JOIV : Int. J. Inform. Visualization, 10(1) - January 2026 57-65

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Enhancing Stand-up Comedy Summarization Using Cohesion and Coherence with Deep Learning

Supriyono [a,c], Aji Prasetya Wibawa [a,*], Suyono [b], Fachrul Kurniawan [c]

[a] *Department of Electrical Engineering and Informatics, Faculty of Engineering, Universitas Negeri Malang, Malang, Indonesia*
[b] *Department of Indonesian Literature, Faculty of Letters, Universitas Negeri Malang, Malang, Indonesia*
[c] *Informatics Engineering, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang, Indonesia*
Corresponding author: [*]aji.prasetya.ft@um.ac.id

***Abstract*—Summarizing stand-up comedy scripts presents a unique problem because of the complex interplay of language, timing, and cultural references that define humor. Traditional summarization techniques sometimes fail to preserve the punchlines, irony, and narrative coherence essential to comedic storytelling. This study introduces a hybrid methodology that integrates linguistic cohesion and coherence methodologies with a deep learning framework employing Long Short-Term Memory (LSTM) networks and Word2Vec embeddings. Our objective is to develop a summarization model that captures both the structural coherence and humorous essence of stand-up performances in a more succinct format. We employed a bilingual dataset of English and Indonesian comic scripts for text preprocessing, embedding, and sequence modelling during system training. The assessment employed metrics such as accuracy, precision, recall, and F1-score, together with a qualitative analysis of the generated summaries. The results demonstrate that the proposed methodology significantly improves humor retention and logical coherence compared to conventional methods. The accuracy increased from 0.65 to 0.83 after training, and validation measures confirmed strong generalization. This approach offers considerable promise for content regulation, customized recommendations, and AI-augmented entertainment solutions. This research improves natural language comprehension in informal, culturally diverse genres by preserving the distinctive structure and intent of comedic language.**

***Keywords*—Stand-up comedy; text summarization; humor retention; cohesion and coherence; natural language processing.**

Manuscript received 19 Nov. 2024; revised 20 Sep. 2025; accepted 23 Nov. 2025. Date of publication 30 Jan. 2026.

I. INTRODUCTION

Stand-up comedy scripts are particularly challenging for automatic summarization due to their unique intricacies and distinctive frameworks. Unlike conventional prose, these scripts utilize timing, cultural references, and subtlety, which would be lost in a condensed version [1], [2]. The equilibrium between humor extraction and retention, with summarization methodologies in NLP, remains an unresolved issue [3], [4]. Contemporary methods face significant constraints in preserving both the humor and the creative elements of the text, particularly in sophisticated content-filtering or summarization tasks.

The field of Natural Language Processing (NLP), along with advancements in deep learning technologies such as Word2Vec and contextual embeddings, has significantly reshaped the landscape of text summarization. These techniques enhance the understanding of lexical relationships and the contextual meaning of specific phrases [5], [6]. While embedding methods have demonstrated their effectiveness in maintaining semantic coherence and context, they often require tailoring when dealing with humor. In stand-up comedy texts, even slight differences between similar phrases can drastically affect the impact of the summary.

This study addresses the complexities of summarizing stand-up comedy content by employing deep learning models in conjunction with techniques that enhance cohesion, coherence, and narrative flow [7], [8], [9]. The novelty of this approach lies in integrating linguistic methods with embedding models, such as Word2Vec, to better preserve humorous elements within text summaries. The proposed hybrid system combines text simplification with humor retention, emphasizing the need to sustain logical structure, textual coherence, and comedic effect.

This issue builds on prior research that highlighted the difficulty of preserving humor in summaries, primarily because humor depends on subtle linguistic cues and

contextual nuances [10], [11], [12], [13]. Unlike traditional summarization techniques, which often fall short in this regard, our approach fuses linguistic strategies with deep learning to improve the retention of humorous content [14], [15]. The key innovation of this method lies in the synergy between sophisticated Word2Vec embeddings and linguistic rules, significantly enhancing the semantic interpretation of humor in textual data.

This study contributes a summarizing framework that preserves the humorous and linguistic integrity of stand-up comedy materials. The model improves comedy detection accuracy, as evidenced by the metrics obtained during both the validation and training phases. The results enrich NLP and content analysis by demonstrating advanced machine learning-based processing of comedic material. This approach opens new avenues for refining content curation, entertainment-oriented recommendation systems, and automated content moderation, where humor plays a crucial role.

II. Materials and Method

The study drew on a curated set of publicly archived stand-up comedy scripts, ensuring sufficient variety in joke structure and delivery. Selected texts represent observational, Anecdotal, and situational humor from both English- and Indonesian-speaking performers [16][17]. Their mix of straightforward and intricate phrasing provides the linguistic range needed to build and assess the summarization system and evaluate the model.

The primary dataset used in this research is the Standup Comedy Data, which can be accessed via DOI: https://data.mendeley.com/datasets/85xgdr7cc7/2 [18]. This dataset provides numerous stand-up scripts annotated with humor elements, making it a potential candidate for future work on humor comprehension and detection [19], [20]. The dataset is critical because it captures scripts from multiple comedians with distinctive styles, languages, and frameworks, thereby providing a solid basis for assessing the model's performance in various comedic contexts.

The preparatory step involved manually removing non-verbal elements, such as laughter or applause, and prominent characters, as well as excessive punctuation that would obstruct the summarization process [21], [22]. To ensure that all relevant details were captured, lemmatization alongside tokenization was applied to basic word forms. To improve the model's efficiency, stop words were removed.

Lemmatization and tokenization were employed to convert words to their basic forms, while the elimination of stop words guaranteed the retention of only pertinent information for a specific model [23]. To improve comprehension of relationships and the context of word usage in the comic scripts, Word2Vec embeddings were employed [24]. Word2Vec was employed due to its capability to convert words into numerical vectors while preserving semantic significance, which is crucial for addressing complex language components such as humor [16]. A model was developed utilizing a hybrid deep learning framework that integrates coherence and cohesion methodologies with embeddings generated by deep learning [25], [26]. The architecture used a Long Short-Term Memory (LSTM) network, which considers the sequential properties of text material to maintain the logical coherence and humor of the summary.

