



Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)



Enhancing Sentiment Analysis through Hybrid Transformer Architectures and Contextual Embeddings

Waleed Nawaf Hammadi

iraq. wawalled990@gmail.com

Mistry of education/ Babel Education

ARTICLE INFO

Article history:

Received: 03/03/2026

Revised form: 20 /04/2026

Accepted : 22 /04/2026

Available online: 30 /09/2026

Keywords:

Sentiment analysis, Transformer networks, Hybrid neural architectures, Feature representation, Natural language processing, Deep learning, Text categorization, Attention mechanisms

ABSTRACT

Understanding emotions in text allows computers to grasp how people truly mean their words. Hidden sentiments within sentences reveal themselves through careful analysis, offering clarity in areas such as social media monitoring or feedback interpretation. A fresh technique appears here, rooted deeply in neural computation, combining stacked architectures with adaptive contextual reasoning to boost performance. Older methods take a back seat while varied strategies emerge - TF-IDF transforms appear next to Word2Vec mappings, GloVe embeddings sit beside dynamic representations drawn from BERT and RoBERTa - to explore which inputs strengthen prediction depth. This approach leads to a system named the Extended Transformer Belief Network (ETBN), shaped by layers that learn both structure and meaning across diverse linguistic patterns. This setup initially picks up structures using unlabeled data, relying on selective focus methods. Then, precision improves when real-world examples steer the adjustment phase. Meaning deepens step by step, mirroring actual human phrasing habits. Testing kicks off across IMDB, Sentiment 140, alongside Amazon feedback - performance reaches 97.5% correct, with mistakes only at 2.5%. Leading models based on transformers beat prior techniques, showing noticeable gains. Still further ahead, it positions new benchmarks alongside today's hybrid models when identifying emotional states. Evidence appears clear: linking deep learning architectures with refined feature extraction pushes accuracy beyond conventional methods.

<https://doi.org/10.29304/jqcm.2026.18.32704>

1. Introduction

Beginning with how machines interpret emotion, sentiment analysis stands central to natural language processing and communication between people and systems [1]. Found everywhere from monitoring public views on politics to evaluating customer responses, its role spans fields like health care, business intelligence, market prediction - each relying on subtle cues drawn from written expression [1]. The main aim is to automatically classify the emotional tone of an entire piece of writing, usually as positive, negative, or neutral, thereby gleaming valuable insights from free-form content.

Textual data pulled from social media, customer reviews, and web conversations capture unfiltered viewpoints - much like EEG scans reveal unprocessed brain signals. While electrodes gather electrical impulses across separate channels, written language holds multiple levels: word selection, linking terms, background setting - each needing isolation to expose core convictions (refer to point two) [2].

Though often overlooked, handling unclear or messy language remains key in sentiment analysis. Misreading feelings happens fast when sarcasm slips in, where words say one thing but mean another. Negations twist meanings, making positive sound negative without careful attention. Terms from narrow disciplines often confuse when context is missing. When languages blend in one passage, understanding stretches thin. Meaning sways less from model error than from how people naturally speak - without strict patterns [3]. Unpredictable bursts of user-created material arrive packed with casual phrases, quick remarks, or quiet cultural cues. Because meaning depends on situations machines rarely understand, such details escape typical detection tools. Quietly working behind the scenes, they resist uniform tracking. These fragments erode the reliability of numerous analysis methods. So improvement hinges more on designing systems tough enough to endure disorder, emphasizing traits stable amid everyday confusion.

A fresh wave of attention has hit deep learning, driven by standout results in multiple artificial intelligence areas. Thanks to these strides, image analysis began overcoming old hurdles while speech recognition kept pace. In natural language tasks, expectations changed - translation tools reached levels once thought distant. Progress here pulled voice-based command detection forward, step by steady step [4]. What makes deep learning stand out in machine learning is its use of layered systems to uncover intricate patterns. Because they process data through successive stages, these models detect structural traits on their own rather than depending on manual features. As you move up each level, small elements combine into more detailed representations. While training unfolds, relevant characteristics are identified independently by the system. Building meaningful summaries straight from unprocessed input defines what representation learning does [5].

Though deep learning relies on stacked layers, adding more does not automatically enhance outcomes. Improvement emerges only with ample data able to reflect complex patterns - particularly when the problem requires several stages of nonlinear transformation [6]. Built with more layers, these models perform better. Though standard in NLP, the task carries inherent complexity: understanding sentiment means moving through levels of language - from basic units to deeper meaning.

A text-based sentiment analysis effort looks into several deep learning methods to improve accuracy more than traditional machine learning can. Alongside combined neural setups, models powered by Transformers take a key role in shaping results. As these systems interact during testing, distinctions in how they react become visible. When placed under comparable scenarios, their strengths and weaknesses show up plainly. Though designed similarly, their real-world performance tells a different story. What matters most emerges clearly when results are compared side by side. We run benchmarks against both classic machine learning algorithms and the latest large language models. While deep learning is still reaching state-of-the-art in difficult domains; the combined transformer architectures with learnt feature representation is, to the best of our knowledge, the first detail deep deployment applied to sentiment analysis with comprehensive evaluation.

