COMPARISON OF DIFFERENT MODELS TO EVALUATE SENTIMENT AND EMOTIONS WELL BEING

[1]Supreet Kaur Sahi,[2]Vandana Kalra, [3]Amaanya Kaur Dhall

[1]Assistant Professor, [2]Professor, [3]Student
[1]Department of Computer Science,
[1]Sri Guru Gobind Singh College of Commerce, University of Delhi, Pitampura, India

Abstract: Much of the human exchange now occurs through modern digital modes – largely replacing long-established channels that rely on paralinguistic and non-verbal expression in favour of text-based means where absent. While the abundance of written human expression has no precedent, reliably parsing text to assess affective state is still a largely unsolved problem — creating both a challenge and opportunity for computational analysis.

The second introduces and assesses within a common multi-level sequential framework, several approaches to emotion recognition together with sentiment analysis and wellbeing inference into a unified processing pipeline. It first classifies the incoming text into one of seven discrete emotion categories, while also filtering low-confidence (50%) predictions related to emotions through a confidence-based mechanism before obtaining a sentiment label conditioned on the emotion and rule.

Methods to examine the four representation models, three model architectures were trained and tested on three progressively more-complex datasets: Sentiment140 (binary tweet classification), GoEmotions (27-category Reddit comment classification) and MELD (multi-speaker conversational emotion recognition from TV dialogue). The results show that for binary classification tasks all the architectures achieve decent performance, with accuracy settling close to 81%. For fine-grained emotion recognition, the performance drops sharply to 50–53% on GoEmotions and even down to 15–28% on MELD. These failures can be analysed and found to result not from insufficiency of architecture but from context, as conversational emotion recognition cannot reliably occur on isolated utterances.

These conclusions drive the essential design choices of the framework suggested, especially with regard to confidence filtering and separating out emotion, sentiment, and well-being stages as separate architectures designed for a specific data context. Ethical considerations of transparency, bias auditing and limiting automated inference to the level of groups rather than individuals are included as fundamental aspects of design instead of secondary or supplementary issues.

Keywords: “sentiment analysis”, “emotion recognition”, “well-being inference”, “conversational AI”, “deep learning”, “natural language processing, “BiLSTM”, “TextCNN”, “transformer models”

INTRODUCTION

The modes through which humans communicate have undergone a structural transformation over the last twenty years. What was once individual communication, in person with all the NonVerbal Cues of vocal tone and facial expression using eye contact and physical proximity now operates primarily between digital text channels: email threads, instant messaging applications, collaborative project platforms, social media feeds. It has led to an asymmetry that is distinct. The immense volume of human-written text that may be computable is larger than at any other point in the history of recorded communications, but the amount of affective information encoded in written language is much lower than what can be achieved through multimodal signals available with embodied communication.

This asymmetry has the most pronounced consequences in institutional contexts where well-being is an issue, such as workplace, clinical or educational settings. At scale the problem is even simpler: whether managers assessing the emotional climate of a distributed team, clinicians predicting psychological change between appointments, or academic counsellors monitoring student stress across months of exams, the most widely available signal is — conditioned at every level — text and nothing more than text yields an incomplete view into emotional state.

The main computational tools that have been developed to deal with this problem consists on sentiment analysis and emotion recognition. The most usual form of sentiment analysis is where you get a polarity assigned to the text — positive, negative or neutral. On the other hand, emotion recognition is trying to identify a particular psychological state like joy, sadness, anger, fear, disgust, surprise or neutral affect. Both the areas had progress in the past decade first by lexicon-based methods and then by machine learning classifiers and most recently deep neural architectures such as transformer models. However, despite these advances the two domains have mostly evolved in parallel rather than in concert and neither has been systematically tied to the downstream question that gives both their applied significance: how mentally healthy a person is.

This study addresses that gap. We introduce a two-stage hierarchical model where the emotion recognition comes before and under the sentiment classification layer, and the joint outputs are forced to predict meaningful well-being beliefs. The framework is based

on the intermediate observation that emotional context changes the meaning of affective expression: "I guess that's okay" means something quite different coming from someone at a level of settled sadness than it would from one at a level of low-level harmony. No system that tries to determine well-being, without first boiling out this kind of emotive confusion, has a complete foundation.

Affective computing research today suffers from three interrelated deficiencies. In fact most of the sentiment analysis and emotion recognition tasks are treated independently, i.e. trained on different datasets, optimised on different objectives and deployed in isolation. It shuts itself off from the causal structure underlying affective expression: emotion is prior and constraining on sentiment, not parallel.

Second, the evaluation of both tasks focuses mainly on isolated utterances rather than conversational sequences. This is a methodologically utile but naïve simplification, which does not match the way emotional meaning would actually be generated in naturalistic communication. They also get their emotional valence from the conversational context they find themselves in which means systems analysing them in isolation are systematically losing out on the single most obvious and discriminative features available.

Thirdly, a persistent mismatch exists between what affective computing systems output (a label or a probability score) and what practitioners in well-being-sensitive contexts really need — an actionability/interpretability signal about psychological state that can guide decision-making about whether to intervene/remotely-monitor more closely. There is little technical literature that explicitly addresses the translation between model output and practical utility, and this translation seldom receives a design requirements status.

This paper formalizes the problem as a pipeline design: how should emotion recognition, sentiment classification and well-being inference be designed such that each stage adds real information gain to the subsequent one and finally producing an output which is both technically-grounded and practically-interpretable?

LITERATURE REVIEW

This research builds upon existing work in sentiment analysis, emotion recognition and computational well-being inference, identifying key limitations and opportunities for integration.

Today how people feel mentally is a lot more important at work than it was ten years ago. We mostly talk to each other by email, messaging apps, online surveys and shared digital spaces. The communication pattern of people can indicate the pressure they are feeling and that they are experiencing burn out or detached. These signs are easier to notice in physical interactions but today these emotions and feelings are hidden behind the words people type on these platforms.

This shift from physical interactions to online communication platforms has enabled us to understand the emotions of workers, how stressed they are feeling and what kind of mood they are in. Despite people not being open about their emotions and mental state these messages they share on communication platforms act like a window into their feelings. These messages are being used by scientists to find signals of strain or strength in people and how to react to certain stimuli. The different techniques to such studies include tracking mood patterns or detecting signs tied to psychological distress and fatigue. Some studies dive into language cues linked to anxiety or exhaustion during work hours. Others explore broader systems meant to monitor team dynamics over time. Privacy questions surface often, especially when machines judge human feelings. Each method carries risks, particularly if conclusions are drawn too quickly. Thoughtful handling matters most where personal expression meets automated review. From basic machine learning up through deep networks, approaches now include transformers, combined model strategies, while also integrating multiple data types. Trends today show recurring design choices alongside unresolved issues - these shape the work presented here.

Back then, classifying feelings in text meant placing it into one of three buckets - positive, negative, or indifferent. Instead of complex models, experts used word lists tied to emotions, along with selected terms and manually built traits. While useful for simple judgments, such methods failed to catch subtle shades of emotion hidden in everyday speech.

Later studies began focusing on distinct feelings - happiness, rage, sorrow, dread, tension - not just general affect. Detecting these states in written words turned into a method for exploring how people act online, especially in posts, messages, or reviews. Written hints about emotion, researchers noticed, tend to reveal deeper layers compared to simple positive-negative ratings.

Starting off, many early setups moved step by step - first cleaning text, then pulling out key traits before assigning categories and reviewing outcomes. Yet when faced with talk that sounded natural or tangled, these sequences often faltered, especially since such patterns fill daily work exchanges. Meaning tied to situation rarely made it into the process.

Deep learning brought change - neural architectures began dominating research areas overnight. Convolutional designs, alongside LSTM units and recurrence-based systems, quickly took root across studies. Textual signals of mental strain like burnout or low mood were now within reach thanks to these approaches. Patterns unfolding over time, along with deeper word relationships, showed clearer results than old techniques ever managed.

