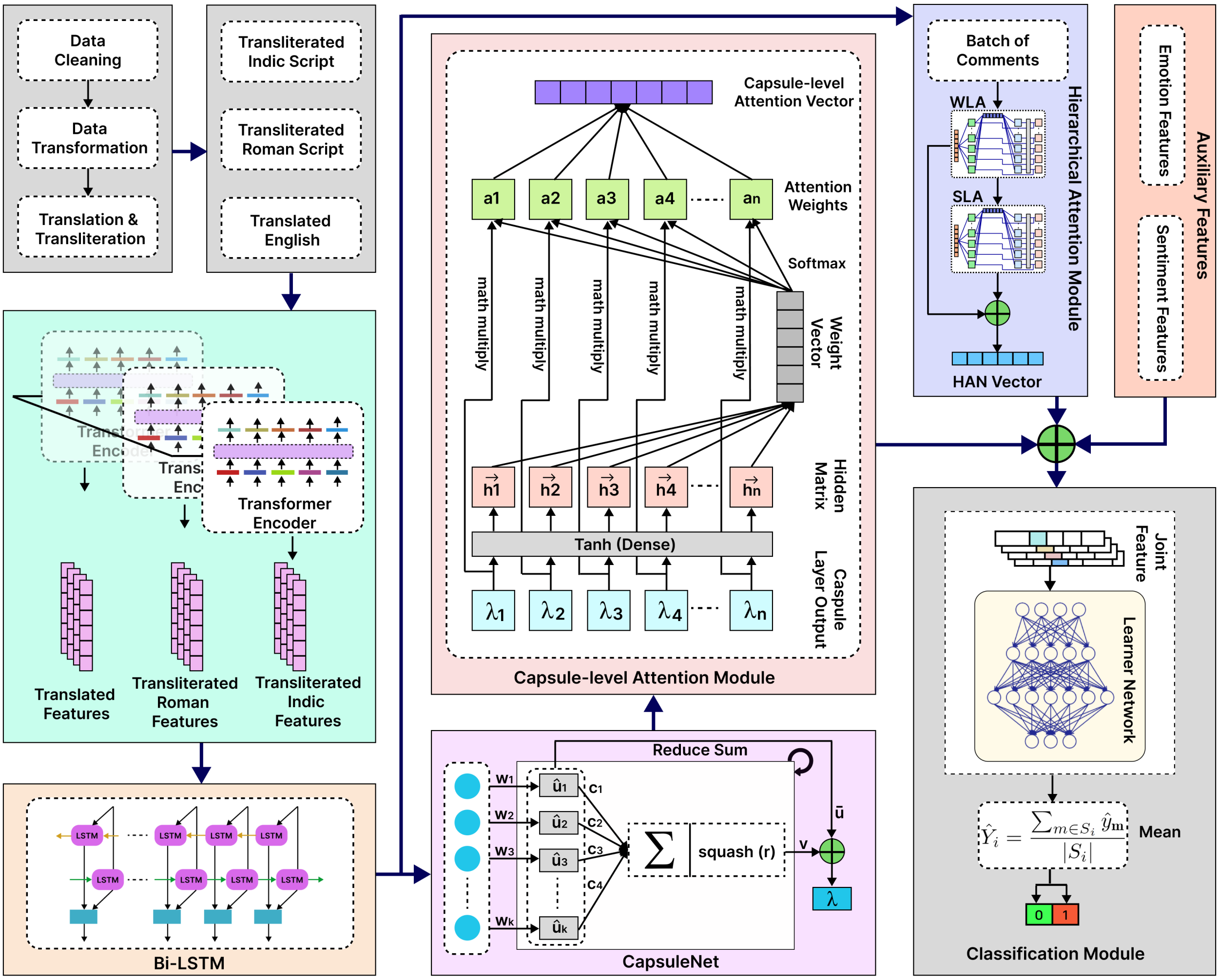


Figure 1: System Architecture of the Proposed Approach

- To devise a novel multilingual hopeful content detection framework for low-resource Indic language.
- To explore the influence of the collective benefit of transliterated and translated comments in code-mixed text classification.

4. Methodology

This section presents a detailed description of the proposed multilingual hope speech detection approach, `HopeCap`. The architecture illustrating our proposed method is depicted in Figure 1. Within our system framework, we initially translate and transliterate the given comment generating three versions. Subsequently, text features are extracted for each of the versions and passed in the pipeline for further processing. The subsequent sections elaborate on the proposed method in detail.

4.1. Feature Extraction

For each comment $s_i \in \mathcal{D}$, we create a set of three sentence variants $S_i = \{\alpha_i, \beta_i, \gamma_i\}$ where α_i signifies the English translation branch of s_i, β_i represents the transliterated version branch of s_i into Roman script, and γ_i denotes the transliterated branch of s_i into the original Indic language script (l). Transliteration is the process of converting text from one writing system into another. This conversion involves mapping the characters of one script to those of another based on their phonetic similarity. The translation process utilizes the Google API[2], while transliteration is performed using IndicXlit (Madhani, Parthan, Bedekar, Nc, Khapra, Kunchukuttan, Kumar and Khapra, 2023), a transformer-based multilingual model.

To extract contextual embeddings from the sentence variants, we employ two state-of-the-art transformer-based models: XLM-R and MuRIL. Each model is strategically chosen based on the nature of the text, XLM-R for the translated variant as it provides better results as shown in the results section (α_i) and MuRIL for the transliterated variants (β_i and γ_i) as it works better for transliterated text (Rehman, Mehta, Singh, Kaushik and Kumar, 2023). The encoder of XLM-R processes α_i, generating a last hidden state matrix $\boldsymbol{H}_{\alpha_i} \in \mathbb{R}^{\boldsymbol{T}\times \boldsymbol{d}}$. Similarly, the encoder of MuRIL processes β_i and γ_i, yielding last hidden states $\boldsymbol{H}_{\beta_i} \in \mathbb{R}^{\boldsymbol{T}\times \boldsymbol{d}}$ and $\boldsymbol{H}_{\gamma_i} \in \mathbb{R}^{\boldsymbol{T}\times \boldsymbol{d}}$, respectively. Each row in these matrices represents a $\boldsymbol{d}$-dimensional embedding vector corresponding to a word in a sequence of length $\boldsymbol{T}$.

These embedding matrices play a crucial role in capturing the semantic and syntactic information inherent in the translated and transliterated sentences. This comprehensive embedding extraction strategy lays the foundation for subsequent hope speech detection, acknowledging the intricacies of language variation and script representation in the diverse linguistic landscape. Subsequently, the last hidden state matrix $\boldsymbol{X}_i \in \{\boldsymbol{H}_{\alpha_i}, \boldsymbol{H}_{\beta_i}, \boldsymbol{H}_{\gamma_i}\}$, is passed to a Bidirectional Long Short-Term Memory (Bi-LSTM) layer for contextual learning.

4.2. Bi-directional Context Learning

Sequential data exhibits temporal dependencies that are crucial for understanding context and capturing long-term relationships. The Bi-LSTM layer is a pivotal component in our model architecture, specifically designed to address the challenges posed by sequential data. While traditional LSTM networks excel at capturing information over extended periods, the bidirectional variant enhances this capability by processing sequences in both the forward and backward directions. This dual-directional processing allows the model to glean contextual information from both past and future inputs, providing a more comprehensive understanding of the data.

$X_i = \{x_{it}\}_{t=1}^{T}$ represents a sequential input of length T. A forward LSTM processes the X_i from x_{i1} to x_{iT}, while a backward LSTM processes the sentence from x_{iT} to x_{i1}. For word embedding x_{it}, the forward LSTM obtains the forward hidden states $\overrightarrow{h_{it}} \in \mathbb{R}^{D}$ (shown in Equation 2) and the backward LSTM obtains the backward hidden states $\overleftarrow{h_{it}} \in \mathbb{R}^{D}$ (shown in Equation 3). The final hidden representation for the word is obtained by concatenating its forward and backward hidden states resulting in an embedding vector $u_{it} \in \mathbb{R}^{2D}$ as shown in Equation 4. It should be noted that throughout the paper, $\oplus$ represents the concatenation operation.

$$\overrightarrow{h_{it}} = \overrightarrow{\text{LSTM}}(x_{it}) \tag{2}$$

$$\overleftarrow{h_{it}} = \overleftarrow{\text{LSTM}}(x_{it}) \tag{3}$$

$$u_{it} = \overrightarrow{h_{it}} \oplus \overleftarrow{h_{it}} \tag{4}$$

This process is repeated for each $x_{it} \in X_i$, resulting in a sequence of concatenated hidden states $U_i = \{u_{it}\}_{t=1}^{T}$.