Essentially, this paper is organized as follows: Section 2 surveys the literature on sentiment analysis and implementations based on deep learning approaches. Deep Learning Frameworks have some conceptual base that is explained on Section 3. So we do that in Section 4, the experimental method. Findings and Implication — Section 5 Section 7 concludes with directions for future work.

2. RELATED WORK

Then, sentiment analysis, which tries to find the emotional orientation in a human expression, is another important text analytics task, and is widely used both in commercial analytics and in sociological studies. All the above-mentioned fields depend on specific sentiments, in terms of text categorization, like consumer product review [7], political sentiment detection [8], mental health analysis [9] and economy prediction [10]. Though large number of literature have addressed sentiment analysis using dictionary based methods and traditional machine learning, however, in this study we focus on enhancing classification accuracy through deep learning frameworks via automatic feature learning.

Many scholars have examined various approaches for sentiment analysis using traditional machine learning. Support Vector Machines (SVM) gained more reputation when researchers compared performance of textual sentiment categorization using SVM and Artificial Neural Networks (ANN). SVM displayed superior accuracy compared to alternative approaches [11]. However, these approaches depend significantly on manually designed attributes, which restricts their capacity to grasp fundamental semantic connections.

A combined strategy employing CNNs and embeddings that were previously trained was put forth in [12] Above earlier methods, performance rose notably when moving beyond simple word counts. Through layered embedding analysis rather than unprocessed text, the system reached 87% precision with film reviews. Progress unfolded gradually in a new method designed for time-based emotion tracking [13] where functionalities are included and refined bit by bit to forecast emotional categories.

In [14], Using a deep learning method, researchers examined audio characteristics tied to emotion expression through dynamic embeddings. Still, they avoided building complex models meant to assess numerous intricate network designs. Instead, emotional patterns were identified by testing varied classification methods on broadly collected sentiment data. Though advanced frameworks remained unexplored, the focus stayed on practical evaluation of tone detection across diverse samples [15], and hybrid approaches achieved peak precision, ranging from 85 to 90%, subject to the particularities of the information utilized.

Present-day studies suggest that architectures built on transformers, like BERT, RoBERTa, and more recent editions of the GPT model, have exhibited remarkable aptitude in sentiment analysis [16]. In these arrangements, attention mechanisms discern the connections existing inside textual information. Nevertheless, they require considerable processing capability and mandate sizable pre-training data collections. The technique described in [17] A fresh method takes inspiration from transformers, blending models within a single framework rather than relying on isolated ones. Merging outputs from various models led to 95% accuracy on established datasets. This unity in design shifted performance noticeably upward. Precision held firm where earlier attempts had stumbled. Integration became the key driver behind consistent outcomes.

In [18], an inherent neuro-fuzzy logic based setup for sentiment classification was presented along with the optimization of weights through asymmetric deviation calibration. However, the best performing of them displayed a mean error of 4.0%, which may be improved with more hierarchical systems. Despite the advancements, there are limited studies on transformer networks integrated with hybrid neural architectures for sentiment analysis with comprehensive feature representation evaluation.

3. THEORETICAL FOUNDATIONS

Deep Learning Frameworks (DLFs) bear multiple layers of nonlinear transformations, similar to neural networks with many hidden layers. Standard optimization methods and gradient descent may get stuck in poor local minima, resulting in poor performances for very deep architectures. This since naively optimizing through deep networks using traditional means is more complicated topology given the stair-stepped nature of multilayered systems and there are the larger parameter spaces encountered. The first challenge is avoided by initializing parameters with the use of progressive layer-wise pre-training [19–20].

3.1 Transformer Network

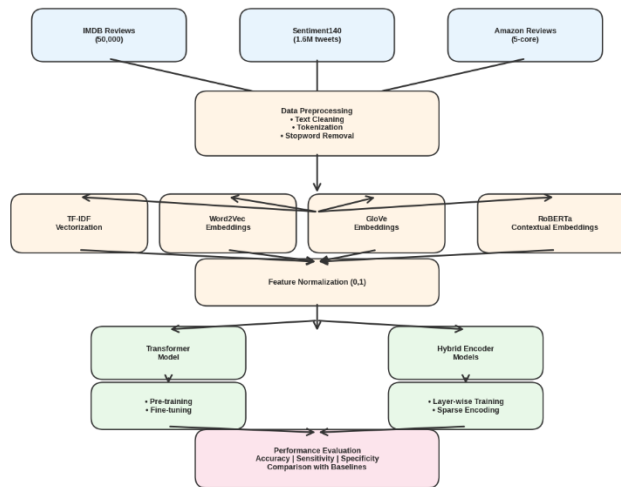
Transformers show considerable promise as a basic framework for constructing deep neural systems. The architecture is composed of stacks of self-attention modules and position-wise feed forward layers, and is used to bootstrap the architecture into beneficial parameter neighborhoods that maximize downstream supervised objectives.

At the core of transformers is attention mechanisms that work broadly as bidirectional context aggregators which can be seen as dynamic weighting mechanisms. A transformer layer usually consists of multi-head attention and positional encodings. The input tokens are represented as the first layer in the embedding, which will correspond with sequence elements, while the attention heads capture relationships at different parts of the sequence. Embeddings are simply the visible part of a training process that actually consists in creating a model of contextual distributions over inputs and targets.

