Comparing DistilBERT and SBERT for Scalable Semantic Search Classification

Jainil Modi
Georgia Institute of Technology
jmodi30@gatech.edu

Aaron Rodrigues
Georgia Institute of Technology
aaronr@gatech.edu

Joshua Belot
Georgia Institute of Technology
jbelot3@gatech.edu

Arun K Tipingiri
Georgia Institute of Technology
atipingiri3@gatech.edu

Abstract

Classification systems based on pre-trained language models often rely on fine-tuned architectures, which assume a fixed label set and require retraining when classes evolve. This presents scalability and maintenance challenges in real-world applications where new classes are frequently introduced. To address this, we explore the use of semantic embeddings and vector search for classification, comparing pre-trained models DistilBERT and Sentence-BERT (SBERT) in base and fine-tuned forms. Model performance on the datasets Amazon Product Reviews, 20 Newsgroups, and TREC Question Classification, were evaluated on standard accuracy metrics as well as few-shot learning performance. Tuned SBERT achieved the highest accuracy in fixed-label settings on 20 Newsgroups (76.4%) and TREC (88.8%), while tuned DistilBERT performed best in few-shot scenarios on Amazon (37.8%) and TREC (81.5%). These results highlight a trade-off between semantic generalization and task-specific adaptation.

1. Introduction

Text classification is a core task in natural language processing (NLP), where the goal is to assign predefined labels to textual inputs such as product reviews, news articles, or questions. Common applications include topic detection, intent classification, spam filtering, and FAQ categorization. For example, an e-commerce platform may classify user reviews into product categories such as ”electronics” or ”books,” whereas a chatbot may classify user intents to determine an appropriate automated response.

Transformer-based models such as BERT [1] have become standard tools for text classification due to their high accuracy across a variety of benchmarks. Typically, a classifier is trained by appending a linear layer to the [CLS] token embedding and optimizing for cross-entropy loss. However, this approach assumes that the label space is fixed. In real-world applications, such as enterprise document tagging, intent detection for virtual assistants, and evolving FAQ systems, the set of classes may change over time. Each time a new class is introduced, these models require retraining, which is computationally expensive and operationally inflexible [2].

As a more flexible alternative, recent work has explored semantic search-based classification. In this framework, both input texts and labeled class examples are encoded into a shared embedding space. Classification is then performed by nearest-neighbor search using distance metrics such as cosine similarity or Euclidean distance. This enables rapid adaptation to new classes by inserting their embeddings into the search index, which reduces the need for retraining [4, 2]. Additionally, this approach improves model interpretability because it enables inspection of the most similar examples that influenced a prediction [2].

In this project, we investigate the ability of embedding-based classifiers to serve as a scalable alternative to traditional fine-tuned models. We compare two embedding-based methods:

1. KNN classifier using DistilBERT embeddings
2. KNN classifier using SBERT embeddings

DistilBERT is a smaller and faster version of BERT that retains approximately 97% of its classification performance [5], while SBERT is optimized for producing sentence embeddings that are semantically meaningful [4]. SBERT was shown to outperform vanilla BERT embeddings in semantic similarity tasks and enables vector search at scale using libraries like `FAISS` [3]. In this study, fine-tuning for DistilBERT was done via supervised classification loss, while SBERT was trained using contrastive learning with student–teacher regularization.

We evaluated these models on three datasets:

Table 1. Summary of dataset properties used in evaluation

Data Domain	**Amazon Reviews** Product Reviews	**20 Newsgroups** News Articles, Forums	**TREC** Open-Domain Questions
# of Entries	20,000	18,846	5,952
Avg Length (chars)	286	1,170	49
Avg Word Count	53	182	10
Vocabulary Size	71,559	288,723	9,775
# of Classes	5	20	50
Class Distribution	4,000 per class	400–600 per class	27–267 per class
# of Held-out Classes	1	3	26

- **Amazon Product Reviews:** Short, informal reviews categorized by product type.
- **20 Newsgroups:** Longer, topic-based news articles and forum posts across 20 balanced classes.
- **TREC Question Classification:** Brief open-domain questions with coarse and fine-grained label types.