4.3. Fusion-based Hierarchical Attention

Hierarchical Attention Network (HAN) (Yang, Yang, Dyer, He, Smola and Hovy, 2016) includes attention at two levels: word-level and sentence-level. The purpose of word-level attention is to identify crucial words and consolidate the representation of these to construct a sentence vector. In our system, the purpose of sentence-level attention is to weigh the importance of each sentence within a batch of samples, facilitating

[2] https://cloud.google.com/translate/docs/reference/rest/

the model to identify and emphasize sentences that are more informative or relevant in the context of the entire batch. The input to HAN consists of the output from the Bi-LSTM layer, denoted as $\boldsymbol{L} = \{\boldsymbol{U}_i\}_{i=1}^{\boldsymbol{B}}$, where $\boldsymbol{U}_i \in \mathbb{R}^{T\times 2D}$ represents the sequence of last hidden states for the $\boldsymbol{i}$-th sample in the batch and the parameter $\boldsymbol{B}$ denotes the batch size.

Word-level Attention Given $U_i = \{u_{it}\}_{t=1}^{T}$, the embedding vector of each word (u_{it}) is subjected to a non-linear transformation to obtain a new hidden representation p_{it} as shown in Equation 5, where W_1 and b_1 are learned weight and bias parameters, respectively. Subsequently, we measure the importance of the word as the similarity of p_{it} with a word-level context vector (z_1) and get a normalized importance weight (a_{it}) through a softmax function as shown in Equation 6. This enables the model to dynamically allocate attention based on the contextual significance of each word. z_1 is randomly initialized and jointly learned during the training process. Finally, we compute q_i, the vector representation of the entire sentence, as a weighted sum of each $u_{it} \in U_i$ based on the attention weights as shown in Equation 7. We refer to $q_i \in \mathbb{R}^{2D}$ as the sentence vector.

$$p_{it} = \tanh(W_1 u_{it} + b_1) \tag{5}$$

$$a_{it} = \frac{\exp(p_{it}^{\top} z_1)}{\sum_k \exp(p_{ik}^{\top} z_1)} \tag{6}$$

$$q_i = \sum_{t=1}^{T} a_{it} u_{it} \tag{7}$$

Sentence-level Attention Once we obtain the sentence vectors for all the sentences in the batch, we again follow the same procedure to apply sentence-level attention. We subject q_i to a non-linear transformation to obtain p_i. Then, similar to the introduction of a word-level context vector (z_1) in word-level attention, we introduce a sentence-level context vector (z_2) and get a normalized importance weight a_i. Finally, we compute the weighted sum of q_i corresponding to each sentence in the batch to get a vector $bb \in \mathbb{R}^{2D}$ that summarizes the information of all the sentences in the batch. We refer to bb as the batch bias.

$$bb = \sum_{i=1}^{B} a_i q_i \tag{8}$$

In order to obtain a comprehensive feature representation (f_{han}) for the $\boldsymbol{i}$-th sample in the batch, that encapsulates a multi-level or hierarchical contextual understanding, we concatenate q_i and bb.

4.4. Capsule Network with Capsule-Level Attention

Capsule networks (Sabour et al., 2017), initially devised to overcome limitations in traditional convolutional neural networks (CNNs) for image recognition, have demonstrated efficacy in capturing hierarchical relationships among visual elements. This unique architecture, featuring capsules that encode both features and spatial relationships, has primarily been applied to image-related tasks. However, extending the use of capsule networks to the domain of natural language processing involves a paradigm shift. In the context of text understanding, each capsule can be seen as a specialized unit responsible for recognizing and encoding specific linguistic features. This approach allows capsule networks to capture the hierarchical relationships between words, phrases, and sentences, which is crucial for understanding the context and semantics of textual data. For instance, a capsule might focus on recognizing syntactic structures, while another could be dedicated to capturing semantic relationships between words. These capsules work collaboratively to form a dynamic routing mechanism, which is a key factor in enabling capsules to form a parse tree-like structure that represents the hierarchical relationships in the data, making capsule networks particularly effective for tasks that require an understanding of spatial hierarchies and relationships.

After processing the output of the Bi-LSTM layer through a dropout layer to prevent overfitting, the last hidden states (U) are passed into the capsule layer, where the following transformation occurs:

$$\hat{u}_{j|i} = W_{ij}u_i \tag{9}$$

Here, $\hat{u}_{j|i}$ signifies the predicted output of capsule j in the upper layer by capsule i in the lower layer, $u_i \in U$ is the output of capsule i in the lower layer, and W_{ij} represents the learned weight matrix connecting capsule i to capsule j. This encapsulates the idea of capsules actively predicting the instantiation parameters of other capsules. We refer to the capsules in the lower layer as child capsules and those in the upper layer as parent capsules. The output r_j of a parent capsule is computed as the weighted sum of predictions from child capsules, each multiplied by a coupling coefficient c_{ij}.

$$r_j = \sum_{i=1}^{n} c_{ij}\hat{u}_{j|i} \tag{10}$$

where n denotes the total number of capsules in the lower layer. The coupling coefficients (c_{ij}) are determined by applying a softmax activation to the prior log probability b_{ij} of the connection between child capsule i and parent capsule j:

$$c_{ij} = \text{softmax}(b_{ij}) = \frac{\exp(b_{ij})}{\sum_{k=1}^{C}\exp(b_{ik})} \tag{11}$$

Then, the squashing function is applied to ensure that the output vector length falls within the range (0, 1).

$$v_j = \text{squash}(r_j) = \frac{\|r_j\|^2}{1 + \|r_j\|^2}\frac{r_j}{\|r_j\|} \tag{12}$$

It is important to note that the coupling coefficients c_{Ij} are updated iteratively over R iterations based on the agreement between the predicted output vectors and the actual output vectors of the capsules in the upper layer. The agreement is measured as the scalar product between the predicted output vector $\hat{u}_{j|I}$ and the actual output vector v_j of capsule j. This iterative update process is a crucial aspect of the capsule network's dynamic routing mechanism, allowing capsules to refine their predictions and adapt to the input data.

For every parent capsule j, we aggregate the predictions $\hat{u}_{j|I}$ made by its child capsules through summation. Following this, we apply the squash function to these aggregated predictions, resulting in $\bar{u_j}$, which is then concatenated with v_j to yield λ_j.

$$\lambda_j = \bar{u_j} \oplus v_j \tag{13}$$

Algorithm-1 shows the implementation of the capsule network, detailing the iterative process of dynamic routing and the subsequent application of CLA to refine the capsule outputs for enhanced contextual understanding.

For the i-th sample, the capsule layer produces an embedding matrix $\mathbf{V}_i = \{\lambda_j\}_{j=1}^{C}$, where λ_j is the final output vector corresponding to the j-th capsule, and C represents the total number of capsules. This matrix serves as the foundation for CLA, where the focus is on adaptively weighting the capsule outputs to

Algorithm 1 Capsule Network with Capsule-level Attention

```
Input:  $U = \{u_i\}_{i=1}^{T}$: Output of Bi-LSTM after applying dropout
        $R$: Number of routing iterations
        $C$: Number of capsules in upper layer
Output: $f_{caps}$: Resultant vector of capsule network after applying CLA
for each capsule $i$ in lower layer, $j$ in upper layer: $\hat{u}_{j|i} = W_{ij} u_i$
for each capsule $j$ in upper layer: $\bar{u}_j \leftarrow \sum_i \hat{u}_{j|i}$
for each capsule $j$ in upper layer: $\bar{u}_j \leftarrow \text{squash}(\bar{u}_j)$
for each capsule $i$ in lower layer, $j$ in upper layer: $b_{ij} = 0$
for $k$ in $1 \rightarrow R$ do
    for each capsule $i$ in lower layer, $j$ in upper layer: $c_{ij} \leftarrow \text{softmax}(b_{ij})$
    for each capsule $j$ in upper layer: $r_j \leftarrow \sum_i c_{ij} \hat{u}_{j|i}$
    for each capsule $j$ in upper layer: $v_j \leftarrow \text{squash}(r_j)$
    for each capsule $i$ in lower layer, $j$ in upper layer: $b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \cdot v_j$
end for
for each capsule $j$ in upper layer: $\lambda_j = \bar{u}_j \oplus v_j$
$\mathbf{V} = \{\lambda_j\}_{j=1}^{C}$
for all $\lambda_j \in \mathbf{V}$ do
    $g_j \leftarrow \tanh(W \lambda_j + b_w)$
    $e_j \leftarrow \text{softmax}(g_j)$
end for
$f_{caps} \leftarrow \sum_{j=1}^{C} e_j \lambda_j$
return $f_{caps}$
```

capture contextually relevant information. In the formulation of CLA, the initial step involves a non-linear transformation within each capsule:

$$g_j = \tanh(W \lambda_j + b_w) \tag{14}$$

Here, W and b_w are learned weight and bias parameters, respectively. Subsequently, the softmax activation is applied to calculate attention weights, assigning significance to each capsule:

$$e_j = \frac{\exp(g_j)}{\sum_{k=1}^{C} \exp(g_k)} \tag{15}$$

This enables the model to dynamically allocate attention based on the contextual significance of each capsule. The final step in CLA involves the computation of a weighted sum, creating a contextually refined representation of the capsule outputs:

$$f_{caps} = \sum_{j=1}^{C} e_j \lambda_j \tag{16}$$

The f_{caps} vector represents the capsule-level attention-weighted sum, capturing the most informative aspects from the entire set of capsules based on the calculated attention probabilities. This adaptability empowers the model to selectively emphasize relevant linguistic features, enhancing its ability to discern context and meaning in the text.