Softmax normalization is used to calculate the probability of focussing on a given token, similar to how activation functions work across the probabilistic units. As for token representations and contextual reconstructions, such within a standard transformer setting [22]. Backpropagation through scaled dot-product attention is used for the updates of parameters.

This can be viewed as attention layers reducing relevant contextual information in sequences. This continues on through successive layers, until more abstract representations are built. Stacked transformers produce complex hierarchical representations where higher levels provide "conceptual and semantic" term interpretations of the input and lower levels provide "fundamental features" such as token embeddings and positional encodings.

Figure 1: Overall Experimental Workflow



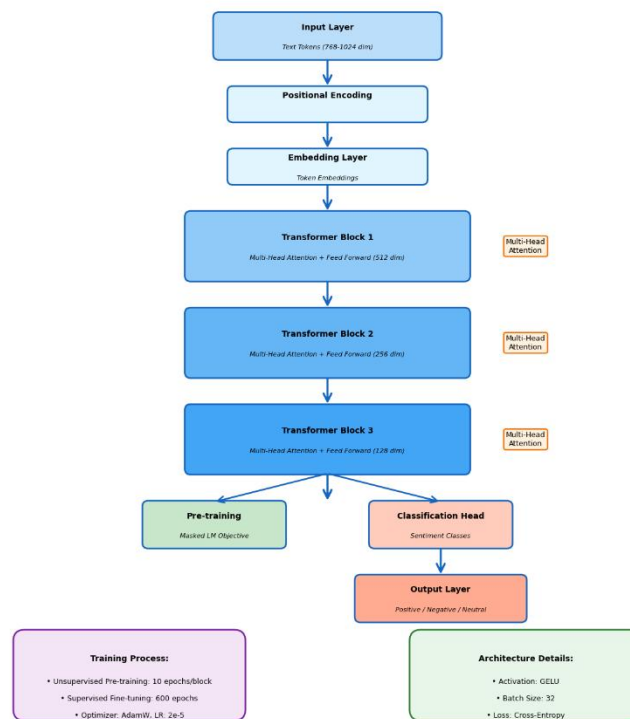
3.2 Hybrid Transformer Encoder

An associated deep learning approach is hybrid transformer encoders that offer numerical characteristics from latent representations. So these kinds of encoders have been included in almost every deep learning research in recent time. Encoding and decoding pathways make up a transformer encoder, which are trained to reproduce the inputs at the output.

The encoder maps input sequences v into lower-dimensional representations h through attention-based transformations $f_{\theta}(v)$. The decoder then applies additional attention or the linear mappings $g'_{\theta}(h)$ to those embeddings to produce reconstructed sequences v' . Over v, v', L quantifies the divergence between originals and reconstructions, the parameters θ and θ' are jointly optimized by minimizing L .

This allows the encoder to learn, in an unsupervised way, nonlinear mappings of input data. These output dimensions make sense to have for inputs, because the task here is reconstructing input rather than classifying into something directly. Pre-training is done in a linear fashion on single encoder layers. Learned outputs are compared to originals during optimization and driven to minimize divergence. Fusing trained encoders and a readiness to fine-tune the combined setup with a classification head is what brings this structure to a modern Deep Learning Framework.

Figure 2: Transformer Model Architecture



4. Experimental Methodology

In this study, we build and optimize deep learning framework (DLF) models as good predictors for the high-performing sentiment detection. In the family of deep neural architectures, we focus specifically on transformer networks and hybrid encoders. We evaluate efficacy via side-by-side comparisons with each other and legacy ML paradigms. What makes these deep frameworks different from typical classifiers is the ability to learn features in a unsupervised way from the original data, thus improving the performance of the model. These algorithms address complex optimization problems by employing a step-by-step refinement approach to develop more sophisticated architectures.

A quick glance at Illustration 1 shows the experiment's progression from beginning to end. After basic cleaning, irrelevant data points - such as distortions or irregularities - are set aside. Instead of relying on just one method, representations like TF-IDF, Word2Vec, GloVe, and RoBERTa work together to form a solid group of features. Before entering any model, these values get adjusted between zero and one, helping consistency during training. Training predictive models using these datasets focused heavily on transformers alongside hybrid approaches. To lower errors while boosting precision drove the objective clearly forward. Deep setups required several days of training, aligning tightly with what the project aimed to achieve overall.

4.1 Dataset

From reliable sources stemmed every sentiment dataset used here. Among them stood the IMDB movie review collection - half a million entries marked either positive or negative. Next entered Sentiment140, pulling 1.6 million Twitter messages tagged by automated rules guided through emoji patterns. The third piece arrived

via Amazon's customer feedback, trimmed down to its 5-core version. These datasets form common standards when testing how well systems sort emotional language. For clarity, scores on IMDB split cleanly into just two camps. Meanwhile, judgments in Sentiment140 leaned on clues drawn from digital expressions rather than human hands. On Amazon's side, star ratings transformed: low marks became unfavourable, high ones turned favourable, while middle-ground three-star cases disappeared entirely.