We allocated 80% of the dataset for training and 20% for validation. During the training phase, the model's retention of humor in summarization was monitored using accuracy and loss metrics. A confusion matrix was used to evaluate the model's proficiency in distinguishing between humorous and non-humorous material, as indicated by true positives, true negatives, false positives, and false negatives [27], [28]. The supplementary assessment of model performance encompassed precision, recall, and F1-score, emphasizing a thorough review of appropriateness, humor summarization, and the cohesiveness and coherence within and between frames.

All experiments were conducted in Python, utilizing the TensorFlow and Keras frameworks for deep learning, together with Gensim for the construction of Word2Vec embeddings [29], [30]. The operation was performed on a high-performance GPU, significantly reducing training and evaluation time.

Figure 1 of the book illustrates the methodology for preserving the humor and coherence of stand-up comedy scripts during summarization. This approach begins by compiling stand-up comedy scripts in both English and Indonesian, with a focus on specific comedic styles. The data undergoes preprocessing, including text sanitization, tokenization, and lemmatization, for analytical purposes. The semantic links among the residual words after lemmatization are represented using Word2Vec embeddings, which are incorporated into a hybrid deep-learning model based on LSTM networks [31]. This method aims to encapsulate screenplays while preserving their semantics, including humor [32]. The model is subjected to training and validation, with its performance assessed by accuracy, loss metrics, and confusion matrix analysis for differentiating humor content [33], [34]. The objective is to develop a summarization technique that preserves humor while ensuring the text remains logical and consistent. The text structure must be coherent while keeping the humor contained.

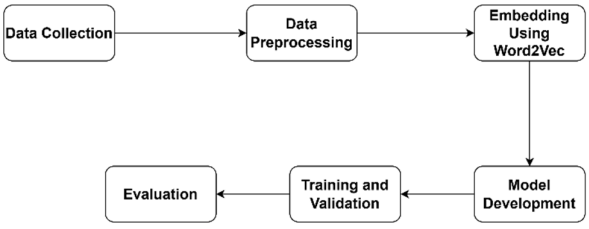

Fig. 1 Research Methodology

Developing a summarization model for stand-up comedy scripts uses a systematic approach indicated in the research methodology [35], [36]. Data collection, which initiates the process, focuses on script collection, primarily from various stand-up comedy genres in English and Indonesian. Each dataset undergoes data cleansing procedures, including tokenization, lemmatization, and the removal of unrelated components. The data is examined using a hybrid deep learning model based on LSTM networks and Word2Vec embeddings, which capture linguistic subtleties [37], [38], [39]. This process is undertaken to maintain coherence and

cohesiveness, ensuring that the summarization is accurate and the text's comedic value is preserved. Preserving the logic flow and humor of the original texts is vital to the accuracy of the summaries, and this process ensures the allure of the texts is retained.

A. Data Collection

Collecting stand-up comedy scripts constitutes the primary data-gathering stage for this study, preceding all other techniques. Most scripts were located online, a choice that introduced minor fluctuations in timing, laughter, and word choice. The data set deliberately spans observational, situational, and anecdotal jokes, allowing scholars to analyze humor across several dimensions [40]. Nearly every performer relies on observational comedy, yet only a small pool of English and Indonesian scripts currently exists [41]. To build a richer, bilingual language model that mirrors real-life joking, researchers must add jokes rooted in local customs, and that material has already been gathered in both target languages.

This step establishes the specific rules for evaluating the value and relevance of each piece of data to the research question. The researchers categorize performance texts by audience popularity, language difficulty, and the complexity of jokes, puns, timing-dependent bits, or culture-bound references [42]. Because the data range widely, a varied mix of comedians was chosen to show different delivery styles and pacing that could strengthen the learning model. Adding scripts by top acts from both countries lets the team spot both differences and common threads in how each culture crafts and reads humor, serving as key comparative markers. By using publicly available material, the approach also respects ethical guidelines regarding access to content.

Working exclusively with public-domain scripts substantially reduced the risk of copyright infringement and helped the project meet key ethical guidelines. At this stage, creating new datasets runs in parallel with gathering material. This setup enables researchers to investigate the characteristics of a specific comedy genre while developing an observational model of the forms laughter can take [43], [44]. The size and variety of the database now serve as a sturdy platform for both the planned analyses and a range of future investigations.

B. Data Preprocessing

Data preprocessing follows data collection and prepares raw information for subsequent analytical modeling. The first task is to clean the textual data by removing performance cues, such as stage directions, applause, and audience laughter, which, although necessary for live shows, do not add to the linguistic humor and can obscure the analysis's goal [45]. Once this clutter is stripped away, stemming and tokenization, which are reported to yield cleaner units of meaning, become feasible. Words are thus recast into basic grammatical forms that function as psychological tokens rather than fully inflected terms.

Another preprocessing step eliminates minor words and emotional markers that clutter summaries yet contribute little to the comedy, notably stop words like 'and', 'the', and 'but'. By filtering these items, the process concentrates the algorithm's attention on heftier linguistic elements that drive the jokes themselves across the script [46]. Still, the procedure is careful to safeguard the distinctive phrases, catchphrases, sarcastic gestures, and similar motifs, ensuring they survive the cleaning and remain available for later analysis.

Maintaining a clear framework of humor during data preparation ensures that any subsequent summary remains accurate and focused. Preprocessing, therefore, balances simplicity of handling with the richness of information, aiming to reduce background noise without losing key nuances. Excessive trimming risks flattening the material, so that subtle comic cues—such as timing, wordplay, and audience reaction—are lost. Careful cleaning thus provides the foundation on which the later model learns to recognize, evaluate, and even admire the layered architecture of humor.

C. Embedding Using Word2Vec

Once the data has been cleaned and standardized, the subsequent phase involves translating the textual material into a numerical format using machine-learning techniques, specifically Word2Vec embeddings. Word2Vec converts each word in a document into an m-dimensional vector that reflects its meaning as inferred from the surrounding text. [47]. By working with these dense vectors, the model becomes sensitive to subtle linguistic ties that govern humor, including synonymy, metaphor, and shifting contextual cues. Because of this vector-based representation, Word2Vec enhances the system's ability to identify jokes hidden within the scripts.