4.5. Auxiliary Features

We incorporate emotion-based and sentiment-based features to enhance the text analysis capabilities of our model. For emotion-based features, we employ the NRCLex[3] library, which allows us to map each word in

[3] https://pypi.org/project/NRCLex/

the text to a specific emotion category {fear, anger, anticipation, trust, surprise, positive, negative, sadness, disgust, joy}. The resulting feature vector for each text sample captures the frequency of occurrences for each emotion category, providing an understanding of the emotional nuances within the text. Additionally, we leverage the VADER[4] sentiment analysis tool to extract a compound sentiment score for each text entry. This score encompasses the overall sentiment polarity, ranging from -1 (most negative) to 1 (most positive), with 0 indicating a neutral sentiment. Subsequently, the emotion-based features are concatenated with the corresponding sentiment score for each sample, resulting in a vector $f_{aux} \in \mathbb{R}^{11}$. This combination enables our model to discern both emotional and sentiment-related nuances within the analyzed text samples. Furthermore, as a preprocessing step, we apply demojize function of PyPI's emoji[5] library. Demojize is a process that involves replacing emojis with their textual equivalents. This step ensures that emojis are appropriately interpreted and accounted for in our text analysis.

4.6. Feature Fusion and Classification

This stage involves the combination of the outputs from HAN (f_{han}) and the capsule network after CLA (f_{caps}), along with the auxiliary features (f_{aux}), through the concatenation operation, resulting in a unified feature vector f_i.

Following the fusion, the combined feature vector f_i undergoes processing through two dense layers with Rectified Linear Unit (ReLU) activation functions. After the first dense layer transformation, f_{d_1} is obtained with an output dimension of 256. Subsequently, the second dense layer transforms f_{d_1} into f_{d_2} with an output dimension of 128. The transformations applied in each dense layer are represented as:

$$f_{d_1} = \mathrm{ReLU}(W_{d_1} f_i + b_{d_1}) \tag{17}$$

$$f_{d_2} = \mathrm{ReLU}(W_{d_2} f_{d_1} + b_{d_2}) \tag{18}$$

where W_{d_1} and b_{d_1} are the weight matrix and bias vector associated with the first dense layer, and W_{d_2} and b_{d_2} are the corresponding parameters for the second dense layer. The processed feature vector f_{d_2} is then forwarded through the classification layer with a sigmoid activation function (σ), generating the predicted output $\hat{y}_i$. Mathematically, this can be expressed as:

$$\hat{y}_i = \sigma(W_c f_{d_2} + b_c) \tag{19}$$

where W_c represents the weight matrix associated with the classification layer, and b_c is the bias vector. For the purpose of training, the model employs the binary cross-entropy loss (L) defined as:

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{20}$$

where N denotes the number of samples, y_i is the ground truth label for the i^{th} sample, and $\hat{y}_i$ is the corresponding predicted probability.

After the model is trained, we obtain the model's predictions and calculate the mean.

$$\hat{Y}_i = \frac{\sum_{m \in S_i} \hat{y}_m}{|S_i|} \tag{21}$$

For each $m \in S_i$ generated using the i-th sample (x_i), resulting in three output probabilities ($\hat{y}_{\alpha_i}, \hat{y}_{\beta_i}, \hat{y}_{\gamma_i}$). To get the final probability, we calculate the mean of these three values.

[4] https://pypi.org/project/vaderSentiment/

[5] https://pypi.org/project/emoji/

Table 2
Dataset Statistics

Dataset	Train	Dev	Test	Total
Tamil	14,199	1,755	1,761	17,715
Malayalam	7,873	974	970	9,817
Kannada	4,940	618	618	6,176

4.7. Hyperparameters and Model Size

HopeCap employs three branches; translated, transliterated Indic, and transliterated Roman script. Three versions of the comments are passed through individual pipelines. A single pipeline uses approximately 19 million trainable parameters. Thus the overall size of the HopeCap in terms of trainable parameters is 51 million. It is to be noted that XML-R is used for feature extraction for the translated version, and MuRIL is used for the other two versions; their parameters are not counted in trainable parameters. Among different sets of hyperparameters tested for the training and testing, we finalize a batch size of 32, a learning rate of 2e-5, and 20 epochs. We employ an AdamW optimizer based on our testing with different optimizers. The final model is trained with these parameters.

5. Experimental Evaluations

This section begins with an overview of the utilized datasets, followed by a detailed comparison of baseline and state-of-the-art methods across multiple languages. Performance metrics such as accuracy, precision, recall, and weighted F1-score are presented for each method, highlighting their efficacy in hope speech detection. In the subsequent results, the weighted F1-score is simply referred to as F1 for brevity. Additionally, ablation analyses are conducted to dissect the impact of various components within HopeCap, including output fusion, attention mechanisms, auxiliary features, and the capsule network. Qualitative evaluations further scrutinize model predictions through test case analyses, shedding light on their strengths and limitations in capturing linguistic nuances and context-specific cues.

5.1. Datasets

The datasets utilized in our research include HopeEDI (Chakravarthi, 2020) and KanHope (Hande et al., 2021). The HopeEDI dataset, primarily derived from social media comments on YouTube, captures discussions related to equality, diversity, and inclusion, spanning topics such as women in STEM, COVID-19, racial conflicts, and geopolitical disputes. The annotations within the dataset are categorized into *Hope*, *Not Hope*, and *Other language* labels. Since we aim to perform a binary classification of hope speech, we exclude the instances with *Other language* label and only consider *Hope* and *Not Hope* as HS and NHS classes, respectively. For our research, we specifically utilize the comments in Tamil and Malayalam languages from the HopeEDI dataset forming the Tamil and Malayalam datasets, respectively. After excluding instances with the *Other language* label, the Tamil dataset comprises 17,715 comments, and the Malayalam dataset comprises 9,817 comments, providing a rich source for investigating hope speech in the context of these languages. Simultaneously, we leverage the publicly available KanHope dataset to investigate hope speech detection in code-mixed Kannada-English texts. This dataset is constructed by collecting 6,176 YouTube comments on diverse topics, such as movie trailers, the India-China border dispute, and social issues. Annotations within the dataset are categorized into *Hope Speech* and *Non-hope Speech* labels, forming the HS and NHS classes, respectively. We refer to this dataset as the Kannada dataset. The distribution of comments across training, development, and test sets for the Tamil, Malayalam, and Kannada datasets is illustrated in Table 2.

Although comparisons with baselines and state-of-the-art models are conducted across all three datasets, for the purpose of ablation analysis, we exclusively employ the Tamil and Malayalam datasets.

5.2. Compared Methods

This section presents a brief description of baseline and state-of-the-art methods employed for comparison with HopeCap.

5.2.1. Baseline Methods

1. IndicBERT (Kakwani, Kunchukuttan, Golla, Gokul, Bhattacharyya, Khapra and Kumar, 2020): IndicBERT, a multilingual model, is trained on a vast monolingual corpus encompassing 12 major Indian languages. With fewer parameters than mBERT (Devlin, Chang, Lee and Toutanova, 2018) and XLM-R (Conneau, Khandelwal, Goyal, Chaudhary, Wenzek, Guzmán, Grave, Ott, Zettlemoyer and Stoyanov, 2019), it achieves comparable or superior performance across diverse tasks.

2. mBERT (Devlin et al., 2018): mBERT is a pretrained transformer model for 104 languages. Trained with masked language modeling and next sentence prediction objectives, it excels in capturing bidirectional representations, making it valuable for diverse downstream tasks.

3. MuRIL (Khanuja, Bansal, Mehtani, Khosla, Dey, Gopalan, Margam, Aggarwal, Nagipogu, Dave et al., 2021): MuRIL is a BERT-based model trained on 17 Indian languages and their transliterations, incorporating translation and transliteration segment pairs during training for enhanced multilingual representation.

4. XLM-R (Conneau et al., 2019): XLM-R, based on RoBERTa (Liu, Ott, Goyal, Du, Joshi, Chen, Levy, Lewis, Zettlemoyer and Stoyanov, 2019) model, is a multilingual model trained on 100 languages. It achieves significant accuracy gains on cross-lingual benchmarks, particularly excelling in low-resource languages.

5. BERT (Devlin et al., 2018): BERT is a pre-trained language representation model that employs bidirectional context to understand word relationships in both directions.