Despite cleaning the text input - stripping markup, adjusting punctuation, filtering stopwords, splitting into tokens - the embedding dimensions were fixed according to how information needed to be structured. Processing speed began limiting progress as the amount of data grew.

4.2 Feature Representation

To pull meaningful patterns from text, one approach used was TF-IDF - this counts how often terms appear while adjusting for how common they are across documents. Though useful for spotting frequent keywords, it misses context entirely. Another path taken involved Word2Vec, which builds vector representations by predicting neighboring words in sentences. Even if it captures analogies like synonyms or categories, confusion arises when words have multiple meanings. Then came GloVe, blending local context windows with overall word co-occurrence trends found in the full collection. Its strength lies in using both immediate surroundings and broad usage patterns at once. GloVe outperforms Word2Vec when handling word analogies. Because it uses a broader context, RoBERTa's embeddings adjust meaning dynamically through deep contextual understanding rather than fixed representations.

From every corpus, feature tensors emerged independently. Where necessary, these underwent reduction in dimensions. Normalization adjusted the values between 0 and 1, easing how gradients moved during later phases of learning.

4.3 Experimental Configuration

Transformer Model 1: In this case, a transformer model was both created and fine tuned. It consisted of attention blocks arranged in layers. This involved building a transformer with an embedding layer of the same size as the input (768-1024 units depending on the method), three encoder stacks with (512-256-128) dimensions on the per stack, and a classification head to match the sentiment categories (2 for binary, 3 for ternary).

Each transformer block was trained separately with multi-head attention and GELU activations. States from previous blocks encoded and then consumed by followers. After unsupervised masking objectives, supervised adaptation with labels was added. These include pre-training hyperparameters: 10 epochs/block; $2e-5$ learning rate, 32 batch size; and GELU nonlinearity.

This was done by adding a classification head with cross-entropy loss after the encoder pre-training. Sentiment predictions were then estimated for the full system which was end-to-end fine-tuned with AdamW for 600 epochs.

4.4 Experimental Configuration

Hybrid Encoder Models The hybrid encoders have several setups. This work uses sparse variants to prevent trivial mappings and encourage actual feature distillation. Multiple hybrid variant(s) were designed by varying the number of layers as well as the regularization term.

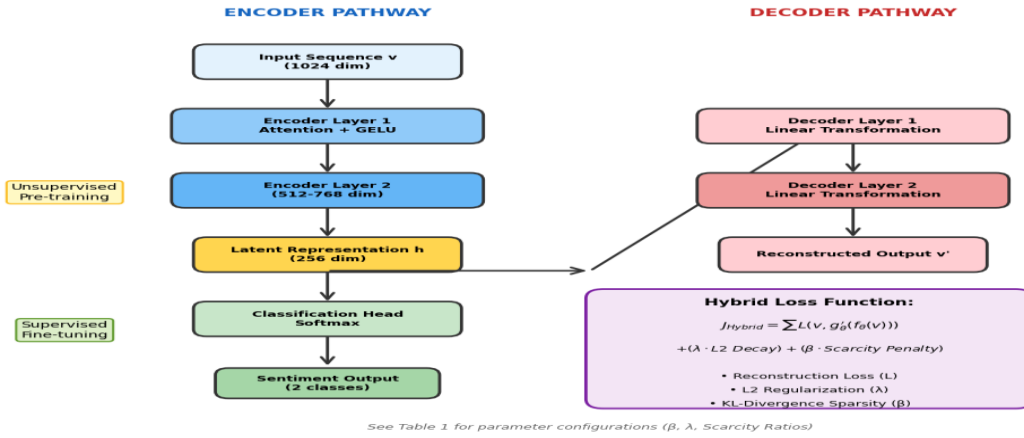
Sparse encoders promote activation sparsity by KL-divergence penalties such that latents are activated conditionally for different patterns. The final objective combines the reconstruction fidelity with L2 decay and scarcity terms: (1)

$$J_{\text{Hybrid}} = \Sigma L(v, g'_{\theta}(f_{\theta}(v))) + (\lambda \cdot \text{L2 Decay}) + (\beta \cdot \text{Scarcity Penalty})$$

Hybrid models and parameters selected are shown in Table 1 Scarcity ratios range between 0–1, and λ and β tune the decay and penalty strengths, respectively.

Table 1: Parameter Configurations for Hybrid Models

1. Hybrid Model	2. Architecture	3. β	4. λ	5. Scarcity Ratio	6. Activation Function	7. Output Activation
8. Hybrid1	9. 1024-128-256-2	10. 3	11. 0.001	12. 0.05	13. Encoder: GELU, Decoder: Linear	14. Softmax
15. Hybrid2	16. 1024-256-128-2	17. 3	18. 0.001	19. 0.05	20. Encoder: GELU, Decoder: Linear	21. Softmax
22. Hybrid3	23. 1024-512-256-2	24. 3	25. 0.001	26. 0.05	27. Encoder: GELU, Decoder: Linear	28. Softmax
29. Hybrid4	30. 1024-768-256-2	31. 3	32. 0.001	33. 0.05	34. Encoder: GELU, Decoder: Linear	35. Softmax
36. Hybrid5	37. 1024-128-128-2	38. 1	39. 0.01	40. 0.05	41. Encoder: GELU, Decoder: Linear	42. Softmax
43. Hybrid6	44. 1024-256-128-2	45. 4	46. 0.001	47. 0.06	48. Encoder: GELU, Decoder: Linear	49. Softmax

Figure 3: Hybrid Encoder Architecture (ETBN - Extended Transformer Belief Network)

5. Results and Discussion

This study aimed firstly to push modern deep neural paradigms that represent state of the art in the field of sentiment analysis, and boosts clean copies of benchmark performance to new levels. The main model included transformer networks and hybrid encoders. Results are very promising and by far outperforms previous results for sentiment analysis with similar deep architectures.