Even so, adjusting deep learning models required patience. When faced with lengthy or intricate texts, they often faltered. Subtle shifts in emotion slipped past them, just as finer points of specialized vocabulary did. Despite their power, nuance remained a challenge.

Though built using layered networks that merged word embeddings with final prediction stages, these architectures showed actual improvements - yet struggled consistently with subtle emotional cues common in office communication. Their accuracy rose, however they frequently misread quiet shifts in tone typical of professional environments.

Studies lately highlight growing use of transformer systems in studying mood, feelings, and psychological states. Instead of reading text piece by piece, they examine full passages together. This structure supports deeper understanding of context.

Across varied sources - online posts, staff feedback, internal messages, clinical notes - transformer-driven methods show stronger outcomes. Detection of nuanced conditions like irritation, worn-out feelings, or thinking strain improves noticeably through these approaches.

With their layered design, transformers turn words into meaning-rich vectors before moving data through several stages of refinement - ending with specialized outputs that make sense for particular goals. These models support nearly every current system built to detect emotions, showing real progress beyond earlier methods in grasping how language works.

Most research on sentiment dives into how workers feel about their jobs, stress levels, and overall wellness. From company surveys to public review boards, data pours in through many paths. By tracing moods across messages, clues emerge about a firm's inner atmosphere.

Looking at different people - office employees, medical teams, learners, those in intense jobs - scientists noticed something consistent. Signs of inner tension often show up first in how someone writes. Emotions hidden in words may signal trouble before it becomes serious. Yet most tools built for workers deliver just one emotional reading at a time. Rarely do they merge more than one feeling signal together. Ethical issues often come after performance goals in these designs. As a result, their value drops when used inside actual companies.

RESEARCH METHODOLOGY

3.1 Proposed Multi-Level Framework

The architecture proposed in this study highlights how the theoretical concepts combine emotions and feelings at different impact levels. The fear of getting something other than expression or sentiment is the main aspect this approach revolves around. In the first assumption, sentiment analysis and classification and emotion recognition are viewed as two similar processes, whereas in the second theory, a step by step process is introduced. First the emotions are recognized and then these emotions are used for the task of sentiment classification. And a fourth stage interprets the outputs of these three into a signal for well-being. The four steps in detail :-

Stage 1: Emotion Classification

An emotion classifier is used to map the input text into one of seven classes: joy/sadness/anger/fear/disgust/surprise/neutral with a probability distribution. These seven categories were chosen because they reflect the annotation schema of the MELD dataset and align with the principles of basic emotion theory. This highest-probability selected category is utilized as the working hypothesis for the next stage. Stage 1 also retains the complete probability distribution for processing under Stage 2.

Stage 2: Confidence-Based Filtering

The confidence filtering mechanism works on the maximum predicted probability from Stage 1. If this probability measures lower than a configurable threshold parameter, the prediction is considered unreliable meaning that the utterance will be either passed to an alternative process or would not make it past the stage of sentiment classification. This mechanism is there to ensure that hypotheses about poorly grounded emotion do not carry over and bias the sentiment and well-being inferences further on downstream. A threshold parameter, treated as a design choice rather than a learnt one; further section gives appropriate empirical guidance for good values.

. Single word utterances like "yes" or discourse markers perform poorly with flat probability distributions indicating genuine uncertainty; in this case, there is no single emotion dominant enough to move the model one way or another. If we let these predictions go forward without checking them the system is going to get messed up with a lot of wrong information. These predictions need to be looked at or they will cause problems in the entire system specifically the predictions will introduce noise, into the pipeline.

Stage 3: Sentiment Classification with Emotional Context

Specifically Stage 2 gives a filtered emotion label to help with the sentiment classification of the text. This is done by adding the labels information to the texts data. The labels information is added by combining it with the texts data in a way or by using a special prompt to guide the network depending on how the network is designed. The label is turned into a code and then added to the texts data. This helps the network understand the emotion, behind the text. The way the label is added can vary depending on the networks architecture. The output of the sentiment classifier is a label of one of three polarity classes: positive, negative or neutral.

This conditioning mechanism puts the theoretical assertion that emotional context alters interpretation of sentiment markers into effect. Say take some feature like less hedging or flat affect in text. Out of the specific emotional context they represent, such

features are open to interpretation. When read with a suppressed-sadness emotion label, they are interpretable as masking of negative affect. The sentiment classifier is supplied with this interpretive frame from the conditioning mechanism.

Stage 4: Well-Being Inference
An interpretable well-being signal is mapped to the combination of emotion label, confidence score and sentiment polarity using a rulebased translation layer. The mapping makes outputs actionable by non-technical users, including managers, counselors and clinicians. The mapping is detailed and rationalized in Section 5.3. Both maintaining modularity and allowing for the mapping to be revised or replaced when domain-specific evidence accrues are satisfied by keeping a translation layer architecturally separated from the classification stages so that models encrypting newly observed training data only need to retrain on instances they have previously classified.

3.2 Data Preprocessing

We kept the same process for preparing the datasets to train models. First, all the text lowercasing — convert everything into lower case making the vocabulary smaller. Output URLs, mentions and HTML entities were also filtered because they did not contribute any emotionally important information for classification. Punctuation was made consistent: multiple instances of marks such as exclamation points, periods or commas were truncated to one instance (since they often represent intensity instead of a particular polarity category) In contrast to other approaches, we chose to include stopwords in the data as research has shown that their distribution can provide information about psychological states. As we are dealing with dirty json data, this preprocessing step is essential for us to avoid noise and discrepancies within our datasets so they will more easily align with the classification models. With this, we wanted to build a strong base for the further analysis and training processes.

Word embeddings were initialised with 300-dimensional GloVe vectors pretrained on a 840-billion-token corpus. Embeddings were allowed to update during training to allow the models to adapt to the target domain. For vocabulary items absent from the pretrained embedding matrix, embeddings were initialised with small random values drawn from a uniform distribution.

The problem of class imbalance in GoEmotions and MELD was tackled by using weighted loss functions during the training process. To do this, class weights were calculated as the inverse of how often each class appeared, and then normalised so that the total weight of all classes added up to the number of classes. This approach makes the model pay more attention to rare classes and less attention to common ones, which helps prevent the most common class from taking over the decision-making process. For example, in MELD, mistakes in the neutral class, which makes up about 46.9% of the training data, are punished about 23 times less severely than mistakes in the disgust category, which is much rarer at 2.0%.

3.4 Model Architectures

3.4.1 TextCNN

The TextCNN takes a fixed-length word sequence of embeddings size d, resulting in a matrix of shape (L, d), L = seq length. Different convolutional filters of window sizes $h \in \{3, 4, 5\}$ (with each filter being $w \times d$ dimension) are applied across this matrix in parallel to learn a certain pattern. A single filter generates a feature map of size $L - h + 1$, and max-over-time pooling takes the highest-activation scalar from each feature map. An intermediate layer is used to concatenate the pooled features of all filters into a fixed-size document representation, and this output passes through a dropout layer and then two fully connected layers. It contains many filters that inspect the data, namely 128 for each window size are generated whereas it creates 384 features. We want to avoid overfitting, so we will use a dropout of 0.5 The last layer uses softmax to assist with classification issues with very many classes. Probably the best thing about TextCNN is that its speed and learn short high emotion phrases easily. However, it has a big limitation - it can't understand how things are related to each other beyond a certain point, which can be a problem. This is because it doesn't have a way to model sequential dependencies that are longer than the filter window size.

