INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

A Systematic Mapping Study of Code-mixed Embedding Technique

Irfan Mubin Shukri[1], Rohayanti Hassan[2] & Zalmiyah Zakaria[3]
Faculty of Computing,
Universiti Teknologi Malaysia,
81310 UTM Johor Bahru, Johor, Malaysia
Email: Irfan.mubin-2001@graduate.utm.my[1]; rohayanti@utm.my[2]; zalmiyah@utm.my[3]

Submitted: 4/5/2025. Revised edition: 21/5/2025. Accepted: 25/5/2025. Published online: 27/5/2025
DOI: https://doi.org/10.11113/ijic.v15n1.550

***Abstract*—The mix of languages with other languages in a conversation or text is a phenomenon known as code-mixing. This paper presents the existing research on code-mixed embedding techniques, with a focus on identifying research gaps, trends, performance, and methodology approaches. A comprehensive review of 44 peer-reviewed publications from 2016 to 2024 was conducted using online digital libraries. The selected studies were analyzed based on publication trends, language pairs, strengths, and limitations. The result shows a growing interest in the publication trend for papers involving code-mixed embedding techniques, with coverage of several common techniques such as character embedding and word embedding, including Hindi-English language pairs being the most studied. However, several limitations remain in unoptimized models and low-resource language coverage. This mapping provides a structured overview of the field and offers direction for future research in developing more robust and inclusive code-mixed embedding techniques.**

***Keywords*—Embedding, Code-Mixed, Model, Language**

I. INTRODUCTION

Code-mixing has become a global linguistic phenomenon, particularly prominent in multilingual and multicultural societies. Social media platforms act as key enablers of this trend, especially in countries such as Malaysia, Indonesia, and India [1], where users frequently switch between languages in both written and spoken communication. This behavior is reflected in casual conversations on social media and everyday interactions among peers, allowing them to have more linguistic freedom [2]

Code-mixed embedding techniques refer to computational methods used to represent words or phrases from multiple languages in a shared vector space. These may involve bilingual combinations such as English–Spanish [2], Hindi–English [3], Malayalam–English [4], or these techniques address the challenges of multilingual text processing, where words from different languages often co-occur within the same sentence—at the beginning, middle, or end. Traditional word embedding models like Word2Vec [5] and GloVe [6] typically perform poorly with such data, necessitating specialized approaches.

Recent advancements in this space include cross-lingual word embeddings [8], subword-level embeddings[19], and contextualized embeddings [21], which offer improved modeling of mixed-language input for various natural language processing (NLP) tasks. These techniques are instrumental in overcoming structural and semantic mismatches across languages in a single discourse.

The significance of code-mixed embeddings is rooted in their capacity to enhance the robustness and adaptability of NLP systems in multilingual contexts. With the growing prevalence of multilingual communication and the exponential rise of social media interactions [7], traditional NLP models face substantial limitations, reducing their real-world applicability.

These embedding models typically utilize machine learning and deep learning architectures [13], projecting semantically similar multilingual words into neighbouring positions within a shared vector space [37]. Contextualized embeddings, particularly those leveraging transformer models, dynamically interpret code-mixed inputs through attention mechanisms, capturing semantic nuances effectively. Such embeddings serve as essential inputs for NLP tasks including classification, translation, and text generation.

Applications of code-mixed embeddings span various domains such as chatbots, machine translation, sentiment analysis, and speech recognition, notably enhancing multilingual interactions. In customer service, chatbots and

virtual assistants leverage these embeddings to deliver accurate, context-sensitive responses. Social media analytics professionals utilize these techniques to interpret informal, multilingual content, significantly improving sentiment detection and opinion mining. This study aims to investigate and map the current landscape of code-mixed embedding techniques through the following research questions:

1. What are the existing code-mixed embedding techniques, and what are the key approaches, methodologies, and state-of-the-art technologies currently used in research?
2. How do current code-mixed embedding techniques perform, and what are their limitations and strengths?
3. What research gaps exist in code-mixed embedding techniques, and what are the emerging trends in this area?

By addressing these questions, this systematic mapping study seeks to contribute to the development of more effective code-mixed embedding strategies, ultimately enhancing the performance of multilingual NLP applications such as chatbots and conversational agents. The findings offer valuable insights for both researchers and practitioners striving to develop culturally adaptive and linguistically inclusive language technologies. The objectives of this study are as follows:

1. To identify and classify existing code-mixed embedding techniques, including language pairs, methodological approaches, and technological trends
2. To analyze the performance, limitations, and strengths of current techniques and
3. To uncover research gaps and emerging directions in this field.

The remainder of this paper is organized as follows: Section 2 describes the materials and methods used in conducting the systematic mapping, including data sources, inclusion and exclusion criteria, search strategy, and filtering techniques. Section 3 presents the findings and discussion, highlighting trends and identifying the techniques currently utilized in research. Finally, Section 4 concludes the paper by summarizing key discoveries and outlining directions for future work.

II. MATERIALS AND METHOD

A. Planning

The identification of literature done throughout the whole paper was done by going through publications from index literature in seven prestigious databases. Based on Table I, the digital libraries are chosen based on their strong scientific foundation, recognition, and relevance to the study.

TABLE I. ONLINE DATABASE OF LITERATURE

No	Digital Library	URL
1	Scopus	https://www.scopus.com/
2	IEEE Xplore	https://ieeexplore.ieee.org/
3	Web of Science	https://www-webofscience.com/
4	ScienceDirect	https://www.sciencedirect.com/
5	ACM Digital Library	https://dl.acm.org/
6	SpringerLink	https://link.springer.com/

Implementing inclusion and exclusion criteria are crucial to identifying the suitable research papers for analysis and mitigating those that are not suitable.

Inclusion Criteria:
- Studies specifically addressing code-mixed embedding techniques
- Published research article
- Studies that demonstrate implementation of embedding techniques
- Publications from 2016 until 2024

The exclusion criteria, if not stated in detail, are the opposite of the inclusion criteria. Further criteria are:
- Non-peer-reviewed conference presentations
- Studies that do not explicitly focus on code-mixed embedding techniques
- Duplicated or redundant publications
- Studies without full text or abstract
- Publications before 2016

The inclusion criteria in this research are papers that specifically address code-mixed embedding techniques that were published between the years 2016 and 2024 to capture recent developments in the code-mixed embedding technology. Conversely, the study excludes non-peer-reviewed conference presentations that do not focus on code-mixed embedding techniques. This paper also denies duplicated or redundant publications found throughout the databases and does not accept studies that do not come with the full text of the paper.