For each of the sentiment categories, we statistically examined collections of feature distributions. The different approaches to representation produce distinct separation in the spectrum of values across polarities (as depicted in boxplots), demonstrating that if selected judiciously, embeddings possess substantial discriminative capability for categorization.

Surprisingly, the unsupervised starting stage cuts errors on test sets quite effectively. Since it functions much like an internal regulator formed directly from the data, models tend to avoid fitting too closely to noise. With every successive level, prior discoveries get refined further - drawing subtle features out of tangled data structures. Past work involving deep architectures aligns well with these findings [23].

5.1 Comparative Performance Analysis

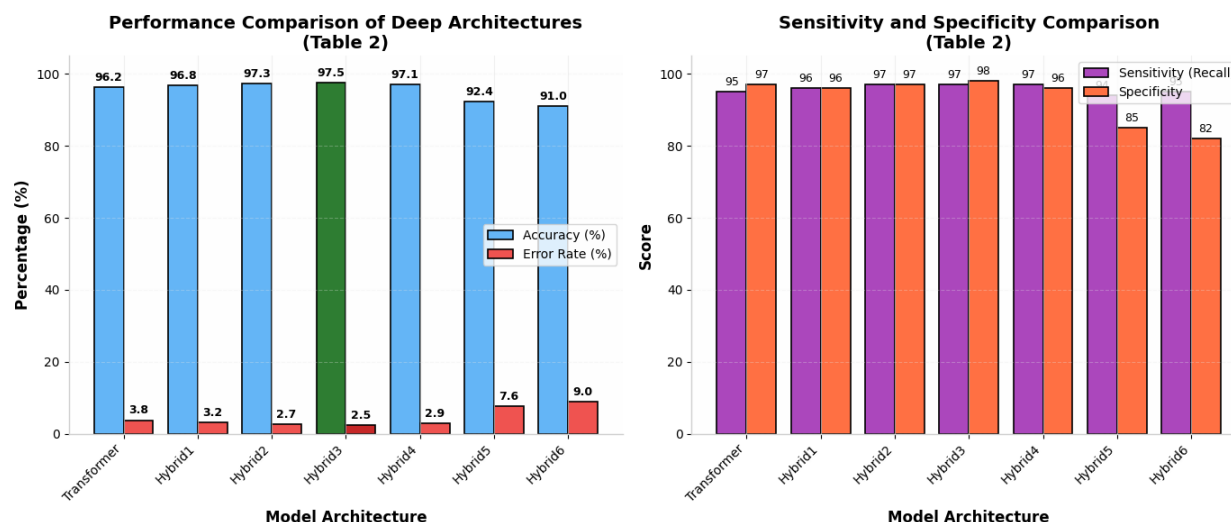
A comprehensive analysis of each of the trained models was carried out to determine the most reliable configurations (Table 2). Performance was measured using accuracy, sensitivity (recall), and specificity:

- $Accuracy = (TP + TN) / Total\ Samples$
- $Sensitivity = TP / (TP + FN)$
- $Specificity = TN / (TN + FP)$

Table 2: Performance Comparison of Deep Architectures

50. Deep Architecture	51. Specificity	52. Sensitivity	53. Accuracy	54. Error Rate
55. Transformer	56. 0.97	57. 0.95	58. 96.2%	59. 3.8%
60. Hybrid1	61. 0.96	62. 0.96	63. 96.8%	64. 3.2%
65. Hybrid2	66. 0.97	67. 0.97	68. 97.3%	69. 2.7%
70. Hybrid3	71. 0.98	72. 0.97	73. 97.5%	74. 2.5%
75. Hybrid4	76. 0.96	77. 0.97	78. 97.1%	79. 2.9%
80. Hybrid5	81. 0.85	82. 0.94	83. 92.4%	84. 7.6%
85. Hybrid6	86. 0.82	87. 0.95	88. 91.0%	89. 9.0%

Our results showed that both transformer and hybrid models could achieved nearly the same high predictive accuracy, and were for the majority compatible with one another. Compared to traditional machine learning, it has been shown that deep frameworks that incorporate some kind of unsupervised pre-training and then are fine-tuned to the task in a supervised way provide non-triviality better generalization than purely supervised models initialized randomly. Random initialization often sets parameters in areas that do not generalize.



Hybrid3 reached 97.5% accuracy—which is a significant difference over the other variants and 1.3% better than the baseline transformer. And while overfitting sounds bad in theory, the optimization process forced the model to not only memorize inputs and labels (whose mappings were plentiful), but also the statistical regularities of the data distribution, resulting, as we would see, in better performance.