5.2.2. State-of-the-art Methods

1. Spartans (Sharma and Arora, 2021): The authors employ a two-step approach, pretraining ULMFiT (Howard and Ruder, 2018) and RoBERTa models on synthetically generated code-mixed data for Tamil hate speech detection and offensive language identification, followed by fine-tuning and an ensemble of classifiers for final predictions.

2. TeamUNCC (Mahajan et al., 2021): The authors utilize a fine-tuning approach with RoBERTa for English and XLM-R for Tamil and Malayalam to address the hope speech detection task.

3. NLP@CUET (Hossain et al., 2021): The authors use machine learning, deep learning, and transformer-based methods to detect hope speech in English, Tamil, and Malayalam, achieving top performance with XLM-R among the transformer models.

4. MHSD (Chakravarthi, 2022b): Multilingual hope speech detection (MHSD) presents a dataset for hope speech in English, Tamil, and Malayalam. Benchmark systems are created using state-of-the-art Machine Learning methods and Deep Learning methods. RoBERTa performs best among all the benchmark systems.

5. Team hub (Huang and Bai, 2021): The authors utilize the XLM-R pre-trained language model, integrate an Inception block, and incorporate the TF-IDF algorithm for hope speech detection in English, Tamil, and Malayalam.

6. MUCS (Balouchzahi et al., 2021): Their model employs an ensemble of classifiers, including Logistic Regression, eXtreme Gradient Boosting (XGB), and Multi-Layer Perceptron (MLP), trained on TFIDF vectors.

7. GCDH (Ziehe, Pannach and Krishnan, 2021): The study employs machine learning techniques, including baseline algorithms such as Naive Bayes and Support Vector Machine, along with a fine-tuned XLM-R model using cross-lingual transfer learning, for the identification of Hope Speech in English, Malayalam, and Tamil short texts.

8. ZYJ (Zhao and Tao, 2021): The authors employ an XLM-R-based model with an attention mechanism, utilizing the weighted sum of 12 layers' output for classification, and employ the Stratified-K-Fold method to enhance training data.

Table 3
Performance Comparison on Tamil Language

Model	Accuracy	Precision	Recall	F1
		Baseline Models		
IndicBERT (Kakwani et al., 2020)	0.5970	0.5958	0.5970	0.5836
mBERT (Devlin et al., 2018)	0.5911	0.5891	0.5911	0.5881
MuRIL (Khanuja et al., 2021)	0.5866	0.5811	0.5866	0.5767
XLM-R (Conneau et al., 2019)	0.6114	0.6127	0.6114	0.6062
BERT (Devlin et al., 2018)	0.5991	0.5988	0.5991	0.5896
		State-of-the-art Models		
Spartans (Sharma and Arora, 2021)	-	0.62	0.62	0.61
TeamUNCC (Mahajan et al., 2021)	-	0.61	0.61	0.61
NLP@CUET (Hossain et al., 2021)	-	0.61	0.61	0.60
MHSD (Chakravarthi, 2022b)	-	0.61	0.61	0.60
Team_hub (Huang and Bai, 2021)	-	0.61	0.61	0.59
MUCS (Balouchzahi et al., 2021)	-	0.59	0.59	0.59
HopeCap	**0.6632**	**0.6608**	**0.6632**	**0.6613**

9. DC-BERT4HOPE (Hande et al., 2021): It is a dual-channel model that combines two language models, utilizing one (model1) fine-tuned on translated English texts and the other (model2) on code-mixed Kannada-English data. By taking the weighted sum of their pooled outputs, the model leverages the strengths of multilingual and monolingual language models to enhance hope speech detection in a code-mixed context.

10. SSN_ARMM (Vijayakumar, Prathyush, Aravind, Angel, Sivanaiah, Rajendram and Mirnalinee, 2022): SSN_ARMM approach is the top performer for Kannada language in LT-EDI-ACL2022 competition. It is based on a pre-trained transformer-based method ALBERT which is trained on 12 Indic languages.

11. DC-LM (Hande, Hegde, Sangeetha, Priyadharshini and Chakravarthi, 2022): Dual Channel Language Modeling (DC-LM) is designed for code-mixed Kannada-English hope speech detection. Two channels are created: one for the Indic language and the other for its translation. Among different models tested, the combination of XLnet and XLM-R performs better than the other combinations.

5.3. Results

This section presents the experimental results and comparisons. First, we present the results of 'HopeCap along with baseline and state-of-the-art methods for each language using four different evaluation metrics. Subsequently, the next section presents an ablation study that demonstrates, in detail, the influence of different components of `HopeCap` on the overall performance.

5.3.1. Performance Comparison on Tamil Language

In our evaluation of baseline models on the Tamil dataset, XLM-R exhibits the highest overall performance, achieving an accuracy of 61.14% and an F1-score of 60.62%. Among the state-of-the-art models, Spartans (Sharma and Arora, 2021) and TeamUNCC (Mahajan et al., 2021) achieve the highest F1-scores of 61.00% each. Notably, `HopeCap` surpasses all baseline and state-of-the-art models, achieving the highest accuracy of 66.32%, F1-score of 66.13%, and superior precision and recall metrics. This marks a notable improvement, as `HopeCap` surpasses the prior best performance by approximately 5.13% in terms of the F1-score. The performance improvement can be attributed to the novel capsule network that learns spatial relationships within text through the integration of two levels of vectors where the child capsule vector is integrated through a skip connection with the vector generated by dynamic routing. This helps capture spatial relationships within text, improving the method's understanding of syntactic structure. Capsule features are further enhanced through CLA, which focuses on discriminative capsule features. Other contributors to enhanced performance are the inclusion of HAN, auxiliary features, and the three versions of the comment for final classification. The detailed breakdown of performance metrics in Table 3 provides insights into the strengths of each model in the context of the Tamil language.

Table 4
Performance Comparison on Malayalam Language

Model	Accuracy	Precision	Recall	F1
		Baseline Models		
IndicBERT (Kakwani et al., 2020)	0.7722	0.7477	0.7722	0.7584
mBERT (Devlin et al., 2018)	0.7535	0.7546	0.7535	0.7538
MuRIL (Khanuja et al., 2021)	0.7749	0.7707	0.7749	0.7728
XLM-R (Conneau et al., 2019)	0.8392	0.8415	0.8392	0.8398
BERT (Devlin et al., 2018)	0.7423	0.7183	0.7423	0.7238
		State-of-the-art Models		
MHSD (Chakravarthi, 2022b)	-	0.87	0.87	0.87
NLP@CUET (Hossain et al., 2021)	-	0.86	0.85	0.85
MUCS (Balouchzahi et al., 2021)	-	0.85	0.85	0.85
GCDH (Ziehe et al., 2021)	-	0.84	0.85	0.85
ZYJ (Zhao and Tao, 2021)	-	0.84	0.84	0.84
Team_hub (Huang and Bai, 2021)	-	0.84	0.85	0.84
HopeCap	**0.9163**	**0.9095**	**0.9163**	**0.9158**

5.3.2. *Performance Comparison on Malayalam Language*

In the Malayalam dataset, XLM-R again demonstrates superior performance among the baseline models, achieving an accuracy of 83.92% and an F1-score of 83.98%. Among state-of-the-art methods, MHSD (Chakravarthi, 2022b) achieves the highest weighted F1-score of 87%. Among other approaches, NLP@CUET(Hossain et al., 2021), MUCS(Balouchzahi et al., 2021), and GCDH(Ziehe et al., 2021) exhibit competitive F1-scores of 85.00%. Again, `HopeCap` outperforms all models, recording the highest accuracy and F1-score of 91.63% and 91.58%, respectively. This significant enhancement is evident as `HopeCap` surpasses the prior best performance by approximately 4.58% in terms of F1-score. The novel CapsuleNet with CLA and HAN contributes to capturing complex relationships within the text. To ensure a comprehensive contextual understanding of each comment, we create three iterations: one translated, another transliterated into the Indic script, and a third transliterated into the Roman script. This approach contributes to improved performance by enhancing the contextual richness available for analysis. Additionally, we incorporate two sets of auxiliary features focused on emotion and sentiment, further enriching the insights derived from each comment.Table 4 details the performance of all the methods for the Malayalam dataset.

5.3.3. *Performance Comparison on Kannada Language*

Lastly, we analyze the performance of different methods on the Kannada dataset. Among the baseline models, mBERT exhibits the highest overall performance, achieving an accuracy of 73.50% and an F1-score of 72.60%. In comparison, among the state-of-the-art models, DC-LM (Hande et al., 2022) with XLnet-XLM-R integration achieves the highest F1-score of 76.66%. However, `HopeCap` surpasses all baseline and state-of-the-art models, securing the highest accuracy of 78.95% and F1-score of 78.84% beating DC-LM (XLnet-XLMR) approximately with 2% in terms of both the accuracy and F1-score. The detailed breakdown of performance metrics in Table 5 provides insights into the strengths of each model specifically within the Kannada language context.