The search string was created from the context tied to the objectives of the research questions. It is used to accurately identify pertinent phrases or synonyms employed in articles about the latest technology used in code-mixed embedding techniques. Test searches were conducted by adjusting the search string and coming up with the best result for the suitable search term as shown in Table II.

TABLE II. SEARCH STRING

No	Search
S1	(“Code-mixed” OR “Code-switching”) AND
S2	(“embedding”) AND
S3	(“techniques” OR “method”)

The literature searching process begins with the first search (S1) with applied filters, which are stated for each database in Table III. These filters consist of several parameters, such as year, language, and document type.

TABLE III. FILTERS ON DATABASES

Databases	Filter details
Scopus	Year: 2016 - 2024 Subject Area: Computer science, engineering, and mathematics Document type: article
IEEE Explore	Year: 2016 - 2024 Document type: journals

Databases	Filter details
Web of science	Year: 2016 – 2024 Document type: Article Subject area: computer science and engineering Language: English Web of Science categories: Computer Science, Artificial Intelligence, and Computer Science Software Engineering
ScienceDirect	Year: 2016 - 2024 Subject Area: Computer Science and Engineering Document type: Review Articles and Research Articles Language: English
ACM Digital Library	Year: 2016-2024 Document type: research article
SpringerLink	Year: 2016-2024 Subject Area: Natural Language Processing Document type: article, research article, and review article Language: English Disciplines: computer science and engineering

B. Execution

This section explains the process of gathering and filtering publications that were found in the digital libraries as shown in Fig. 1. Search A is the first step in the filtering process that implements the search string of S1 in Table II to all the online databases involved. A total of 8,756 publications were discovered in this initial search. This huge number of publications was mostly presented by SpringerLink, as it has the most publications coming from its digital library. The lowest number of publications comes from IEEE Explore, with a total of 108 publications. The rest of the initial search can be seen in Table IV.

The search is continued by moving on to search B and search C, which implement the expansion of the search string by including S2 and S3 from Table II, respectively. Both searching processes have reduced the number of literature outputs by a significant amount, such as the results from both search B and search C are 1828 and 456, respectively. The progressive reduction in the number of publications from Search A to Search C demonstrates the increasing specificity of the search strings. While Search A casts a wide net to ensure coverage, Search C ensures that only domain-specific and highly relevant works are retained. The decrease reflects a narrowing of scope and a refinement of relevance to the research objectives. The remaining 456 publications from Search C were further filtered based on title, abstract, publication date, and language for inclusion in the final analysis.

TABLE IV. LITERATURE FILTRATION

Database	Search A	Search B	Search C	Inclusion	Exclusion
Scopus	266	41	24	18	6
IEEE Explore	108	17	11	4	7
Web of Science	691	70	27	2	25
ScienceDirect	277	171	169	8	161
ACM Digital Library	360	159	156	12	144
SpringerLink	6,534	1249	69	0	69
Total	8756	1828	456	44	412

A three-stage quality assessment process was implemented to exclude publications that did not meet the inclusion criteria. This process consists of three phases: eliminating duplicates and retracted publications, filtering based on title and abstract and conducting a full-text review. Table V presents the number of articles filtered at the conclusion of the third phase.

The first stage of the process focuses more on eliminating duplicates and retracted publications. From the initial 456 publications that came from the search string filtration, 36 papers were excluded for redundancy purposes. In the second stage, an amount of 420 publications underwent the title and abstract filtering. 353 publications were excluded from the dataset due to irrelevant titles, abstract, or keywords. The papers excluded did not mention anything about code-mixed embeddings or did not have any relevance to the research questions. The last stage of the process involves filtering the overall text of the article to ensure that the context of the article is relevant to the research questions. That leaves us with 44 articles to work with for this study.