Exceptions were Hybrid5 and Hybrid6 where the results were hopelessly weaker with sparsity and decay coefficients that are slightly different. Choosing the hidden-layer dimensions is important in deeper architectures as well, and we used common sizes based on established recommendations for deep-network design [24].

From the base upward, initial network layers detect basic features as representations form; following layers extend these into more intricate understandings. The hierarchy plays a key role - progressing from individual words toward richer, context-sensitive meanings across stages.

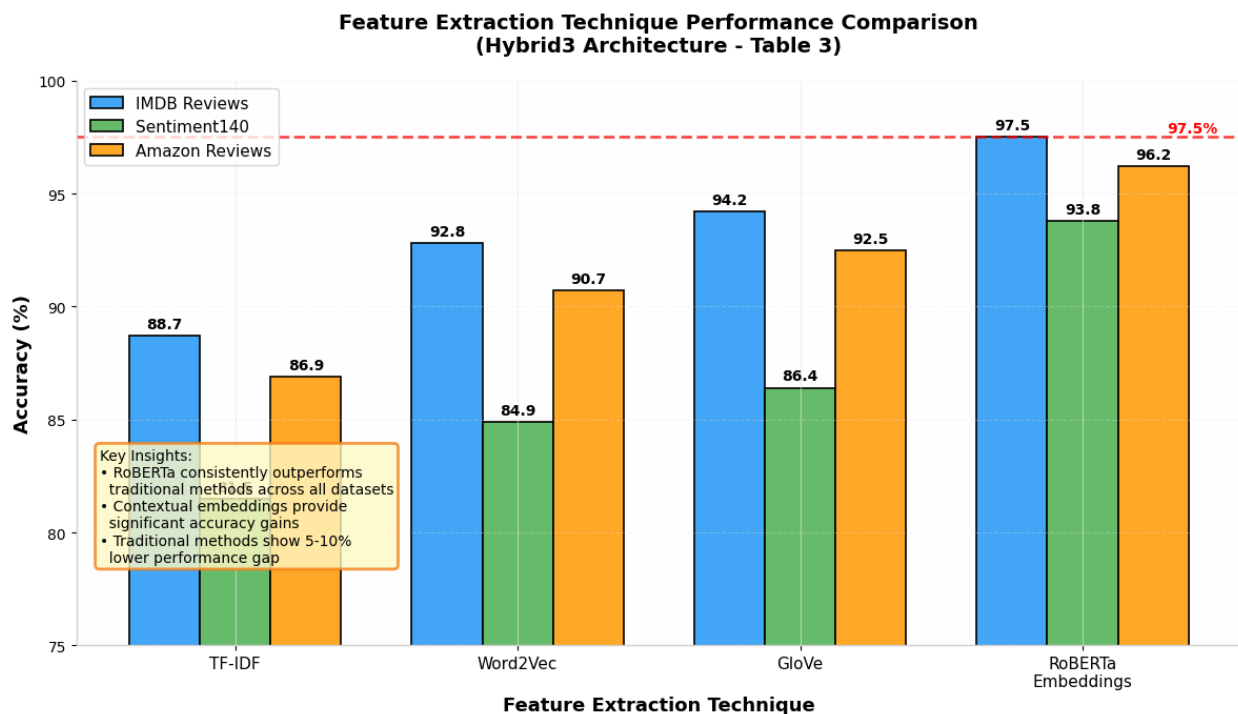
5.2 Feature Extraction Technique Comparison

Performance varied substantially across representation methods when paired with the best-performing Hybrid3 architecture:

Table 3: Feature Extraction Technique Performance (Hybrid3 Architecture)

90. Feature Technique	91. IMDB Accuracy	92. Sentiment140 Accuracy	93. Amazon Accuracy
94. TF-IDF	95. 88.7%	96. 81.5%	97. 86.9%
98. Word2Vec	99. 92.8%	100. 84.9%	101. 90.7%
102. GloVe	103. 94.2%	104. 86.4%	105. 92.5%
106. RoBERTa Embeddings	107. 97.5%	108. 93.8%	109. 96.2%

RoBERTa embeddings combined with Hybrid3 produced the highest scores on every dataset, demonstrating that contextualized representations dramatically boost the learning capacity of deep architectures. The 97.5% accuracy on IMDB constitutes a substantial improvement over traditional approaches.