5.4. Ablation Analysis

In the ablation analysis, our primary objective is to assess the influence of specific components on the overall performance of the system. To ensure reliable comparisons, we systematically modify the system while keeping other factors constant.

5.4.1. *Effect of Output Fusion*

As mentioned in subsection 4.1, to classify a sample as *hope speech* or *non-hope speech*, the model's predictions from all three data branches (α, β, γ) are aggregated in order to obtain the final probability. Now, we analyze the impact of fusing these predictions by evaluating the performance of the model on individual branches and different combinations of branches. The results are presented in Table 6. Initially, the impact of each branch is evaluated independently, i.e., we consider the prediction of a single branch as the final output of the model. Then, we combine the branches in pairs to explore their joint impact on

Table 5
Performance Comparison on Kannada Language

Model	Accuracy	Precision	Recall	F1
	Baseline Models			
IndicBERT (Kakwani et al., 2020)	0.6909	0.6800	0.6900	0.6647
mBERT (Devlin et al., 2018)	0.7350	0.7280	0.7350	0.7260
MuRIL (Khanuja et al., 2021)	0.7152	0.7100	0.7200	0.6922
XLM-R (Conneau et al., 2019)	0.7136	0.7100	0.7100	0.7037
BERT (Devlin et al., 2018)	0.7040	0.7010	0.7040	0.7020
	State-of-the-art Models			
DC-BERT4HOPE (BERT-mBERT) (Hande et al., 2021)	0.740	0.734	0.740	0.735
DC-BERT4HOPE (RoBERTa-mBERT) (Hande et al., 2021)	0.756	0.752	0.756	0.752
SSN_ARMM (Vijayakumar et al., 2022)	-	0.740	0.760	0.750
DC-LM(XLnet-mBERT) (Hande et al., 2022)	0.700	0.700	0.701	0.726
DC-LM(XLnet-XLMR) (Hande et al., 2022)	0.770	0.758	0.767	0.766
HopeCap	**0.7895**	**0.7865**	**0.7895**	**0.7884**

Table 6
Effect of Fusing Transliterated Indic, Translated, and Transliterated Roman Branches

Data Branch	Tamil				Malayalam			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
α	0.6411	0.6398	0.6411	0.6388	0.8783	0.8898	0.8783	0.8790
β	0.6206	0.6194	0.6206	0.6195	0.8835	0.8808	0.8835	0.8820
γ	0.6337	0.6333	0.6337	0.6320	0.8897	0.8851	0.8897	0.8815
$\beta+\gamma$	0.6417	0.6408	0.6417	0.6375	0.8938	0.8894	0.8938	0.8902
$\alpha+\beta$	0.6502	0.6490	0.6502	0.6487	0.8979	0.8946	0.8979	0.8957
$\alpha+\gamma$	0.6536	0.6530	0.6536	0.6499	0.8948	0.8918	0.8948	0.8929
$\alpha+\beta+\gamma$ (HopeCap)	**0.6632**	**0.6608**	**0.6632**	**0.6613**	**0.9163**	**0.9095**	**0.9163**	**0.9158**

predictive performance. In this setting, the final output of the model is obtained by taking the mean of the two output probabilities. We observe that combining the branches in pairs yields improvements in accuracy and F1-score compared to individual branches. Finally, as part of our proposed model, we combine all three branches to assess the collective effect of leveraging diverse sources of information. This combination ($\alpha + \beta + \gamma$) exhibits higher accuracy and F1-score compared to individual and pairwise combinations of branches. This indicates that leveraging the original, translated, and transliterated versions and combining their predictions through mean aggregation enhances the overall predictive capability of our model.

5.4.2. *Effect of Attention Components*

Given that HopeCap integrates two attention components, HAN and CLA, we aim to study their impact on the overall model architecture. Our analysis begins by evaluating the performance of the model after selectively excluding the HAN component while keeping the remainder of the architecture unchanged. This modification results in the direct transmission of the output from the BiLSTM layer solely to the capsule network. Subsequently, when the CLA component is removed while keeping the rest of the architecture unchanged, we flatten the output of the capsule network before concatenating it with the outputs from the HAN component and auxiliary features. Table 7 illustrates the outcomes derived from the systematic removal of attention components within our model architecture. Upon exclusion of HAN, the Tamil dataset witnesses a reduction of 1.08% in accuracy and 1.23% in the F1-score, while the Malayalam dataset experiences a decrease of 1.40% in accuracy and 1.48% in the F1-score. Similarly, the removal of CLA results in a decline of 0.97% in accuracy and 1.06% in the F1-score for the Tamil language and a decrease of 1.23% in accuracy and 1.35% in the F1-score for the Malayalam language.

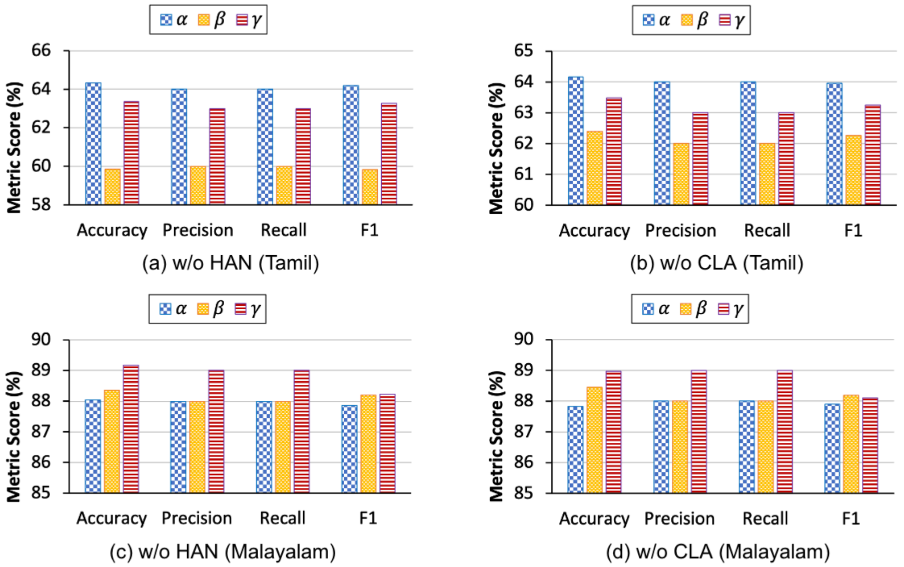


Figure 2: Ablation Analysis of Attention Components

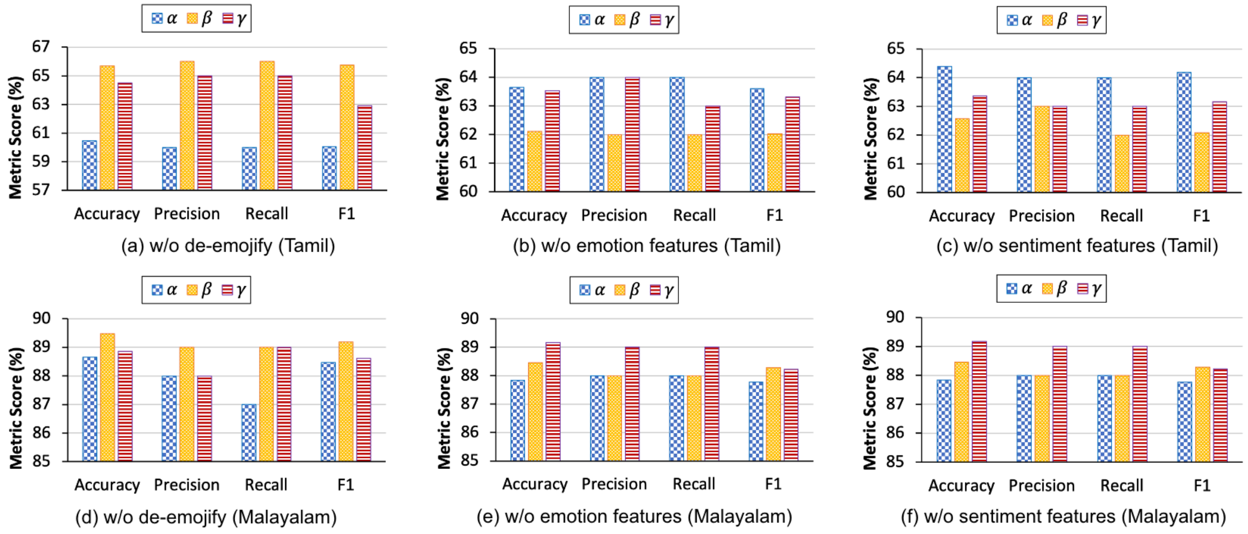
Table 7
Effect of Attention Components and Auxiliary Features