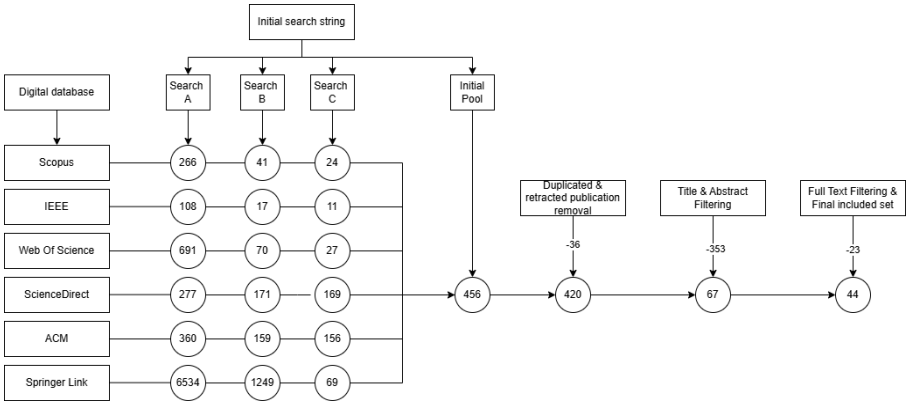

Fig. 1. Publication dataset filtering summary

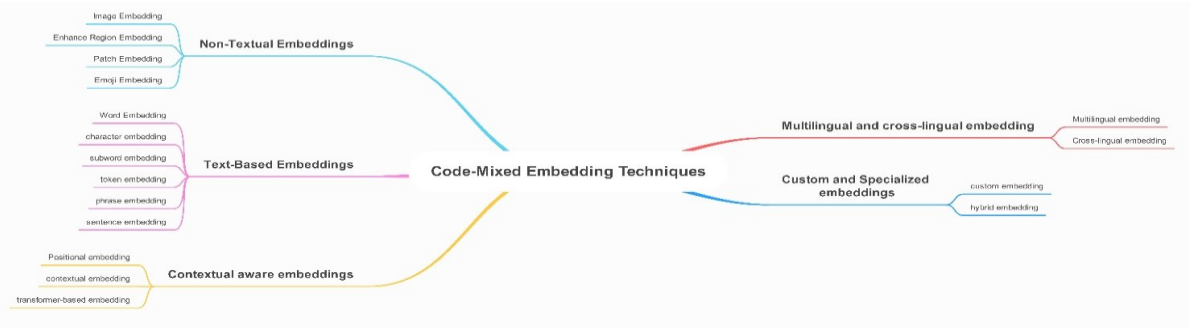

Fig. 2. Taxonomy of code-mixed embedding technique

III. RESULT AND DISCUSSION

This section presents the findings of the primary studies on code-mixed text embeddings. This study divided the discussion into three subsections in response to the respective research questions explained earlier. The first subsection addresses RQ1 regarding current code-mixed embedding techniques, key approaches, methodologies, and state-of-the-art technologies used in research. In the second subsection, this study presents findings concerning RQ2, which investigates the performance of modern embeddings and identifies their strengths and weaknesses. Subsequently, this research provides findings regarding research gaps and emerging trends in RQ3 that could help us visualize the upcoming technologies in the embedding area.

A. (RQ1) What are the existing code-mixed embedding techniques, and what key approaches, methodologies, and state-of-the-art technologies are currently used in research?

This study reviewed 44 primary studies and identified five key categories of embedding techniques developed to address the unique challenges presented by code-mixed language data as illustrated in Fig. 2. These categories are text-based embeddings, contextual-aware embeddings, multilingual and cross-lingual embeddings, custom and specialized embeddings, and non-textual embeddings that represent the major thematic directions in current research. The following discussion synthesizes insights drawn from each category.

1. Text-Based Embeddings

Text-based embeddings, encompassing word-level [8], character-level [9], and subword representations, serve as foundational approaches in early studies on code-mixed data. While these methods offer computational simplicity and interpretable vector structures, their limitations in multilingual environments are well-documented.

Specifically, static embeddings such as GloVe [16], [22] and Word2Vec [10], [11] fail to model semantic shifts across languages and lack contextual sensitivity. Two of the most common learning methods used in Word2Vec are CBOW [13], [16] and skip-gram [12], [17]. Code-mixed scenarios, in which a single token may assume different meanings based on the language context, pose significant challenges to these representations [18]. However, studies indicate that such embeddings remain valuable as input layers or when combined with more advanced models in hybrid systems. Their ability to capture surface-level lexical features makes them useful in preliminary stages of embedding pipelines.

2. Contextual-Aware Embeddings

The emergence of contextualized embedding models has significantly advanced the treatment of code-mixed language. Transformer-based models such as BERT [14], [21], mBERT [15], [19], and BART [23] have been widely adopted due to their capacity to generate dynamic representations based on word context. These models incorporate attention mechanisms and positional encoding, enabling the capture of syntactic and semantic dependencies across multiple languages within a single sentence. Research highlights their effectiveness in resolving lexical ambiguity, improving the interpretability of tokens with multiple meanings depending on linguistic context. Such embeddings are particularly advantageous for high-level NLP tasks, including sentiment analysis, intent classification, and machine translation in code-mixed environments.

3. Multilingual and Cross-Lingual Embeddings

Another significant development involves multilingual and cross-lingual embedding strategies. These embeddings aim to project words from different languages into a shared semantic space, thereby enabling cross-lingual information transfer. Techniques such as bilingual word alignment and multilingual pretraining [25] have demonstrated strong potential in low-resource code-mixed settings, where annotated corpora are scarce. These embeddings also allow for zero-shot and few-shot learning across languages, making them suitable for scalable multilingual NLP systems. Studies have shown that these models enhance generalizability and robustness, particularly in language pairs with overlapping syntactic or lexical features [24].

4. Custom and Specialized Embeddings

Recognizing the limitations of general-purpose models, several studies have explored custom and task-specific embedding strategies. These include embeddings trained exclusively on code-mixed corpora that include emojis or tailored for specific application domains such as healthcare, education, or programming [20]. Customized embeddings offer improved performance by incorporating language-specific cues and domain knowledge. Moreover, hybrid embedding architectures that combining static and contextual models have emerged as effective solutions, balancing efficiency and accuracy. Such approaches are particularly relevant in real-time systems where computational resources and latency are concerns.

5. Non-Textual Embeddings

A growing body of work has begun to explore non-textual embedding techniques, such as those incorporating visual elements (e.g., image [3] or emoji embeddings [30]). These techniques are especially relevant in social media and informal communication, where meaning is often derived from both textual and visual cues. Integrating non-verbal signals enhances semantic grounding and supports more nuanced interpretation of user-generated content. For instance, emojis can disambiguate emotion or sentiment in code-mixed text, while image embeddings can provide additional context for content classification. Although still an emerging area, multimodal embedding strategies show promise for improving performance in real-world, informal, and dynamic communication settings.

Building upon the categorization of embedding strategies, the current landscape of research reveals a broad spectrum of performance outcomes, as summarized in Table V. Notably, several studies employing contextual embeddings and hybrid architectures report high precision, F1-score, and accuracy across diverse code-mixed language pairs, particularly Hindi-English [37], Tamil-English [38], and Malayalam-English [39]. For instance, hierarchical transformer-based models and combinations involving mBERT, BiLSTM, and MuRIL consistently achieve precision rates exceeding 90% [28], with some models such as the BLSTM-IndoBERT-XLMRoBERTa pipeline reaching up to 93.69% [44] accuracy in Indonesian Javanese English contexts.

Character and subword-level embeddings also demonstrate robust performance when integrated with deep learning models such as CNNs, GRUs, and BiLSTMs. These approaches have proven effective in handling orthographic variations and agglutinative structures in regional code-mixed texts, yielding F1-scores as high as 96.67% [4] in multilingual contexts. Conversely, while simpler models based on traditional word embeddings and shallow classifiers like SVM [26] or Random Forest [29] still yield competitive results in specific tasks, their performance tends to plateau, particularly in low-resource and linguistically diverse environments.

Custom and domain-specific embeddings, including image, emoji, and patch-based representations, are emerging as valuable supplements to textual features, though their reported performances remain variable. For instance, multimodal or positional embeddings occasionally underperform, with F1-scores and accuracy metrics falling below 40% [33] in complex multilingual settings. These findings suggest that while contextualized and transformer-based embeddings currently dominate the state-of-the-art, further innovation is required to generalize these models effectively across underrepresented language pairs, domains, and modalities.