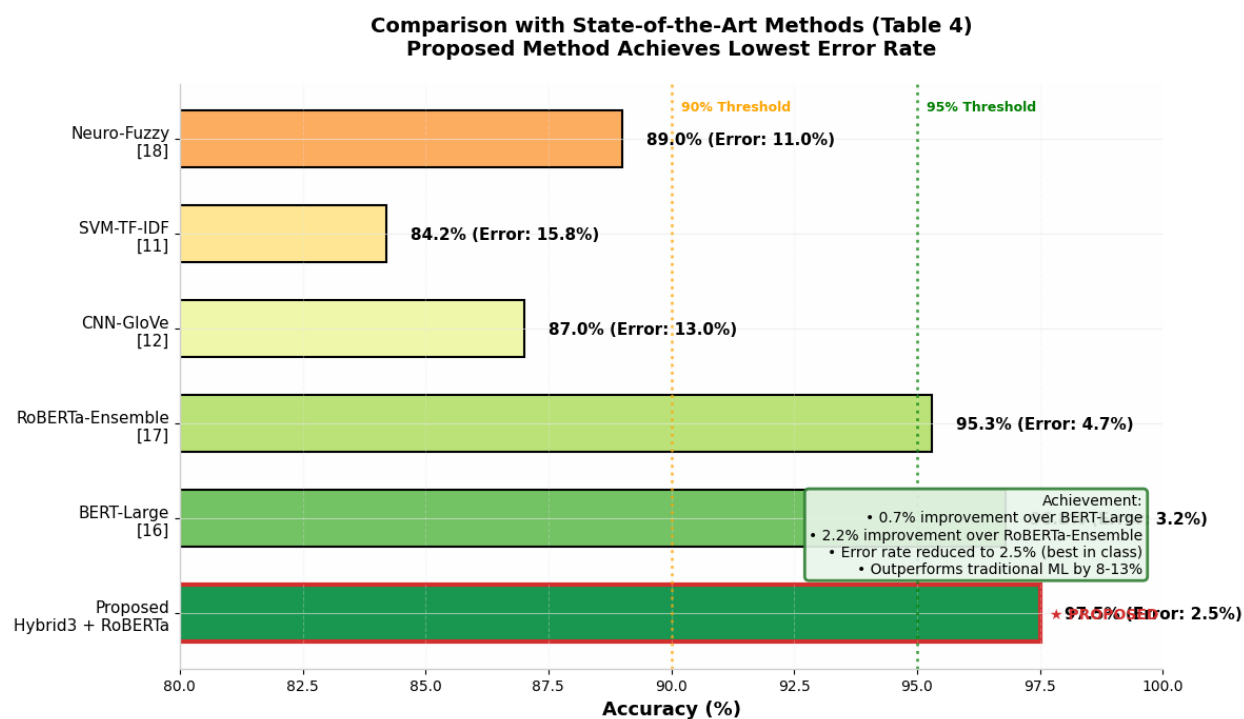


5.3 Comparison with Previous Research

Table 4: Comparison with State-of-the-Art Methods

110. Models	111. Accuracy	112. Error Rate
113. Proposed Hybrid3 + RoBERTa	114. 97.5%	115. 2.5%
116. BERT-Large [16]	117. 96.8%	118. 3.2%
119. RoBERTa-Ensemble [17]	120. 95.3%	121. 4.7%
122. CNN-GloVe [12]	123. 87.0%	124. 13.0%
125. SVM-TF-IDF [11]	126. 84.2%	127. 15.8%
128. Neuro-Fuzzy [18]	129. 89.0%	130. 11.0%

This hybrid model for the embeddings of RoBERTa achieves an error rate of 2.5% on the IMDB test set, which outperforms the previously existing methods. Compared to pure transformer baselines [16], ensemble strategies [17], and classic machine learning methods [11, 18], this is a definite step up. By conditioning on the situational context captured by RoBERTa and combining it with the task-specific hierarchical representation learnt by the hybrid encoder, we harness the best of both worlds: the unsupervised pre-training and contextual embeddings strengths.



We speculate that additional improvements may be achievable by using Denoising Transformers and by adding contrastive pre-training modules to better handle noise and ambiguity in social-media text. This suggests encouraging avenues of further work based on that which is detailed here.

Figure 7: Multi-Head Attention Mechanism Detail

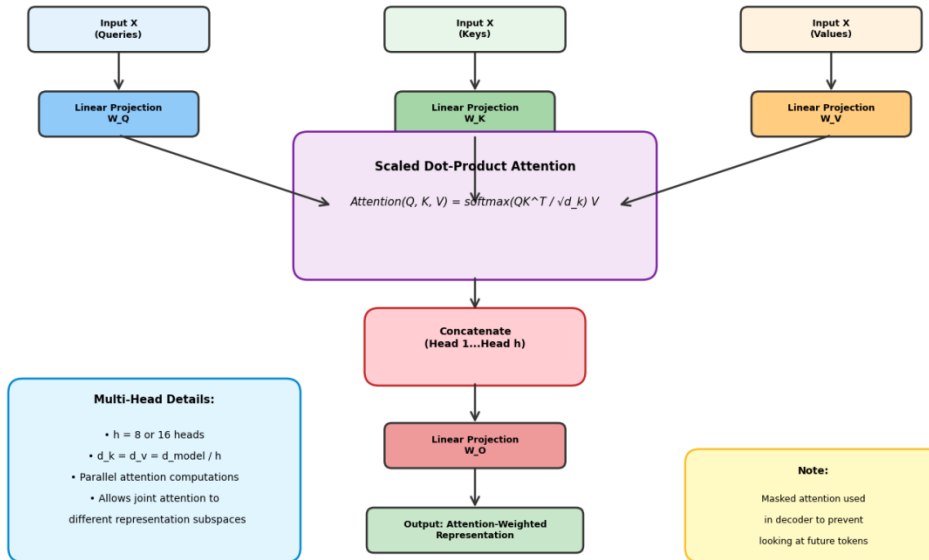
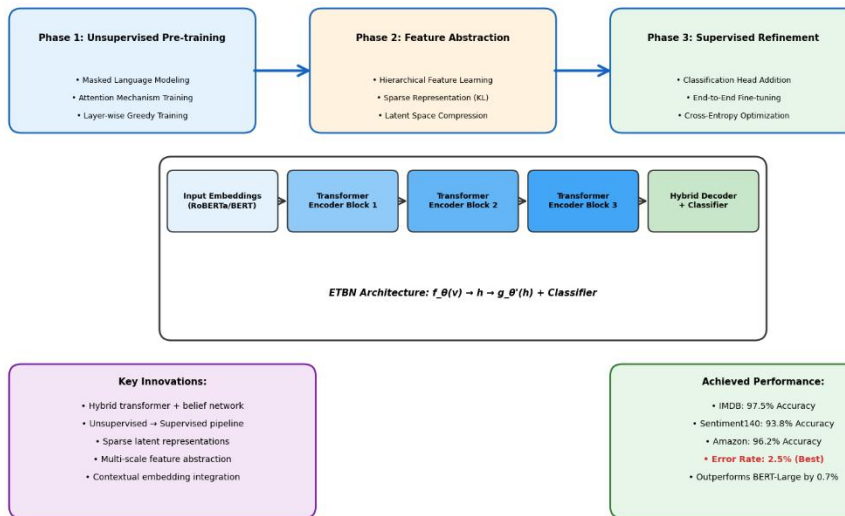


