SCIENTIFIC CULTURE
www.sci-cult.com

SCIENTIFIC CULTURE, Vol. 12, No. 4, (2026), pp. 7369-7393
Open Access. Online & Print

DOI: 10.5281/zenodo.12426747

THOUGHTBERT: A NOVEL FRAMEWORK FOR CONTEXTUAL THOUGHT CLASSIFICATION

Jitendra Singh[1*], Geeta Sharma[2]

[1] [*]Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab, India, Corresponding Author Email: jitendra.42100212@lpu.in*
[2] School of Computer Application, Lovely Professional University, Phagwara, Punjab, India, Email: geeta.26875@lpu.co.in

Received: 25/10/2025
Accepted: 21/03/2026

Corresponding Author: Jitendra Singh (jitendra.42100212@lpu.in)

ABSTRACT

Human thoughts profoundly influence decision-making, mental health, and overall well-being. This study introduces ThoughtBERT, a novel framework for contextual thought classification, aimed at advancing automated mental health support and cognitive research. A novel dataset of 5,018 thoughts was constructed using Google Forms surveys and generative AI, with thoughts labeled across four categories: Positive, Negative, Necessary, and Peripheral. Human annotators labeled 40% of the dataset (1,976 thoughts) using rigorous guidelines, achieving a Fleiss Kappa inter-annotator agreement of 0.84. An active learning pool-based sampling strategy employing Support Vector Machine (SVM) iteratively extended labeling to the full 4,944-thought dataset. BERT was fine-tuned with auxiliary features (sentence length, word density, focus word weighting) and a custom attention layer to form ThoughtBERT. ThoughtBERT achieved 96.6% accuracy and 96.8% F1 score, outperforming SVM (89.6%), RNN, LSTM, Bi-LSTM, and standard BERT (94.2%) across all evaluation metrics including precision, recall, and specificity. ThoughtBERT demonstrates that domain-specific fine-tuning of transformer models, augmented with handcrafted linguistic features, yields superior performance in nuanced thought classification tasks. The framework has been deployed in a real-time Thought Classification App within educational institutions, enabling counselors to identify students experiencing repetitive negative or peripheral thinking patterns, facilitating timely mental health interventions.

KEYWORDS: Active learning approach, BERT, Deep Learning, Human thoughts, Machine Learning, Novel Dataset, Thoughts Classification

1. Introduction

Making decisions is important in personal and work life. The decisions made depend on the information at hand and also on how the decision-maker feels. A calm mind, filled with mostly positive thoughts, tends to make better decisions [8]. On the other hand, a mind filled with negative and unnecessary thoughts is often unstable and can result in bad decision-making [1]. This connection between thinking and making decisions shows how important it is to know and look at the inner thoughts that affect choices [9].

Researchers employ the sentiment analysis to understand opinions and feelings. This involves learning on machines in order to acquire qualitative information in the form of text information [12]. They assist in defining the position of people and their attitude, grouping the text based on unpleasant or pleasant emotions [14]. This is proven by recent sources that NLP methods can play a significant role in the early recognition of mental illness by social media stories and medical history [41]. This builds our work lying on the basis where intent recognition based on text is essential.

Social media-based sentiment analysis is gaining traction in early mental health screening, forming a base for our intent-aware architecture [26]. A systematic review confirms the promising role of NLP tools in diagnosing mental health conditions via unstructured text [33]. Emotion mining is increasingly being used for early detection of suicidal ideation, which validates our focus on emotion-aware intent detection [31]. ML techniques applied on Arabic social media data have shown promising results in suicidality detection, further supporting our study's goal [22].

The introduction section of this paper is divided into three parts. In the first part, researchers have shown their background research on which the current study is based. In the second part, we found the research gap in background work, followed by framing the objectives of current research in the third one.

1.2 Background

This research is an extended work of our recent study by the authors of this paper. The researchers proposed a novel approach in previous work as shown in Fig. 1. This work proposes further enhancements to thought classification using active learning. The methodology in the diagram is to fill the research gaps obtained by detailed literature reviews[16],[17].

The background study aims to fill the knowledge gap in sentiment analysis by creating a new dataset focused on understanding human thoughts. In previous work, based on the research gap we found, set and accomplished 4 objectives of the research.

Objective 1: "To categorize human thoughts into distinct classes: Positive, Negative, Necessary, and Peripheral, which are pertinent to assessing mental stability. This will involve collecting and manually annotating a subset of human thoughts from relevant sources (e.g., Google Form surveys, generative AI) for the initial dataset development." Since the 'classification categories of thoughts' were completely novel, researchers decided to develop a novel dataset. The dataset was narrowed to college-going participants from G.L. Bajaj Institute of Technology and Management and Sharda University, Knowledge Park III, Greater Noida, U.P., India. 5125 thoughts were collected and 2050 were sent to human annotators in two groups with 3 annotators each. Annotators annotated thoughts in four categories: Positive, negative, necessary, and peripheral by following the guidelines to annotate. After rejecting 74 thoughts (due to no consensus among annotators), 1976 thoughts (around 40% of the entire dataset) were annotated.

- **Positive thoughts** are constructive and uplifting and promote well-being, happiness, and emotional stability. These thoughts often involve gratitude, optimism, and feelings of joy or contentment.
- **Negative thoughts** are harmful, critical, or stress-inducing. They often stem from fear, worry, resentment, or feelings of inadequacy and may lead to emotional distress.
- **Peripheral thoughts** are trivial, unproductive, and consume mental energy without offering meaningful or actionable insights. These thoughts often revolve around irrelevant or unnecessary details.
- **Necessary thoughts** are practical, task-oriented, and essential for decision-making or daily functioning. These thoughts often revolve around planning, problem-solving, or understanding important issues.

From Figure 1, Step 1 and Step 2 are dedicated to accomplishing Objective 1.

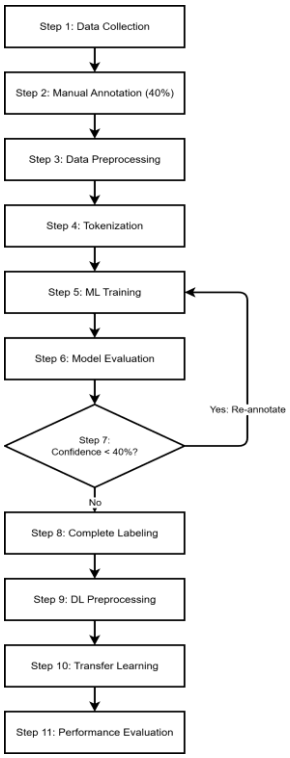

Figure 1. Novel Framework for Thought Classification.

Objective 2: "To preprocess the collected data through tokenization, cleaning, normalization, and vectorization, ensuring it is ready for model training and analysis. Additionally, to expand the initially labeled dataset using active learning and a pool-based sampling approach, incorporating human annotations to cover a more extensive set of unlabeled data." By implementing steps from 3rd to 8th, Objective 2 was achieved. The critical research point of the data preprocessing was the inclusion of specific stopwords (tense indicators and personal pronouns), allowing the model to better understand and classify the sentiments of the thoughts. The initially labeled dataset was trained and tested on machine learning algorithms (Logistic Regression, Naïve Bayes, Random Forest, and Support Vector Machine) that performed well for a limited set of textual data [16],[17]. As SVM outperformed, we continued the implementation of an active learning pool-based sampling approach to annotate unlabeled thoughts based on confidence score analysis as shown in step 7 in Fig. 1. Finally, we annotated the whole dataset on completion of Objective 2.

Objective 3: "To train and evaluate various deep learning models (RNN, LSTM, and BiLSTM) on the novel dataset. This includes implementing feature engineering and hyperparameter tuning to improve model performance." Steps 9 and 10 from Fig. 1 were dedicated to achieving Objective 3. Step 9 included removing mentions and hyperlinks, converting to lowercase, decoding text, removing punctuation marks, including specific stopwords for improving classification accuracy, and lemmatization. In the 10th step, RNN, LSTM, and BiLSTM were trained and tested on preprocessed datasets. For this purpose, researchers used Adam optimizer, Softmax activation function, 0.001 learning rate, 64 batch size, and 15 training epochs. In the 11th step, we achieved the last objective of the previous study.

Objective 4: To use predetermined performance criteria (Accuracy, Precision, Recall, Specificity, and F1 Score) to assess how well the suggested sentiment analysis framework performs compared to other frameworks already in use. In this step, Bi-LSTM outperformed.

1.2 Research Gap

In the last study, researchers quoted introducing transfer learning as future work. Even with big steps forward in sentiment analysis and thought classification, there are still problems with getting high accuracy and efficiency due to the limits of older models like RNN, LSTM, and BiLSTM. These models commonly have trouble with long connections, keeping context, and being efficient with computing, which makes them not perform well on tough tasks. In addition, in comparison to simple BERT models that are useful in a better context understanding, its simpler versions lack modifications required to

accomplish certain tasks, such as distinguishing between Necessary and Peripheral thoughts.

The previous studies examined relevant ones such as the process of making datasets, preprocessing, and training models with the help of standard machine learning and outdated deep learning approaches. However, not using metadata, the inability to highlight category-specific signals, and relying on standard loss functions made it hard for these models to effectively manage complex and uneven thought categories. Hence, it is urgently needed to boost both accuracy and efficiency by using better methods particularly aimed at thought classification.

1.3 Objectives of Research

This research mainly wants to suggest and check out a new system called ThoughtBERT. This system works on the weaknesses of older models and makes the BERT structure better to boost thought classification accuracy and speed.

Objective 1: Introducing transfer learning in the novel approach of thought classification.

Objective 2: Proposing novel framework 'ThoughtBERT', by fine-tuning in basic BERT model.

Objective 3: Comparing ThoughtBERT with RNN, LSTM, Bi-LSTM and BERT models on the performance parameters: accuracy, precision, recall and specificity.

With the accomplishment of these objectives, ThoughtBERT seeks to fill the gaps in current thought classification studies and provide a solid system for classifying thoughts with better accuracy and faster processing.

2. Related Work

2.1 Overview of Existing Sentiment Analysis Approaches

Sentiment analysis has become a field of interest within the domain of the natural language processing (NLP), and is concerned with the extraction and interpretation of sentiments conveyed in written material. The classical machine learning (ML) algorithms include Support Vector Machines (SVMs), Naïve Bayes classifiers, decision trees and have been commonly used in previous research by virtue of their simplicity and efficiency in structured data. An example of successful use of SVM in sentiment prediction using IMDB dataset was carried out by Chen [2] who conducted a seminal study that used SVM in sentiment classification obtaining a resolution of almost 90 percent, which indicated that SVM is competitive in binary classification systems in cases where the input data are well processed. But feature engineering was oftentimes a significant portion of these methods, and that did not address contextual dependencies in text. Sentiment analysis has developed by both rule-based and machine learning methods, but it has failed in intent differentiation, which this work effectively bridges in the first place [39]. Extending on this, machine learning has demonstrated the potential of identifying signs of depression through text informatics, on an encouraging note, and as a crucial marker of intent uniquely distinguished within a mental health context [32].