The embedding process effectively compresses a wide range of comical features into a single vector representation, allowing the model to map words and phrases to their intended meanings rather than treating them as isolated tokens. For example, the algorithm can learn standard comedic moves, such as set-up followed by punchline, alongside culturally specific terms that travel across different joke traditions. Armed with these enriched vectors, the classifier discerns how words relate within humorous discourse far better than it could if it relied solely on bag-of-words vectors. This deeper insight is crucial for humor detection because jokes often hinge on inference and expectation rather than the dense linguistic redundancy found in literary texts.

Training the Word2Vec architecture also drew on jokes dating back to Lakota traditions, enabling the algorithm to recognize power dynamics, identity play, and irony that traverse different cultural contexts [48]. The Word2Vec vectors serve as follows:

- They provide the model's most basic layer of language representation.
- They do so while being mindful of the cultural and linguistic cues in the input.
- and they influence how well the system grasps jokes, brevity, and gist.

D. Model Development

Once the embeddings have been put in place, there remains only one step: model development. This stage aims to construct a hybrid deep-learning model comprising Long Short-Term Memory [49]. LSTM networks are best suited for summarizing humorous texts since they can preserve context over time and the order of information presentation. This

detail is significant in stand-up comedy scripts because the order and incorporation of jokes at certain moments contain the humor itself. Owing to the strategic role the LSTM played in examining and retaining these comic episodes, the text could be summarized while still maintaining a reasonable order of ideas.

The hybrid model also builds a cohesive, coherent approach in every summary, ensuring the meaning is not compromised. Stand-up jokes are structured so that there is perfect continuity in the story, with all the jokes building on one another and the punch lines depending on the context. This means that purposes demonstrating the relevant linguistic aspects of this cohesion show that the crucial phrase repetitive language or literary parallelism can keep up for passages while shortening them. As a result of this choice of model, the model's performance improves at summarizing information while retaining humor.

Also, the system can run scripts in both languages and is built with multilingual support. That is because this deep learning model integrates both sequential and hierarchical features for sequential internal Hidden Markov Models. These components also help ensure that the model, while brief, can still convey the required amount of humor, as anticipated, because the summary is dense.

E. Training and Validation.

This means the model will learn on 80% of the data set, and 20% will be used for validation. With this split, an institution can put the model to test generalization with data that has not been part of the learning process, and yet uses a significant portion of the available data for learning [50]. Training processes use backpropagation and optimize the model parameters to reduce the difference between the accurate and generated summaries. Other parameters, such as accuracy and loss, are measured as training progresses to ensure the resulting model heads towards a correct or optimal state.

The validation stage of any model is a good way to check for overfitting, a common problem in deep learning where the model performs well on the training set but poorly on new data. During the validation stage, where performance is assessed on the validation set, it becomes evident that actions must be taken to prevent overfitting; for example, dropout layers can be added, or regularization can be employed. In addition, this stage enables the establishment of the first half of a cup by presenting a model for humor summation, which helps adjust timing and structure in comedic content.

Also, during training and validation, there is a rigorous check on how the model is preserved across many runs and how much ridicule can be consolidated into laughs. Each epoch allows the model to learn from the data more effectively and gradually improves its ability to separate relevant from irrelevant information in a humorous context. In summary, the model's training is a cycle in which its summarization is improved, resulting in educational highlights that align with the original comedy content.

F. Evaluation

The assessment phase includes evaluating the model's ability to accurately detect and summarize humorous content utilizing several metrics, such as precision, recall, and F1-score. There is also a confusion matrix that tabulates and represents the number of true positives, true negatives, false positives, and false negatives classifications made by the model [51]. This matrix can help determine the model's strengths and identify areas for improvement, such as distinguishing between fine humor teasing and plain text.

The precision vs. recall trade-off is invaluable for evaluating the model's humor detection. The measure of precision seeks the accuracy of the humorous predictions. In contrast, in the recall measure, the model treats humorous events within the story as all instances of humor in the text. These measurements provide a comprehensive perspective on the model's performance, highlighting the areas where humorous summation was performed accurately. An F1-score combines precision and recall into a single score for how effective the model is.

Last but certainly not least, qualitative analysis is conducted to determine whether the summaries retain the humor. During the evaluation phase, summarizing example summaries allows us to assess the presence of humor in the model's final format by checking whether crucial sarcasm and punchlines are retained in the humor's condensed structure. In this way, a comprehensive evaluation process incorporating both numerical and subjective assessment techniques was sufficient to demonstrate that the model is adequate at producing quantitative and acceptable stand-up comedy summaries.

III. Results and Discussion

Stand-up comedy texts present a unique and complex challenge for analysis due to their distinctive language framework and often contextual and implicit humor. Summarizing these texts requires a unique approach that maintains the comedic elements while simplifying the content. This challenge becomes especially apparent when comic language needs to be abridged without compromising its essence, a task that engages the audience in the complexity of stand-up comedy analysis.

This study highlights that cohesion and coherence are essential for generating text summaries that maintain a funny quality. Cohesion, the linkage of textual elements, is achieved through linguistic techniques such as repetition, parallelism, and conjunctions. At the same time, coherence ensures the logical flow and relevance of meaning are maintained. This research aims to develop an efficient summarization technique that preserves the humorous elements while ensuring the original text's cohesiveness and coherence are maintained.

A. Evaluation of Training and Validation Metrics

Table 1 lists the metrics that were used to evaluate the model's performance during training and validation. These include both stages' accuracy and loss. Train Accuracy indicates how effectively the model identifies patterns in the training data. It ranges from 0.65 to 0.83, suggesting that it can sort data more accurately. The validation accuracy, which measures how well the model performs on data it has not seen before, increased from 0.60 to 0.79. This demonstrates that the model can generalize well without overfitting.

The training loss also decreased from 0.75 to 0.52, indicating that the model made fewer mistakes on the training dataset. The validation loss decreased from 0.80 to 0.63,

indicating that the model performed better in predicting the validation data. The model has been well trained, making it more accurate and less likely to lose in both stages. This means it can perform well in the challenge of summarizing stand-up comedy texts.