TABLE V. CODE-MIXED LANGUAGE, METHOD USED, MACHINE LEARNING/DEEP LEARNING, AND REPORTED BEST PERFORMANCE

Type of performance	Reference	Code-mixed Language	Method Used	Machine Learning/Deep Learning	Reported Best Performance
Precision	[1]	Hindi-English, Spanish-English, Bengali-English, Telugu-English	Word embedding	Hierarchical transformer-based architecture	93.1%
	[2]	Hindi-English	Word embedding	MuRIL, LSTM, TabNet, CNN-LSTM, BiLSTM, SVM, Decision tree, Random forest, XGBoost	93%
	[7]	Hindi-English	Enhance region embedding	Deep Pyramid CNN, Pooled BiLSTM, Disconnected RNN	90.52%
F1 Score	[3]	Hindi-English	Image embedding	BERT, ResNet, CNN, GRU, Capsule network	74.11%
	[4]	Hindi-English, Malayalam-English	Character embedding	LSTM, Bi-LSTM, CNN, mBERT	96.67%
	[6]	Hindi-English, Dutch-English, Spanish-English, Bengali-English, Dutch-Limburgish, Turkish-German	Character embedding	Rule-based, Lexicon-based, CNN, RNN, LSTM, GRU Transformer model	91.03%
	[9]	Hindi-English	Word embedding	Bi-LSTM CNN	93.97%
	[10]	Hindi-English	Word embedding	Bi-LSTM	93.97%
	[12]	Indonesian-English	Word embedding	GRU, Bi-LSTM, SVM ET classifier, DT, RF	92.65%
	[13]	Hindi-English	Character embedding	SVM	93.91%

Type of performance	Reference	Code-mixed Language	Method Used	Machine Learning/Deep Learning	Reported Best Performance
	[14]	French-English, Spanish-English, Dutch-English, Russion-English	Multilingual embedding	XABSA	55.93%
	[15]	Hindi-English	Hybrid embedding	CMHE Self-attention mechanism	77.54%
	[16]	Bangla-English	Word embedding	Logistic Regression, Decision Tree, Support Vector Machine, Extreme Gradient Boosting, 1DConv-LSTM, BERT-Multilingual, Distill BERT, Base BERT	87%
	[19]	Hindi-English	Subword embedding	Mbert, C-HAN, Logistic Regression, Random Forest SVM	96.8%
	[21]	Roman Urdu-English	Transformer-based embedding	SVM, RF, DT, NB, LR, ME CNN, RNN, LSTM, Bi-LSTM, At-Bi-LSTM, TRS	68.4%
	[23]	Hindi-English	Word embedding	BART Seq2seq	25.17%
	[25]	Chinese-english	Token Embedding	BERT models	62%
	[28]	Hindi-English	Subword embedding	HIT FAME	91.5%
	[31]	Bengali-English, Hindi-English, Telugu-English	Word embedding	BiLSTM	42.1%
	[34]	Hindi-English	Word embedding	BERT LiteBERT HingBERT-LID	31%
	[39]	Malayalam-English	Word Embedding	CNN, LSTM, GRU	76.17%
	[41]	Hindi-English	Word embedding	CNN, LSTM, mBERT IndicBERT, MuRIL	78%
	[42]	Tamil-English	Phrase embedding	HAN, BiLSTM BiGRU	93%
	[44]	Indonesian-Javanese-English	Character embedding	BLSTM, IndoBERT mBERT, XLMRoBERTa	93.69%
Accuracy	[5]	Telugu-English	Word embedding	BiLSTM mBERT BERT Transformer model	86%
	[8]	Tamil-English	Custom embedding	LaBSE,, SVM MLP K-Nearest classifier	74%
	[11]	Hindi-English	Word embedding	LSTM Bi-LSTM	95.96%
	[17]	Hindi-English	Word embedding	Linear SVM SVM-RBF Random Forest	85.81%
	[18]	Moroccan Darija-French-English	Word Embedding	CCE, CCF, SPF	62%
	[22]	Hindi-English	Word embedding	CNN-BiLSTM	83.21%
	[24]	Javanese-Indonesian, Sundanese-Indonesian	Word embedding	LSTM, GRU, LSVC, DT, LR, RF	82.2%
	[26]	Indonesian-javanese, Indonesian-sundanese	Word embedding	LSTM, GRU LSVC, LR, RF	86.3%
	[27]	English-hindi	Character embedding	HLSTM	97.49%
	[29]	Hindi-English	Word embedding	mBERT, , XLM-R, MuRIL	52.4%
	[30]	Hindi-English, Chinese-English	Emoji Embedding	M-BERT BiLSTM BiGRU	71.4%
	[32]	Malayalam-English	Subword embedding	LR, SVM XGBoost, BiLSTM CNN	96.4%
	[33]	English-Hindi, English-Bengali, English-Telugu, English-Gujarati, English-Spanish, English-French	Positional Embedding	Reinforced transformer, Utterance Encoder	32.11%
	[36]	Hindi-English, Spanish-English, Tamil-English, Malayalam-English, Kannada-English, Bambara-French	Word embedding	BERT, RoBERTa, mBERT	91.5%

Type of performance	Reference	Code-mixed Language	Method Used	Machine Learning/Deep Learning	Reported Best Performance
	[37]	Hindi-English	Cross lingual embedding	XLMR	71.61%
	[38]	Tamil-English	Sentence embedding	LaBSE SVM	74%
	[40]	English-isiZulu, English-isiXhosa, English-Setswana, English-Sesotho	Word embedding	SLMs RNN	45.9%
	[43]	English-Hindu, English-Telugu, English-Bengali	Character embedding	SVM	87%
Recall	[20]	Not stated	Sentence embedding	CodeBERT CDR4Tag	92.3%
	[35]	Hindi-English	Patch embedding	ALBERT, RoBERTa, CNN	83.2%

B. (RQ2) How do current code-mixed embedding techniques perform, and what are their limitations and strengths?

Fig. 3 shows a heatmap consisting of several embedding types with their performance result based on their respective metric. The heatmap consists of values ranging from 30 to 100 and a colour gradient that indicates the performance level. Darker colour indicates a higher value in performance, meanwhile brighter colour indicates a lower value in performance. There are 4 main performance metrics in the figure : Accuracy, F1 Score, Precision and Recall. The value in the coloured box shown is the maximum performance for the embedding category based on their respective metrics. The Best performance value is achieved by character embedding with the value of 97.49[27] for their accuracy.