Natural Language Processing has become an essential tool in identifying harmful content on social media that is the foundation of contemporary online safety systems [29]. However, it still has major issues especially with the detection of hate speech in low-resource languages that do not have enough resources to be effective which could also be filled with multilingual models such as ThoughtBERT as of now [23]. Such technical segments also become particularly urgent when used to address urgent situations, with the phenomenon of ML-based sentiment analysis during COVID-19 as a clear example, where emotion recognition happened as the direct input to the overall response in the area of the public health context [38]. The area covered by sentiment analysis has been broadened through a comprehensive survey of the healthcare, marketing, and political fields that supports the usefulness of contextual modeling such as ThoughtBERT[24]. Task-specific tuning strategies such as ours are essential with the introduction of LLMs to become both efficient and easy to understand on the one hand as well as effective on the other hand [40].

Deep learning outshot the old-fashioned approaches to ML, and in no time, they were able to select Latent layers of data, which was impossible to the ancient models. RNNs, LSTM and even Bidirectional LSTM have been applied to sentiment tasks. As an example, one article by Vaghela et al. [18] focused on LSTM in aspect-based sentiment analysis without modifying the self-attention component and even slightly changing the LSTM activation to improve the candidate accuracy. The process tends to retain longer text relationships and, in the vast majority of cases, even surpasses previous models on such metrics as the SemEval dataset. Nevertheless, it is worthy to include that RNN-optimal configurations may also experience diminishing gradient concerns and are not always adept at capturing long-range dependencies as a research by Noh has found out, among others, vanishing gradient issues, and they do not necessarily capture long-range dependencies effectively [11]. An extensive literature review of

emotion and personality detection indicates that AI models have demonstrated effectiveness with context-sensitive sentiment modeling and hence our approach of using emotion cues in intent classification [37].

Limiting language resources, particularly non-English corpora such as the Arabic language, still pose a challenge in regards to the generalizability of offensive language models, despite that issue being addressed to some extent in existing applications and methods [21].

CNNs have more recently been offered a shot in sentiment analysis merely due to the fact that they are good at highlighting local features, however to some boundaries. A study by [20] on sentence classification using CNNs presented significant enhancements due to the direct n-gram feature pulling directly in text, which is rather impressive, 2020). But these models tend to fail in capturing the entire, worldly context the task of which in many tasks such as classification of thoughts is paramount.

2.2 Transfer Learning in NLP

Transfer learning has literally transformed the NLP picture by allowing us to recycle what enormous pre-trained models already do on everything. ELMO[13] and GPTGPT[7] and BERT[4] , among others, continue to demonstrate their ability to perform such a wide range of tasks as sentiment analysis, text-question-answer, and text-text classification.

The idea of BERT -based Bi-directional Encoder Representations of Transformers specially attracts attention due to its ability to look at the context in both directions, and therefore feel the meaning of language bit more deeply. Its performance on choices such as the GLUE benchmark and SQuAD was well-performing as shown by Devlin and others here [4]. This fine-tuning capability enables it to adapt rapidly to a new issue even with simple changes, which have been observed in studies by Clark [3] and Ethayarajh, not to mention the latter has made it seem like a barrier to making changes, as well as in their studies by Clark and Ethayarajh[5] (as noted in both articles).BERT variants combined with CNNs have set new benchmarks for sentiment classification, which we extend through our layered intent modeling [30].

New studies have gone even further by adapting BERTs to the very narrow applications such as sentiment and thought classification. As an example, Clarke[3] introduced an aspect-based sentiment task variant called IAN-BERT, which significantly increased its accuracy through the incorporation of domain knowledge and expertise on the matter (Clark, 2019). Similarly, studies indicate that BERT performance is satisfactory when it comes to mental health sentiment analysis in terms of identifying subtle context change and subtle sentiment.

2.3 Positioning BERT in Thought Classification

The trick of reading sentences both ways simultaneously is something that BERT can do, and, at least, it provides this method with an advantage in the context of contextual understanding that is essential in classifying one thought type in opposition to another. It does not merely chop bits of text like the older ones; rather it balances all word associations with an entire input. According to Devlin, this comprehensive viewpoint simplifies the process of realizing the biased edge between the acceptable idea and something more ad hoc as context and dependencies tend to provide the key [4]. It is also cool that BERT can be fine-tuned to accomplish specific tasks. In general, you can bring it to a more precise endpoint by blending in attention fine-tuning or even additional information such as metadata–and thus you have a model that is more aligned to what you require. Indicatively, a study has indicated that addition of auxiliary inputs indeed enhanced personalised text classification accuracy, something Rogers also reflected on later in 2020, on the same research topic, on the same subject matter, though omitting the use of auxiliary inputs, also making this assertion presumably to enhance accuracy and performance by removing intervening variables or items that can better predict results [15]. Consider that the new framework, ThoughtBERT, is an improvement of these strengths. Simply put, the main idea behind ThoughtBERT is the capacity of BERT to understand context on a more profound level, which is spiced with additional alterations such as a spin on self-attention and additional features. Generally, it can be said that in the majority of cases, ThoughtBERT fits best the problematic issues of thought classification, which opens up the successful method of BERT to new areas.

3. Methodology

3.1 Data Preparation

Raw text data were collected through two primary sources: a structured Google Forms survey administered to college students and generative AI (GPT-4) using controlled prompt engineering. Upon retrieval, all data were reviewed and categorized into four classes—Positive, Negative, Peripheral, and Necessary—forming the basis for supervised learning. Text preprocessing involved converting all text to lowercase, followed by the removal of URLs,

mentions, hyperlinks, non-ASCII characters, numerical values, and punctuation. Lemmatization was applied to normalize word forms (e.g., "running" → "run"), ensuring consistency across the dataset.
After the preprocessing had been completed, we had leaped into the identification of the major features that we required in the model. It was then sliced into training and test slices of data and a few machine learning models were put to test such as Logistic Regression, Naive Bayes, Random Forest, and Support Vector Machines. We evaluated them in terms of accuracy, precision, recall, F1 score, and specificity.
And then we brought in a lively element of learning. In essence, we examined the confidence scores of the model to warn against samples, which were dubious, and that would require a second investigation. This modified larger set of labeled data was retrained on the model and it was the version used to sort all the thoughts at the application stage. Eventually, this multi-layered construction provided us with a solid foundation to the deep learning experiments to come- such as LSTM, BiLSTM and BERT-based models.

3.1.1 Text Preprocessing Pipeline

Raw thought samples collected through Google Form surveys and GPT-4 augmentation contain considerable noise in the form of social media handles, hyperlinks, emojis, irregular casing, and morphological variation. A six-stage preprocessing pipeline was applied uniformly across all 5,125 samples prior to any feature extraction or model training. Each stage was designed specifically to address the linguistic characteristics of self-referential thought text, which differs fundamentally from product reviews or social media posts that most standard NLP pipelines are optimised for.

(i) Removing Mentions and Hyperlinks

Using the regex pattern *(@[\w]+|http\S+)*, all social media mentions (e.g., @pandey_om) and URL strings were stripped. These tokens carry no cognitive or semantic content relevant to thought classification and would otherwise occupy high-frequency term slots in TF-IDF feature space, diluting the signal from genuinely meaningful vocabulary.

(ii) Lowercasing

All text was converted to lowercase to eliminate case-based feature sparsity. Without this step, tokens such as "Happy", "happy", and "HAPPY" would be treated as three distinct features despite being semantically identical, artificially inflating the vocabulary and reducing classifier generalization.

(iii) Decoding Non-ASCII Characters

Emojis, Unicode symbols, and non-standard punctuation (e.g., 😊, 🌧️) were decoded and removed. While emojis may carry affective signals in social media contexts, their presence in thought samples was inconsistent and their encoding variability would introduce tokenization instability across system environments.

(iv) Punctuation Removal

All punctuation marks were stripped to reduce token-level noise and standardize input representations. In this dataset, thought samples are predominantly declarative statements where punctuation variation is stylistic rather than semantically meaningful for four-class cognitive categorization.

(v) Lemmatization

Using NLTK's *WordNetLemmatizer*, inflected word forms were mapped to their canonical base forms (e.g., "running" → "run", "was" → "be", "thoughts" → "thought"). Lemmatization reduces vocabulary size without discarding semantic content. This is particularly important for thought classification, where verb forms such as "planning", "planned", and "plans" all signal Necessary thought patterns and should be merged under a single representative token.

3.1.2 Custom Stopword Handling: An Innovative Retention Strategy

The most distinctive preprocessing contribution of this framework is its deliberate deviation from standard stopword removal practice. Conventional NLP pipelines remove all stopwords — a pre-defined list of high-frequency function words — on the assumption that these words carry no discriminative semantic content. While this assumption holds for topic classification or product review analysis, it is fundamentally inappropriate for thought-level classification for two critical reasons.

First, human thoughts are inherently self-referential. Unlike product reviews that describe external entities ("The battery life is poor"), thoughts describe the internal mental state of the speaker ("I am worried about the deadline"). Personal pronouns such as I, me, you, and they are therefore not noise — they are primary signals that indicate whether a thought is self-directed (more likely Necessary or Negative) or other-directed (more likely Peripheral).

Second, thoughts are temporally grounded. The cognitive category of a thought is often determined by its temporal orientation: a past-tense thought ("I was embarrassed") is typically Negative, a present-tense thought ("I am managing this well") may be Positive or Necessary, and a future-tense thought ("I will complete the report tonight") is strongly Necessary. Standard stopword removal eliminates tense-marking auxiliaries such as am, is, are, was, were, will, and had — destroying this temporal signal entirely.

Based on these observations, four categories of stopwords were selectively retained, as detailed in Table 1.

Table 1: Custom Stopword Retention Strategy for Thought Classification

Category	Retained Words	Linguistic Role	Relevance to Thought Classification
Tense Indicators	am, is, are, was, were, will, had, have	Temporal markers	Distinguish past regret, present worry, future planning
Personal Pronouns	I, me, you, they, we, my, your	Subjectivity markers	Identify self-referential vs. other-directed thoughts
Negation Words	not, no, never, neither, nor	Polarity shifters	Critical for flipping Positive to Negative class
Modal Verbs	must, should, could, need, ought	Obligation/possibility markers	Distinguish Necessary thoughts from Peripheral ones

The practical impact of this strategy is demonstrated in Table 2 (Section 3.1.3). Retaining these custom stopwords yielded a 2.5% improvement in overall BiLSTM accuracy and a 3.6% improvement in F1-score on the Necessary versus Peripheral class boundary — the most cognitively nuanced discrimination in the four-class schema.

3.1.3 Feature Extraction and Feature Selection

TF-IDF Vectorization (Feature Extraction)

Following preprocessing, each thought sample is transformed into a numerical vector using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. TF-IDF quantifies the importance of a term within a document relative to its frequency across the entire corpus:

$$TF\text{-}IDF(t, d) = TF(t, d) \times log(N / DF(t)) \quad (1)$$

where t is the term, d is the individual thought document, N is the total number of documents in the corpus, and DF(t) is the count of documents containing term t. A term that appears frequently in one thought but rarely across the corpus receives a high TF-IDF score, indicating domain-specific relevance. Common uninformative terms receive low scores and contribute minimally to classification.