Each dataset varied in domain, size, length, and label complexity. A summary of key dataset features is provided in Table 1. We prioritized diversity in input length, vocabulary, and class balance to stress-test our methods across different real-world scenarios.

Our experiments focused on traditional accuracy benchmarks and also included few-shot learning scenarios, where models classify samples from novel classes using only a few labeled examples. Prior research showed that model frameworks like SetFit can achieve strong results in such settings by leveraging SBERT's embedding space [6].

2. Approach

We evaluated classification performance using semantic embeddings from DistilBERT and Sentence-BERT (SBERT), comparing base models with fine-tuned versions. To decouple inference from parametric classifiers, we applied a non-parametric K-nearest neighbors (KNN) approach using the `FAISS` library [3].

For DistilBERT, the `distilbert-base-uncased` model [5] was fine-tuned using the HuggingFace `transformers` library [7] by adding a classification head and optimizing cross-entropy loss with the Trainer API. After training, the classification head was removed, and the hidden representation of the `[CLS]` token from the final layer was extracted and used as the input embedding for KNN classification.

For SBERT, the `all-MiniLM-L6-v2` model provided by the `sentence-transformers` library [4] was used. It was initially fine-tuned with `CosineSimilarityLoss`, but we observed collapsed embeddings and poor performance. Switching to `MultipleNegativesRankingLoss`, which contrasts

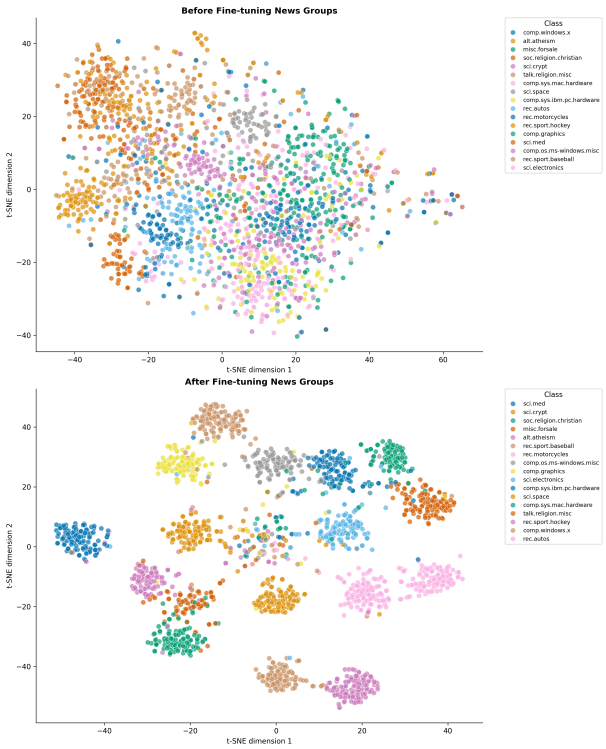

Figure 1. Visualized DistilBERT embeddings for 20 Newsgroups, before and after fine-tuning

each positive pair against in-batch negatives, produced sharper separation and improved accuracy. We also observed signs of catastrophic forgetting, where fine-tuning distorted the pretrained embedding space and hurt generalization. To mitigate this, we introduced a student–teacher objective: both the student and a frozen teacher computed pairwise cosine similarity distributions, and a KL divergence loss penalized their divergence. The final loss combined contrastive and KL divergence terms, balancing task-specific adaptation with structural stability. Batch size, learning rate, and other hyperparameters were tuned to optimize performance.

All models were evaluated using a nearest-neighbor classification pipeline. Embeddings from training samples were

stored in a FAISS index. During inference, test samples were encoded and classified by majority vote among their top-k nearest neighbors ($k = 5$).

To evaluate generalization, certain classes were withheld during training. At test time, a few (5) labeled examples from these holdout classes were embedded and added to the FAISS index (see Section 3.2). The models were then used to classify additional examples from the unseen classes without retraining.