Method	**Tamil**				**Malayalam**			
	Accuracy	**Precision**	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**
Attention Components								
w/o HAN	0.6524	0.6512	0.6524	0.6490	0.9023	0.9014	0.9023	0.9010
w/o CLA	0.6535	0.6510	0.6535	0.6507	0.9040	0.9028	0.9040	0.9023
Auxiliary Features & de-emojify								
w/o de-emojify	0.6581	0.6565	0.6581	0.6560	0.9121	0.9086	0.9121	0.9080
w/o emotion features	0.6546	0.6528	0.6546	0.6524	0.9058	0.9065	0.9058	0.9049
w/o sentiment features	0.6572	0.6553	0.6572	0.6562	0.9111	**0.9110**	0.9111	0.9102
HopeCap	**0.6632**	**0.6608**	**0.6632**	**0.6613**	**0.9163**	0.9095	**0.9163**	**0.9158**

To further investigate the impact of attention components, we conduct a performance evaluation of `HopeCap` on individual data branches subsequent to the removal of attention components (illustrated in Figure 2). A notable observation arises: upon the removal of HAN, there is a decrease of 2.21% in accuracy and 2.13% in the F1-score, specifically for the β branch of the Tamil dataset. Subsequent analyses involving the removal of HAN and CLA reveal marginal differences in performance for both datasets, ranging from 0%-0.34% in both accuracy and F1-score across all branches. This analysis underscores the significance of both HAN and CLA components in sustaining optimal model performance across different language datasets.

5.4.3. Effect of Auxiliary Features

Table 7 illustrates the impact of auxiliary features and de-emojify on the performance of our model. In this analysis, we systematically evaluate the impact of de-emojify, emotion features, and sentiment features on our model's performance by excluding each one individually. When not using de-emojify, we simply remove the emojis from the text in the preprocessing step. Without de-emojify, the Tamil dataset experiences a decrease of 0.51% in accuracy and 0.53% in the F1-score, while the Malayalam dataset shows a decrease of 0.42% in accuracy and 0.78% in the F1-score. Similarly, excluding emotion features results in a decrease



(a) w/o de-emojify (Tamil)

(b) w/o emotion features (Tamil)

(c) w/o sentiment features (Tamil)

(d) w/o de-emojify (Malayalam)

(e) w/o emotion features (Malayalam)

(f) w/o sentiment features (Malayalam)

Figure 3: Ablation Analysis of Auxiliary Features

Table 8
Influence of Capsule Network

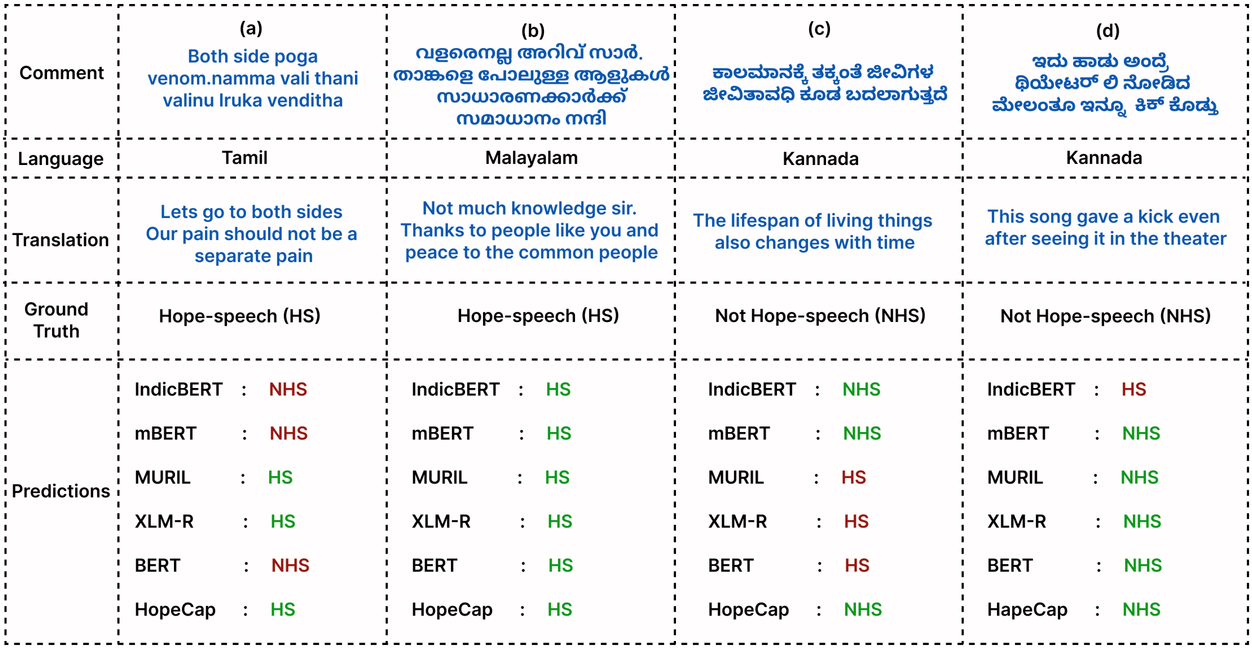
Method	Tamil				Malayalam			
	Accuracy	**Precision**	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**
Dense layers	0.6452	0.6433	0.6452	0.6441	0.8927	0.8941	0.8927	0.8917
CNN	0.6374	0.6399	0.6374	0.6369	0.8848	0.8865	0.8848	0.8839
LSTM network	0.6498	0.6481	0.6498	0.6475	0.8952	0.8963	0.8952	0.8945
CapsuleNet (HopeCap)	**0.6632**	**0.6608**	**0.6632**	**0.6613**	**0.9163**	**0.9095**	**0.9163**	**0.9158**

of 0.86% in accuracy and 0.89% in the F1-score for the Tamil dataset, and 0.42% in accuracy, and 0.78% in the F1-score for the Malayalam dataset. Moreover, when sentiment features are excluded, the Tamil dataset witnesses a decrease of 0.6% in accuracy and F1-score each, and the Malayalam dataset experiences a decrease of 0.52% in accuracy and 0.56% in F1-score.

Similar to the analysis conducted on attention components, we evaluate the performance of HopeCap across individual data branches by iteratively removing de-emojify, emotion features, and sentiment features, as depicted in Figure 3. Our investigation reveals an interesting observation upon the removal of de-emojify: a notable decrease of 3.64% and 3.83% in the model's accuracy and F1-score, respectively, is observed for the α branch of the Tamil dataset. Conversely, for the β branch of the Tamil dataset, a corresponding increase in accuracy and F1-score by 3.64% and 3.79%, respectively, is noted. For the analyses involving the removal of emotion and sentiment features, the impact is comparatively marginal across all branches, with variances in the range of 0%-1.14% in both accuracy and F1-score across both datasets.

5.4.4. Effect of Capsule Network

This section analyses the impact of the capsule network within HopeCap framework by conducting a comprehensive performance analysis on the Tamil and Malayalam datasets. This involves substituting the capsule network and CLA with alternative components such as Convolutional Neural Network (CNN), Dense layers, and Long Short-Term Memory (LSTM) network. Through this comparative assessment (shown in Table 8), we aim to elucidate the distinctive contributions and effectiveness of the capsule network in enhancing our model's capabilities. In the case of the Tamil dataset, LSTM exhibits slightly superior performance in both accuracy and F1-score compared to Dense layers and CNN, achieving 64.98% accuracy and 64.75% F1-score. Nevertheless, the utilization of the capsule network instead of LSTM resulted in a



Comment	(a) Both side poga venom.namma vali thani valinu Iruka venditha	(b) വളരെനല്ല അറിവ് സാർ. താങ്കളെ പോലുള്ള ആളുകൾ സാധാരണക്കാർക്ക് സമാധാനം നന്ദി	(c) ಕಾಲಮಾನಕ್ಕೆ ತಕ್ಕಂತೆ ಜೀವಿಗಳ ಜೀವಿತಾವಧಿ ಕೂಡ ಬದಲಾಗುತ್ತದೆ	(d) ಇದು ಹಾಡು ಅಂದ್ರೆ ಥಿಯೇಟರ್ ಲಿ ನೋಡಿದ ಮೇಲಂತೂ ಇನ್ನೂ ಕಿಕ್ ಕೊಡ್ತು
Language	Tamil	Malayalam	Kannada	Kannada
Translation	Lets go to both sides Our pain should not be a separate pain	Not much knowledge sir. Thanks to people like you and peace to the common people	The lifespan of living things also changes with time	This song gave a kick even after seeing it in the theater
Ground Truth	Hope-speech (HS)	Hope-speech (HS)	Not Hope-speech (NHS)	Not Hope-speech (NHS)
Predictions	IndicBERT : NHS mBERT : NHS MURIL : HS XLM-R : HS BERT : NHS HopeCap : HS	IndicBERT : HS mBERT : HS MURIL : HS XLM-R : HS BERT : HS HopeCap : HS	IndicBERT : NHS mBERT : NHS MURIL : HS XLM-R : HS BERT : HS HopeCap : NHS	IndicBERT : HS mBERT : NHS MURIL : NHS XLM-R : NHS BERT : NHS HapeCap : NHS

Figure 4: Qualitative Analysis of HopeCap and Other Baseline Methods

notable enhancement, with a margin of 1.34% in accuracy and 1.38% in F1-score, yielding 66.32% accuracy and 66.13% F1-score. Similarly, for the Malayalam dataset, LSTM surpasses Dense layers and CNN in accuracy and F1-score, reaching 89.52% and 89.45%, respectively. For the Malayalam language, too, the capsule network demonstrates superior performance, achieving a higher accuracy of 91.63%, surpassing LSTM by 2.11%, and an enhanced F1-score of 91.58%, outperforming LSTM by 2.13%. These findings affirm the effectiveness of the capsule network with CLA in augmenting the overall efficacy of HopeCap for hope speech detection in multilingual settings.