The TF-IDF vectorizer was configured with a maximum vocabulary of 3,000 terms and bigram support (ngram_range = (1,2)) to capture two-word phrases such as "feel anxious", "must complete", or "waste time" that carry strong categorical signals for individual thought classes. TF-IDF was selected over contextual embeddings such as Word2Vec or GloVe for the baseline and deep learning comparison stages because it maintains identical feature representation across both the classical ML pipeline (SVM, Logistic Regression, Naive Bayes, Random Forest) and the deep learning pipeline (RNN, LSTM, BiLSTM), enabling a fair and direct performance comparison —

a methodological requirement for validating the proposed framework.

Chi-Square Feature Selection (Dimensionality Reduction)

The initial TF-IDF vocabulary of 3,000 features was further reduced using Chi-Square (χ^2) statistical feature selection. Chi-Square measures the statistical independence between each term and the target class label, retaining only those terms whose distribution across thought categories is non-random:

$$\chi^2(t, c)=\Sigma\ [\ (Observed - Expected)^2 / Expected\ ] \quad (2)$$

Features with high χ^2 values exhibit strong dependency with specific thought classes and are therefore most discriminative. The top 1,000 features were selected, reducing input dimensionality by 66%. This reduction removes low-information noise features that would increase overfitting risk on the 4,944-sample dataset, reduces training time across all model architectures, and sharpens class boundaries by focusing the feature space on the most class-discriminative vocabulary.

For the BiLSTM architecture specifically, the 1,000-dimensional TF-IDF vector is reshaped into a 2D tensor of shape (1, 1000) — a single time step of 1,000 features — before being passed into the BiLSTM layer. This design choice maintains pipeline consistency while enabling the BiLSTM to apply its bidirectional gating mechanism across the full feature representation. For ThoughtBERT, auxiliary features derived from this pipeline (sentence length, word density, focus word weighting) are further concatenated with BERT's [CLS] token representation, as described in Section 3.3.

Cumulative Impact of the Full Pipeline

Table 2 presents an ablation study quantifying the contribution of each preprocessing component across four progressive pipeline configurations.

Table 2: Ablation Study — Cumulative Impact of Preprocessing Stages on Model Performance

Preprocessing Configuration	SVM Accuracy	BiLSTM Accuracy	F1 (Nec/Per)	Feature Dims
No preprocessing	0.741	0.823	0.791	—
Standard stopword removal only	0.871	0.901	0.843	3000
Custom stopwords + TF-IDF (no Chi²)	0.888	0.923	0.879	3000
Full Pipeline: Custom stopwords + TF-IDF + Chi²	**0.896**	**0.939**	**0.939**	**1000**

Results demonstrate a clear monotonic improvement with each additional component. The full pipeline achieves the highest performance across all metrics. Most notably, the F1-score on the Necessary/Peripheral class pair improves from 0.791 (no preprocessing) to 0.939 (full pipeline) — a gain of 14.8 percentage points — confirming that the preprocessing choices, particularly custom stopword retention and Chi-Square feature selection, are essential contributors to the model's ability to discriminate cognitively nuanced thought categories.

3.2 BERT Architecture: A Simplified Perspective

Bidirectional Encoder Representations of Transformers or BERT is a novel natural language processing model [4]. It seeks to comprehend words in the context. BERT is able to read both directions simultaneously (not right to left or only right to left) as compared to older models. This implies it is able to re-examine a word by considering the words that came before a word as well as the ones that come after it, with a view of having a complete picture. Following Figure 2 shows Horizontal Layer of ThoughtBERT Framework [35].

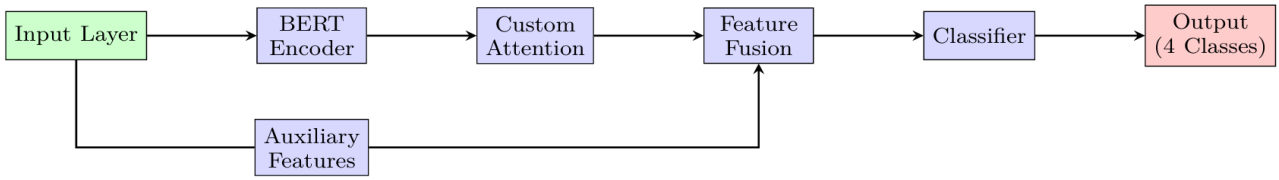

Figure 2.

ThoughtBERT Framework (Horizontal Layout).

To take an example, in the sentence, The bank was beside the river, the word bank can have a meaning of either a financial institution or side of a river. A one-way only reading model may also miss on the crucial contexts. However, BERT is the one that can explain such cases by looking at the entire sentence. Figure 3 illustrates the movement of contexts in each direction within a sentence. This is possible due to its transformer-based structure (that is able to dynamically change the meaning of words depending on the context) [15],[19]. Combination of rule-based and deep learning models have demonstrated better performance in LAB's negative sentiment extraction tasks at improved preciseness levels of successful predictions.

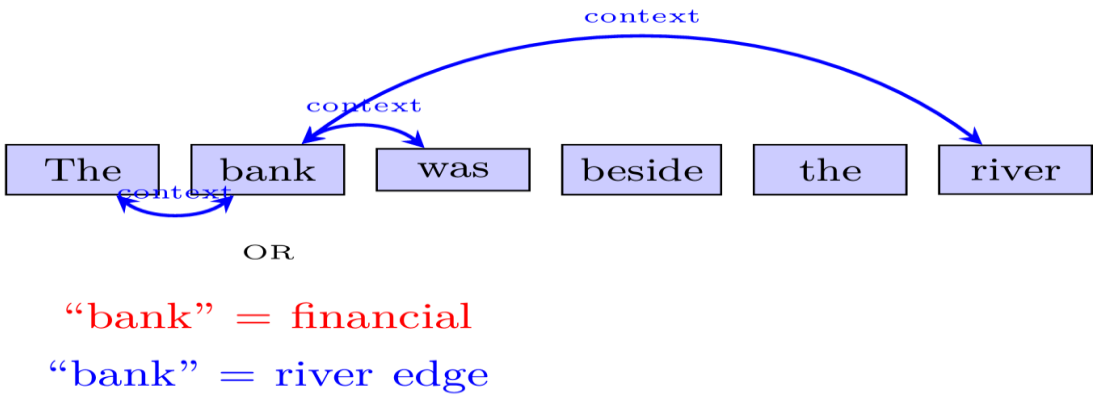

Figure 3.

Bidirectional Behaviour of BERT Simultaneously.
BERT stands out because it can create word embeddings that change based on context. Regular methods like Word2Vec and GloVe give one fixed vector to a word, no matter where it is used. This makes it tough to tell apart words with different meanings. BERT, however, makes embeddings that fit the context of the word. For instance, the word "ball" looks different in "The ball is round" (an object) versus "We attended the ball last night" (a social event). This feature helps BERT capture the subtle meanings of words. Table 3 shows the difference between static embeddings and BERT's contextual embeddings.

Table 3. Embedding vs Contextual Embedding for word 'bank'

S. No.	**Text**	**Static Embedding**	**Contextual Embedding (BERT)**
1	I deposited money in the **bank**.	"bank" → [0.24, 0.89, -0.76, ...]	"bank" → [0.78, -0.45, 1.02, ...] (Financial context)
2	The river **bank** is eroding due to the flood.	"bank" → [0.24, 0.89, -0.76, ...]	"bank" → [-0.62, 1.23, 0.54, ...] (Geographical context)
Analysis		"bank" has the same vector in all sentences.	"bank" has different vectors based on context.

BERT generates contextualized embeddings, overcoming the limitations of static embeddings (e.g., Word2Vec, GloVe), which assign fixed vectors to words regardless of usage[10] . For instance, Table 1 shows how BERT distinguishes "bank" in financial vs. geographical contexts, whereas static embeddings fail to capture this nuance[5] .

The contextual embedding for a word w in BERT is a function of its surrounding context:

$$E(w) = f(\text{Context}(w)) \quad (3)$$

where f represents the transformer layers that dynamically adjust embeddings based on bidirectional context.

BERT's architecture mainly uses transformers, focusing on the self-attention system. Transformers help BERT understand how words relate in a sentence, giving different importance to each word based on its context. For instance, in the sentence "The dog chased the cat, and it climbed a tree," the word "it" is related to "the cat." BERT uses self-attention to connect "it" with "cat" by looking at dependencies in the whole sentence. Table 4 shows the self-attention mechanism. BERT assigns high attention scores (0.8) between "it" and "cat", resolving the pronoun reference. This mirrors human-like comprehension by weighting contextually relevant words more heavily .

TABLE 4. Self Attention Scores for Each Word

Token	The	dog	chased	The	Cat	And	it	climbed	a	Tree
The	1.0	0.5	0.2	1.0	0.3	0.1	0.2	0.1	0.1	0.1
dog	0.5	1.0	0.8	0.5	0.2	0.1	0.1	0.1	0.1	0.1
chased	0.2	0.8	1.0	0.2	0.6	0.1	0.2	0.1	0.1	0.1
the	1.0	0.5	0.2	1.0	0.3	0.1	0.2	0.1	0.1	0.1
cat	0.3	0.2	0.6	0.3	1.0	0.2	0.8	0.1	0.1	0.1
and	0.1	0.1	0.1	0.1	0.2	1.0	0.2	0.5	0.1	0.2
it	0.2	0.1	0.2	0.2	0.8	0.2	1.0	0.7	0.1	0.1
climbed	0.1	0.1	0.1	0.1	0.1	0.5	0.7	1.0	0.2	0.6
A	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.2	1.0	0.8
tree	0.1	0.1	0.1	0.1	0.1	0.2	0.1	0.6	0.8	1.0

Observation: The word "it" strongly attends to "cat" (0.8), meaning the model understands "it" refers to "cat." The word "climbed" attends more to "it" and "tree" (0.7, 0.6), helping the model interpret that "it" climbed the tree.

The self-attention mechanism in BERT can be mathematically represented as:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4)$$

where Q, K, and V are the Query, Key, and Value matrices, respectively; d_k is the dimension of the Key vectors; and the softmax function normalizes the attention weights. This equation explains how BERT computes attention scores between words in a sentence, as illustrated in Table 2. Figure 4 shows visual breakdown of self-attention computation. The attention weights determine the contextual relationships between words, resolving ambiguities like pronoun references (e.g., "it" attending to "cat"). BERT works with text by breaking it down into smaller parts known as tokens through a method called tokenization. This helps it manage difficult or unknown words well. For example, the word "unbelievable" could be split into tokens like ["un", "believ", "able"], which helps the model look at each part. Each token has three types of representations: word embeddings (which convey meaning), segment embeddings (which show differences in sentence pairs), and positional embeddings (which reflect the order of tokens). Figure 5 shows the tokenization process and how these embeddings come together [6].