Models were evaluated using Overall Accuracy, Precision (weighted), Recall (weighted), F1-Score (weighted), Mean Reciprocal Rank (MRR), and Top-k Accuracy ($k = 5$). Accuracy and F1-Score summarized general performance, while precision and recall captured trade-offs between false positives and false negatives. MRR and Top-k accuracy assessed the ranking behavior of the KNN models by measuring how often the correct class appeared near the top of the similarity list. To further interpret the embedding space, we visualized the learned representations using t-SNE. Figure 1 shows the 2D projection of DistilBERT embeddings on the 20 Newsgroups dataset, illustrating how well the tuned model separates class into distinct clusters.

3. Experiments and Results

3.1. Experiment 1: Finetuning Pre-trained Models

3.1.1 Amazon Reviews Dataset

Results

The results for the Amazon Reviews Dataset are shown in Table 2. The base SBERT model achieved an accuracy of 78.6%, indicating that the embeddings effectively separated the product review classes. The fine-tuned DistilBERT classification model outperformed the base SBERT across all metrics. After fine-tuning, DistilBERT achieved an accuracy of approximately 85%. This improvement was expected, due to DistilBERT's ability to directly optimize for classification, learning decision boundaries that best separate the training classes. The 2D, t-SNE projections of the embedding space before and after tuning the two models can be seen in Supplementary Figures 2 and 3.

Discussion

The success of fine-tuned DistilBERT over the base SBERT was largely driven by the nature of the dataset. Amazon reviews are relatively short (average of 53 words) and domain-specific, with a low topic complexity. This favors classification layers that directly exploit token-level patterns in text as opposed to relying on general-purpose embeddings.

The base SBERT model performed well despite no direct label supervision during training. Its high MRR and Top-5 accuracy indicate that the sentence embeddings capture much of the necessary information for distinguishing classes. However, without a task-specific classification head, the decision boundary remains less tightly optimized.

After tuning, DistilBERT showed improved classification performance and successfully adapted to the product review domain. We did not observe overfitting, likely due to the balanced class distribution and the relatively low number of classes (5). However, as the number of classes increases or the domain becomes more complex, the margin between tuned and untuned models would likely widen further. See below with the results from the TREC dataset.

SBERT excels in transfer-learning scenarios, where semantic similarity is valuable and new classes may emerge after training, because its pre-trained embedding space already encodes rich sentence-level relationships without needing additional supervision. However, this strength becomes a weakness when applied to fixed-class classification tasks, especially when fine-tuning on noisy or highly variable data like Amazon Reviews. The semantic space that SBERT builds was optimized for general sentence similarity, not for discriminating between fine-grained, domain-specific labels like "Electronics" vs. "Home and Kitchen". In our experiments, fine-tuning SBERT on Amazon Reviews often led to semantic drift: the embedding space, originally structured for semantic proximity, was distorted as the model tried to force entries into rigid class boundaries. Amazon Reviews' relatively short inputs (286 characters) and overlapping vocabulary between product categories made this problem worse. The [CLS] representations or pooled embeddings that were once semantically meaningful lost their flexibility after fine-tuning, reducing generalization to new examples and degrading classification metrics.

3.1.2 20 Newsgroups

Results

Table 2 shows the success metrics of the four models. Base SBERT performed well without any training, as seen by the relatively high metrics and the 77.33% accuracy. The base DistilBERT performed much more poorly; however, fine-tuning DistilBERT allowed the model to perform closer to base SBERT with a 72.6% accuracy. The 2D, t-SNE projections of the embedding space before and after tuning the two models can be seen in Figure 1 and Supplementary Figure 4.

Discussion

SBERT is already designed and trained with contrastive loss, meaning that it learned well which embeddings belong close together and which ones do not inside of the embedding space. It excels at extracting the core semantics from text, which is why it can perform well even with the large

Table 2. Experiment 1 Results.