5.5. Qualitative Analysis

The performance of various methods in hope speech detection, including HopeCap, IndicBERT, mBERT, XLM-R, BERT, and MuRIL, is assessed using four comments as test cases as shown in Figure 4.

The first comment, shown in Figure 4(a), "Let's go to both sides. Our pain should not be a separate pain," is originally a Tamil language comment written in Roman script. MuRIL, XLM-R, and HopeCap correctly predict the comment as hope speech, demonstrating their proficiency in identifying the Indic language in Roman script. HopeCap benefits from three versions of the comment, translated, transliterated in Roman script, and transliterated in Indic script. The three versions mitigate the error in classification; hence HopeCap obtains accurate predictions.

The second comment, shown in Figure 4(b), "Not much knowledge sir. Thanks to people like you and peace to the common people," is originally a Malayalam language comment written in Malayalam script. All the methods, including MuRIL, XLM-R, IndicBERT, mBERT, BERT, and HopeCap, correctly identify the presence of hope speech. We argue this is due to the fact that the comment stands out as a hope speech by explicitly using hopeful terms such as peace. However, the context of each word is equally important, which in this case, all the methods capture correctly. This reflects their ability to discern positive sentiments and expressions where some common terms of gratitude or hope are used explicitly.

In the third comment, "The lifespan of living things also changes with time," MuRIL, XLM-R, and BERT exhibit inconsistencies in predicting hope speech, incorrectly labeling the comment as such. This suggests a potential limitation in understanding the nuanced context or semantics related to hope speech, highlighting areas for improvement in these models' performance.

Lastly, the fourth comment, "This song gave a kick even after seeing it in the theater," is correctly identified as not-hope-speech by all methods except IndicBERT. This variation in predictions indicates differences in the models' interpretation of linguistic cues. It emphasizes the importance of further refinement to enhance the accuracy of hope speech detection across diverse linguistic nuances.

In conclusion, the evaluation of various methods for hope speech detection, including MuRIL, XLM-R, IndicBERT, mBERT, and `HopeCap`, on four diverse comments sheds light on their respective strengths and weaknesses. While MuRIL, XLM-R, and `HopeCap` demonstrate consistent and accurate predictions for hope speech in most cases, there are instances where certain models, such as IndicBERT, show discrepancies. These findings underscore the importance of considering the linguistic nuances and context-specific characteristics inherent in hope speech detection.

6. Discussion

The proposed approach, `HopeCap`, for hopeful content detection shows performance gain across different evaluation metrics as compared to state-of-the-art methods. This is to be noted that a more recent competition, LT-EDI-2022, for hope speech detection on Indic languages was organized as part of ACL 2022 (Chakravarthi et al., 2022). Interestingly, the methodologies introduced in this competition failed to match the performance achieved by methods presented in the previous competition despite having the same dataset. Due to this reason, we take methods from the previous competition, LT-EDI-2021 (Chakravarthi and Muralidaran, 2021), for comparison. `HopeCap` achieves an average performance gain of 7.24% in Tamil, 14.6% in Malayalam, and 9.06% in the Kannada language over baseline transformer-based models in terms of F1-score. The performance gain can be attributed to the synergy of different mechanisms employed in `HopeCap`. The proposed approach uses three versions of a comment that provide more contextual information. The custom CapsuleNet helps capture spatial relationships in the comment. It combines the probability vector of the child capsule and the final vector of CapsuleNet through a skip connection providing two-level information. To gain insights into the hierarchical structure of comments, a hierarchical attention mechanism is employed to get word-level and sentence-level attention vectors. Moreover, emotion and sentiment features provide useful insights related to different emotions and the overall sentiment of the comments. Ablation analysis of the proposed approach demonstrates that each component contributes to enhancing the model's capability to identify hopeful instances in low-resource languages. We occasionally observe cases where removing certain model components leads to slightly improved performance on individual branches compared to when no components are removed. However, these occurrences are rare and inconsistent in our empirical findings. Therefore, we deem them negligible and confidently assert that `HopeCap` consistently outperforms in terms of performance and reliability. The influence of individual modules and components on the performance of `HopeCap` is shown in the ablation study in subsection 5.4.

7. Conclusion

This study introduces, `HopeCap`, a framework for detecting hope speech across multiple low-resource Indic languages. `HopeCap` employs a novel capsule network that outputs a joint representation of the predicted child capsule vector and the vector obtained through dynamic routing. This helps to learn spatial relationships within the text. Further, the integration of attention components, hierarchical attention, and capsule-level attention, the inclusion of sentiment-based and emotion-based features, and the fusion of outputs from the translated, transliterated Indic, and transliterated Roman data branches contribute collectively to enhanced performance. The proposed method achieves consistent and competitive outcomes across various linguistic contexts. In the Tamil, Malayalam, and Kannada datasets, `HopeCap` surpasses the state-of-the-art by approximately 6%, 6.5%, and 4% in weighted F1-score, respectively. The proposed method outperforms alternative architectures, such as those relying on dense layers, CNN, and LSTM networks, emphasizing the effectiveness of the capsule-level attention in capturing nuanced patterns in hope-related comments. Ablation analyses further demonstrate the significance of individual components in our model. Overall, our findings underscore the efficacy of `HopeCap` in accurately identifying hope speech across multiple languages, highlighting its potential for application in real-world scenarios such as social media monitoring and sentiment analysis. For future work, expanding the `HopeCap` framework to include additional languages could enhance its applicability and impact, facilitating hope speech detection in a more diverse range of linguistic communities. Additionally, integrating additional modalities, such as audio, video, or user interactions, could enhance the framework's depth and efficacy in detecting hope speech across diverse contexts. Investigating how different modalities complement each other and contribute to overall performance would be an interesting area of exploration.

8. Acknowledgment

The authors are thankful for the Young Faculty Research Catalysing Grant (YFRCG) by the Indian Institute of Technology Indore for providing the resources to the project (Project ID: IITI/YFRCG/2023-24/03). The authors are also thankful to the UGC-Junior Research Fellowship (UGC-JRF) scheme, an initiative of the Government of India, for providing the Ph.D. fellowship to Mohammad Zia Ur Rehman (NTA Ref. No.: 190510846524).

CRediT authorship contribution statement

Mohammad Zia Ur Rehman: Conceptualization, Methodology, Software, Investigation, Writing - Original Draft, Writing - review & editing. **Devraj Raghuvanshi:** Conceptualization, Methodology, Software, Investigation, Writing - Original Draft, Writing - review & editing. **Harshit Pachar:** Conceptualization, Methodology, Software, Investigation, Writing - Original Draft, Writing - review & editing. **Chandravardhan Singh Raghaw:** Software, Formal analysis, Data Curation, Writing - Original Draft. **Nagendra Kumar:** Conceptualization, Methodology, Supervision, Writing - review & editing.

References

Anshul, A., Pranav, G.S., Rehman, M.Z.U., Kumar, N., 2023. A multimodal framework for depression detection during covid-19 via harvesting social media. IEEE Transactions on Computational Social Systems .

Arunima, S., Ramakrishnan, A., Balaji, A., Thenmozhi, D., et al., 2021. ssn_dibertsity@ lt-edi-eacl2021: hope speech detection on multilingual youtube comments via transformer based approach, in: Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion, pp. 92–97.

Balaji, V., Kannan, A., Balaji, A., Singh, B.B., 2023. Nlp_ssn_cse at hope2023@ iberlef: Multilingual hope speech detection using machine learning algorithms .

Balouchzahi, F., Aparna, B., Shashirekha, H., 2021. Mucs@ lt-edi-eacl2021: cohope-hope speech detection for equality, diversity, and inclusion in code-mixed texts, in: Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion, pp. 180–187.