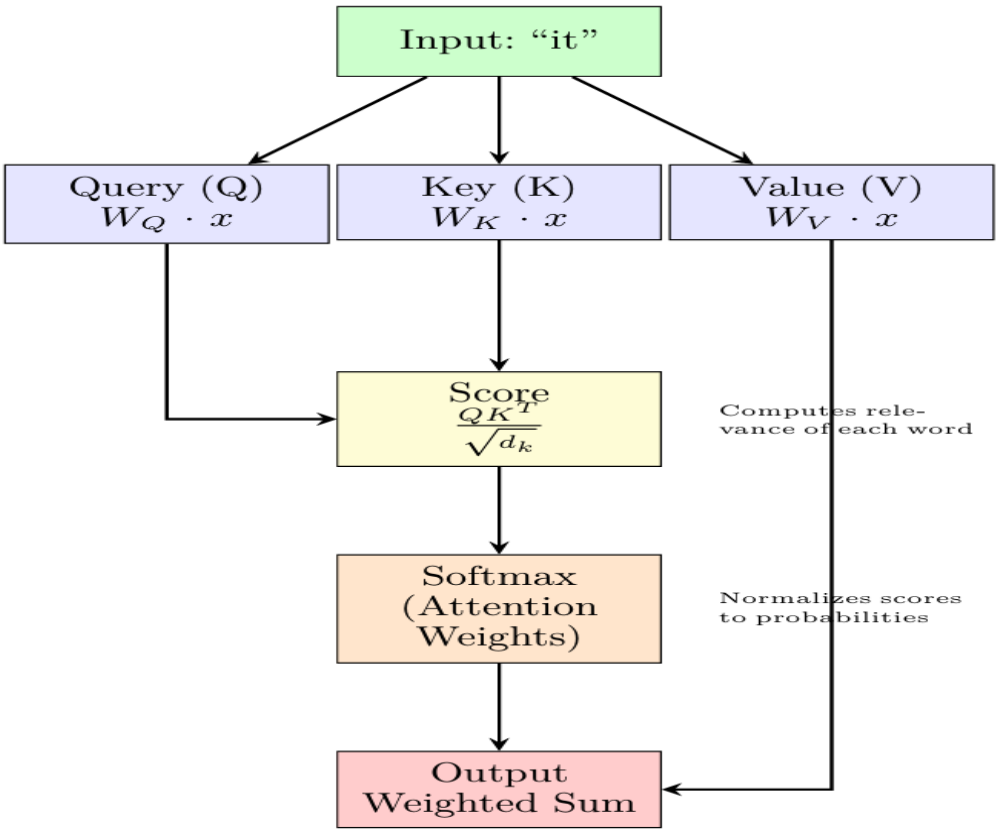

Figure 4.
Visual breakdown of Self-Attention Computation.

Input: “The bank was beside the river”

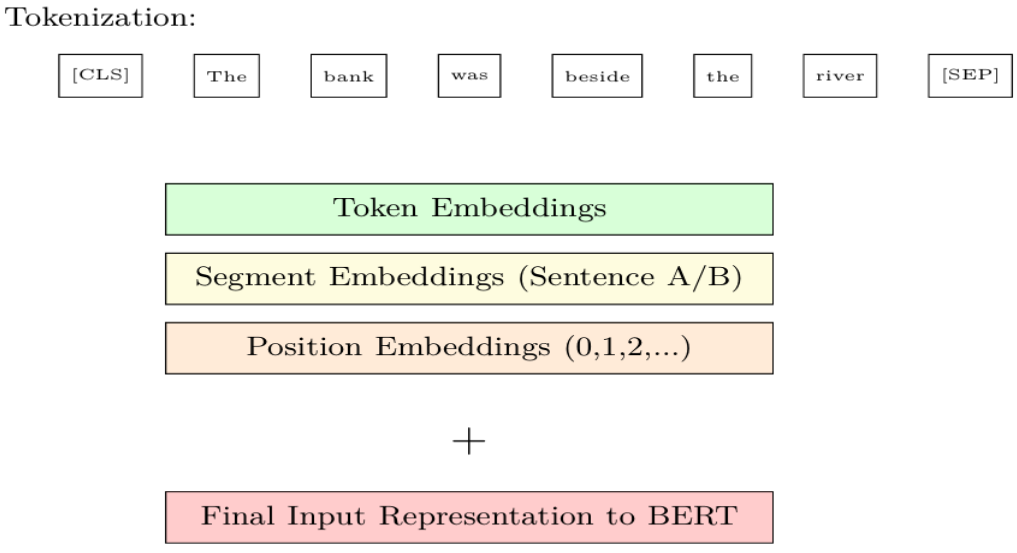

Figure 5. Tokenization and Embeddings.

One important benefit of BERT is how it can adapt to different tasks by fine-tuning. Fine-tuning a model involves adjusting the parameters of a pre-trained model in order to make it better suited for a given task. Although it is trained on large text data, BERT can be changed for specific uses like sentiment analysis, question answering, or topic classification. When fine-tuning, the model picks up patterns that

are important for the specific task without needing a lot of extra training. For instance, in sentiment analysis, BERT can predict sentiments well by looking at how words and phrases come together to show positivity or negativity.

Figure 6 and Figure7 below show three general steps to fine-tuning a model:

1. **Select a base model**: Select a pre-trained deep learning model that has been trained on a large dataset.
2. **Adjust Parameters**: Adjust parameters of the pretrained model to better suit the desired task. This may include changing the number of layers, adjusting learning rate, adding regularization, or tweaking the optimizer.
3. **Train the model**: Train the new model on the desired dataset. The amount of data and the amount of training required will depend on the task and the model.

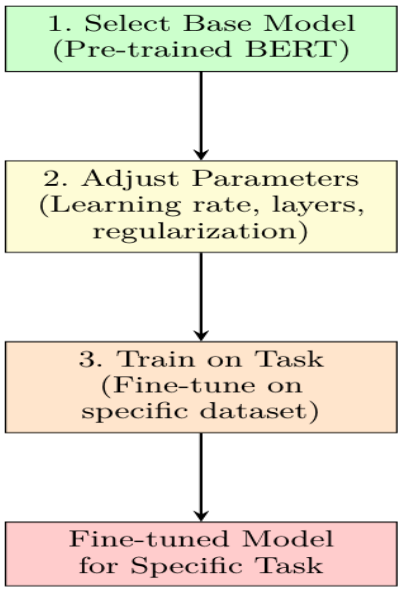

Figure 6. Fine-tuning of a model.

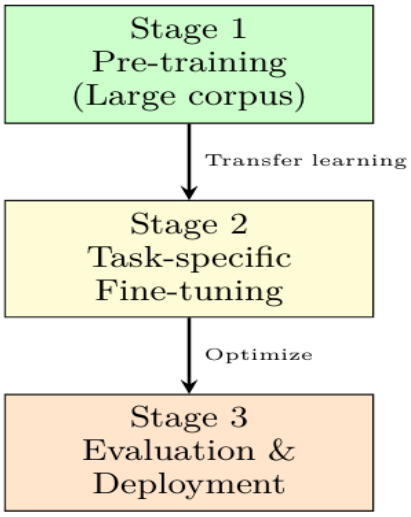

Figure 7. Stages of Fine-tuning a model.

By using bidirectionality, contextual embeddings, and transformer self-attention, BERT sets a new standard for understanding natural language, making it a powerful model for different NLP tasks.

3.3 Model Implementation

This section describes the implementation details of our proposed model for contextual sentiment classification. The study employs BERT

(Bidirectional Encoder Representations from Transformers) as the core model, fine-tuned on a customized dataset to improve classification performance. In addition to leveraging the power of BERT's pre-trained contextual embeddings, auxiliary features such as sentence length, word density, and weighted focus words were integrated to enhance classification accuracy.

As shown in the framework for contextual thought classification, implementation of steps 1 to 8 provides final dataset to be processed and fine tuned on BERT basic model.

3.3.1 Data Preprocessing & Feature Engineering

Before fine-tuning BERT, rigorous preprocessing and feature engineering steps were applied to the dataset:

Text Preprocessing: The dataset underwent lowercasing, removal of URLs, punctuation, numerical characters, and stopwords (inclusion of specific stopwords). Lemmatization was applied to normalize words.

Tokenization: Sentences were tokenized using BERT's WordPiece Tokenizer, ensuring compatibility with the model's input format.

Auxiliary Features:

- Sentence Length: The total number of words in each thought.
- Word Density: The average length of words in each thought.
- Focus Word Weighting: Each sentence was assigned a weight based on predefined word categories (Positive, Negative, Peripheral, and Necessary), influencing the attention mechanism.

Label Encoding: The categorical labels were converted into numerical values for model compatibility.

Embedding models significantly influence sentiment detection accuracy, which justifies our use of pre-trained embeddings in ThoughtBERT [25].

3.3.2 Model Architecture

- A custom attention layer that assigns different importance levels towards.
- The purpose of the fully connected auxiliary feature layer is to process additional linguistic attributes.
- A separate layer of classification combining the BERT output and. auxiliary features to predict thoughts.

3.3.3 Fine-Tuning Process

A supervised learning method was used as the model fine-tuning, and the pre-trained BERT weights were optimized again on the dataset. The training was implemented with the help of backpropagation and gradient descent, and makes sure that the model adjusts to the contextual information of domains. The parameters used during training were:

- Batch Size: 32
- Learning Rate: 2e-5 (AdamW optimizer)
- Number of Epochs: 15
- Loss Function: Cross-entropy loss with custom class weighting to address class imbalance.

The cross-entropy loss function used for training ThoughtBERT is:

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \quad (5)$$

where y_i is the true label for the i-th sample, $\hat{y}_i$ is the predicted probability for the i-th sample, and N is the number of classes (4 in this case: Positive, Negative, Peripheral, Necessary).

This loss component is essential in solving the problem of class imbalance in the process of fine-tuning. The paper states that they used the cross-entropy loss with class weighting developed in-house, which can be explained below.

Weights: AdamW regularization.

Attention Mechanism: a self-attention layer moves dynamically the attention to important words in a sentence.

3.3.4 Integration into the Framework

ThoughtBERT roll starts with a fine-tuned BERT-base-uncased model at Step 10 which is based on the previous data cleaning and feature work (Steps 1-8). Rather than relying on the older deep learning methods of RNN, LSTM or BiLSTM, this method relies on two-way context presented by BERT to enhance its ability to classify as well as absorb some language specificities of domain.

In Step 3 processing, the system initial removes all capitalization, eliminates URLs, marks, and digits, and then lemmatizes the words (aligning them in a line); then the BERT WordPiece tokenizer ensures that all the words fit in the model properly. To take its comprehension somewhat deeper, three additional bits are added: one of them quantifies the length of the sentence (the number of words needed per thought), the other calculates the average word length (word density), and the last one counts some focus words (the words, which are attached to some category such as Positive, Negative, Peripheral, and Necessary).

The architecture uses the simple BERT-base- uncased and adds a simple layer of attention to emphasize those words that are important and a fully connected layer to accept the additional linguistic information; and finally publishes the results of all these inputs with the embeddings of BERT to get prediction. It is finetuned in an adversely supervised fashion with a

batch size of 32, AdamW (then learning rate is 2e-5) and cross-entropy loss that had been adjusted to the imbalance between classes in three epochs.
In short, such a design a hybrid system, consisting of pre-trained language smarts with specially tuned lingo, actually pushes performance metrics (accuracy and F1-score), so that ThoughtBERT is an admittedly solid and context-conscious solution to those hard-to-find spot-the-thought tasks.