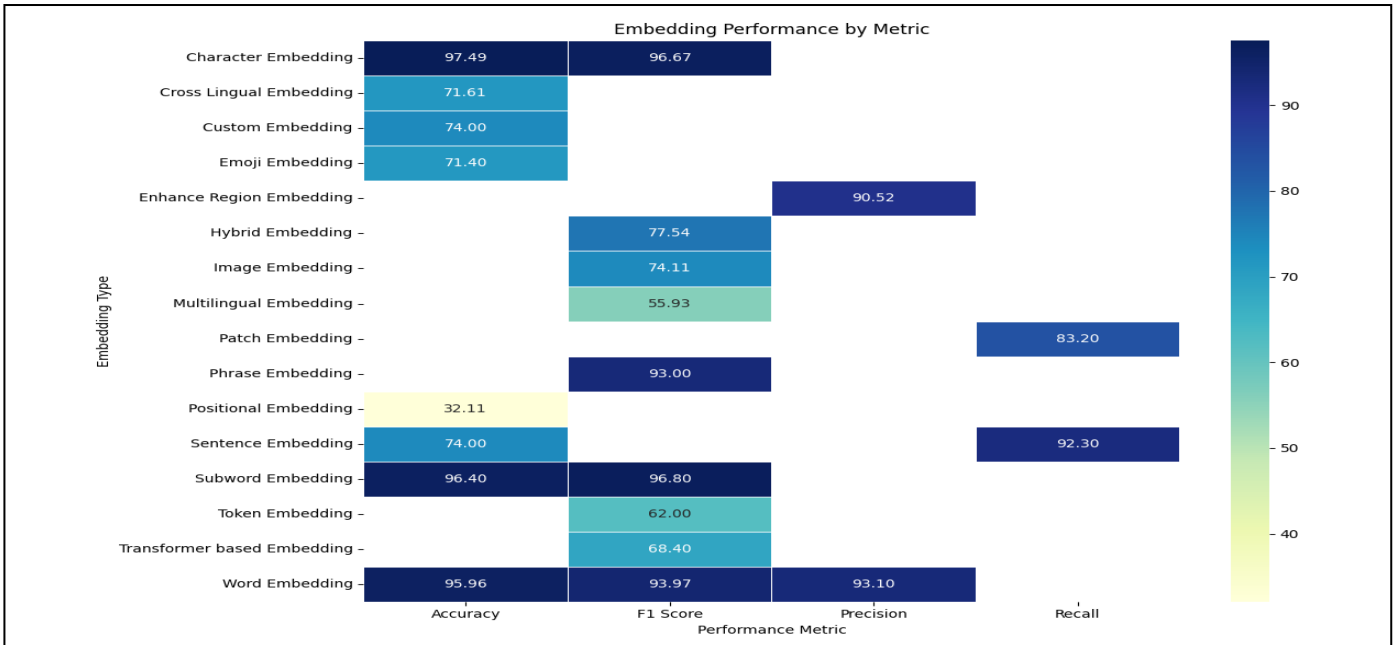

Fig. 3. Heatmap of embedding performance by metric based on embedding type

1. Accuracy

Custom embedding[8], emoji embedding[30], and cross-lingual embedding [37] are some of the examples of embedding techniques that are only available in the accuracy metric. 37.5% of results in the accuracy category achieved 95% and above; meanwhile, 50% of the results reached above 70% in performance. Some of the improvisations that are being implemented in high-performance research are optimizing neural network architecture, using custom training vocabulary, and using pre-trained models. Paper with low performance value are a result of poor quality in the dataset that affects the training process.

2. F1 Score

A total of 21 papers use the F1 score to measure the performance of their model, with about 44% of them performing excellently. The bilingual pairs involved in this figure are Hindi-English [3], Malayalam-English [4], Dutch

English [6], Spanish English [14], Bengali English [31], Dutch-Limburgish [6], Turkish German [6], Indonesian English [12], and Bangla-English [16]. A research paper conducted by [23] has the lowest performance value of 25.17 due to the complexity of sequential labeling tasks and noisy language in the dataset. The best performance for F1 score matrices comes from a research paper conducted by [44] due to the well-equipped and large amount of dataset being used in the research.

3. Precision

Precision metrics consist of 2 word embedding techniques papers [1], [2] and 1 enhanced region embedding techniques paper [7]. The results of the embedding technique performance for the first 2 word embedding techniques papers are 93.1 and 93, respectively, and the performance value for the enhanced region embedding is 90.52. The performance results show a significant strength acquired by both the word embedding and enhanced region embedding for their respective predictions. There are several deep learning architectures identified, such as Hierarchical transformer-based architecture [1], MuRIL architecture [2], Deep Pyramid CNN [3], Pooled BiLSTM, and Disconnected RNN. The code-mixed data includes Hindi-English, Spanish-English, Bengali-English, and Telugu-English.

4. Recall

Recall metrics consist of 2 embedding techniques [20], [35] that show a high value in performance. The 92.3 performance score from sentence embedding by [20] is a product of the utilization of the CDR4Tag method that optimizes deep semantic correlations. A research paper conducted by [35] has a high-performance value of 83.2, which involves the Hindi-English language pair and uses an optimized framework that helps in identifying potential misogynistic content

C. What research gaps exist in code-mixed embedding techniques, and what are the emerging trends in this area?

The study of code-mixed embedding techniques has gained significant traction in recent years, driven by the increasing prevalence of multilingual and code-mixed communication on digital platforms. However, despite the progress, several research gaps remain, and emerging trends are shaping the future of this field.