3.4.2 BiLSTM

So basically, the BiLSTM architecture is a way of processing sequences of data, like words in a sentence. It uses two different LSTMs, one that looks at the data from left to right, and another that looks at it from right to left. At each point in the sequence, the left-to-right LSTM looks at the current piece of data and everything that came before it, and the right-to-left LSTM looks at the current piece of data and everything that comes after it. This helps the model understand the context of each piece of data. The two LSTMs produce two different hidden states, which are then combined to create a more complete representation of the data at each point. Finally, the model uses this combined representation to make a classification, either by looking at the last hidden states of the two LSTMs or by averaging out the representations from all points in the sequence. This approach allows the model to capture complex patterns and relationships in the data.
The implementation uses hidden state dimension 256 in each direction, for a combined representation dimension of 512. A dropout rate of 0.5 was applied to the output layer. BiLSTM is well suited to tasks where sentiment or emotion depends on long-range sequential dependencies within the utterance, but its capacity to exploit conversational context is limited to what is encoded in the current utterance's embedding sequence.

3.4.3 CNN-BiLSTM

The hybrid architecture works by first using TextCNN to extract features from the text, which gives us a sequence of representations that are informed by the local context. Then, it feeds this sequence into a BiLSTM, which models the dependencies between these local features. The idea behind this is that the features extracted by the convolutional layer provide more useful inputs to the recurrent layer than the raw word embeddings, because each convolutional feature already captures a pattern at the phrase level. This way, the recurrent layer can focus on modeling the relationships between these patterns, rather than trying to learn them from

scratch from the raw word embeddings. By combining the strengths of both layers and recurrent layers this hybrid architecture can learn more effective representations of the text.

When we built this system, we used a layer that you can think of as taking a unique It uses something called a convolutional layer, with 128 individual helpers that look at little pieces of the data. This layer assists with locating crucial attributes in the data that is always forwarded to a dedicated a sort of long short-term memory layer by the name of BiLSTM, which can observe the data in both directions. You have a BiLSTM with hidden dimension of 256, which means that it can remember a lot of information. Finally, we are given the last hidden state of the BiLSTM which is then fed into a unique And then it passes to a fully connected layer that helps, followed by a dropout helper that reduces overfitting contents us make predictions. This system has a lot of parts, which means it takes a long time to train, but surprisingly, it doesn't always do better than its individual parts on the tasks we tested, and we talk more about why that is in further section.

3.5 Training Configuration and Evaluation Metrics

All models were trained using Adam optimizer with early stopping when validation performance stopped improving. Results are reported using accuracy and macro averaged F1 score the latter is more informative when classes are imbalanced because it treats every category equally regardless of how common it is.

Training Curves – TextCNN

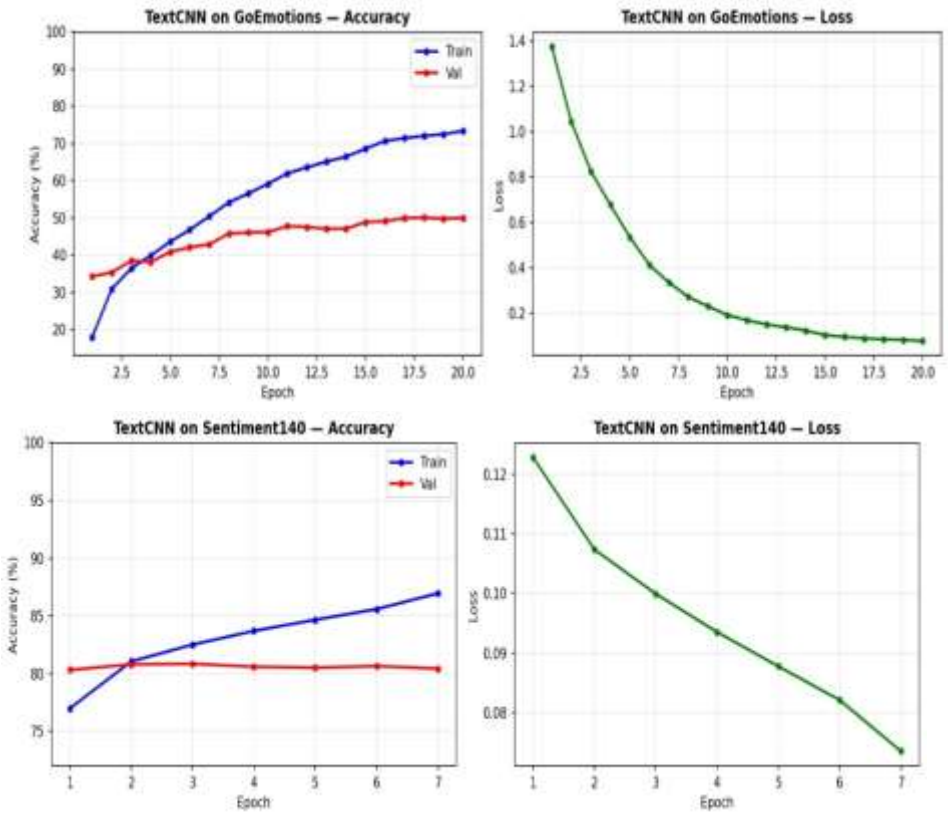
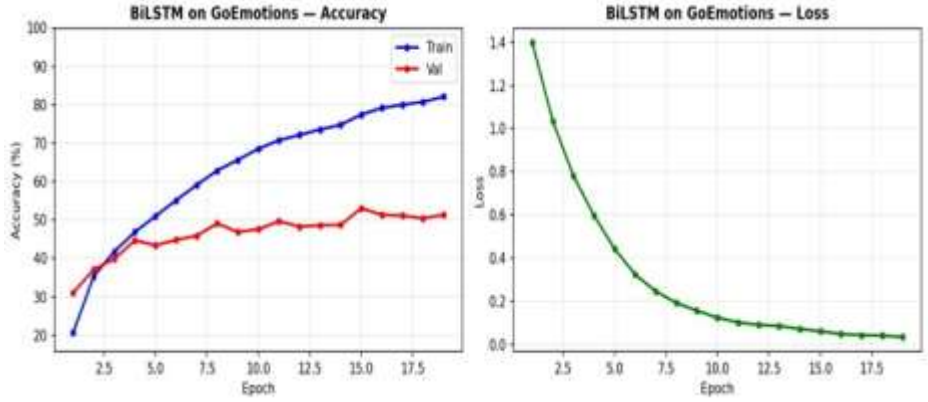