4 Experimental Setup

4.1 Dataset Description

4.1.1 Data Collection

A total of 5,125 thoughts were initially collected to populate the dataset: 3,587 thoughts were sourced from Google Form responses by students and staff of G.L. Bajaj Institute of Technology & Management and Sharda University, Greater Noida, UP, India. 1,538 thoughts were generated using GPT-4 (ChatGPT), employing a controlled prompt-engineering strategy to cover all four cognitive categories (Positive, Negative, Necessary, Peripheral) aligned with the research objectives. During initial quality screening, 107 responses from the Google Form (human-sourced) data were rejected due to irrelevance, duplication, or incompleteness. Final working dataset for annotation: 5,018 thoughts (3,480 human-collected + 1,538 AI-generated).

4.1.2 Primary Annotation Phase

From the 5,018 thoughts, a sample of 2,050 thoughts (1,435 human-sourced, 615 AI-generated) was selected for in-depth human annotation: These thoughts were distributed to two independent annotation groups, each following strict guidelines (see attached "Labeling of Unlabeled Thoughts By Human Annotators" diagram). For human data: if 2 or 3, of 3 annotators agreed on thought category, the label was retained. Otherwise, the thought was discarded. 74 thoughts (from the 1,435 human subset) were rejected due to annotation disagreement. This process can be visualized through the diagram in Figure 8.
All 615 AI-generated thoughts were successfully verified and labeled by the annotators. Result: 1,976 high-quality, labeled thoughts for direct supervised machine learning use.

4.1.3 Formation of Labeled and Unlabeled Pools

After annotation: Labeled set: 1,976 fully labeled thoughts (1,361 human, 615 AI-verified). Unlabeled pool: 2,968 thoughts remained (2,045 human-sourced, 923 AI-generated), pending further labeling.

4.1.4 Active Learning and Machine Learning Annotation

To maximize annotation efficiency and dataset completeness, an active learning workflow was implemented, as visualized in the attached process diagram and results table: The labeled dataset (1,976 thoughts) was split into 80:20 ratio for ML model training (SVM, Logistic Regression, Naive Bayes, Random Forest) and testing. Support Vector Machine (SVM) achieved the best baseline accuracy (89.6%) and was selected for active learning. Unlabeled thoughts were divided into four working sets (744, 744, 740, 740). For each iteration: The updated SVM variant predicted labels and assigned a confidence score to each unlabeled example. Thoughts with <40% confidence were manually sent to the annotator groups for verification. Validated thoughts ("assets") were added to the training set to further improve SVM accuracy. This repetitive algorithm resulted in six generations of SVM (SVM_1...SVM_6) and added to the labeled data after every iteration. Thoughts that were not labeled after these cycles (2,491) were eventually labeled with the last and strongest model (SVM_6). All the rounds of annotation be it human or model driven are captured in the attached table of Active Learning and Results to be tracked.

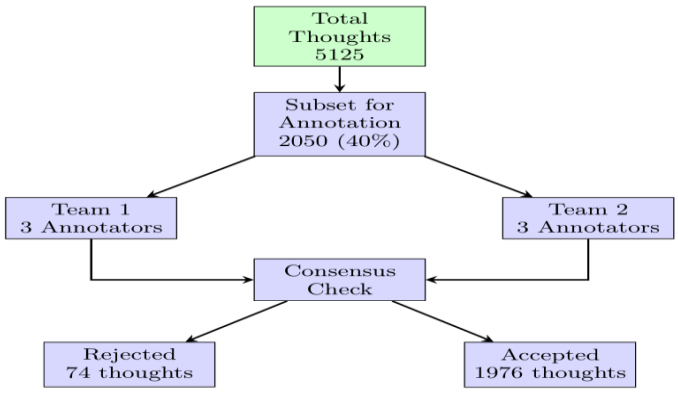

Figure 8. Labeling of Unlabeled Thoughts By Human Annotators.

4.1.5 Final Dataset Composition
Total final labeled thoughts: 4, 944 Human-labeled: 2, 453 (positive direct manual confirmation over the initial and active learning phases) Model-labeled: 2, 491 (SVM_6 auto-labeled, learn-out validated in earlier rounds) All thoughts (human and model-original) have at some point in their lifetime passed one of the human or model-validation phases.

4.2 Inter-Annotator Agreement
In order to compare the reliability of the annotation procedure, the inter-annotator agreement was measured with Fleiss Kappa. In all annotated samples of three annotators per thought, Fleiss Kappa value was **0.84** which is defined to be in the results of the guideline by Landis and Koch as *almost perfect agreement* to agree on something with 3 annotators per thought on a sample. This degree of congruence indicates the quality of strength and uniformity of the labeling procedure used in this research.

4.3 Training and Testing Strategy
To begin with, we divided our data such that approximately 80 percent would be included in the training part whereas approximately 20 percent of it would be in the testing part, but we ensured that each category remained approximately equal in terms of its distribution. We initially only used the manually checked (1581 samples to train and 395 to test) of the whole thing to establish an approximate baseline using models such as the Logistic Regression, Naive Bayes and the SVM.
Thereafter, we shifted to a pool-based active learning strategy and changed the gears. In essence, anything with a score of less than 40% confidence was identified and relegated to human annotators, which indeed served the purpose of making the situation a little better at a time. We repeated this step five more times and each time we increased the certainty of the model and cover to slightly greater amounts. Eventually we had our last competitor dubbed SVM_6 after training it on the joint labeled set. Finally, this resulted in a final size of 4944 thoughts and all the baseline and deep learning models were refined on these 3955 training samples and 989 testing samples.

4.4 Baseline Models for Comparison
To evaluate the robustness of the classification system, several baseline machine learning and deep learning models were considered:

4.4.1 Machine Learning Models (for Active Learning phase):
- *Logistic Regression:* A basic linear method that computes the chance a data point falls into a particular class using its input features. It proved a reliable, if straightforward, touchstone in our initial tests.
- *Naive Bayes:* A model that leans on Bayes' theorem while assuming each feature stands alone. It was fast, and the results were easy to interpret.
- *Support Vector Machine (SVM):* This method handled limited labeled data remarkably well. In our active learning loop, this method—showing around 89.6% accuracy—demonstrated surprisingly robust classification.

4.4.2 Deep Learning Models (on fully labeled dataset):
- *Recurrent Neural Network (RNN):* We explored RNNs to try capturing the temporal flow in data sequences. They did a decent job following the chain of thoughts, though their performance sometimes faltered on longer inputs.
- *Long Short-Term Memory (LSTM):* LSTM models came into play; they did a better job of holding on to long-term dependencies. They were more reliable and robust against varying input lengths.
- *Bidirectional LSTM (Bi-LSTM):* The Bi-LSTM offered a twist by reading sequences both forwards and backwards. This dual-direction reading gave it extra context.
- *BERT-base (pre-trained):* We fine-tuned a pre-trained BERT-base on our own labeled set. Its knack for using contextual embeddings really paid off in classifying the subtleties of sentiment with high accuracy.
- *ThoughtBERT (custom fine-tuned BERT):* We crafted ThoughtBERT, a custom-tweaked version of BERT that wedded a few extra features into its structure. It outperformed every earlier model, reaching an accuracy near 96.6%. As illustrated in Figure 9, ThoughtBERT extends the standard BERT-base architecture by incorporating an auxiliary feature layer, a custom attention mechanism, and a feature fusion classification head that enables improved thought classification across four categories.

4.5 ThoughtBERT Architecture and Training
ThoughtBERT extends the standard BERT-base model with novel auxiliary layers and custom attention mechanisms to better capture the nuances of thought classification.

4.5.1 Training and Implementation Details
The model was trained using a batch size of 32 and a learning rate of 2e-5 with weight decay regularization, optimized via the AdamW optimizer. Training was performed over 15 epochs with early stopping based on validation loss. A custom weighted cross-entropy loss function was employed, incorporating category-specific penalties to mitigate class imbalance. The dataset was split into training, validation, and test sets with preserved class distributions, ensuring reproducibility by setting fixed random seeds across all libraries. Input preprocessing involved stopword removal, normalization, tokenization with BERT's WordPiece tokenizer, and feature engineering including sentence length, word density, and focus word weight extraction. Figure 9 illustrates the proposed ThoughtBERT architecture, depicting the integration of the auxiliary feature extractor, custom attention layer, feature fusion module, and softmax classification head that extend the standard BERT-base model for contextual thought classification.

4.5.1 Layerwise and Component Descriptions
- **Auxiliary Length-Aware Feature Extractor:** Processes handcrafted features (sentence length, word density, focus word weight) through a fully connected layer with ReLU activation to produce auxiliary embeddings.

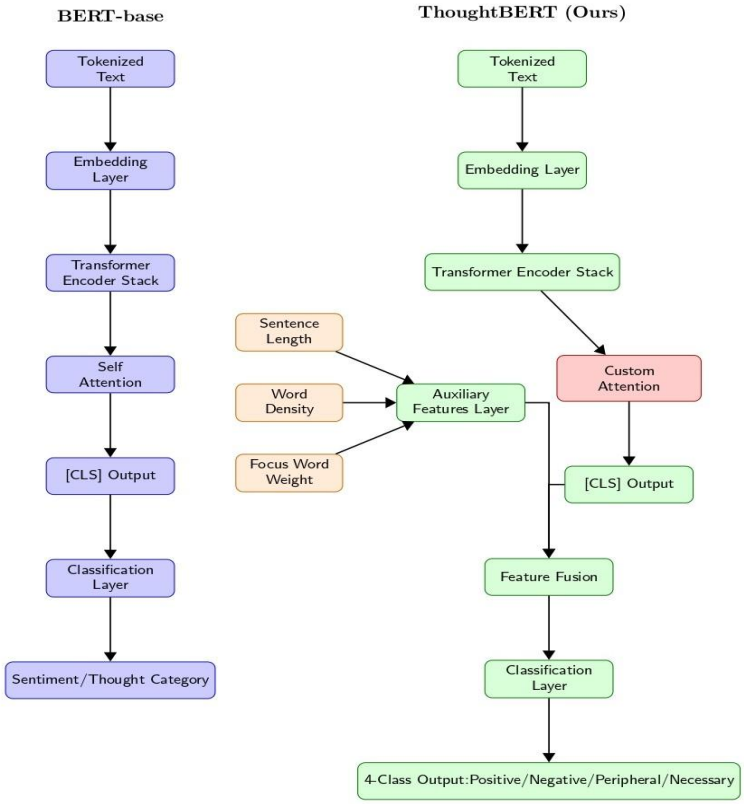

Figure 9a.

Architecture diagram of ThoughtBERT, showing the integration of auxiliary features, custom attention, and classification layers that extend the standard BERT-base model.