Fig. 4 complements this by showcasing the diversity of embedding techniques over the same period, with word embedding being the most dominant like shown by [1], [2], [9], [10], and [11], appearing consistently across all years. Character embedding [4], [13], [27] and subword embedding [19], [28], [32] also show steady usage, particularly in 2022–2024, reflecting their effectiveness in handling the morphological complexity of code-mixed text. Emerging techniques like transformer-based embedding [21], patch embedding [36], and contextual embedding [35] appear in later years (2023–2024), signaling a shift towards more advanced, context-aware methods. However, techniques such as emoji embedding [30], positional embedding [33], and cross-lingual embedding [38] remain underexplored, each appearing only once, indicating potential research gaps in leveraging these embeddings for code-mixed scenarios.

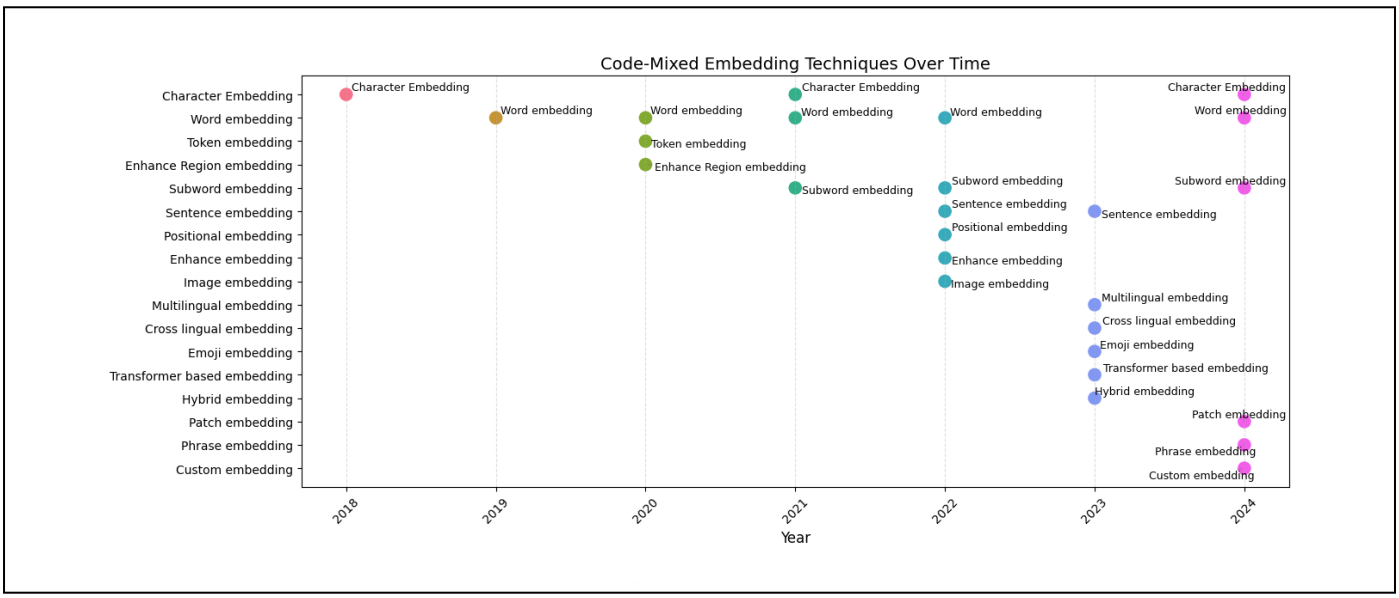

Fig. 4. Types of code-mixed embedding techniques papers published from 2018 to 2024

Fig. 5 reveals a fluctuating yet overall upward trend in publications on code-mixed embedding techniques between 2018 and 2024. Starting with 2 publications in 2018, the number dips to 1 in 2019, rises sharply to 7 in 2020, and peaks at 14 in 2022, before stabilizing at 8 publications per year in 2023 and 2024. This trajectory indicates growing interest in the field, particularly around 2022, likely driven by advancements in deep learning and transformer-based models, as evidenced by the references.

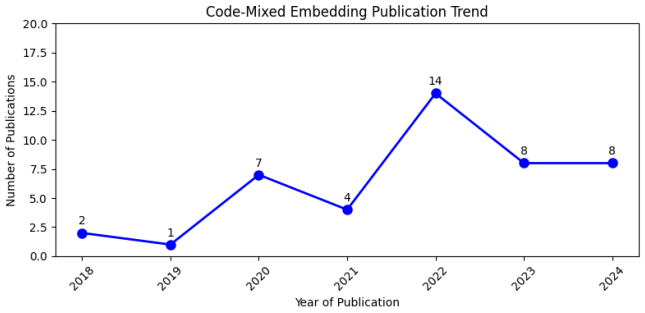

Fig. 5. Publication trend for code-mixed embedding technique papers

The insights from Figs. 4 and 5 highlight several research gaps and emerging trends. The dominance of word and character embeddings suggests a reliance on traditional methods, which may not fully capture the semantic nuances of code-mixed text, especially in diverse language pairs like Roman Urdu-English or Tamil-English. The limited use of emoji embedding, despite its relevance in social media contexts, points to a gap in addressing multimodal code-mixed data, where emojis play a significant role in sentiment and intent. Similarly, the scarcity of positional and cross-lingual embeddings indicates an underexplored area for improving context retention and cross-lingual transfer in code-mixed settings, particularly for low-resource languages like isiZulu or Sesotho [41].

IV. CONCLUSION

This systematic mapping study has analyzed current methodologies, performance metrics, and trends in code-mixed embedding techniques across various language pairs. Transformer-based and contextual embeddings have emerged prominently between 2023 and 2024, notably through advanced models like BERT and mBERT, demonstrating superior handling of mixed-language semantics and achieving high accuracy. Despite their success, variability in model performance highlights opportunities for further optimization tailored specifically for code-mixed scenarios. The observed peak in research interest around 2022, followed by stabilization, indicates that the field might be reaching maturity, suggesting a need for novel approaches such as hybrid or multimodal embeddings. Future studies should prioritize exploring these innovative methods, particularly in low-resource language contexts, and investigate multimodal data integration to enhance the robustness of code-mixed embedding techniques.

ACKNOWLEDGEMENT

This work was supported by the Ministry of Higher Education, Malaysia (MOHE) under the Fundamental Research Grant Scheme (FRGS), FRGS/1/2024/ICT02/UTM/02/11.

