Interactive Analysis of Sentiment Retention under BERT Embedding Compression and Deletion

Dylan OBrien (obriend7@wit.edu)
Wentworth Institute of Technology, School of Computing and Data Science, Boston, Massachusetts

Link to Project:
https://github.com/obriend7atwit/NLP_Project_Dylan_OBrien.git

WENTWORTH INSTITUTE OF TECHNOLOGY

Problem and Motivation

Goal:
- Automatically classify **IMDb movie reviews** as **positive** or **negative** using modern NLP methods.

Baseline:
- BERT-based models offer **high accuracy** but are **computationally heavy** and **memory-intensive.**

Real-world constraint:
- Many applications run on **resource-limited devices** (laptops, embedded systems, web dashboards) where full BERT is costly.

Central question:
- ***How much can we compress BERT embeddings or shorten inputs while still preserving sentiment predictions?***

Motivation:
- Reduce **inference cost** and **storage footprint** without severe accuracy loss.
- Enable **faster, lighter** sentiment systems for deployment and teaching.
- Provide an **interactive tool** that helps users see what compression does to text, embeddings, and predictions.

Dataset

Source:
- IMDb 50K Movie Reviews dataset (standard binary sentiment benchmark).
- https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

Size:
- 50,000 labeled reviews
- **25,000 positive, 25,000 negative**

Splits used in this project:
- **Train:** 20k reviews → further split into train/validation
- **Validation:** 5k reviews
- **Test:** 5k reviews, held-out for final evaluation

Properties:
- Reviews are free-form English text with varied length and style.
- Test set contains no overlap in movies with the training set.

Methodology

System Overview

Input:
- Raw IMDb review text.

Encoder:
- Pre-trained ***bert-base-uncased*** (Hugging Face Transformers).

Sentence representation:
- Use the final-layer ***[CLS]*** token as a 768-dimensional embedding.
- BERT is frozen (no fine-tuning) → acts as a **feature extractor**.

Compression stage:
- **PCA-based compression** in embedding space.
- **IDF-based minimal deletion** in input space.

Classifier:
- **Logistic Regression** trained on BERT embeddings.
- Same classifier architecture used for full and compressed representations

Text → BERT → Embedding → (PCA/Deletion) → Logistic Regression → Sentiment

Compression Strategies

PCA Embedding Compression

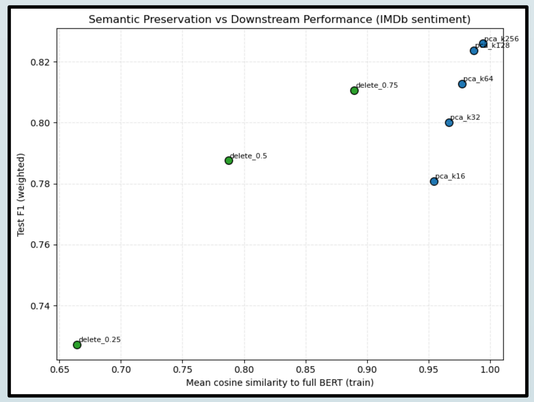
Train Principal Component Analysis (PCA) on training embeddings.
Test multiple target dimensions: k**=16,32,64,128,256**
For each embedding:
- Project to **k-dimensional** space and **reconstruct** back to 768-dim.
- Feed reconstructed embeddings into the **same logistic regression classifier.**

Goal: Reduce embedding dimensionality while keeping sentiment information intact.

IDF-Based Token Deletion

Compute inverse document frequency (IDF) for tokens over the training set.
For each review:
- Rank tokens by **IDF score** (higher IDF = rarer, more informative).
- Keep only the top tokens according to a **keep ratio** r.

Keep ratios explored: r**=1.0,0.75,0.5,0.25** (and others).
Reassemble the shortened review and re-encode with BERT.
Effect: Shorter input, potentially less context, but faster / smaller overall system.



Macro-averaged F1 – balances performance on positive and negative classes.
Mean cosine similarity – how close compressed embeddings are to original ones.

Results

Key Results

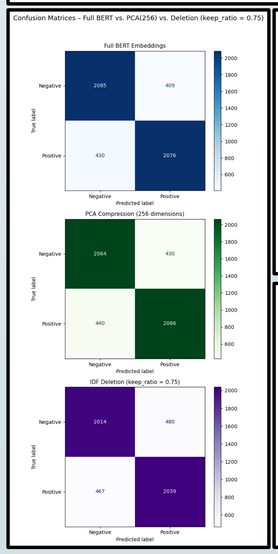
Full BERT baseline:
- Strong **accuracy** and **macro-F1** on IMDb.
- Serves as **upper bound** for compressed variants.