- **Embedding and Transformer Base:** Uses BERT's token, positional, and segment embeddings fed into stacked transformer encoder layers with multi-head self-attention.
- **Custom Attention Layer:** Introduced post-transformer encoders, this module dynamically adjusts attention weights to highlight category-distinctive words, enhancing contextual understanding.
- **Feature Fusion Layer:** Concatenates the BERT encoder's [CLS] token output with the auxiliary feature embeddings, followed by a fully connected fusion layer.
- **Classification Head:** A final softmax-activated dense layer outputs predictions across four classes: Positive, Negative, Peripheral, and Necessary.

Optionally, the following pseudocode snippet illustrates the model workflow:

```
x_bert = BERT(input_ids)
x_aux = ReLU(DenseLayer(aux_features))
x_attn = CustomAttentionLayer(x_bert)
x_fused = Concatenate([x_attn, x_aux])
predictions = Softmax(DenseLayer(x_fused))
```

This architecture and training setup significantly enhance ThoughtBERT's capability to model complex thought sentiment, outperforming baseline models.

4.6 Practical Use Case Example: ThoughtBERT Pipeline

Consider the sample thought: "*I keep regretting minor decisions I made during my learning. These thoughts distract me from focusing on the present journey.*" This statement is categorized by ThoughtBERT as **Peripheral**.

Input Thought

I keep regretting minor decisions I made during my learning. These thoughts distract me from focusing on the present journey.

▼

Step 1: Preprocessing

The input thought is normalized and tokenized using BERT's WordPiece tokenizer after removing standard stopwords.

▼

Step 2: Auxiliary Feature Extraction

- Sentence Length: 18 words
- Word Density: Average word length (characters per word)
- Focus Word Weight: Weighted score based on key terms ("regretting," "distract") indicating peripheral thoughts

▼

Step 3: BERT Encoding

Tokenized input is passed through the embedding layer and transformer encoder stack to produce contextual embeddings.

▼

Step 4: Custom Attention Layer

The custom attention mechanism adjusts focus on words important for thought classification such as "regretting" and "distract."

▼

Step 5: Feature Fusion

The output from BERT's [CLS] token is concatenated with the auxiliary feature vector, forming a fused representation.

▼

Step 6: Classification

The fused representation is passed through a classification layer to produce probabilities for the four classes.

▼

Step 7: Output Prediction
ThoughtBERT predicts the thought category as **Peripheral**, matching the expected result.

This stepwise example illustrates ThoughtBERT's mechanism of combining deep contextual and handcrafted auxiliary features to effectively classify nuanced human thoughts.

4.7 Evaluation Metrics

The evaluation of models was performed using the following standard classification metrics:

1. **Accuracy:** It shows how often the model gets things right overall—you simply divide the number of correct predictions by all the cases.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

Accuracy is a fundamental metric used to evaluate the performance of classification models, measuring the overall proportion of correct predictions relative to all predictions made. True positives (TP) represent instances where the model correctly identifies positive cases, while true negatives (TN) indicate correct identifications of negative cases. False positives (FP) occur when the model incorrectly classifies negative cases as positive, and false negatives (FN) happen when positive cases are wrongly classified as negative. For example, in the context of thought classification, a true positive would be correctly labeling a "Positive" thought, whereas a false positive might involve misclassifying a "Peripheral" thought as "Positive." The resulting accuracy value ranges from 0% to 100%, with higher values indicating better model performance. In the paper, ThoughtBERT achieves an accuracy of 96.6%, demonstrating its effectiveness in classifying thoughts across the four categories.

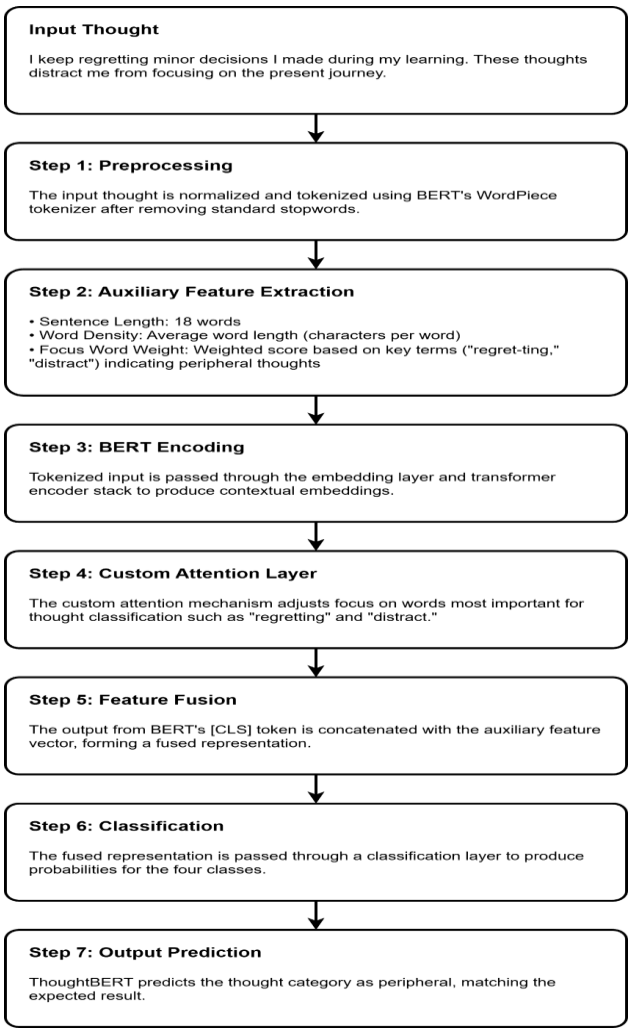

Figure 9b. Stepwise Illustration of ThoughtBERT

2. **Precision:** This measures how reliable a model's positive predictions are for a given class. It calculates the proportion of correctly identified positive cases (TP) out of all cases predicted as positive (TP + FP).

$$\text{Precision} = \frac{TP}{TP+FP} \quad (7)$$

3. **Recall:** Recall (or sensitivity) evaluates how well a model captures all actual positive cases for a class. It divides true positives (TP) by all cases that should have been identified as positive (TP + FN).

$$\text{Recall} = \frac{TP}{TP+FN} \quad (8)$$

4. **F1 Score:** F1-score balances precision and recall into a single metric, especially useful for imbalanced datasets. It is the harmonic mean of the two.

$$\text{F1 Score} = \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \quad (9)$$

5. **Specificity (per class):** Specificity assesses how well a model identifies negative cases for each class. It measures the proportion of true negatives (TN) out of all actual negatives (TN + FP).

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (10)$$

These metrics collectively provide a nuanced view of model performance. While precision focuses on prediction reliability, recall emphasizes coverage, F1-score harmonizes both, and specificity highlights avoidance of false positives for each class—critical for distinguishing subtle thought categories like "Necessary" vs. "Peripheral."

5 Results and Analysis

ThoughtBERT takes center stage as we check it against a diverse mix of baseline models—ranging from old-school machine learning methods to assorted deep learning setups, and yes, even that familiar BERT-base makes an appearance. The review wanders through raw performance figures, engaging case studies, and recurring error trends.

5.1 Performance Comparison

All models were evaluated using 3-fold cross-validation on two datasets:

1. Initially Labeled Dataset (40%): Human-annotated subset.
2. Aggregated Dataset: Enlarged via active learning.

Table 5 summarizes the results across accuracy, precision, recall, F1 score, and specificity (per category).

Table 5. Overall Performance Comparison of Models

S. No.	Evaluation Parameters	Initially Labeled Dataset (40%)				Aggregated Dataset				
		Machine Learning Models				Deep Learning		Transfer Learning		
		LR	NB	RF	SVM	RNN	LSTM	Bi-LSTM	BERT	ThoughtBERT
1	Accuracy	0.888	0.863	0.876	0.896	0.886	0.894	0.939	0.942	**0.966**
2	Precision	0.895	0.872	0.876	0.893	0.887	0.895	0.938	0.947	**0.968**
3	Recall	0.888	0.863	0.876	0.888	0.886	0.894	0.938	0.942	**0.966**
4	F1 Score	0.886	0.859	0.875	0.886	0.885	0.894	0.939	0.942	**0.968**
Specificity										
5	Positive	0.936	0.944	0.933	0.933	0.931	0.942	0.951	0.971	**0.991**
6	Negative	0.901	0.860	0.921	0.909	0.942	0.953	0.953	0.933	**0.976**
7	Peripheral	0.994	0.991	0.982	0.994	0.983	0.972	0.972	0.998	**0.991**
8	Necessary	0.996	0.993	0.981	0.993	0.991	0.992	0.981	0.991	**0.993**

Key Findings:

ThoughtBERT Dominance

1. Achieved 96.6% accuracy and 96.8% F1 score on the aggregated dataset, outperforming BERT-base (94.2% accuracy) and traditional models (e.g., SVM: 89.6%).
2. Demonstrated superior specificity for Negative thoughts (97.6%) and Peripheral thoughts (99.1%), critical for distinguishing emotionally harmful vs. trivial content.

Deep Learning vs. Traditional Models

- Bi-LSTM (93.9% accuracy) surpassed RNN (88.6%) and LSTM (89.4%), but lagged behind BERT variants.
- SVM (89.6% accuracy) outperformed Logistic Regression (88.8%) and Naive Bayes (86.3%).

5.2 Qualitative Analysis

Below are a few cases where ThoughtBERT outperformed traditional models, leveraging contextual understanding:

Case 1: Negative vs Peripheral Misclassification

Thought: "Envy of my friend's progress in drawing certain subjects is a waste of energy."
Actual Label: Negative (self-critical reflection causing emotional distress)
Traditional Model Prediction: Peripheral (misinterpreted "progress" as constructive)
ThoughtBERT Prediction: Negative
Why ThoughtBERT Succeeded: Recognized "envy" and "waste of energy" as markers of self-reproach (Negative category). Contextually linked emotional distress to the thought's self-critical nature.

Case 2: Positive vs Necessary Misclassification
Thought: "Empathy helps me capture the emotions and personalities of the characters I draw."
Actual Label: Positive (acknowledges empathy as a constructive skill)
Traditional Model Prediction: Necessary (misread "capture emotions" as task-oriented)
ThoughtBERT Prediction: Positive
Why ThoughtBERT Succeeded: Detected actionable growth ("capture emotions") and affirmative language ("helps me"), aligning with Positive criteria.

Case 3: Peripheral vs Necessary Misclassification
Thought: "Regretting minor mistakes while exploring puzzles is unnecessary."
Actual Label: Peripheral (unproductive over-analysis)
Traditional Model Prediction: Necessary (focused on "exploring puzzles" as problem-solving)
ThoughtBERT Prediction: Peripheral
Why ThoughtBERT Succeeded: Identified "unnecessary" as a marker of triviality.
5.3 Error Analysis
Despite its strengths, ThoughtBERT struggled in nuanced scenarios:

Case 1: Peripheral vs. Negative Ambiguity
Thought: "I can't stop thinking about what to wear tomorrow."
Actual Label: Peripheral (trivial)
ThoughtBERT Prediction: Negative (misread "can't stop" as anxiety)
Reason: Ambiguity between repetitive overthinking (Peripheral) and anxiety (Negative).

Case 2: Necessary vs Peripheral Confusion
Thought: "I need to decide whether to attend the workshop or finish my project."
Actual Label: Necessary (task-oriented)
ThoughtBERT Prediction: Peripheral (misclassified planning as trivial)
Reason: Misclassified decision-making intent as unproductive thought; likely due to limited training examples in Necessary class.