CONFLICTS OF INTEREST

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1] S. Maddu and V. R. Sanapala. (2022). A survey on NLP tasks, resources and techniques for low-resource Telugu-English code-mixed text. *ACM Trans. Asian Low-Resour. Lang. Inf. Process*. Doi: 10.1145/3695766.

[2] A. Sengupta, T. Suresh, M. S. Akhtar, and T. Chakraborty. (2022). A comprehensive understanding of code-mixed language semantics using hierarchical transformer. *arXiv:2204.12753.* Doi: 10.48550/arXiv.2204.12753.

[3] A. Chopra, D. K. Sharma, A. Jha, and U. Ghosh. (2023). A framework for online hate speech detection on code-mixed Hindi-English text and Hindi text in Devanagari. *ACM Trans. Asian Low-Resour. Lang. Inf. Process, 22*(5), 1–21. Doi: 10.1145/3568673.

[4] K. Maity, P. Jha, S. Saha, and P. Bhattacharyya. (2022). A multitask framework for sentiment, emotion and sarcasm aware cyberbullying detection from multi-modal code-mixed memes. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Madrid Spain: ACM, 1739–1749. Doi: 10.1145/3477495.3531925.

[5] S. Dowlagar and R. Mamidi. (2021). A survey of recent neural network models on code-mixed Indian hate speech data. *Forum for Information Retrieval Evaluation, Virtual Event* India: ACM, 67–74. Doi: 10.1145/3503162.3503168.

[6] A. F. Hidayatullah, A. Qazi, D. T. C. Lai, and R. A. Apong. (2022). A systematic review on language identification of code-mixed text: techniques, data availability, challenges, and framework development. *IEEE Access, 10*, 122812–122831. Doi: 10.1109/ACCESS.2022.3223703.

[7] A. Khandelwal and N. Kumar. (2020). A unified system for aggression identification in English code-mixed and uni-lingual texts. *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, Hyderabad India: ACM, 55–64. Doi: 10.1145/3371158.3371165.

[8] G. G L, K. Swaminathan, D. Krishnakumar, T. D, and B. B. (2024). Abusive comment detection in tamil code-mixed data by adjusting class weights and refining features. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.,* 3664619. Doi: 10.1145/3664619.

[9] S. Shekhar, D. K. Sharma, and M. M. Sufyan Beg. (2020). An effective Bi-LSTM word embedding system for analysis and identification of language in code-mixed social media text in English and Roman Hindi. *CyS, 24*(4). Doi: 10.13053/cys-24-4-3151.

[10] S. Shekhar, D. K. Sharma, and M. M. Sufyan Beg. (2019). An effective cybernated word embedding system for analysis and language identification in code-mixed social media text. *KES*, *23*(3), 167–179. Doi: 10.3233/KES-190409.

[11] S. Shekhar, D. K. Sharma, D. K. Agarwal, and Y. Pathak. (2022). Artificial Immune Systems-Based Classification Model for Code-Mixed Social Media Data. *IRBM*, *43*(2), 120–129. Doi: 10.1016/j.irbm.2020.07.004.

[12] A. Suciati and I. Budi. (2020). Aspect-based sentiment analysis and emotion detection for code-mixed review. *IJACSA,* 11(9). Doi: 10.14569/IJACSA.2020.0110921.

[13] P. V. Veena, M. Anand Kumar, and K. P. Soman. (2018). Character Embedding for Language Identification in Hindi-English Code-mixed Social Media Text. *CyS, 22*(1). Doi: 10.13053/cys-22-1-2775.

[14] N. Lin, Y. Fu, X. Lin, D. Zhou, A. Yang, and S. Jiang. (2023). CL-XABSA: Contrastive learning for cross-lingual aspect-based sentiment analysis. IEEE/ACM Trans. Audio Speech Lang. Process, *31*, 2935–2946. Doi: 10.1109/TASLP.2023.3297964.

[15] S. Mundra and N. Mittal. (2023). CMHE-AN: Code mixed hybrid embedding based attention network for aggression identification in hindi english code-mixed text. *Multimed Tools Appl, 82*(8), 11337–11364. Doi: 10.1007/s11042-022-13668-4.

[16] M. Tareq, Md. F. Islam, S. Deb, S. Rahman, and A. A. Mahmud. (2023). Data-augmentation for bangla-english code-mixed sentiment analysis: Enhancing cross linguistic contextual understanding. *IEEE Access*, *11*, 51657–51671. Doi: 10.1109/ACCESS.2023.3277787.

[17] K. Sreelakshmi, B. Premjith, and K. P. Soman. (2020). Detection of hate speech text in hindi-english code-mixed data. *Procedia Computer Science*, *171*, 737–744. Doi: 10.1016/j.procs.2020.04.080.

[18] M. Aghzal and A. Mourhir. (2021). Distributional word representations for code-mixed text in Moroccan Darija. *Procedia Computer Science, 189*, 266–273. Doi: 10.1016/j.procs.2021.05.090.

[19] A. Sengupta, S. K. Bhattacharjee, Md. S. Akhtar, and T. Chakraborty. (2022). Does aggression lead to hate? Detecting and reasoning offensive traits in hinglish code-mixed texts. Neurocomputing, *488*, 598–617. Doi: 10.1016/j.neucom.2021.11.053.

[20] L. Li, P. Wang, X. Zheng, Q. Xie, X. Tao, and J. D. Velásquez. (2023). Dual-interactive fusion for code-mixed deep representation learning in tag recommendation. *Information Fusion*, *99*, 101862. Doi: 10.1016/j.inffus.2023.101862.

[21] A. Ilyas, K. Shahzad, and M. Kamran Malik. (2023). Emotion detection in code-mixed Roman Urdu-English Text. *ACM Trans. Asian Low-Resour. Lang. Inf. Process, 22*(2), 1–28. Doi: 10.1145/3552515.

[22] T. T. Sasidhar, P. B, and S. K. P. (2020). Emotion Detection in Hinglish (Hindi+English) code-mixed social media text. *Procedia Computer Science*, *171*, 1346–1352. Doi: 10.1016/j.procs.2020.04.144.

[23] K. Maity, R. Jain, P. Jha, and S. Saha. (2024). Explainable cyberbullying detection in Hinglish: A generative approach. *IEEE Trans. Comput. Soc. Syst., 11*(3), 3338–3347. Doi: 10.1109/TCSS.2023.3333675.