Figure 1: TextCNN on Sentiment140 – Accuracy and Loss over training epochs

Figure 2: TextCNN on GoEmotions – Accuracy and Loss over training epochs



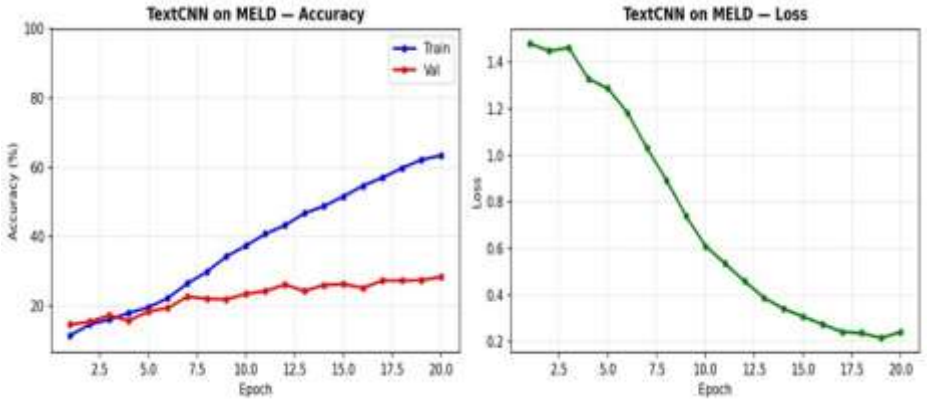


Figure 3: TextCNN on MELD – Accuracy and Loss over training epochs

Training Curves – BiLSTM

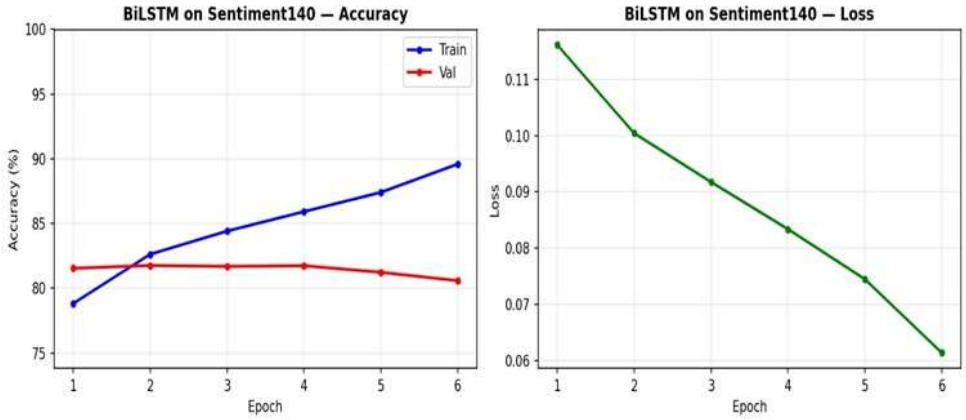


Figure 4: BiLSTM on Sentiment140 – Accuracy and Loss over training epochs

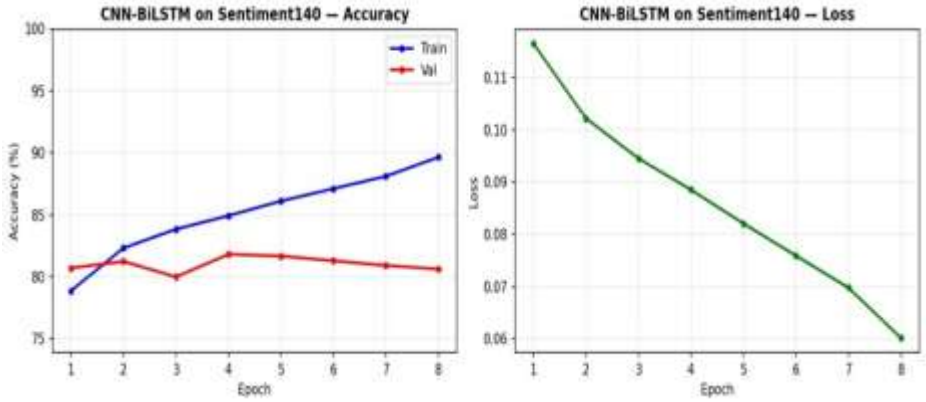


Figure 5: BiLSTM on GoEmotions – Accuracy and Loss over training epochs

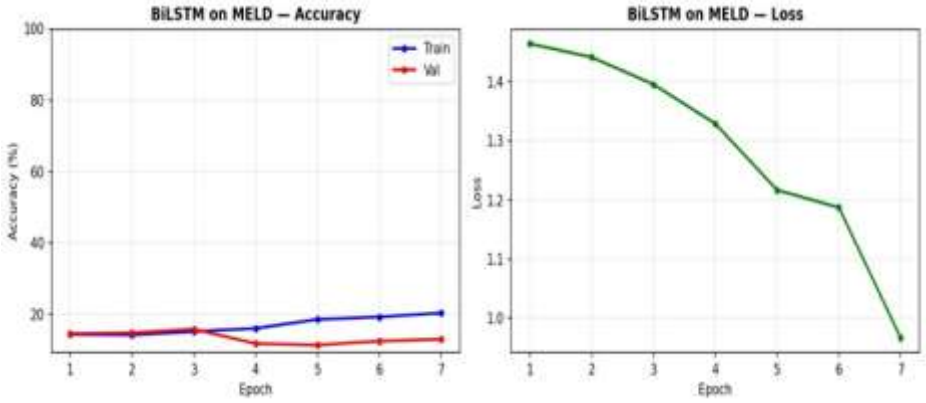


Figure 6: BiLSTM on MELD – Accuracy and Loss over training epochsTraining Curves – CNN-BiLSTM

Training Curves – CNN-BiLSTM

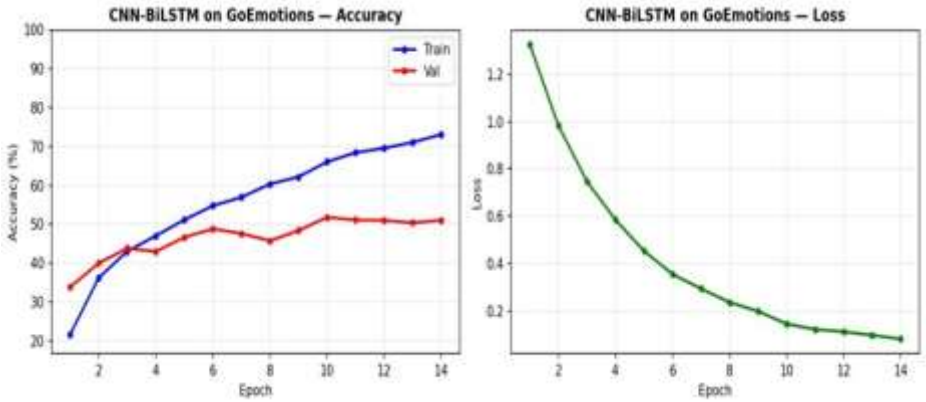


Figure 7: CNN-BiLSTM on Sentiment140 – Accuracy and Loss over training epochs

Figure 8: CNN-BiLSTM on GoEmotions – Accuracy and Loss over training epochs

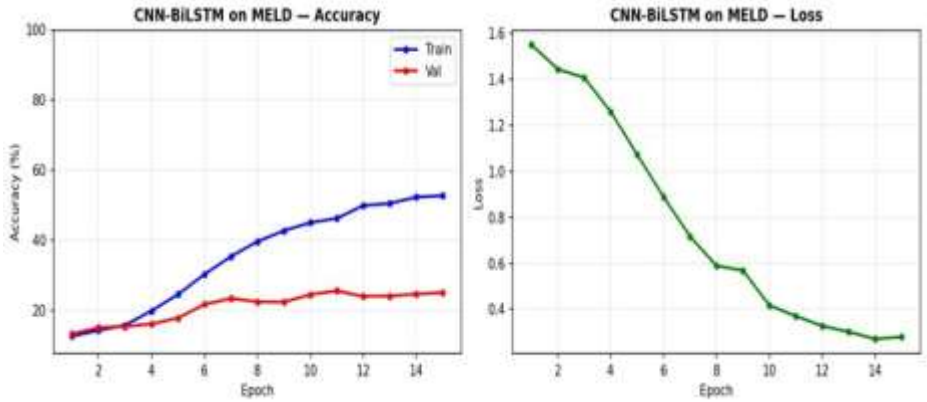


Figure 9: CNN-BiLSTM on MELD – Accuracy and Loss over training epochs

IV. RESULTS AND DISCUSSION

4.1 Overall Performance Comparison

Table 4.1 presents the accuracy and macro-averaged F1 scores for all three model architectures across all three datasets. The most salient pattern in the results is a consistent and steep degradation in performance as the task transitions from binary tweet classification to fine-grained Reddit emotion recognition to conversational utterance-level emotion classification. This pattern holds across all three architectures and is more pronounced than the differences between architectures within any single dataset.

Table 4.1: Performance Comparison

Dataset	Model	Accuracy
Sentiment140	TextCNN	80.86%
Sentiment140	BiLSTM	81.76%
Sentiment140	CNN-BiLSTM	81.82%
GoEmotions	TextCNN	50.10%
GoEmotions	BiLSTM	53.07%
GoEmotions	CNN-BiLSTM	51.78%
MELD	TextCNN	28.35%
MELD	BiLSTM	15.82%
MELD	CNN-BiLSTM	25.56%

4.2 Performance on Sentiment140 Dataset

The classification task on Sentiment140 dataset across the three models used converged to an average accuracy of around 81%. Training curves demonstrated a stable convergence of the models between 5 and 8 epochs, the validation loss was similar to the training loss which indicates that the model doesn't overfit to the training data. The confusion matrices that amount of positive class instances misclassified as negative and negative classes misclassified as negative was approximately equal.