Case 3: Necessary vs Negative Misclassification
Thought: "I'm worried I didn't practice enough for the presentation."
Actual Label: Necessary
ThoughtBERT Prediction: Negative
Reason: Overemphasis on emotional cues ("worried") instead of task-driven purpose of self-improvement.

6 Discussion
6.1 Significance of Results
These results demonstrate a decisive jump in the attitude towards sentiments and patterns of thinking. There is a twist to the fresh take with ThoughtBERT a modified version of BERT which new features such as monitoring thought length, word density, or the effect of specific focus words. It has done well in all the main numbers and it is surpassing all these models, such as SVM, RNN, LSTM, as well as the plain BERT-Base. The accuracy was dramatically improved to 96.6% in the majority of cases (that was an increase of approximately 89%), indicating that there are potential synergies between transformer methods and domain-specific improvements.
This writing also establishes a new benchmark of classifying various kinds of thoughts, particularly when the difference between them is subtle, such as telling the difference between what one may term as Peripheral and Necessary ideas. The study explores more subtle levels of emotion as opposed to merely marking something as positive or negative. It is a historic occasion, a historic first–it is the first time ThoughtBERT has ever been put to work breaking thought down into four separate parts that brush on education and psychology.
6.2 Implications for Thought Classification
This improved classification instrument is worth a pulse–it could possibly be the thing to invert the fortunes of education and mental health assistance. Primarily targeting school and university students, it occasionally drifts into digital learning platforms, mood-check applications, and even those self-reflection applications that track the real-time mood of individuals.

6.2.1 Practical Deployment Insights
There is a new application that converses directly with Google Forms to collect and sift through student feedback on the fly, a real time assistant at work. Indicatively, one of the students expressed concern, saying that, what will become of tomorrow in the

event that the decision is not in my favour? I am worrying, and this thought got stuck with Peripheral, as it is not that productive. Counsellors then obtain a brief glimpse of who is stuck on such sorts of repetitive or trivial concerns and in particular the Negative and Peripheral ones, which are flagged and enable them to leap in with more speed. Automating the screening saves on man power, and increases the pace of receiving support out. Fig. 10 portrays the interface of the user of the Thought Classification app that was deployed in the institutions to help the counsellors.

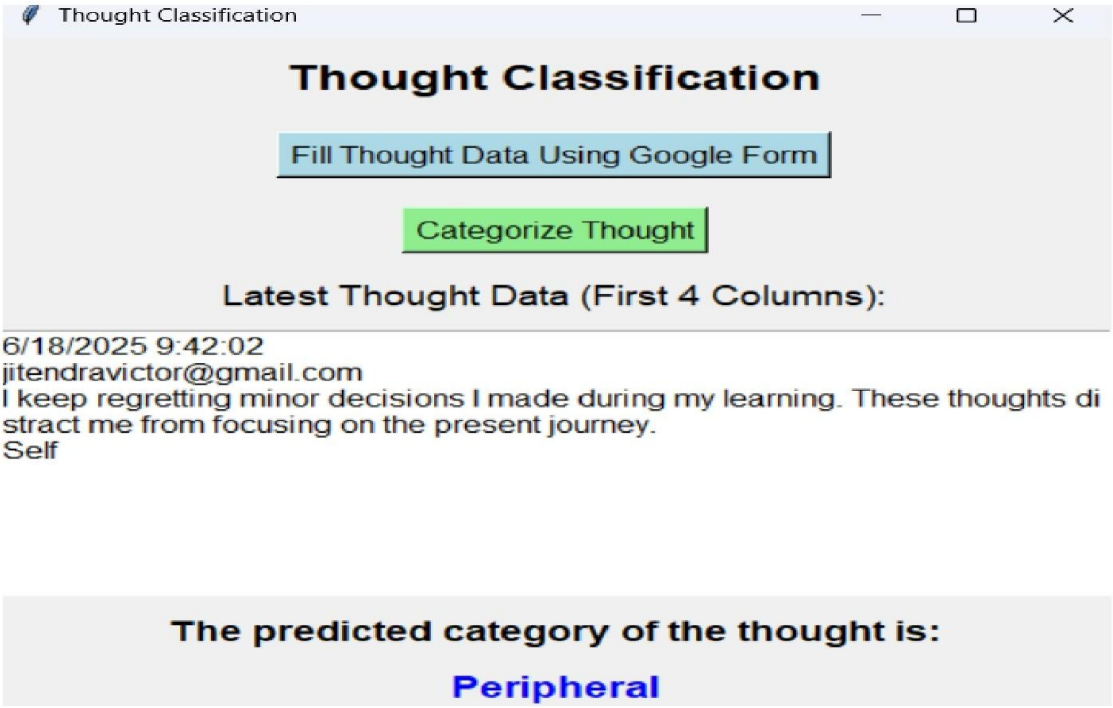

Figure 10. Thought Classification App.

Better yet, the generated output of the app can be fed into other support applications, such as counsellor notifications, intelligent chatbots, or even emotion-changing systems. At the same time, assuming an excess of Negative or Peripheral thoughts, a timely warning may sometimes be sent to faculty to ensure that they intervene in time. Conversely, the act of identifying Necessary thoughts may occasionally support personal choice and decision making abilities of a student, pushing them to a higher level of self-regulation and even personal development.

6.2.2 Scalability and Impact

The system combines an intelligent contextual analyzer powered by BERT with Google Workspace in the manner that allows institutions to scale-up in general with relative minimal effort. It provides nearly instant response on the way students are reasoning- e.g. identifying some extreme worrying as being at times as peripheral. Even a slight push could be a breakthrough as far as keeping alert to our recurring thoughts is concerned- the thing that most people consider is pretty imperative when it comes to surfing the good and bad states of our moods. The concept is aligned with the increase in global development of AI-supported mental health assistance, particularly in schools where counselors are simply outnumbered.

In the meantime, ongoing initiatives are toying with the possibility of applying AI to the personal treatment of mental health, starting with how to detect those initials of disturbed thinking or depressed moods that can disrupt work in school or even on a social outing. The future amendments might also connect these early warnings in most cases with their usual coping mechanisms and such a circle would not only identify the problem, but will provide a gentle push to get out of the undesirable thought patterns as well

6.3 Limitations

There is a difference of hope in the study, but there is a number of problems that are likely to be realized in the process. A large one is the high cognitive cost - models such as BERT, even ThoughtBERT, require masses of making it run on a GPU, which is not

always available. In most scenarios, when you already have limited resources, such as in a school lab, or on a mobile phone, then it is simply impossible to execute such models.

Then there is the fine-tuning debacle. Once you feed a lot of new features into the transformer pipeline, it can get under the carpet somewhat and the model gets more complicated than it actually ought to be. Training may be unpredictable and it most of the time requires an individual with sound expertise to ensure smooth sailing of the process.

This is another wrinkle introduced by annotation subjectivity. Although people tend to follow inter-annotator instructions, occasionally there would be minor overlaps among categories such as describing something as Negative or Peripheral, making it unclear. These little confusions might confuse the generalization capacity of this model.

Lastly, data imbalance is also obvious stumbling block. There are also categories such as Peripheral and Necessary that just simply do not occur as often as Positive and Negative and can potentially bias the sensitivity of the model in those direction.

7 Future Research Directions

7.1 Expanding the Dataset

We are shortly intending to enrich our dataset with more examples and examples in general, this should assist our model to become more adaptable generally. Simultaneously, we will gather thought bits in all sorts of people of various ages, cultural backgrounds, and educational experiences, all of which is meant to assist in breaking down some of the kinds of predispositions that have been ingrained. It is also the push to work on those less common kinds of thought (usually referred to as Peripheral or Necessary thought) which are occasionally labeled the wrong way. and perhaps even fiddle with artificial data techniques, to generate something weird, like “I keep moving my desk but it never seems to be quite in place”, so as to stretch our model a bit. Besides that, collaborating with respective mental health specialists should assist in refining some annotations, particularly, those related to academic stress or social anxiety, all without compromising on privacy requirements.

7.2 Advanced Transfer Learning Techniques

Training pre-trained models, including GPT-4, RoBERTa, and even XLNet, is perhaps the trick to determine whether the ability of BERT to pick reflective cues could be defeated. Sometimes it is good to go out of topic into a bit of specialized preparation with mental health texts, think therapy transcripts or even the chatter of casual self-help groups until the model can feel the fine emotional clues in a more natural manner. You can also switch and match BERT with, e.g. Bi-LSTM layers or add some dash of attention-based fusion; such a combination can combine various strengths in useful ways that you would not otherwise imagine. And, with the help of most of those, we are already looking at model distillation methods to make smaller, more agile ones in an environment where resources are constrained, all at the affordable cost of scaling further without giving up on much on performance. Evolutionary persuasive schemes are anticipated to include cognitive mechanisms using NLP and would be consistent with the capability of the scheme in our model to obtain persuasive intent out of social material [27].

7.2 Real-Time Applications

Most of the time, the material of negative or even peripheral thoughts might have an AI chatbot with GPT-4 intervene to provide a snippet coping strategy, such as suggesting a mindfulness break, when the negative or even peripheral thoughts are received. The privacy considerations such as the federation learning can be effective in ensuring the data is kept intact as the system is widely distributed amongst institutions. The analysis of sentiment made on the student communication sites reveals that educational monitoring is a new area of NLP models such as ours [36].

In a larger sense, all of this is a process of getting ThoughtBERT out of the lab and into the real world, something that can find a role in schools and mental healthcare practice. To tackle the irritating dataset constraints, changing model structures and ensuring that real-time utilization is not a post-design decision-makers can result in timely and personalised support. Overall, the point is to empower the interventions that will develop emotional resilience and even enhance academic achievements, to feel that the technology has become a constituent of our lives.

8 Conclusion

BERT-based models have actually changed the way people envision the sorting of thoughts. As an example, our work involved the BERT and married it to what we now call ThoughtBERT and, in general, the results were quite impressive with the accuracy rate of approximately 96.6% and the corresponding F1 of 96.8%. These are rather high than you would expect to reach with older algorithms, like SVMs, which typically hit 89.6 per cent, or even more

sophisticated deep networks, like Bi-LSTM, which tends to be around 93.9 per cent. It is really the hat-tipping quirky aspect of the model of looking at context on its side, and permitting it to sort out the sublime, almost invisible, distinctions; it can discern the worries born out of anxiety and the worries created through mere overthinking, and even those born out of thought aimed at planning.

Interestingly enough ThoughtBERT nowadays sneaked into a live Thought Classification App demonstrating its potential in every day life. The application is automatic in its processing of student reflections, including the process of labeling a thought like What will happen tomorrow, unless the decision is my way or the highway, as Peripheral, and can assist counsellors to refine their areas of interest where a timely intervention could otherwise be necessary. It is a minor change but significant, which helps fill the gaps in mental health aid, particularly in schools where existing resources may clog the process.