[24] E. W. Pamungkas, D. Purworini, D. Priyawati, and R. R. B. Chasana. (2023). Exploring the impact of lexicon-based knowledge transfer for hate speech detection in Indonesia code-mixed languages. *Proceedings of the 2023 7th International Conference on Natural Language Processing and Information Retrieval*, Seoul Republic of Korea: ACM. 85–90. Doi: 10.1145/3639233.3639247.

[25] T. Tang, X. Tang, and T. Yuan. (2020). Fine-tuning BERT for multi-label sentiment analysis in unbalanced code-switching text. IEEE Access, *8*, 193248–193256. Doi: 10.1109/ACCESS.2020.3030468.

[26] E. W. Pamungkas, A. Fatmawati, and F. D. Salam. (2022). Hate speech detection on indonesian social media: A preliminary study on code-mixed language issue. *Proceedings of the 2022 6th International Conference on Natural Language Processing and Information Retrieval,* Bangkok Thailand: ACM, 104–109. Doi: 10.1145/3582768.3582771.

[27] S. Shekhar, H. Garg, R. Agrawal, S. Shivani, and B. Sharma. (2023). Hatred and trolling detection transliteration framework using hierarchical LSTM in code-mixed social media text. Complex Intell. Syst., *9*(3), 2813–2826. Doi: 10.1007/s40747-021-00487-7.

[28] A. Sengupta, S. K. Bhattacharjee, T. Chakraborty, and Md. S. Akhtar. (2021). HIT-A hierarchically fused deep attention network for robust code-mixed language representation. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 4625–4639. Doi: 10.18653/v1/2021.findings-acl.407.

[29] A. Bagora, K. Shrestha, K. Maurya, and M. S. Desarkar. (2022). Hostility detection in online Hindi-English code-mixed conversations. *14th ACM Web Science Conference 2022,* Barcelona Spain: ACM, 390–400. Doi: 10.1145/3501247.3531579.

[30] T. Tang and K. Nongpong. (2023).“Impact of emojis in emotion analysis on code-mixed text. *Proceedings of the 2023 7th International Conference on Natural Language Processing and Information Retrieval,* Seoul Republic of Korea: ACM, 25–30. Doi: 10.1145/3639233.3639342.

[31] S. N. Bhattu, S. K. Nunna, D. V. L. N. Somayajulu, and B. Pradhan. (2020). Improving code-mixed POS tagging using code-mixed embeddings. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.,* 19(4), 1–31. Doi: 10.1145/3380967.

[32] L. K. Dhanya and Dr. K. Balakrishnan. (2024). Integrating hybrid neural networks and domain-specific embeddings for detecting hate content in code mixed social media comments. *JISIS, 14*(3), 316–329. Doi: 10.58346/JISIS.2024.I3.019.

[33] G. V. Singh, M. Firdaus, Shambhavi, S. Mishra, and A. Ekbal. (2023). Knowing what to say: Towards knowledge grounded code-mixed response generation for open-domain conversations. *Knowledge-Based Systems, 249*, 108900, Doi: 10.1016/j.knosys.2022.108900.

[34] G. Takawane, A. Phaltankar, V. Patwardhan, A. Patil, R. Joshi, and M. S. Takalikar. (2023). Language augmentation approach for code-mixed text classification. *Natural Language Processing Journal, 5,* 100042. Doi: 10.1016/j.nlp.2023.100042.

[35] A. Singh, D. Sharma, and V. K. Singh. (2024). MIMIC: Misogyny identification in multimodal internet content in Hindi-English code-mixed language. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.,* 3656169. Doi: 10.1145/3656169.

[36] K. R. Mabokela, T. Celik, and M. Raborife. (2023). Multilingual sentiment analysis for under-resourced languages: A systematic review of the landscape. *IEEE Access, 11*, 15996–16020. Doi: 10.1109/ACCESS.2022.3224136.

[37] S. Ghosh, A. Priyankar, A. Ekbal, and P. Bhattacharyya. (2023). Multitasking of sentiment detection and emotion recognition in code-mixed Hinglish data. *Knowledge-based Systems, 260,* 110182. Doi: 10.1016/j.knosys.2022.110182.

[38] K. Swaminathan, D. K, G. G L, T. Durairaj, and B. B. (2022). PANDAS@Abusive comment detection in tamil

code-mixed data using custom embeddings with LaBSE. *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, Dublin, Ireland: Association for Computational Linguistics, 112–119. Doi: 10.18653/v1/2022.dravidianlangtech-1.18.

[39] S. Thara and P. Poornachandran. (2022). Social media text analytics of Malayalam–English code-mixed using deep learning, *J Big Data, 9*(1), 45. Doi: 10.1186/s40537-022-00594-3.

[40] E. Van Der Westhuizen and T. R. Niesler. (2019). Synthesised bigrams using word embeddings for code-switched ASR of four South African language pairs. *Computer Speech & Language*, *54*, 151–175. Doi: 10.1016/j.csl.2018.10.002.

[41] D. Sharma, A. Singh, and V. K. Singh. (2024). THAR-targeted hate speech against religion: A high-quality Hindi-English code-mixed dataset with the application of deep learning models for automatic detection. *ACM Trans. Asian Low-Resour. Lang. Inf. Process,* 3653017. Doi: 10.1145/3653017.

[42] V. S. Devi, S. Kannimuthu, and A. K. Madasamy. (2024). The effect of phrase vector embedding in explainable hierarchical attention-based tamil code-mixed hate speech and intent detection. *IEEE Access*, *12*, 11316–11329. Doi: 10.1109/ACCESS.2024.3349958.

[43] A. K. Madasamy and S. K. Padannayil. (2023). Transfer learning based code-mixed part-of-speech tagging using character level representations for Indian languages. *J Ambient Intell Human Comput,* 14(6), 7207–7218. Doi: 10.1007/s12652-021-03573-3.

[44] A. F. Hidayatullah, R. A. Apong, D. T. C. Lai, and A. Qazi. (2024). Word level language identification in Indonesian-Javanese-English code-mixed text. *Procedia Computer Science, 244*, 105–112. Doi: 10.1016/j.procs.2024.10.183.