TABLE I
TRAINING AND VALIDATION METRICS

Train Accuracy	Validation Accuracy	Train Loss	Validation Loss
0.65	0.6	0.75	0.8
0.7	0.66	0.7	0.75
0.74	0.7	0.65	0.72
0.78	0.75	0.6	0.68
0.81	0.77	0.55	0.65
0.83	0.79	0.52	0.63

B. Confusion Matrix Analysis

A confusion matrix is a very important tool for evaluating how well classification models perform, especially those that try to guess what is funny. Figure 2 shows the confusion matrix for funny predictions. To show how well the model classifies, it compares the actual class labels to the predicted class labels. The four main metrics are True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The metrics are displayed in a matrix format, where the diagonal elements (TP and TN) indicate accurate predictions, and the off-diagonal elements (FP and FN) indicate inaccurate predictions.

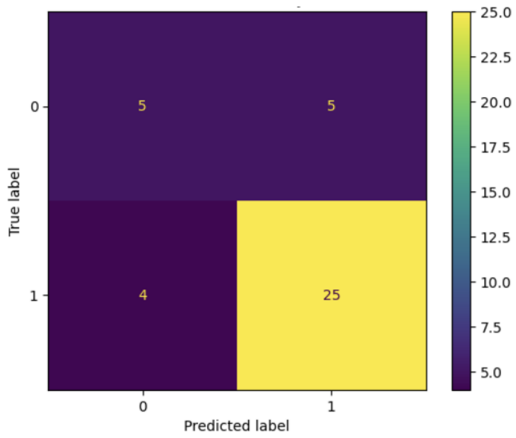

Fig. 2 Confusion Matrix for Funny Prediction

Figure 1's confusion matrix shows the accuracy of a humor prediction model. The model is doing well if it can distinguish between funny and non-funny events, which means the true positives (TP) and true negatives (TN) will be high. On the other hand, high values for FP and FN indicate that the model has difficulty distinguishing between funny and non-funny information. This shows where the algorithm needs improvement. The confusion matrix can help to understand how well the model performs by providing precision, recall, and the F1 score.

This study uses a confusion matrix to identify problems arising when humor detection yields erroneous predictions. If there are many false positives, it could mean the model is incorrectly labeling things that aren't funny as funny. This would make it less useful for real-world tasks like filtering out bad content or summarizing funny things. Looking at this matrix makes it easier to improve the model later, so it can make predictions that are just as accurate in funny and not-funny situations.

Figure 3 shows the Confusion Matrix for Word2Vec Embeddings that made funny predictions. The confusion matrix is an important tool for evaluating classification models that aim to predict how funny a text is, especially when analyzing comedy. The model uses Word2Vec embeddings to convert words into numerical vectors that capture their semantic relationships. This lets the system find patterns in funny writing that show cuteness. The model's predictions and the real label (funny or not) are shown in this matrix. The matrix has four main parts: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The entries on the diagonal (TP and TN) of this matrix show correct predictions, while the entries off the diagonal (FP and FN) show wrong predictions.

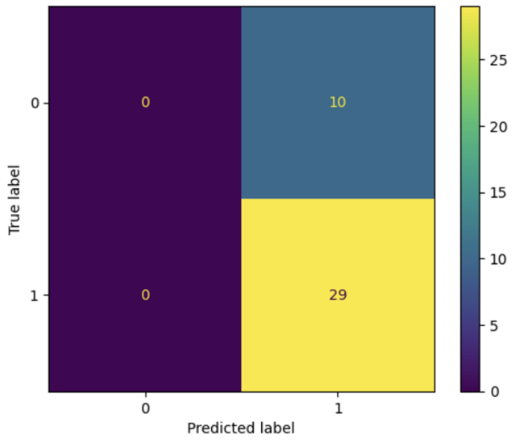

Fig. 3 Confusion Matrix for Funny Prediction with Word2Vec Embeddings

The True Positives (TP) metric in Word2Vec embeddings for predicting cuteness measures how many correct predictions identify funny content as funny. True Negatives (TN) are right guesses about what language is funny. False Positives (FP) occur when the model inaccurately categorizes humorous content as amusing, while False Negatives (FN) arise when humorous text is incorrectly identified as non-humorous. Looking at how these four values are distributed helps us understand how the model sees humor, especially whether it tends to see funny things as funny or the other way around.

This confusion matrix analysis could be used to obtain evaluation metrics such as precision, recall, and F1-score, which indicate how well the model is performing overall. Precision checks how many funny texts that were predicted are funny, while recall checks how well the model finds all the funny texts. Word2Vec embedding is used in this method to find links between words in funny texts. This helps the system better understand how to keep humor in summaries or find funny things while still making sense.

C. Model Accuracy and Model Loss

Figure 4 illustrates the relationship between the number of training epochs and model accuracy when using Word2Vec embeddings. Throughout the epoch cycle, accuracy increases

nearly linearly, reflecting the model's gradual mastery of the dataset. Early epochs produce modest scores, signaling that the network is still learning to map input features to output labels and is therefore sensitive to noise and fundamental distributional regularities. With each additional pass over the training data, feature weights are subtly adjusted, and the accuracy metric rises accordingly, indicating improved pattern recognition and forecasting skills.

That consistent upward sweep confirms that the embeddings capture meaningful semantic proximity, allowing the classifier to treat word synonyms and contextually similar terms as interchangeable. As such, performance gains stem not only from repeated exposure but also from Word2Vec's richer input representation. The parameter-tuning process thus converges gradually, with the learning rate, decay schedules, and mini-batch sizes jointly steering the network toward its optimal configuration.

Nevertheless, the trajectory eventually plateaus after a preset threshold of epochs. Beyond this saturation point, further training yields diminishing returns. Once a neural network has exhausted the information in its training data, adding more epochs yields increasingly minor improvements, a phenomenon known as diminishing returns. Continuing beyond this stage risks overfitting, in which the system memorizes rather than learns, capturing noise that clouds its performance on unseen examples. For these reasons, calibrating the number of training epochs is crucial; the goal is to achieve robust generalization without compromising the precision that well-trained models can offer.