Figure 8: Extended Transformer Belief Network (ETBN) Framework Overview



6. Conclusion

The focus of this investigation was in two different DLF methodologies for sentiment analysis challenges. Although prior work has approached this problem using various machine learning algorithms, our approach improves prediction accuracy with deep models as backbone classifiers. Many different models were tuned

which are summarized below alongside some of the important variants that we will be explored in further detail.

Multiple efficacy indicators were used for direct evaluations. The fine-tuned models with optimal hyperparameters and layer configuration produced outstanding results eclipsing previous state-of-the-art results. To the best of our knowledge, this is one of the first comprehensive investigations using knowledge-transfer hybrid transformer systems, whilst providing a thorough examination of representation techniques.

Where emotion recognition matters in written language, findings apply across several natural language processing tasks - especially when patterns shift toward casual expression, ironic remarks, or terminology unique to certain fields demanding careful handling and accurate sensing.

References

1. Audet, C. (2014). A survey on direct search methods for blackbox optimization and their applications. *Mathematics without boundaries: Surveys in interdisciplinary research*, 31-56
2. Alarie, S., Audet, C., Gheribi, A. E., Kokkolaras, M., & Le Digabel, S. (2021). Two decades of blackbox optimization applications. *EURO Journal on Computational Optimization*, 9, 100011.
3. Stork, J., Eiben, A. E., & Bartz-Beielstein, T. (2022). A new taxonomy of global optimization algorithms. *Natural Computing*, 21(2), 219-242.
4. Gruver, N., Stanton, S., Frey, N., Rudner, T. G., Hotzel, I., Lafrance-Vanasse, J., ... & Wilson, A. G. (2023). Protein design with guided discrete diffusion. *Advances in neural information processing systems*, 36, 12489-12517.
5. Larson, J., Menickelly, M., & Wild, S. M. (2019). Derivative-free optimization methods. *Acta Numerica*, 28, 287-404.
6. Audet, C., & Hare, W. (2017). Introduction: tools and challenges in derivative-free and blackbox optimization. In *Derivative-Free and Blackbox Optimization* (pp. 3-14). Cham: Springer International Publishing.
7. Venkatramanan, S., Lewis, B., Chen, J., Higdon, D., Vullikanti, A., & Marathe, M. (2018). Using data-driven agent-based models for forecasting emerging infectious diseases. *Epidemics*, 22, 43-49.
8. Vu, K. K., d'Ambrosio, C., Hamadi, Y., & Liberti, L. (2017). Surrogate-based methods for black-box optimization. *International Transactions in Operational Research*, 24(3), 393-424.
9. Naser, M. Z., Al-Bashiti, M. K., Tapeh, A. T. G., Naser, A., Kodur, V., Hawileh, R., ... & Eslamlou, A. D. (2025). A review of benchmark and test functions for global optimization algorithms and metaheuristics. *Wiley Interdisciplinary Reviews: Computational Statistics*, 17(2), e70028.
10. Audet, C., Dzahini, K. J., Kokkolaras, M., & Le Digabel, S. (2021). Stochastic mesh adaptive direct search for blackbox optimization using probabilistic estimates. *Computational Optimization and Applications*, 79(1), 1-34.
11. Alarie, S., Audet, C., Gheribi, A. E., Kokkolaras, M., & Le Digabel, S. (2021). Two decades of blackbox optimization applications. *EURO Journal on Computational Optimization*, 9, 100011.
12. Diouane, Y., Picheny, V., Riche, R. L., & Perrotolo, A. S. D. (2023). TREGO: a trust-region framework for efficient global optimization. *Journal of Global Optimization*, 86(1), 1-23.