Dataset	Model	Performance metrics					
		Accuracy	F1-Score	Precision	Recall	MRR	Top-5 Acc
Amazon Reviews	Base SBERT	0.7856	0.7829	0.7873	0.7856	0.8527	0.9336
	Tuned SBERT	0.6332	0.6235	0.6283	0.6332	0.7557	0.9039
	Base DistilBERT	0.6748	0.6669	0.6747	0.6748	0.7855	0.9139
	Tuned DistilBERT	**0.8366**	**0.8357**	**0.8365**	**0.8366**	**0.8835**	**0.9402**
20 Newsgroups	**Base SBERT**	**0.7733**	**0.7770**	**0.7894**	**0.7733**	**0.8438**	**0.9144**
	Tuned SBERT	0.7643	0.7664	0.7769	0.7643	0.8286	0.8980
	Base DistilBERT	0.5898	0.5929	0.6114	0.5898	0.6944	0.8234
	Tuned DistilBERT	0.7260	0.7291	0.7354	0.7260	0.7714	0.8259
TREC	Base SBERT	0.589	0.5782	0.5821	0.589	0.6922	0.8305
	Tuned SBERT	**0.8876**	**0.8838**	**0.8856**	**0.8876**	**0.9165**	**0.9438**
	Base DistilBERT	0.4204	0.4018	0.4188	0.4204	0.5478	0.7079
	Tuned DistilBERT	0.883	0.877	0.8848	0.883	0.9083	0.9382

entries (average of 182 words) found in 20 Newsgroups. As stated earlier, 20 Newsgroups has a very large and noisy vocabulary, and SBERT is well equipped to ignore this noise as seen in the results.

Instead of capturing semantic information, DistilBERT used the `[CLS]` token to encode an entry's patterns, which often included uncommon words. This may cause problems because it can view two documents as similar if they have similar uncommon tokens, even if the semantic meaning is completely different. This makes base DistilBERT especially sensitive to large vocabularies of rare words because it is much more difficult to identify noisy entries belonging to the same class. This poor performance is reflected in Table 2, which shows that DistilBERT necessitated fine-tuning to adapt the embedding space for effective classification on 20 Newsgroups.

We used a strong KL weight ($\alpha = 0.85$) in fine-tuning SBERT to help the previous embedding space stay intact, but it still could not match the SBERT base model. The amount of noise in these longer posts is likely at fault for shifting the embedding space to be worse. Techniques such as chunking longer posts or more rigorous cleaning may assist to improve the fine-tuned model's performance.

3.1.3 TREC

Results

Evaluation with the TREC test set reveals a clear performance gap between baseline and fine-tuned models. As shown in Table 2, the base SBERT model achieved 58.9% accuracy, which improved to 88.76% after fine-tuning. The base DistilBERT model performed worse initially at 42.04%, but fine-tuning raised its accuracy to 88.3%, closely matching the tuned SBERT model. The 2D, t-SNE projections of the embedding space before and after tuning the two models can be seen in Supplementary Figures 5 and 6.

Discussion

The performance gains from tuning can be attributed to several data and task specific factors. The short average sequence length (10 words) and low topic complexity made the dataset ideal for models focused on sentence-level semantics like SBERT and DistilBERT. However, the class imbalance and moderate label overlap introduced ambiguity and difficulty in distinguishing fine-grained classes.

The base models used general-purpose embeddings that were not specialized for the TREC task, which limited their ability to capture subtle distinctions between similar classes and explains the poor baseline scores. Tuning the models addressed this, enabling them to learn discriminative representations that better separated semantically similar labels.

Across models, SBERT exhibited stronger baseline performance due to its pretraining on sentence-level similarity tasks, making it more suitable for out-of-the-box use. In contrast, DistilBERT required tuning to reach comparable performance but ultimately matched or slightly trailed SBERT once optimized.

Interestingly, both tuned models converged to near-identical results in every metric, which suggests that the existing model weights and tuning method becomes less critical once task-aligned fine-tuning is applied. Additionally, the significant performance improvements of DistilBERT after tuning highlights its capacity to serve as a lightweight yet effective model in resource-constrained settings, provided sufficient tuning is performed.