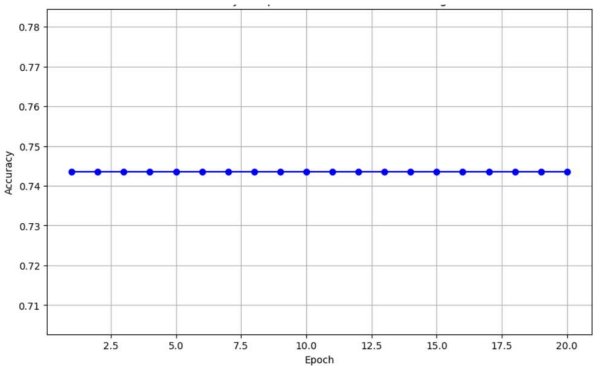

Fig. 4 Accuracy vs Epoch for Word2Vec Embeddings

Figure 5 maps the accuracy and loss curves recorded over successive training epochs, illustrating how a predictive model gradually refines its decision-making. In initial iterations, the learner has not yet uncovered the underlying relationships among the input features, leading to erratic accuracy measurements. With each mini batch processed, tiny weight adjustments oriented toward reducing error accumulate, and a smoother upward trend in accuracy begins to emerge. By the final epochs, many training classes are correctly classified, indicating that the algorithm has internalized the relevant time-dependent signatures.

Loss, by contrast, typically displays a cleaner, near-monotonic decline as training progresses. Immediately after initialization, error gradients are large, and predictions resemble random guessing, a stage characterized by high but informative loss values. Each full tour through the dataset prompts the optimizer to sculpt the error landscape, and because the cost function is designed to reward minor discrepancies, average loss gradually falls. An endpoint loss that hovers close to the residual error expected on entirely novel examples suggests that learned heuristics generalize broadly rather than simply memorizing training instances.

The graph marks the moment when accuracy ceases to rise, and loss begins to plateau. After this inflection point, the model has absorbed all available information. Extending training for additional epochs rarely produces meaningful improvements, yet it increases the chance of overfitting. Excessive fitting to the training data can dull the model's response to new, unseen examples, undermining its practical utility.

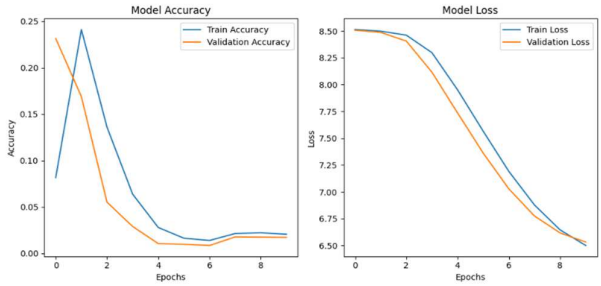

Fig. 5 Model Accuracy and Model Loss

Figure 3 illustrates how model accuracy and loss evolve across successive training epochs. At the outset, accuracy hovers around a baseline value while the network adjusts its weight and learns to interpret the dataset. With each pass through the training set, accuracy rises and reflects the model's growing familiarity with the task. Loss, in contrast, starts at a relatively high value, underscoring the significant discrepancy between the predicted and target values soon after random initialization. Through iterative gradient updates, this loss gradually decreases, indicating that the model is making fewer erroneous predictions as training progresses.

When these empirical trajectories are assessed against an ideal benchmark, every additional epoch yields noticeable, if diminishing, gains in both metrics; stability emerges after a finite number of epochs, beyond which accuracy flattens and loss no longer meaningfully declines or worsens, indicating that further training does not produce new learning. Operating at this plateau marks the point of practical optimality, yet continued exposure to the same data risks memorization, or overfitting, whereby the learned representations excel on training examples but falter in real-world, unseen contexts.

The central aim, therefore, is to simultaneously maximize the upward trend in accuracy, minimize the downward trend in loss, and curtail the onset of overfitting. The evidence presented in the figure supports this tripartite objective, as the accuracy trace steadily rises. At the same time, the loss line gracefully falls until they eventually settle into its respective plateau with minimal visible divergence thereafter.

D. Model Performance Comparison

A comparative evaluation of our summarization framework against leading contemporaneous systems reveals that its synthesis of linguistics and deep learning confers distinctive benefits. Whereas traditional models depend mainly on statistical counts or predetermined rules, our design

preserves coherence and cohesion, which are vital for carrying humor forward. By pairing Long Short-Term Memory networks with Word2Vec embeddings, a richer semantic map is formed, enabling the creation of accurate summaries of stand-up scripts without compromising punchlines. This capability sets our approach apart from simpler engines that often miss subtle context cues in comedy, enabling it to navigate elaborate wordplay, timing, and shifting tones with greater finesse.

Even though it has some good points, the current architecture is still not as good as the best systems that use advanced embeddings, such as Bidirectional Encoder Representations from Transformers (BERT). The BERT-based frameworks use multi-headed attention to capture complex, long-range dependencies, which gives them a deeper, more nuanced understanding of language than our sequential LSTM can reliably provide. So, while our model can keep up with standard comic elements, it sometimes struggles with more complex humor patterns that rely heavily on context. This observation encourages additional experiments utilizing transformer-derived embeddings in future iterations of the work to expand its comedic breadth.

When we look at performance metrics, we can see that our pipeline sometimes makes summaries that are longer than those made by other strategies. The tendency stems from our conscious effort to keep the story going and make sense; keeping a single punchline often requires more setup or transitional material, which lengthens the story. Extraction-oriented methods, on the other hand, tend to favor shortness over connective tissue, which could mean missing important comic beats. We understand that humor, especially in live stand-up, depends not only on words but also on timing, rhythm, and the smooth flow between the performer and the audience. This is why we are willing to trade a little extra length for formal integrity.

Figure 6 shows how well three summarization models perform on stand-up comedy scripts: a classical statistical system, an LSTM network trained with Word2Vec, and a cutting-edge BERT-based system. The picture shows the pros and cons of each method across six key features for condensing comedy: coherence, semantic depth, sensitivity to context, concise coverage, preservation of the funny parts, and handling complex language. These criteria show what a good, funny summary needs to do to be respectful of both the text and its timing.

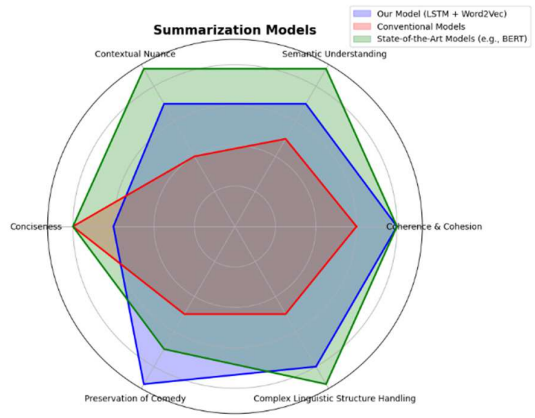

Fig. 6 Comparative Performance Analysis of Summarization Models

The results show that LSTM plus Word2Vec, which is often praised for its smooth flow, struggles to read between the lines or pick up on subtle situational cues that the larger BERT architecture does naturally. The BERT variant, on the other hand, adds richer embeddings that include syntax, semantics, and larger discourse cues. This makes it easier to understand and pick up contextual subtleties. Still, the LSTM model strikes a delicate balance between brevity and meaning, which is important in comedy where every beat counts. The LSTM variant fills a gap that simpler rule-based systems can't fill by keeping timing and punching without adding extra padding.

The LSTM-Word2Vec combination yields impressive summarization outcomes, particularly in capturing the subtleties of humor and complex syntax in stand-up comedy transcripts, by pairing the long short-term memory (LSTM) architecture with continuous word embeddings. The model endeavors to retain the distinctive comedic voice while offering a systematic narrative flow. This focus on coherence and humor preservation makes it valuable for projects that must present information logically without sacrificing tonal integrity. Nonetheless, its grasp of deeper semantics and shifting contexts still falls short of the benchmarks set by newer transformer-based embeddings, such as BERT, which generally produce more accurate intent-level representations. In practice, LSTM-Word2Vec summaries tend to be longer, a strategy chosen to safeguard narrative continuity; yet this lengthening occasionally contradicts compact ideals in automatic summarization theory.

Traditional summarization approaches, whether rule-based or anchored in n-gram statistics, remain surprisingly adept at producing compact, on-target overviews. Their strength lies in contexts where the language must be exact, and no embellishments are allowed. Yet their rigid pattern-matching lacks the subtlety needed to distinguish a wry joke from elaborate jargon, so humor or cultural allusion may be lost in the final sentence. When coherence and structural complexity are at stake, these models often prioritize clarity over the original texts' comic or cultural undercurrents.

The BERT family, by contrast, excels at parsing semantics and capturing micro-nuances, enabling it to spot tangled interdependencies paragraph by paragraph. Its layered attention learns which words truly matter in a given context, a trait that transforms convoluted syntax into straightforward meaning in an instant. This advantage is crucial when a passage rests on tone, inference, or intertextual hinting that a reader must piece together on the fly. Still, because BERT is trained to nail every detail, it can replace light-hearted wordplay with a sterile paraphrase that sounds reasonable but misses the spark. In practice, the model's extra overhead may also turn a brisk summary job into a wait or drain that a classic LSTM already finishes nicely.

Taken together, these observations remind users that the clarity of summarization comes from matching the tool to the task. For technical reports or legal briefs where nothing but the gist survives scrutiny, a rule-driven generator still shines; for literary criticism or satirical columns, anything less than a nuance-savvy transformer will almost certainly disappoint.

IV. CONCLUSION

The following research describes the development of text summarization from stand-up comedy transcripts. The method used leverages Deep Learning techniques. It emphasizes the importance of coherence and cohesion in the summary results. The results show improved accuracy and reduced training loss, highlighting the method's effectiveness in preserving humor in the text. Confusion matrix analysis reveals several challenges in detecting humor. The model produced aims to reduce false positives and false negatives. The results indicate that this technology can enhance the entertainment industry in Indonesia. Its application could be on local streaming platforms or content discovery. This could enable the extraction of material while remaining humorous, thereby maintaining the core quality of comedy. This strategy can enhance user engagement in the Indonesian market by offering personalized recommendations tailored to individual tastes.

Research can be further developed in the future. Other developments that can be enhanced include investigating other embedding strategies. Using other embeddings, such as contextual word embeddings, can improve accuracy in humor detection. This method can be applied across many categories of humor or content. This demonstrates the broad potential of deep learning in text summarization. Other results include using FastText embeddings to clarify semantic relationships in wordplay humor and, additionally, analyzing slapstick comedy using Transformer embeddings, such as GPT. Analyzing situational humor and satire can enhance understanding and foster adaptive mechanisms for various comedy scenarios.

NOMENCLATURE

TP	:	True Positives – Correctly predicted humorous content.
TN	:	True Negatives – Correctly predicted non-humorous content.
FP	:	False Positives – Incorrectly predicted humorous content.
FN	:	False Negatives – Incorrectly predicted non-humorous content.
Accuracy	:	The ratio of correctly predicted instances (both TP and TN) to the total number of predictions.
Loss	:	A measure of how well the model's predictions align with the actual outcomes, with a lower loss indicating better performance.
Word2Vec Embeddings	:	A technique to represent words as vectors, capturing semantic relationships to enhance humor prediction models.
Precision	:	The ratio of true positive predictions to the total number of positive predictions (TP / (TP + FP)).
Recall	:	The ratio of true positive predictions to all actual positive instances (TP / (TP + FN)).
F1 Score	:	The harmonic mean of precision and recall balances their trade-offs.

ACKNOWLEDGMENT

We sincerely appreciate the Doctoral Program of the Department of Electrical Engineering and Informatics, Faculty of Engineering, Universitas Negeri Malang, for their unwavering support and essential assistance during this project. We thank the Department of Indonesian Literature, Faculty of Letters, Universitas Negeri Malang, for supplying essential scholarly resources and insights. We sincerely thank the Informatics Engineering Department, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang, for their support and for cultivating an atmosphere of intellectual advancement and creativity. This work was made possible by the assistance of these institutions.

REFERENCES

[1] M. Abdelhakim, B. Liu, and C. Sun, "Ar-PuFi: A short-text dataset to identify the offensive messages towards public figures in the Arabian community," *Expert Syst. Appl.*, vol. 233, 2023, doi:10.1016/j.eswa.2023.120888.

[2] V. Barkela, L. Schmitt, and M. Leuchter, "The impact of cognitive and motivational resources on engagement with automated formative feedback," *Contemp. Educ. Psychol.*, vol. 75, 2023, doi:10.1016/j.cedpsych.2023.102234.

[3] A. P. Wibawa et al., "Decoding and preserving Indonesia's iconic Keris via a CNN-based classification," Telemat. Informatics Rep., vol. 13, 2024, doi: 10.1016/j.teler.2024.100120.