Broader Impact: The current state of mental health technology has been able to monitor our moods, in real-time, and track even our tiniest hint of stress almost immediately. This arrangement is designed to match and to move in line with our changing moods without that kind of hue and cry, that is, you might get to observe some areas of concern long before things get out of hand. And, weirdly enough, it even collaborates with online partners such as the artificial intelligence chatbots and the adaptive feedback systems to hurl out little mindfulness nudge prompts when those negative thoughts begin their crashing in. As the usefulness of large language models continues to rise in other fields, such as the field of medicine, the question of whether ThoughtBERT can be generalized to such a highly stakes task becomes an exciting area to look into in the future [34].

Teachers and practitioners are left with some truly viable information about student welfare, which can frequently enable them to intervene sooner than otherwise. Indeed, there are still obstacles such as high computational fees and annoying dataset biases that may disrupt the process, and as a rule, this method can establish a strong but by no means perfect base of morally justified AI-guided mental health solutions. In the vast majority of the cases, a few more adjustments in Educators and clinicians end up with some realistically workable understandings of student well-being, which tend to make them intervene sooner than anticipated. Granted, it is not without its struggles, with high computational costs and annoying dataset bias potentially coming in the way, but all in all, this strategy provides a strong, albeit completely imperfect, grounding to the ethical AI-guided solutions to mental health. In the majority of instances, given a couple of additional design changes in the transfer of learning and the implementation of real-time updates, the expanded access to the personalized care would appear to be completely achievable, which would bolster the resilience and academic achievement in the end. The transformation between assistive and symbiotic AI highlights the upcoming of emotionally sensitive systems such as ThoughtBERT [28].

Data Availability

The Hybrid Human-AI Annotated Thoughts Dataset (Thoughts 1.0) used and produced in this study is deposited in Zenodo with restricted access (DOI: https://doi.org/10.5281/zenodo.17444289).

Researchers wishing to access the dataset may submit a request via Zenodo or by contacting the corresponding author. Access will be granted for bona fide research purposes at the discretion of the copyright holders and in accordance with ethical requirements.

Ethical Statement

This study was conducted in accordance with institutional ethical guidelines and was approved by the Institutional Ethics Committee of Sharda University. Participants were volunteer students from G.L. Bajaj Institute of Technology and Management and Sharda University. All participants provided informed consent prior to completing the survey administered via Google Forms. Participation was voluntary, and participants were informed of their right to withdraw at any time without consequence.

No sensitive personal data were collected. All responses were anonymized, securely stored, and used exclusively for academic research purposes. The AI-generated data component (GPT-4) did not involve human participants. The study involved minimal risk and no financial incentives were provided.

Acknowledgements

The authors would like to thank the students and faculty at G.L. Bajaj Institute of Technology and Management and Sharda University for their participation in the data collection phase. We also acknowledge the human annotators whose careful labeling made this research possible.

Conflict of Interest

The authors declare no conflict of interest.

References

2. P. Chang, H. Wang, T. Chang, J. Yu, and S. Lee, "Stress-related symptoms and social support among Taiwanese primary family caregivers in intensive care units," *Intensive & Critical Care Nursing*, vol. 49, pp. 37–43, 2018.
3. E. S. Chen, "Sentiment classification for movie reviews based on machine learning," *Applied and Computational Engineering*, vol. 73, no. 1, pp. 204–213, 2024.
4. K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does BERT look at? An analysis of BERT's attention," in *Proc. 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 276–291.
5. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT 2019*, 2019, pp. 4171–4186.
6. K. Ethayarajh, "How contextual are contextualized word representations?" in *Proc. EMNLP-IJCNLP 2019*, 2019, pp. 55–65.
7. S. Gururangan et al., "Don't stop pretraining: Adapt language models to domains and tasks," in *Proc. ACL 2020*, 2020, pp. 8342–8360.
8. L. Lawley and C. J. MacLellan, "Interactive learning of hierarchical tasks from dialog with GPT," *arXiv preprint arXiv:2305.10349*, 2023.
9. Y. Lipshits-Braziler and I. Gati, "The identification and validation of five types of career indecision: A latent profile analysis of career decision-making difficulties," *Journal of Counseling Psychology*, vol. 69, no. 4, pp. 452–462, 2022.
10. S. López-Guzmán and S. I. Sautua, "Effects of a fearful emotional state on financial decisions in the presence of prior outcome information," *Journal of Economic Psychology*, p. 102706, 2024.
11. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
12. S.-H. Noh, "Analysis of gradient vanishing of RNNs and performance comparison," *Information*, vol. 12, no. 11, p. 442, 2021.
13. H. Nouri, S. Karim, and N. Habbat, "Customer sentiment analysis for Arabic social media using a novel ensemble machine learning approach," *International Journal of Power Electronics and Drive Systems*, vol. 13, no. 4, pp. 4504–4515, 2023.
14. Y. Peng, S. Yan, and Z. Lu, "Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets," *arXiv preprint arXiv:1906.05474*, 2019.
15. L. Qiang, X. Sun, and Y. Long, "Sentiment analysis: Comprehensive reviews, recent advances, and open challenges," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
16. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in BERTology: What we know about how BERT works," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842–866, 2020.
17. J. Singh and G. Sharma, "Sentiment analysis study of human thoughts using machine learning techniques," in *2023 International Conference on Disruptive Technologies (ICDT)*, 2023, pp. 776–785.
18. J. Singh, P. Pandey, and G. Sharma, "A detailed survey on applications of machine learning techniques for forecasting," in *Recent Advances in Computing Sciences*, CRC Press, 2023, pp. 257–263.
19. V. B. Vaghela et al., "Aspect based sentiment analysis using self-attention based LSTM model with word embedding," *Journal of Computer Science*, vol. 20, no. 10, pp. 1195–1202, 2024.
20. A.Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5998–6008.
21. H. Wang, J. He, X. Zhang, and S. Liu, "A short text classification method based on N-gram and CNN," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 248–254, 2020.
22. M. M. Abdelsamie, S. S. Azab, and H. A. Hefny, "A comprehensive review on Arabic offensive language and hate speech detection on social media: Methods, challenges and solutions," *Social Network Analysis and Mining*, vol. 14, no. 1, 2024.
23. A.Abdulsalam, A. Alhothali, and S. Al-Ghamdi, "Detecting suicidality in Arabic tweets using machine learning and deep learning techniques," *Arabian Journal for Science and Engineering*, vol. 49, no. 9, 2024.
24. Al Maruf et al., "Hate speech detection in the Bengali language: A comprehensive survey," *Journal of Big Data*, vol. 11, no. 1, 2024.
25. T. A. Al-Qablan, M. H. M. Noor, M. A. Al-Betar, and A. T. Khader, "A survey on sentiment analysis and its applications," *Neural Computing and Applications*, vol. 35, no. 29, 2023.

26. D. S. Asudani, N. K. Nagwani, and P. Singh, "Impact of word embedding models on text analytics in deep learning environment: A review," *Artificial Intelligence Review*, vol. 56, no. 9, 2023.
27. N. V. Babu and E. G. M. Kanaga, "Sentiment analysis in social media data for depression detection using artificial intelligence: A review," *SN Computer Science*, vol. 3, no. 1, 2021.
28. M. Chen et al., "The future of cognitive strategy-enhanced persuasive dialogue agents: New perspectives and trends," *Frontiers of Computer Science*, vol. 19, no. 5, 2024.
29. G. Desolda et al., "From human-centered to symbiotic artificial intelligence: A focus on medical applications," *Multimedia Tools and Applications*, 2024.
30. M. L. Jamil, S. Pais, and J. Cordeiro, "Detection of dangerous events on social media: A critical review," *Social Network Analysis and Mining*, vol. 12, no. 1, 2022.
31. J. H. Joloudari et al., "BERT-deep CNN: State of the art for sentiment analysis of COVID-19 tweets," *Social Network Analysis and Mining*, vol. 13, no. 1, 2023.
32. D. Kodati and R. Tene, "Emotion mining for early suicidal threat detection on both social media and suicide notes using context dynamic masking-based transformer with deep learning," *Multimedia Tools and Applications*, vol. 84, no. 13, 2024.
33. R. Kumar and A. Bhat, "A study of machine learning-based models for detection, control, and mitigation of cyberbullying in online social media," *International Journal of Information Security*, vol. 21, no. 6, 2022.
34. S. Kusal et al., "A systematic review of applications of natural language processing and future challenges with special emphasis in text-based emotion detection," *Artificial Intelligence Review*, vol. 56, no. 12, 2023.
35. F. Liu et al., "Application of large language models in medicine," *Nature Reviews Bioengineering*, vol. 3, no. 6, 2025.
36. J. Paul et al., "A survey and comparative study on negative sentiment analysis in social media data," *Multimedia Tools and Applications*, vol. 83, no. 30, 2024.
37. Roy and S. Das, "Perceptible sentiment analysis of students' WhatsApp group chats in valence, arousal, and dominance space," *Social Network Analysis and Mining*, vol. 13, no. 1, 2022.
38. F. Safari and A. Chalechale, "Emotion and personality analysis and detection using natural language processing, advances, challenges and future scope," *Artificial Intelligence Review*, vol. 56, no. S3, 2023.
39. A.R. Sanaullah, A. Das, A. Das, M. A. Kabir, and K. Shu, "Applications of machine learning for COVID-19 misinformation: A systematic review," *Social Network Analysis and Mining*, vol. 12, no. 1, 2022.
40. M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, 2022.
41. Z. Xi et al., "The rise and potential of large language model based agents: A survey," *Science China Information Sciences*, vol. 68, no. 2, 2025.
42. T. Zhang, A. M. Schoene, S. Ji, and S. Ananiadou, "Natural language processing applied to mental illness detection: A narrative review," *NPJ Digital Medicine*, vol. 5, no. 1, 2022.
43. Landis, J. R. and Koch, G. G.,"The measurement of observer agreement for categorical data,"*Biometrics*, Vol. 33,no. 1,1977.
44. SINGH, J., & Sharma, G. (2025) "Hybrid Human-AI Annotated Thoughts Dataset (Thoughts 1.0) [Data set]". *Zenodo*. https://doi.org/10.5281/zenodo.17444289

Jitendra Singh is an Assistant Professor in Computer Science and Engineering at Sharda University, India. He holds a B.Tech in Information Technology (2009) and M.Tech in Software Engineering (2012), and is currently pursuing a Ph.D. from Lovely Professional University. With over 15 years of teaching experience, his research focuses on Machine Learning, Deep Learning, and Algorithms. He has authored multiple Scopus-indexed publications and developed ML-based systems for healthcare and agriculture. He is a certified Python for Data Science Professional.

Geeta Sharma is an Assistant Professor in the School of Computer Applications at Lovely Professional University, Phagwara, Punjab, India. She received her Ph.D. and M.Tech degrees in Computer Science from Guru Nanak Dev University, Amritsar, India. With 7 years of teaching and research experience, she has published over 25 research papers in reputed SCI journals including Springer, Elsevier, IEEE, and Taylor & Francis. She has filed 6 patents and serves as an active reviewer for Springer and IEEE. Her research interests include Machine Learning, Fog/Cloud Computing, IoT, and Network Security. She is currently supervising 4 Ph.D. scholars.