Balouchzahi, F., Sidorov, G., Gelbukh, A., 2023. Polyhope: Two-level hope speech detection from tweets. Expert Systems with Applications 225, 120078.

Chakravarthi, B.R., 2020. Hopeedi: A multilingual hope speech detection dataset for equality, diversity, and inclusion, in: Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media, pp. 41–53.

Chakravarthi, B.R., 2022a. Hope speech detection in youtube comments. Social Network Analysis and Mining 12, 75.

Chakravarthi, B.R., 2022b. Multilingual hope speech detection in english and dravidian languages. International Journal of Data Science and Analytics 14, 389–406.

Chakravarthi, B.R., Muralidaran, V., 2021. Findings of the shared task on hope speech detection for equality, diversity, and inclusion, in: Proceedings of the first workshop on language technology for equality, diversity and inclusion, pp. 61–72.

Chakravarthi, B.R., Muralidaran, V., Priyadharshini, R., Cn, S., McCrae, J.P., García, M.Á., Jiménez-Zafra, S.M., Valencia-García, R., Kumaresan, P., Ponnusamy, R., et al., 2022. Overview of the shared task on hope speech detection for equality, diversity, and inclusion, in: Proceedings of the second workshop on language technology for equality, diversity and inclusion, pp. 378–388.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V., 2019. Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116 .

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

Dowlagar, S., Mamidi, R., 2021. Edione@ lt-edi-eacl2021: Pre-trained transformers with convolutional neural networks for hope speech detection., in: Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion, pp. 86–91.

Hande, A., Hegde, S.U., Sangeetha, S., Priyadharshini, R., Chakravarthi, B.R., 2022. The best of both worlds: Dual channel language modeling for hope speech detection in low-resourced kannada, in: Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion, pp. 127–135.

Hande, A., Priyadharshini, R., Sampath, A., Thamburaj, K.P., Chandran, P., Chakravarthi, B.R., 2021. Hope speech detection in under-resourced kannada language. arXiv preprint arXiv:2108.04616 .

Hossain, E., Sharif, O., Hoque, M.M., 2021. Nlp-cuet@ lt-edi-eacl2021: Multilingual code-mixed hope speech detection using cross-lingual representation learner, in: Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion, pp. 168–174.

Howard, J., Ruder, S., 2018. Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146 .

Huang, B., Bai, Y., 2021. Team hub@ lt-edi-eacl2021: hope speech detection based on pre-trained language model, in: Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion, pp. 122–127.

Junaida, M., Ajees, A., 2021. Ku_nlp@ lt-edi-eacl2021: a multilingual hope speech detection for equality, diversity, and inclusion using context aware embeddings, in: Proceedings of the first workshop on language technology for equality, diversity and inclusion, pp. 79–85.

Kakwani, D., Kunchukuttan, A., Golla, S., Gokul, N., Bhattacharyya, A., Khapra, M.M., Kumar, P., 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages, in: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 4948–4961.

Kamal, A., Anwar, T., Sejwal, V.K., Fazil, M., 2023. Bicapshate: attention to the linguistic context of hate via bidirectional capsules and hatebase. IEEE Transactions on Computational Social Systems .

Kazienko, P., Bielaniewicz, J., Gruza, M., Kanclerz, K., Karanowski, K., Miłkowski, P., Kocoń, J., 2023. Human-centered neural reasoning for subjective content processing: Hate speech, emotions, and humor. Information Fusion 94, 43–65.

Khanuja, S., Bansal, D., Mehtani, S., Khosla, S., Dey, A., Gopalan, B., Margam, D.K., Aggarwal, P., Nagipogu, R.T., Dave, S., et al., 2021. Muril: Multilingual representations for indian languages. arXiv preprint arXiv:2103.10730 .

Kim, J., Jang, S., Park, E., Choi, S., 2020. Text classification using capsules. Neurocomputing 376, 214–221.

Liu, L., Xu, D., Zhao, P., Zeng, D.D., Hu, P.J.H., Zhang, Q., Luo, Y., Cao, Z., 2023. A cross-lingual transfer learning method for online covid-19-related hate speech detection. Expert Systems with Applications 234, 121031.

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

Madhani, Y., Parthan, S., Bedekar, P., Nc, G., Khapra, R., Kunchukuttan, A., Kumar, P., Khapra, M.M., 2023. Aksharantar: Open indic-language transliteration datasets and models for the next billion users, in: Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 40–57.

Mahajan, K., Al-Hossami, E., Shaikh, S., 2021. Teamuncc@ lt-edi-eacl2021: Hope speech detection using transfer learning with transformers, in: Proceedings of the first workshop on language technology for equality, diversity and inclusion, pp. 136–142.

Mahmud, T., Ptaszynski, M., Eronen, J., Masui, F., 2023. Cyberbullying detection for low-resource languages and dialects: Review of the state of the art. Information Processing & Management 60, 103454.

Malik, M.S.I., Nazarova, A., Jamjoom, M.M., Ignatov, D.I., 2023. Multilingual hope speech detection: A robust framework using transfer learning of fine-tuning roberta model. Journal of King Saud University-Computer and Information Sciences 35, 101736.

Min, C., Lin, H., Li, X., Zhao, H., Lu, J., Yang, L., Xu, B., 2023. Finding hate speech with auxiliary emotion detection from self-training multi-label learning perspective. Information Fusion 96, 214–223.

Mittal, H., Verma, B., 2023. Cat-capsnet: A convolutional and attention based capsule network to detect the driver's distraction. IEEE Transactions on Intelligent Transportation Systems .

Noorian, Z., Ghenai, A., Moradisani, H., Zarrinkalam, F., Alavijeh, S.Z., 2024. User-centric modeling of online hate through the lens of psycholinguistic patterns and behaviors in social media. IEEE Transactions on Computational Social Systems .

Pan, R., Alcaraz-Mármol, G., Garcıa-Sánchez, F., 2023. Umuteam at hope2023iberlef: Evaluation of transformer model with data augmentation for multilingual hope speech detection, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2023), co-located with the 39th Conference of the Spanish Society for Natural Language Processing (SEPLN 2023), CEUR-WS. org.

Rajasegaran, J., Jayasundara, V., Jayasekara, S., Jayasekara, H., Seneviratne, S., Rodrigo, R., 2019. Deepcaps: Going deeper with capsule networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10725–10733.

Rehman, M.Z.U., Mehta, S., Singh, K., Kaushik, K., Kumar, N., 2023. User-aware multilingual abusive content detection in social media. Information Processing & Management 60, 103450.

Roy, P.K., 2024. Deep ensemble network for sentiment analysis in bi-lingual low-resource languages. ACM Transactions on Asian and Low-Resource Language Information Processing 23, 1–16.

Sabour, S., Frosst, N., Hinton, G.E., 2017. Dynamic routing between capsules. Advances in neural information processing systems 30.

Saumya, S., Mishra, A.K., 2021. Iiit_dwd@ lt-edi-eacl2021: hope speech detection in youtube multilingual comments, in: Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion, pp. 107–113.

Sharma, M., Arora, G., 2021. Spartans@ lt-edi-eacl2021: inclusive speech detection using pretrained language models, in: Proceedings of the first workshop on language technology for equality, diversity and inclusion, pp. 188–192.

Vijayakumar, P., Prathyush, S., Aravind, P., Angel, S., Sivanaiah, R., Rajendram, S.M., Mirnalinee, T., 2022. Ssn_armm@ lt-edi-acl2022: hope speech detection for equality, diversity, and inclusion using albert model, in: Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion, pp. 172–176.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E., 2016. Hierarchical attention networks for document classification, in: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pp. 1480–1489.

Zhao, Y., Tao, X., 2021. Zyj@ lt-edi-eacl2021: Xlm-roberta-based model with attention for hope speech detection, in: Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion, pp. 118–121.

Ziehe, S., Pannach, F., Krishnan, A., 2021. Gcdh@ lt-edi-eacl2021: Xlm-roberta for hope speech detection in english, malayalam, and tamil, in: proceedings of the first workshop on language Technology for Equality, diversity and inclusion, pp. 132–135.

ORCID Information

Author Name	ORCID ID
Mohammad Zia Ur Rehman	https://orcid.org/0000-0001-6374-8102
Devraj Raghuvanshi	https://orcid.org/0009-0005-9956-2111
Harshit Pachar	https://orcid.org/0009-0009-7543-6013
Chandravardhan Singh Raghaw	https://orcid.org/0009000322689507
Nagendra Kumar	https://orcid.org/0000-0003-4644-3168

Highlights

- A novel framework for hope speech detection in low-resource code-mixed languages.
- A custom CasuleNet integrates two-level probability vectors via a skip connection.
- Capsule-level and hierarchical attention help focus on discriminative information.
- Translated, transliterated Indic, and Roman script versions are taken into account.
- Qualitative and quantitative experiments on three low-resource languages.

Declaration of Interest Statement

Declaration of interests

☒The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: