Word Embedding with BERT

Chien-An Chiu [1]

[1]Department of Electrophysics, Department of Electrical Engineering, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan

March 19, 2025

Abstract

This challenge explores the mechanism of attention in BERT and its impact on sentiment classification using the IMDB dataset. We investigate attention scores, fine-tuning behavior under partial and contrastive data, the structure of contextual embeddings, and how they can be clustered to reflect sentiment categories. Embeddings are further analyzed using dimensionality reduction techniques such as PCA and t-SNE.

1 Introduction

Transformer-based models like BERT rely on the attention mechanism to understand contextual relationships within text. In this challenge, we explore how BERT performs sentiment classification and how the learned word embeddings reflect semantic structures.

1.1 Multi-Head Attention Architecture

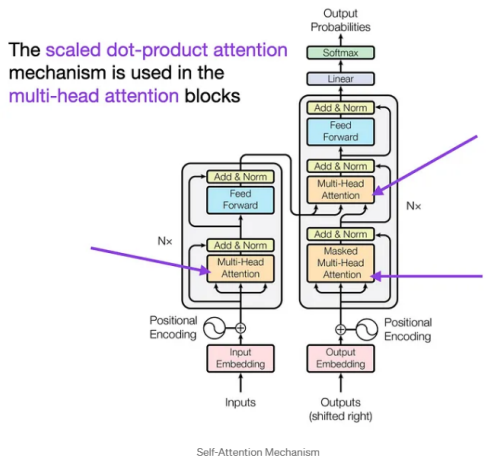

Figure 1: Multi-Head Attention in Transformer Architecture [1]

Figure 1 illustrates the attention mechanism within the Transformer architecture, specifically in BERT. Each input token is first projected into three separate matrices: the Query Q, Key K, and Value V.

In each multi-head attention layer:

- Attention scores are computed via scaled dot-product attention.
- Multiple heads (typically 12 or more in BERT) run attention in parallel.
- Their outputs are concatenated and linearly projected.

This design allows different heads to specialize in capturing different types of linguistic dependencies such as syntactic structures (e.g., subjectverb) and semantic patterns (e.g., adjectivenoun pairing). Each head independently computes:

$$\text{Attention}(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i$$

The final output is:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) W^O$$

This module is essential to BERT's ability to model contextual meaning in both directions simultaneously, as all positions can attend to one another within the same sequence.

2 BERT Architecture and Pretraining

BERT (Bidirectional Encoder Representations from Transformers) is composed entirely

of Transformer **encoder blocks** there is no decoder in its architecture. In the base version (BERT-base), the model consists of:

- 12 encoder layers
- 12 attention heads per layer
- Hidden size of 768
- Total of 110 million parameters

Each encoder layer contains two main subcomponents:

1. **Multi-Head Self-Attention Layer:** Enables each token to attend to all other tokens in the sequence in parallel. Multiple attention heads allow the model to capture different linguistic dependencies simultaneously.
2. **Position-wise Feed-Forward Network:** Applies two linear transformations with a non-linearity (typically GELU) in between, independently to each token position.

Both subcomponents are wrapped with **residual connections** followed by **layer normalization**:

$$\text{LayerNorm}(x + \text{Sublayer}(x))$$

This stabilizes training and helps gradients flow across deep layers.

BERTs input representation is formed by summing:

- **Token Embeddings:** WordPiece tokenized input.
- **Segment Embeddings:** Indicates sentence A or B (used in NSP).
- **Positional Embeddings:** Since the Transformer has no recurrence, position is encoded explicitly.

2.1 Pretraining Objectives

BERT is pretrained on unlabeled text from Wikipedia and BookCorpus using two unsupervised objectives:

1. Masked Language Modeling (MLM) Randomly masks 15% of input tokens and trains the model to predict the masked words. Unlike standard left-to-right models, this enables BERT to use bidirectional context:

$$\text{Prediction: } p(w_i \mid w_{<i}, w_{>i})$$

2. Next Sentence Prediction (NSP) Given two segments A and B, the model predicts whether B is the actual next sentence that follows A in the original document. This trains the model to understand inter-sentence coherence, which is critical for tasks like QA and NLI.

2.2 Token Output

The final output of BERT is a contextualized vector for each token. The special `[CLS]` token's final hidden state is used as the aggregate sequence representation and typically passed to a classifier head for downstream tasks.

3 Fine-Tuning on IMDB Dataset

We fine-tuned BERT using a combination of a small custom dataset and a subset of the IMDB movie review dataset. Instead of training on the entire corpus, we selected 2500 positive and 2500 negative samples, and combined them with 8 handcrafted samples (4 positive, 4 negative) containing sarcasm and contrastive patterns.

```
# Load IMDb
imdb = load_dataset("imdb")
df_train_full = pd.DataFrame(imdb["train"])

# --- ✅ Select 2500 positive + 2500 negative samples
df_pos = df_train_full[df_train_full["label"] == 1].sample(2500, random_state=42)
df_neg = df_train_full[df_train_full["label"] == 0].sample(2500, random_state=42)
df_balanced = pd.concat([df_pos, df_neg], ignore_index=True)

# Add 'sentiment' column for consistency
df_balanced["sentiment"] = df_balanced["label"]

# Combine with custom data
df_total = pd.concat([df_custom, df_balanced], ignore_index=True)

# Split into train and validation
train, val = train_test_split(df_total, test_size=0.2, stratify=df_total["sentiment"], random_state=42)
```

Figure 2: Partial dataset selection and preprocessing for fine-tuning

3.1 Effect of Contrastive Data on Sarcasm Understanding

We evaluated the model's ability to handle sarcasm by comparing training with and without contrastive data. When sarcastic or ambiguous examples were excluded, the model exhibited poor generalization on these cases.

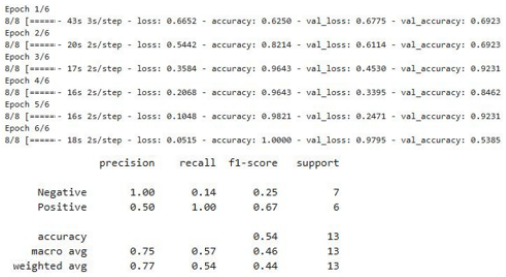

Figure 3: Low validation performance without contrastive examples (e.g., sarcasm)

The F1-score for the negative class dropped significantly despite high precision. The model overfitted on literal sentiment but failed to recognize cases such as: *"Blew my mind how bad it was. What a waste."*

3.2 Improved Performance with Full Dataset

When we included contrastive examples in training, performance across both positive and negative classes became more balanced and robust. As shown in the figure below, both accuracy and F1-scores improved across all metrics:

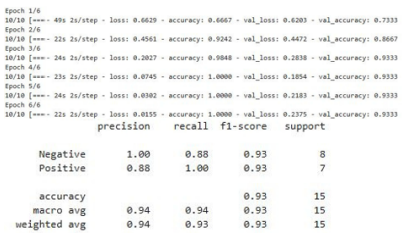

Figure 4: Improved validation performance after including contrastive (sarcastic) samples

This result suggests that BERT benefits significantly from seeing linguistically challenging data such as sarcasm. Even partial fine-tuning of only the classification head was sufficient to learn subtle sentiment features, demonstrating BERT's strong generalization capacity when guided by appropriate examples.

4 Embedding Analysis and Visualization

In BERT, each input sentence is transformed into a sequence of contextualized token embeddings through multiple self-attention and feed-forward layers. For sentence-level tasks, we extract the embedding corresponding to the special `[CLS]` token from the final layer, which aggregates semantic information from the entire sentence.

4.1 Dimensionality Reduction using PCA and t-SNE

Since each embedding vector lies in a high-dimensional space ($\mathbb{R}^{768}$), we employ dimensionality reduction techniques to visualize the geometric structure of the learned representation:

- **Principal Component Analysis (PCA)**: A linear projection technique that maximizes variance along principal axes, revealing global structure.
- **t-distributed Stochastic Neighbor Embedding (t-SNE)**: A nonlinear method that preserves local neighborhood structure, suitable for clustering analysis.

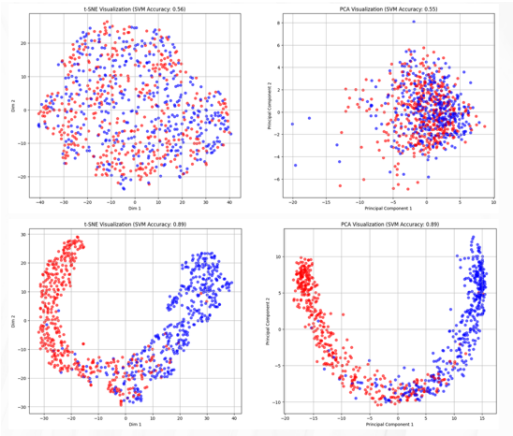

Figure 5: PCA and t-SNE visualizations of sentence embeddings before (top) and after (bottom) fine-tuning

As shown in Figure 4, embeddings before fine-tuning exhibit overlapping clusters with poor separation between sentiment labels. After fine-tuning, the red (positive) and blue (negative) embeddings are mapped to more distinct regions in the feature space, suggesting that BERT has learned to encode sentiment polarity into geometric structure.

4.2 Visualizing Extreme Sentiments

To better understand the output behavior, we also examined the most confidently classified examples. After applying softmax on prediction logits, we collected the top 5 most positive and most negative reviews based on prediction confidence:

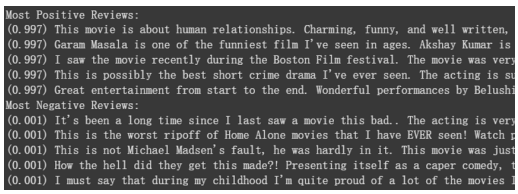

Figure 6: Examples of reviews with maximum and minimum sentiment scores

Positive samples typically contain emotionally rich, descriptive language (e.g., Charming, funny, and well written), whereas negative samples are more blunt and emphatic (e.g., This is the worst ripoff I have EVER seen!). These outputs indicate that BERT not only learns general sentiment direction, but also captures strength and extremeness in tone.

—

The combination of embedding visualization and extreme case inspection confirms that the fine-tuned model organizes sentence semantics meaningfully and effectively within the embedding space.

5 Clustering Embeddings

To analyze the semantic structure of BERT sentence embeddings, we apply K-Means clustering to the [CLS] token embeddings after fine-tuning. The goal is to determine whether sentence embeddings naturally organize according to sentiment.

5.1 Clustering and Visualization

We applied K-Means with $k = 2$ and visualized the clusters using both t-SNE and PCA. As shown in Figure 7 and Figure 8, the embeddings are projected into two distinct regions.

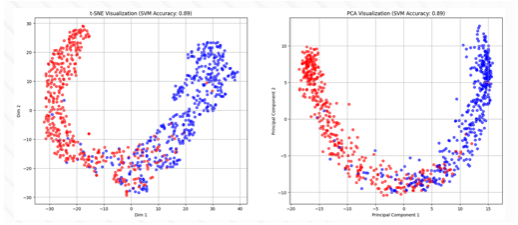

Figure 7: t-SNE and PCA of sentence embeddings colored by ground truth sentiment

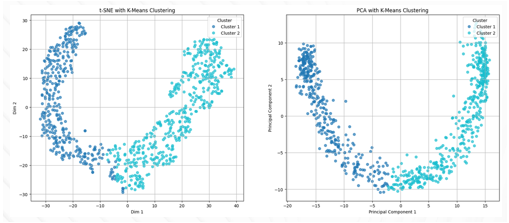

Figure 8: t-SNE and PCA of sentence embeddings colored by K-Means clustering results

We observe that the embeddings separate clearly into two distinct regions corresponding to positive and negative sentiment, even when clustering is done without label supervision.

5.2 Cluster Evaluation with Confusion Matrix

To evaluate the correspondence between the discovered clusters and the true sentiment labels, we compute a confusion matrix, shown in Figure 9.

Figure 9: Confusion matrix comparing clusters with true sentiment labels

Although not perfect, the alignment demonstrates BERTs capability to encode sentiment information geometrically in embedding space.

5.3 Cluster Interpretation through Samples and Keywords

To interpret the content of each cluster, we examine representative reviews (Figure 10) and the most frequent keywords (Figure 11).

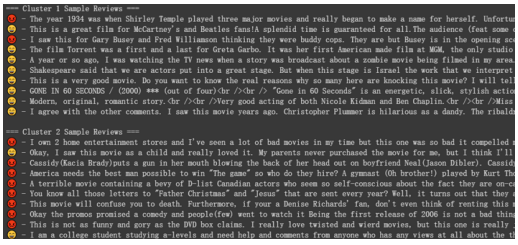

Figure 10: Sample reviews from Cluster 1 and Cluster 2

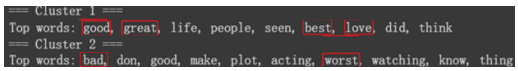

Figure 11: Top keywords extracted from each cluster

Cluster 1 is dominated by strongly positive words such as *good, great, love*, while Cluster 2 contains typical negative sentiment indicators such as *bad, worst, boring*. These results confirm that BERTs embeddings reflect sentiment polarity in a spatially meaningful way, and that unsupervised clustering is able to recover it to a large extent.

6 Results and Evaluation

Our fine-tuned BERT model demonstrates strong performance in sentiment classification, with balanced precision, recall, and F1-scores across both classes. Incorporating even a small number of contrastive examplesespecially sarcastic or ambiguous onessignificantly improved generalization to challenging cases.

Without such data, models struggled with sentiment reversals, but their inclusion boosted validation accuracy and robustness. Dimensionality reduction techniques (PCA and t-SNE) revealed clear separation between sentiment classes, further confirmed by over 89% accuracy using an SVM classifier in 2D space.

K-Means clustering on [CLS] embeddings also produced distinct sentiment groups, with high alignment to true labels and polarized keywords (e.g., *love* vs. *boring*), highlighting BERTs ability to encode sentiment geometrically and semantically.

7 Conclusion

This challenge demonstrates BERTs strong ability to capture both local and global semantic relationships through its attention-based architecture. Even with limited data, fine-tuning on contrastive and challenging examples enables robust performance, particularly in handling sentiment and sarcasm.

Visualizations using PCA, t-SNE, and K-Means clustering show that BERTs embeddings naturally separate sentiment classes and encode interpretable structures. In summary, fine-tuned BERT models not only excel at sentiment classification but also reveal meaningful internal representations that support deeper analysis and interpretability.

References

[1] R. Shaikh, "A comprehensive guide to understanding BERT: from beginners to advanced," Medium, Apr. 2023. [Online]. Available: https://medium.com/@shaikhrayyan123/a-comprehensive-guide-to-understanding-bert-from-begi

[2] A. Vaswani *et al.*, "Attention is all you need," in *NeurIPS*, 2017.

[3] J. Devlin *et al.*, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.