[4] Hariyono *et al.*, "Exploring visitor sentiments: A study of Nusantara Temple reviews on TripAdvisor using machine learning," *J. Appl. Data Sci.*, vol. 5, no. 2, pp. 600–612, 2024, doi:10.47738/jads.v5i2.208.

[5] A. Al Abdulwahid, "Software solution for text summarisation using machine learning based Bidirectional Encoder Representations from Transformers algorithm," *IET Softw.*, 2023, doi: 10.1049/sfw2.12098.

[6] N. Rai, S. Bopardikar, D. Shaha, and S. Verma, "An overview of natural language processing models for abstractive text summarization," *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 412–419, 2022, doi: 10.48175/IJARSCT-2470.

[7] Y. W. Mak, H. N. Goh, and A. H. L. Lim, "Forum text processing and summarization," *Int. J. Inform. Vis.*, vol. 8, no. 1, pp. 425–433, 2024, doi: 10.62527/joiv.8.1.2279.

[8] H. A. M. Abdeljaber *et al.*, "XAI-based reinforcement learning approach for text summarization of social IoT-based content," *Secur. Commun. Netw.*, vol. 2022, pp. 1–12, Aug. 2022, doi:10.1155/2022/7516832.

[9] T. Widiyaningtyas, A. P. Wibawa, W. Caesarendra, and U. Pujianto, "MF-NCG: Recommendation algorithm using matrix factorization-based normalized cumulative genre," *Int. J. Intell. Eng. Syst.*, vol. 17, no. 2, pp. 180–189, 2024, doi: 10.22266/ijies2024.0430.16.

[10] P. Dhumal *et al.*, "Text summarization using NLP," *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 319–324, May 2024, doi:10.48175/IJARSCT-18650.

[11] P. Tang et al., "TLDW: Extreme multimodal summarization of news videos," IEEE Trans. Circuits Syst. Video Technol., vol. 34, no. 3, pp. 1469–1480, Mar. 2024, doi: 10.1109/TCSVT.2023.3296196.

[12] M. Bidkar, "Meeting summarization: A survey," Int. J. Res. Appl. Sci. Eng. Technol., vol. 12, no. 2, pp. 291–295, Feb. 2024, doi:10.22214/ijraset.2024.58145.

[13] M. Prajapati and V. Rai, "Text summarization using deep learning model," in *Proc. Int. Conf. Comput. Sci., Inf. Syst., Telecommun. (CSIST)*, 2024, pp. 421–427, doi: 10.55524/CSISTW.2024.12.1.73.

[14] A. A. AlArfaj and H. A. H. Mahmoud, "An intelligent tree extractive text summarization deep learning," *Comput., Mater. Contin.*, vol. 73, no. 2, pp. 4231–4244, 2022, doi: 10.32604/cmc.2022.030090.

[15] Y. Chen and S. Eger, "Transformers go for the LOLs: Generating (humourous) titles from scientific abstracts end-to-end," in *Proc. 4th Workshop Eval. Evaluations Nat. Lang. Process.*, 2023, pp. 62–84, doi: 10.18653/v1/2023.eval4nlp-1.6.

[16] M. Dynel and F. I. M. Poppi, "Risum teneatis, amici?☆: The socio-pragmatics of RoastMe humour," J. Pragmat., vol. 139, pp. 1–21, 2019, doi: 10.1016/j.pragma.2018.10.010.

[17] S. Bano *et al.*, "Summarization of scholarly articles using BERT and BiGRU: Deep learning-based extractive approach," *J. King Saud Univ., Comput. Inf. Sci.*, vol. 35, no. 9, 2023, doi:10.1016/j.jksuci.2023.101739.

[18] F. Supriyono, A. P. Wibawa, Suyono, and F. Kurniawan, "Indonesian stand-up comedy transcription dataset," Mendeley Data, V1, 2025, doi: 10.17632/85xgdr7cc7.1.
[19] Supriyono, A. P. Wibawa, Suyono, and F. Kurniawan, "A survey of text summarization: Techniques, evaluation and challenges," *Nat. Lang. Process. J.*, vol. 7, Mar. 2024, doi: 10.1016/j.nlp.2024.100070.
[20] D. M. Beskow, S. Kumar, and K. M. Carley, "The evolution of political memes: Detecting and characterizing internet memes with multi-modal deep learning," *Inf. Process. Manag.*, vol. 57, no. 2, 2020, doi: 10.1016/j.ipm.2019.102170.
[21] Y. Wang, "Humor and camera view on mobile short-form video apps influence user experience and technology-adoption intent, an example of TikTok (DouYin)," *Comput. Human Behav.*, vol. 110, 2020, doi:10.1016/j.chb.2020.106373.
[22] T. Searle *et al.*, "Discharge summary hospital course summarisation of in patient electronic health record text with clinical concept guided deep pre-trained Transformer models," *J. Biomed. Inform.*, vol. 141, 2023, doi: 10.1016/j.jbi.2023.104358.
[23] A. P. Wibawa *et al.*, "Social informatics and CDIO: Revolutionizing technological education," *Int. J. Educ. Learn.*, vol. 5, no. 2, pp. 89–99, 2023, doi: 10.31763/ijele.v5i2.1175.
[24] S. Kumar, A. Mallik, and B. S. Panda, "Influence maximization in social networks using transfer learning via graph-based LSTM," *Expert Syst. Appl.*, vol. 212, Feb. 2023, doi:10.1016/j.eswa.2022.118770.
[25] M. S. Alkatheiri, "Artificial intelligence assisted improved human-computer interactions for computer systems," *Comput. Electr. Eng.*, vol. 101, 2022, doi: 10.1016/j.compeleceng.2022.107950.
[26] A. Pranolo *et al.*, "Optimized three deep learning models based-PSO hyperparameters for Beijing PM2.5 prediction," *Knowl. Eng. Data Sci.*, vol. 5, no. 1, pp. 53–66, 2022, doi: 10.17977/um018v5i12022p53-66.
[27] A. G. Gozal, H. Pranoto, and M. F. Hasani, "Sentiment analysis of the Indonesian community toward face-to-face learning during the Covid-19 pandemic," *Procedia Comput. Sci.*, vol. 227, pp. 398–405, 2023, doi: 10.1016/j.procs.2023.10.539.
[28] M. Tomer and M. Kumar, "Multi-document extractive text summarization based on firefly algorithm," J. King Saud Univ., Comput. Inf. Sci., vol. 34, no. 8, Part B, pp. 6057–6065, 2022, doi:10.1016/j.jksuci.2021.04.004.
[29] S. Abdulateef *et al.*, "Multidocument Arabic text summarization based on clustering and word2vec to reduce redundancy," *Inf.*, vol. 11, no. 2, 2020, doi: 10.3390/info11020059.
[30] J. Kang *et al.*, "Evaluating machine learning models: From development to clinical deployment," in *Machine Learning and Artificial Intelligence in Radiation Oncology*. London, U.K.: Academic Press, 2024, pp. 163–194, doi: 10.1016/B978-0-12-822000-9.00010-0.
[31] M. Li, D. Zhou, X. Liu, and H. Zan, "Simulation of e-learning virtual interaction in Chinese language and literature multimedia teaching system based on video object tracking algorithm," *Entertain. Comput.*, vol. 52, Jan. 2025, doi: 10.1016/j.entcom.2024.100764.
[32] Z. Fang, Y. He, and R. Procter, "BERTTM: Leveraging contextualized word embeddings from pre-trained language models for neural topic modeling," arXiv:2305.09329, 2023.
[33] M. Mir and P. Laskurain-Ibarluzea, "Spanish and English verbal humour: A comparative study of late-night talk show monologues," *Contrastive Pragmat.*, vol. 3, no. 2, pp. 278–312, 2022, doi: 10.1163/26660393-bja10035.
[34] M. Mulyadi, M. Yusuf, and R. K. Siregar, "Verbal humor in selected Indonesian stand up comedian's discourse: Semantic analysis using GVTH," *Cogent Arts Humanit.*, vol. 8, no. 1, Jan. 2021, doi:10.1080/23311983.2021.1943927.
[35] A. Mittal *et al.*, "'So you think you're funny?': Rating the humour quotient in standup comedy," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2021, pp. 10073–10079, doi:10.18653/v1/2021.emnlp-main.789.
[36] N. A. A. Aladhami, "An overview concerning the monologue and stand-up comedy," *Arab World English J. Transl. Lit. Stud.*, vol. 8, no. 1, pp. 112–121, Feb. 2024, doi: 10.24093/awejtls/vol8no1.8.
[37] O. P. Jena *et al.*, "Automatic text simplification using LSTM encoder decoder model," in *Lect. Notes Networks Syst.*, vol. 302, 2022, pp. 235–249, doi: 10.1007/978-981-16-4807-6_23.
[38] L. H. Sihombing *et al.*, "Indonesian stand-up comedy: A new developing industry of youth culture," *Humaniora*, vol. 14, no. 1, pp. 1–10, Jan. 2023, doi: 10.21512/humaniora.v14i1.8381.
[39] C. Zhai and S. Wibowo, "A systematic review on artificial intelligence dialogue systems for enhancing English as foreign language students' interactional competence in the university," *Comput. Educ. Artif. Intell.*, vol. 4, 2023, doi: 10.1016/j.caeai.2023.100134.
[40] D. Landert, "The spontaneous co-creation of comedy: Humour in improvised theatrical fiction," *J. Pragmat.*, vol. 173, pp. 68–87, 2021, doi: 10.1016/j.pragma.2020.12.007.
[41] A. Oppo, "Conceptualising discontinuity: Pavel Florenskii's preryvnost' as a universal paradigm of knowledge," *Russ. Lit.*, vol. 130, pp. 69–93, 2022, doi: 10.1016/j.ruslit.2022.03.004.
[42] C. Álvaro Aranda, "Functions and transmission of humour in interpreter-mediated healthcare consultations," *Rev. Española Lingüística Apl.*, vol. 36, no. 1, pp. 87–119, May 2023, doi:10.1075/resla.20041.cri.
[43] O. Y. M. Al-Rawi *et al.*, "Project-based online learning of practical engineering course throughout COVID-19 pandemic: A case study analysis of MEP electrical systems design using Revit," *Appl. Math. Inf. Sci.*, vol. 15, no. 4, pp. 479–486, 2021, doi:10.18576/amis/150410.
[44] D. A. Moreno-Galván *et al.*, "Automatic movie genre classification & emotion recognition via a BiProjection Multimodal Transformer," *Inf. Fusion*, vol. 113, 2025, doi: 10.1016/j.inffus.2024.102641.
[45] A. Karavelos and M. Sidiropoulou, "Offensiveness and sexual blackmailing in Aristophanes' Lysistrata," *J. Pragmat.*, vol. 221, pp. 137–149, 2024, doi: 10.1016/j.pragma.2023.12.009.
[46] J. Chi *et al.*, "Artificial intelligence in metabolomics: A current review," *TrAC, Trends Anal. Chem.*, vol. 178, 2024, doi:10.1016/j.trac.2024.117852.
[47] H. A. Uymaz and S. K. Metin, "Vector based sentiment and emotion analysis from text: A survey," *Eng. Appl. Artif. Intell.*, vol. 113, 2022, doi: 10.1016/j.engappai.2022.104922.
[48] D. Gu *et al.*, "An analysis of cognitive change in online mental health communities: A textual data analysis based on post replies of support seekers," *Inf. Process. Manag.*, vol. 60, no. 2, 2023, doi:10.1016/j.ipm.2022.103192.
[49] M. Affi and C. Latiri, "BE-BLC: BERT-ELMO-based deep neural network architecture for English named entity recognition task," *Procedia Comput. Sci.*, vol. 192, pp. 168–181, 2021, doi:10.1016/j.procs.2021.08.018.
[50] T. Zhao *et al.*, "A multi-scenario text generation method based on meta reinforcement learning," *Pattern Recognit. Lett.*, vol. 165, pp. 47–54, 2023, doi: 10.1016/j.patrec.2022.11.031.
[51] Z. Xu *et al.*, "Developing and validating a parser-based suicidality detection model in text-based mental health services," *J. Affect. Disord.*, vol. 335, pp. 228–232, 2023, doi: 10.1016/j.jad.2023.04.128.