The performance of different architectures only this task was nearly equal. If the problem is simple, such as only two categories used, text is very short and informal and straight to fan aspect type of semantic meaning (such as describing an image), then architecture selection makes little difference on the output. The reason those models can only do so well is not because the models are flawed, but rather that there were errors in how data was previously labeled automatically.

This is especially true for tweets that're ironic or sarcastic. The labeling method can make mistakes in these cases. The result shows that all three architectures are working correctly. They are doing what we expected. This gives us a base to understand results from more complex datasets. We can trust our method, for datasets. Now we can move on to complex ones.

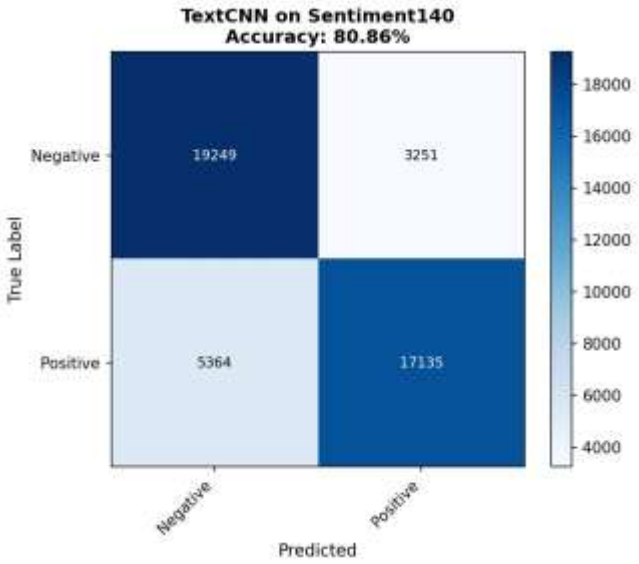


Figure 1: TextCNN on Sentiment140 – Confusion Matrix (Accuracy: 80.86%)

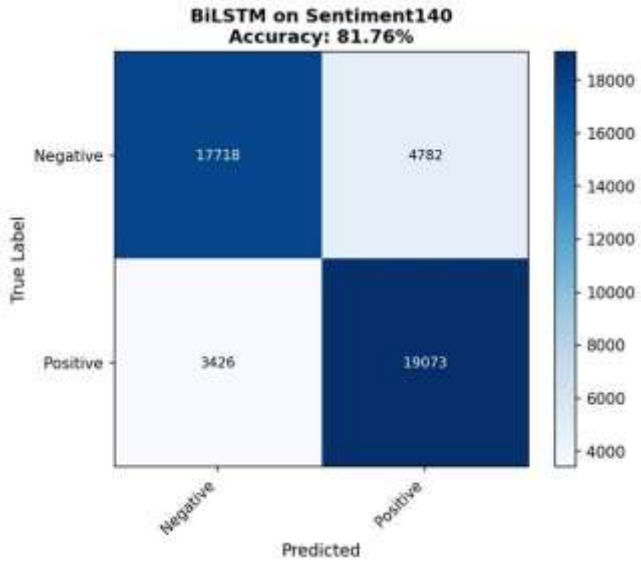


Figure 11: BiLSTM on Sentiment140 – Confusion Matrix (Accuracy: 81.76%)

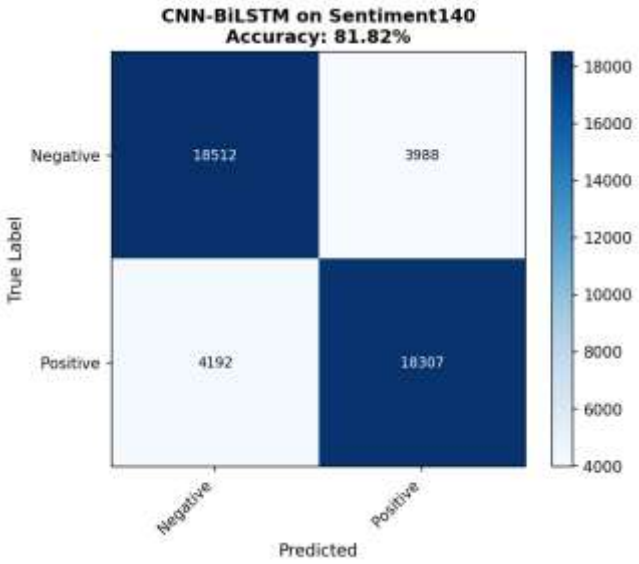


Figure 12: CNN- BiLSTM on Sentiment140 – Confusion Matrix (Accuracy: 81.82%)

4.3 Performance on GoEmotions Dataset

The performance on GoEmotions wasn't great, with accuracy ranging from 50 to 53% and macro F1 scores between 32 and 35%. But considering it's a 27-category classification problem, this is still way better than chance, which would be around 3.7%. This suggests that the models are actually learning some useful features. However, when you look at the macro F1 scores, it's clear that the models are doing okay with the more common categories, but struggling with the finer details between less common categories that are similar in meaning. This means the models are good at telling apart the big differences, but not so good at picking up on the smaller ones.

BiLSTM achieved the highest performance on GoEmotions (53.07% accuracy, 34.90% macro F1), providing marginal but consistent improvement over TextCNN. This advantage is attributable to the sequential modelling capacity of the recurrent architecture: fine-grained emotion distinctions in Reddit comments frequently depend on sentence-level structure and the sequential relationship between phrases, which BiLSTM captures more effectively than the local n-gram features of TextCNN.

When we look at how well the model does for each class in GoEmotions, we see some patterns that keep happening. For example, the model often gets "remorse" mixed up with "sadness", "admiration" with "approval", and "curiosity" with "interest". But this isn't just the model making mistakes - it's actually because these emotions are really similar and can be hard to tell apart, even for humans. When people were labeling the data, they didn't always agree on what emotion something was, and the model is just reflecting those same disagreements. So, it's not that the model is failing, it's just that the lines between these emotions can be blurry.

The BiLSTM model showed signs of overfitting when used with the GoEmotions dataset. What happened was that the training accuracy just kept getting better and better, even after the validation accuracy had stopped improving at around epoch 7. The limitation of the model to memorize training samples rather than learning general patterns was showcased. This indicates that the model performs well on the training data but since it didn't learn patterns the model could not be applied to new unseen data. In general context, this may be due to the smaller size of the GoEmotions dataset with only 58,000 examples compared to the large number of evaluation parameters present in the BiLSTM architecture.