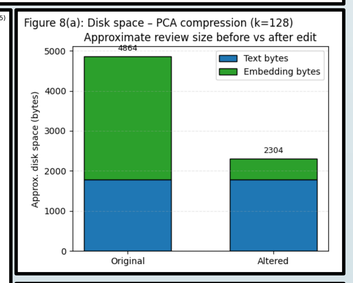
PCA compression:
- At **128–256 dimensions**, performance remains close to baseline.
- Mean cosine similarity stays high, indicating minimal semantic distortion.
- At **≤ 32 dimensions**, F1 and similarity drop somewhat, showing over-compression.
- **CM PCA(256): slight increase in errors but similar pattern to baseline.**

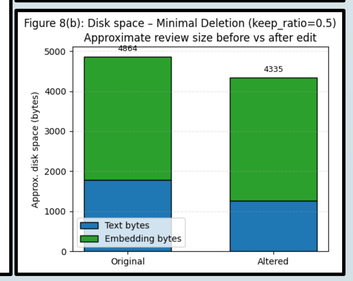
IDF-based deletion:
- Keep ratio **0.75–1.0**: modest deletion, small performance loss.
- Keep ratio **≤ 0.5**: larger drops in accuracy/F1 as important sentiment cues are removed.
- **CM Deletion (0.75): more false negatives or false positives when key phrases are removed.**

Main Takeaways:
- PCA mainly reduces **embedding size**.
- Deletion mainly reduces **text length**.
- **Approximate disk usage (text bytes + embedding bytes) illustrates different compression benefits.**
- **Confusion matrices highlight how compression changes the types of mistakes, not just the overall score.**

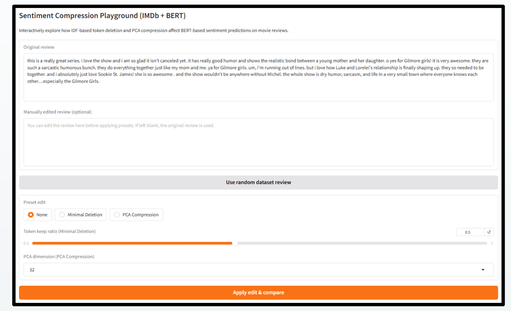
	method	val_accuracy	val_f1	test_accuracy	test_f1	method_type	pca_dim	keep_ratio	approx_size	mean_cosine_train
0	bert_full	0.835667	0.835665	0.8322	0.832199	bert_full	NaN	NaN	768.0	NaN
1	pca_k16	0.780000	0.779960	0.7808	0.780792	pca	16.0	NaN	16.0	0.954354
2	pca_k32	0.803000	0.802999	0.8002	0.800200	pca	32.0	NaN	32.0	0.966361
3	pca_k64	0.815000	0.815001	0.8128	0.812797	pca	64.0	NaN	64.0	0.977239
4	pca_k128	0.821333	0.821326	0.8238	0.823800	pca	128.0	NaN	128.0	0.986770
5	pca_k256	0.828000	0.828001	0.8260	0.826000	pca	256.0	NaN	256.0	0.994124
6	delete_0.25	0.732333	0.732235	0.7272	0.727197	deletion	NaN	0.25	192.0	0.664440
7	delete_0.5	0.793000	0.793001	0.7876	0.787600	deletion	NaN	0.50	384.0	0.787424
8	delete_0.75	0.814333	0.814334	0.8106	0.810598	deletion	NaN	0.75	576.0	0.889541







Interactive Demo (Gradio)



Conclusions, Limitations, and Future Work

Key Insights

BERT embeddings for sentiment are **highly redundant**:
- Significant dimensionality reduction via PCA is possible before performance collapses.

Embedding-space compression (PCA) is generally **safer** than input-space deletion:
- **PCA preserves structure while shrinking representation size.**
- **Deletion can remove critical negation, intensifiers, or context.**

Interactivity helps users **understand and trust model** behavior under compression.

Limitations

Single dataset: IMDb, English movie reviews.
Encoder is frozen (no fine-tuning); results may differ for fully fine-tuned transformers.
Only **two** compression strategies explored:
- **PCA in embedding space.**
- **Static IDF-based deletion in input space.**

Human evaluation is **qualitative** via GUI usage, not a controlled user study.

Future Work

Investigate learned **token pruning**, **knowledge distillation**, and **quantization** for more advanced compression.
Extend framework to:
- **Other domains** (e.g. social media, product reviews, multi-lingual data, larger texts).
- **Aspect-based sentiment analysis** and other NLP tasks.

Conduct user studies on:
- Perceived quality, fairness, and trust in compressed sentiment models.
- How interactive tools influence model understanding.

References

[1] Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," NAACL, 2019.
[2] Wolf et al., "Transformers: State-of-the-art NLP," EMNLP, 2020.
[3] Maas et al., "Learning Word Vectors for Sentiment Analysis," ACL, 2011.
[4] Huertas-García et al., "Exploring Dimensionality Reduction Techniques in Text Embedding Spaces," Appl. Sci., 2022.
[5] Kim et al., "Learned Token Pruning for Transformers," KDD, 2022.
[6] Abid et al., "Gradio: Hassle-free Sharing and Testing of ML Models in the Wild," arXiv:1906.02569, 2019.