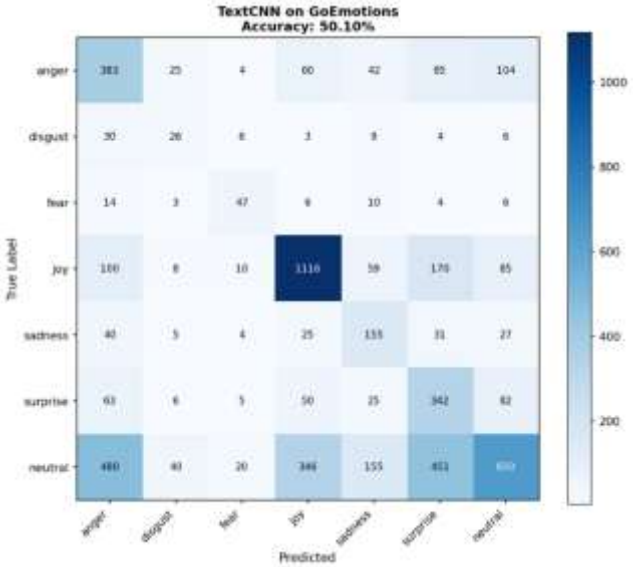


Figure 13: TextCNN on GoEmotions – Confusion Matrix (Accuracy: 50.10%)

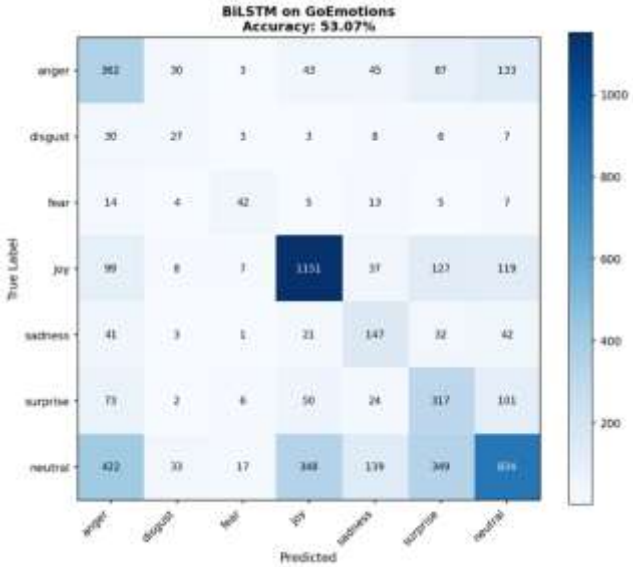


Figure 14: BiLSTM on GoEmotions – Confusion Matrix (Accuracy: 53.07%)

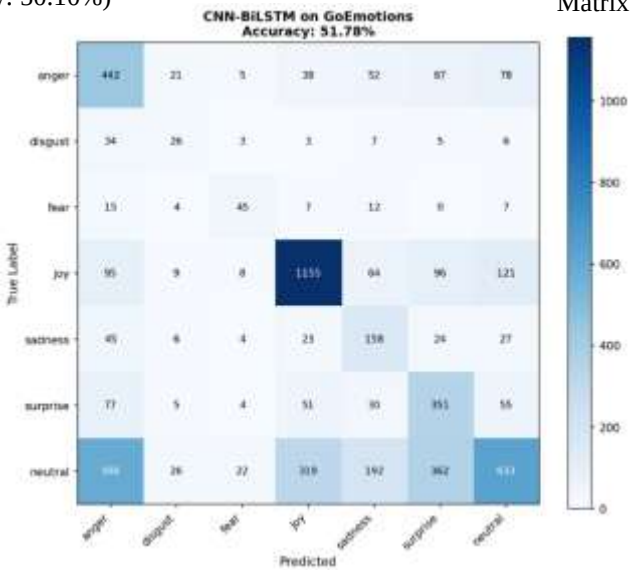


Figure 15: CNN- BiLSTM on GoEmotions – Confusion Matrix (Accuracy: 51.78%)

4.4 Performance on MELD Dataset

Because the MELD results demonstrate two counterintuitive but clinically significant patterns, they deserve an in-depth investigation. One, TextCNN (28.35% accuracy) far performs BiLSTM (15.82% accuracy) and CNN-BiLSTM (25.56% accuracy),

so the simple model outperforms better their complex models in a non-intuitive manner scenario that does with regard to the expected complexity ordering regarding the theoretical expressiveness of these models. Second, as expected from the macro F1 scores observed for all three models, validation accuracy is consistently substantially below those scores (confirms systematic misclassification of rare categories).

The reversal in architecture performance ordering of the two main findings can be explained via properties of MELD utterances. MELD has a very high percentage of very short utterances (single word responses, discourse markers, fragments). We can see that the median utterance length in MELD is about 7 tokens; this makes it a lot shorter than Sentiment140 or GoEmotions. At such lengths, where the number of time steps is potentially much larger than 17, BiLSTM's ability to model long-range sequential dependencies within an utterance does not really carry over - there is too little sequence on which a recurrent mechanism can operate within an utterance. TextCNN, which deals with local n-gram features rather than maintaining recurrent state, is less penalised in this case and by default more accurate.

The second finding, i.e. the near-zero recall on low-frequency categories, is indicative of one known failure mode of standard supervised classifiers on highly imbalanced datasets. Even after using weighted loss functions during training, all these three models revert to predicting the majority neutral class when they are given inputs that are somewhat ambiguous. Because utterances in the dataset are ambiguous enough when taken out of conversational context, a majority class default is almost always more cost effective than attempting to differentiate fear from surprise or disgust from sadness, 46.9% of MELD training examples are neutral cases.

MELD has poor performance not due to its architecture but due to its structure. Meaning is not a property of utterances in isolation, even though emotional meaning (or at least emotional relevance) derives from the interaction of particular types of responses to individual utterances. The mere word "okay" takes its emotional valence from who said it, what went before, and where the conversation naturally has been heading in one direction or another. At test time, all of this information was unavailable to the modeling techniques evaluated here, which operated on each utterance independently. This is not a problem that can be solved by better model architecture or more astute hyperparameter tuning; it calls for an input representation that quantitatively encodes the conversational history with the target utterance.

These results directly inspired the design of the confidence filtering step in the proposed framework. If a model is associated with high uncertainty on such short, contextually dependent utterances then it is, in an important sense, doing nothing wrong as it has identified there to be very little or nothing that can reliably be inferred from the input available. This recognition is formalised by a confidence filtering mechanism that mitigates the propagation of noisy inferences downstream.

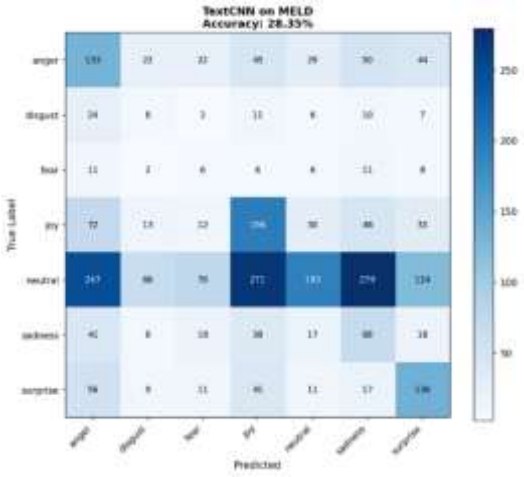


Figure 16: TextCNN on MELD – Confusion Matrix (Accuracy: 28.35%)

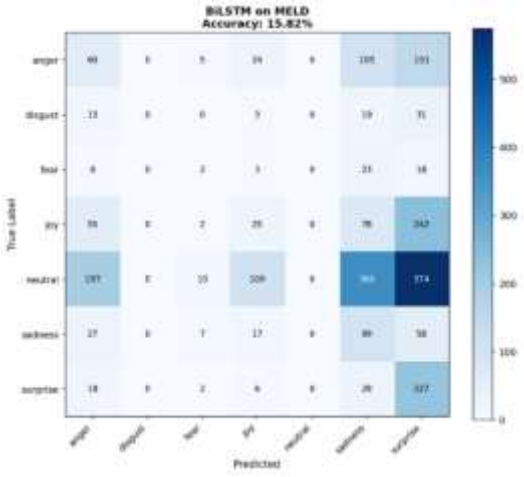


Figure 17: BiLSTM on MELD – Confusion Matrix (Accuracy: 15.82%)

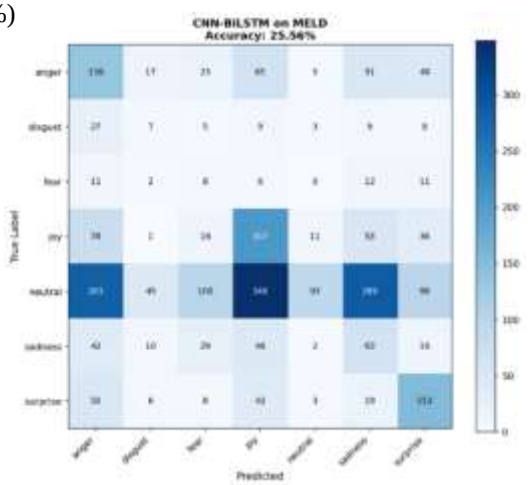


Figure 18: CNN- BiLSTM on MELD – Confusion Matrix (Accuracy: 25.56%)

5. DISCUSSION

5.1 Importance of Context in Emotion Understanding

For decades, linguistics and psychology have established a theoretical claim about the meaning of an utterance —specifically its emotional valence— that was strongly supported by experimental results reported in the previous section the meaning of an utterance is not an intrinsic property of the utterance in isolation, but a relational property jointly determined by the utterance and its communicative context. The interpretation of ambiguous phrases such as “that is fine” maybe different for people in varying

contexts. People may display phases of acceptance, surrender, sarcasm, anger or discomfort, which of these views is correct majorly depends in what context the sentence was used.

The implications of the contextual dependence don't limit themselves to the MELD performance metrics along with being theoretically interesting. It cleverly follows that any system intended to detect emotional state from text will fail systematically: in a way, any such signal tends to be expressed as contextually embedded, pragmatically indirect forms of speech. A message about burnout is not going to say "I'm burnt out." Instead, it is more likely to come in the form a series of short, one-word answers; delayed responses; less participation with team communications; and slight changes in language that can only be recognised as pulling clues when looking at that feedback against the context of how that person typically communicates.

This prompts dual-tracked framework to tackle such challenges. Our confidence filtering mechanism also avoids associating individual utterances with misinformation by suppressing inferences when they are uncertain. The structural modification required to solve the dilemma at the sequence level is the extension into genuine conversational context modelling we outline in next section. Collectively, these design choices represent a recognition that context is not a complicating factor to be designed around but rather an essential part of the problem at hand.

5.2 Interpretability and Explainability

One of the main criticisms levelled against deep neural networks approaches used for affective computing is that they are black boxes: They output results without providing explanations of why, meaning that it is impossible to audit their reasoning, failure modes, or which features contributed to arriving at a particular conclusion. This is a hassle for an engineer in low-stakes settings but an actual ethical concern in contexts of well-being inference where the model outputs may be applied to making employment, treatment or welfare allocation decisions about individuals.

We design this pipeline architecture in this thesis with interpretability as a first-class design constraint instead of an afterthought. The framework delineates the steps linking input text to well-being output by performing emotion classification, sentiment classification, and well-being inference in separate inspectable stages. A user can analyze the emotion distribution of Stage 1 output, determine whether the prediction with highest probability is reasonable, see if passed through the confidence filter, what sentiment classification it received, and which combination led to which well-being inference.

This intermediate-prediction transparency does not address the core opacity of the neural classifiers themselves—why, for a given input, one would assign high probability to sadness in a run BiLSTM; what it has learned internally cannot be directly inspected from that score alone. The use of attention visualisation techniques to identify the impact of input tokens towards specific emotion predictions would be an impactful development in this direction. It should be surfaced to end users, as a feature of the well-being monitoring interface, not kept as an internal model diagnostic.

The explainability requirement entails another consideration when it comes to model selection. Because of this, Transformer-based models like BERT would probably perform significantly better than the evaluated architectures on GoEmotions and MELD. Nevertheless, they are orders of magnitude more complex to interpret than for their CNN and LSTM counterparts. We must measure explicitly the interpretability trade-off with a deployment-oriented mindset in mind to make the decision about future system design rather than basing it on benchmark performance alone.

5.3 Well-Being Inference Framework

The fourth step converts a pair of emotion label, confidence score and sentiment polarity, in a well-being interpretation expressed with chunks that are easily understandable by non-technical users.

Emotion	Sentiment	Interpretation	Priority
Joy	Positive	Engaged, healthy	Low
Sadness	Negative	Emotional fatigue or stress	High
Anger	Negative	Frustration or conflict	High
Fear	Negative	Anxiety or uncertainty	High
Neutral	Neutral	Stable baseline	Low
Disgust	Negative	Dissatisfaction	Moderate
Surprise	Mixed	Unexpected event	Low–Moderate

Mapping Principles — A few design principles go into this mapping. First, it provides a mapping between high priority acute well-being concerns (High priority: sadness-negative, anger-negative, fear-negative) and those that are less clear from the data and need monitoring over time (Moderate priority: disgust, suppressed emotional expression). This distinction is important, operationally: it enables the system to detect true alarms without creating alert fatigue from a larger volume of observation priorities at the moderate level.

Second, the mapping is explicit about instances where emotion and sentiment are in conflict. Sad, neutral sentiment or anger, neutral sentiment may mean the general emotional suppression is taking place: the employee is hiding negative affect under professionally neutral language. Rather, these patterns are designated Moderate priority with the explicit rationale of being tracked over time and not requiring immediate action.

Third, and importantly, mapping is provisional. What a particular emotional pattern means to an individual depends on contextual factors, such as individual differences, cultural background, communication style and professional context. This mapping is intended to produce hypotheses for human evaluation, not conclusions for machine implementation. This distinction is maintained by the interface recommendations.

For applications in the workplace, it is recommended that well being inference layers operate at an aggregate team level as opposed to individual level. Monitoring the distribution of wellness signals among a team through time is much more accurate for indicating group well-being and significantly less ethically problematic than tracking individuals. Changes over time in the percentage of a team that have High-priority observations is more informative and less idiosyncratic than any one person being identified as a red flag.

6. FUTURE WORK

Several clear priorities emerge from this work.

The most immediate improvement would be replacing the BiLSTM emotion classifier with a fine-tuned BERT encoder. The performance gains on MELD, in particular, would likely be substantial.

The most important structural extension is genuine conversational context modeling – processing dialogue sequences rather than individual utterances, tracking speaker emotional state across turns and incorporating the five to ten preceding exchanges as context for each prediction.

Over a longer time horizon, adding audio features – tone of voice, pace, hesitation – would bring the system closer to how humans actually read emotion in real-time communication.

The well-being inference layer needs real training data: annotated corpora in which text is paired with validated psychological assessments. Building this kind of dataset is genuinely difficult but it is the only way to move beyond static rules.

Finally, the interpretability work needs to go deeper. Attention visualization and counterfactual explanation tools should be systematically evaluated – not just as technical tools but as user-facing features that help non-expert users understand and appropriately trust the system's outputs.

7. CONCLUSION

The aim of this study was to identify, implement and test a multi-level sequential approach to text-based inference of psychological wellbeing. It combines emotion recognition, a confidence-based filtering and classification approach dependent upon emotional context, and a pragmatic rule-governed well-being inference component into one technical but interpretable pipeline.

Three deep learning models, TextCNN, BiLSTM and CNN− were examined in this study. For binary sentiment classification, BiLSTM performed reasonably well achieving ~81% on the Sentiment140 dataset. However, they performed at the level of a coin flip when identifying emotions with more granular detail, achieving only 50-53% accuracy on the GoEmotions dataset. When they tried to stop detecting emotions in conversations, it got worse at 15-28% on the MELD dataset. Diving deeper into these failures revealed that the core issue was insufficient context for the models; Although they scored utterances in isolation, these lacked the context of a conversational history that could indicate emotion.

This research determined three main decisions we took into consideration when designing the framework. We previously incorporated emotion recognition as a prerequisite problem to sentiment classification, since emotions are an important component of how we express feelings. And this is how it plays out: first emotions, then we can figure out the sentiment. Second, we put a filter on it such that emotion predictions were only used when they seemed fairly confident. And if we're not sure of a word, we do not use it because nobody wants to be wrong. We consider it informative uncertainty not the thing to be overcome! Third, we separated the components of the framework that determines well-being from the classification portions.

The conversation highlighted two factors that are frequently dismissed as peripheral to technical research but should be viewed as core issues in responsibly developing systems for inferring well-being. The interpretability analysis showed that the intermediate predictions in our proposed framework showcased that such systems work more effectively when their reasoning process is defendable and auditable. With regards to the ethics assessment, four critical attributes of the ethics system, namely, transparency, participation, bias evaluation, and limitation of uses of the system, should form the support system of the framework.

We need to make systems that care about people than they care about being in control. This requires a lot of knowledge and we have to be honest about what we do not know. We also have to be open with the people who are using these systems. We have to remember that we are doing this to help people not to control them.

By being transparent thinking about what's right and wrong and giving people power we can make systems that really help people do well. We want to make systems that care about well-being and we want human well-being to be the most important thing. Human well-being is what matters most. We have to keep that in mind when we make these systems.

REFERENCES

[1] Merhbene et al. 2022. BurnoutEnsemble: Augmented Intelligence to Detect Indications for Burnout in Clinical Psychology. Frontiers in Big Data.

[2] Järvinen et al. 2024. A Behavior and Emotion Recognition Framework for Emotion-Aware Services in Physical Spaces. IEEE CBMI.

[3] de Boa Esperança. 2021. AE Thesis Fall 2021. Purdue University.

[4] Naik et al. 2025. AI-Driven Burnout Detection & Eco-Volunteering: A Sustainable Employee Well-Being Model. Journal of Environmental Science

[5] Emexidis et al. 2025. Analyzing Employee Job Satisfaction Through Sentiment Analysis for Enhanced Workplace Improvement and Business Success. Theoretical and Applied Ergonomics.

[6] Ebayan et al. 2025. Analyzing Employee Sentiment of Companies via Social Media and Job Sites for Informed Job Hunting. IEEE IAICT.

[7] Dixit et al. 2023. Analyzing Textual Data for Mental Health Assessment: NLP for Depression and Anxiety. IEEE UPCON.

[8] Pradhan et al. 2021. Application of Deep Learning Techniques to Detect Workplace Burnout. Academic Journal.

[9] Li et al. 2025. Assessing Occupational Burnout in Psychiatric Professionals Using Multimodal Emotion Recognition Methods. IEEE ICHMS.

[10] Kaur et al. 2025. BERT-based Deep Learning Approach for Classification of Mental Health Disorders. IEEE InCACCT.

[11] Krabashini et al. 2024. Enhancing Emotion Detection in Chatbots for Improved Mental Health Support. IEEE ICISBME.

[12] Pereira et al. 2025. Deep Emotion Recognition in Textual Conversations: A Survey. Artificial Intelligence Review.

[13] Wang et al. 2023. DeepEmotionNet: Emotion Mining for Corporate Performance Analysis. Information Processing & Management.

[14] Hasan et al. 2025. Early Detection of Occupational Stress. PLOS ONE.

[15] Ghandeharioun et al. 2019. EMMA: An Emotion-Aware Wellbeing Chatbot. ACII.

[16] Madampe et al. 2025. EmoReflex: An AI-Powered Emotion-Centric Developer Platform. Automated Software Engineering.

[17] Roemmich et al. 2023. Emotion AI at Work: Implications for Workplace Use. CHI (ACM).

[18] Yadav et al. 2025. Emotion-aware Ensemble Learning (EAEL). IEEE Access.

[19] Dubey et al. 2025. Employee Satisfaction Mining Using LLMs. IEEE CIACON.

[20] Costa & Veloso. 2015. Employee Analytics through Sentiment Analysis. ACM SBBD.

[21] Hermawan et al. 2024. Mental Health with Machine Learning: A Prediction-Based Intervention Chatbot for Mental Health Conversations. IEEE EECSI.

[22] Ben Chaabene et al. 2025. Ethical and Explainable Pedagogical Interventions. Procedia Computer Science.

[23] Üveges & Ring. 2023. HunEmBERT: A Fine-Tuned BERT Model for Classifying Sentiment and Emotion in Political Communication. IEEE Access.

[24] Bhadauriya et al. 2025. Leveraging Emotional Intelligence Metrics and NLP-Driven Sentiment Analysis for Predictive Workplace Mental Health Monitoring. IEEE SENNET.

[25] Garg et al. 2024. Machine Learning Driven Analysis of Mental Health Indicators. IEEE ICESC.

[26] Poddar et al. 2024. Mental Health Monitoring in Students: A Classifier-Based Machine Learning Approach. IEEE CIACON.

[27] Sujal et al. 2022. Mental Health Analysis of Employees Using Machine Learning Techniques. IEEE COMSNETS.

[28] Sehgal et al. 2023. Mental Health Awareness Using Machine Learning. IEEE Conference.

[29] Poddar et al. 2024. Mental Health Monitoring in Students (Extended Study). IEEE CIACON.

[30] Hemnath. 2025. Integrating Natural Language Processing with BERT and LSTM for Employee Sentiment Analysis in HRM. IJAHSS.

[31] Yang et al. 2024. Personalized Mental Health Interventions Using Generative AI and Multimodal Data. IEEE Conference.

[32] Zhou et al. 2022. Predicting Meeting Success With Nuanced Emotions. IEEE Pervasive Computing.

[33] Bíró et al. 2023. Real-time Artificial Intelligence Text Analysis for Identifying Burnout Syndromes in High-Performance Athletes. IEEE Conference.

[34] Ye et al. 2023. Textual Emotion Recognition Based on ALBERT-BiLSTM and SVM-NB. Soft Computing.

[35] Alruily. 2023. Sentiment Analysis for Predicting Stress Among Workers. Alexandria Engineering Journal.

[36] Sharma et al. 2024. Sentiment Analysis: Decoding Workspace Emotions. IEEE Conference.

[37] Shameen & Geetha. 2025. Sentiment Analysis of Work-From-Home Practices Among Indian IT Professionals Using Transformer-Based Models. IEEE Conference.

[38] De Silva et al. 2022. Solution to Measure Employee Productivity with Employee Emotion Detection. IEEE Conference.

[39] Nijhawan et al. 2022. Stress Detection Using Natural Language Processing and Machine Learning Over Social Interactions. Journal of Big Data.

[40] Metallinou et al. 2013. Two-Stage Dimensional Emotion Recognition by Fusing Predictions of Acoustic and Text Networks Using SVM. IEEE ICASSP.

[41] Men & Yue. 2019. Creating a Positive Emotional Culture: Effect of Internal Communication and Impact on Employee Supportive Behaviors. Public Relations Review.

[42] Gupta et al. 2023. Emotion Detection from Text Data Using Machine Learning for Human Behavior Analysis. IEEE Conference.

[43] Alzahrani et al. 2024. Leveraging Large Language Models for Emotion Recognition: Case of Employee Satisfaction Surveys. Springer Conference.

[44] World Health Organization. 2022. Mental Health and Well-Being at the Workplace. WHO Report.

[45] Singh et al. 2021. Performance Evaluation of Reddit Comments Using ML and NLP in Sentiment Analysis. Journal of Big Data.

[46] Yadav et al. 2020. A Review on Sentiment Analysis and Emotion Detection from Text. ACM Computing Surveys.

[47] Pang & Lee. 2008. Automatic Sentiment Analysis in Online Text. Foundations and Trends in Information Retrieval.

[48] Medhat et al. 2014. Text Sentiment Analysis: A Review. Journal of Artificial Intelligence Research.

[49] Bokhari et al. 2022. An Efficient Classification Algorithm for Employee Well-Being Prediction Using Deep Learning. IEEE Access.

[50] Li et al. 2024. Transforming Emotions: A Comprehensive Review of Text Emotion Detection with Transformer Models. Artificial Intelligence Review.

[51] Kalra, Sahi & Kaur. 2024. Elevating Happiness by Analyzing Socioeconomic Factors. International Research Journal of Multidisciplinary Scope.