3.1.4 Cross-Dataset Discussion

Comparing model performance across datasets highlights how input characteristics affect results. Fine-tuned DistilBERT performed best when entries were short and classes were few, as in Amazon Reviews. TREC, despite having more classes, still favored DistilBERT because shorter entries made it easier for the `[CLS]` token to represent the full input without losing critical information. In contrast, for longer, noisier entries with overlapping classes, as in 20 Newsgroups, SBERT's flexible semantic embedding space was better suited for classification.

3.2. Experiment 2: Few-Shot Learning

In Experiment 2, we evaluated each model's ability to generalize to new, unseen classes during training. This scenario mimics a real-world few-shot learning setting, where new classes of data emerge dynamically post-deployment.

3.2.1 Amazon Reviews Dataset

For this experiment, we held out the "Books" category from both SBERT and DistilBERT during the training phase and reintroduced it for testing.

Results

As seen in Table 3, the tuned DistilBERT model achieved the strongest performance across all metrics on the Amazon Reviews dataset. It reached the highest accuracy at 37.79% indicating effective class separation and improved generalization. In contrast, the base DistilBERT model performed noticeably worse, with an accuracy of only 29.94%. The tuned SBERT model underperformed in overall accuracy at 28.59% compared to its base counterpart.

Discussion

SBERT is built for sentence similarity tasks using contrastive loss and does not rely on direct supervision for classification. However, in a fixed-label scenario like Amazon product categories, the inability to fine-tune effectively may limit its upper performance bound. The results suggest that DistilBERT's architecture, when adapted through supervised learning, can learn highly discriminative class boundaries even in relatively noisy, user-generated text. SBERT, while semantically powerful out-of-the-box, underperformed when fine-tuned in this setting, potentially due to the degradation of its pretrained embedding space during supervised optimization.

The structure of the Amazon dataset also played a key role in these results. The input reviews are relatively short (averaging 53 words), class distribution is balanced, and there is modest semantic overlap between categories. These characteristics made the dataset a good candidate for models that can benefit from task-specific adaptation, as DistilBERT did.

SBERT is ideal in scenarios where flexibility and semantic proximity matter—such as few-shot classification or retrieval tasks using FAISS. However, these results suggest that DistilBERT performs better in environments with fixed, well-represented labels that can directly be used for tuning.

3.2.2 20 Newsgroups

The held out set for 20 Newsgroups comprised of three subgroups under the talk general category. These subgroups were politics-mideast, politics-guns, and politics-misc.

Results

Table 3 shows the results from classifying the hold-out set. The tuned SBERT model performed the best, with an accuracy of 44.1%. This was not a significant lift over the base SBERT, but it indicates the effectiveness of the fine-tuning process. The tuned DistilBERT performed much better on classifying the hold-out set than the base DistilBERT model yet still performed about as well as the base SBERT.

Discussion

The hold-out set is difficult to classify because the three subclasses have much in common. Another issue is that the hold-out set is very different from the other classes, so placing political topics in an embedding space full of computer or sports entries is difficult, even when fine-tuned. These are also relatively long entries with noisy, unique words, which hurt DistilBERT's performance.

SBERT generally performs better on this task because it provides a well-structured embedding space focused on capturing semantic meaning. Given the inherent noise and variability in language, preserving semantic content is critical. In contrast, during fine-tuning, DistilBERT can overemphasize uninformative tokens at the expense of more meaningful features. The pooling strategy while tuning SBERT mitigates this issue by effectively summarizing relevant information while filtering out noise.

With both models, we see that tuning has a positive effect on the few-shot learning task. Fine-tuning DistilBERT enabled the model to recognize identifying terms in an entry. For SBERT, student-teacher tuning was used to prevent overfitting because the base model was already decent, and this helped the model stay flexible enough to generalize more to the hold-out set. However, SBERT still struggled with the held out group. This is likely because the three held out classes all belong to the same overall group, which puts the entries closer in the embedding space and makes them more difficult to distinguish by a model which has never

Table 3. Experiment 2 Results.

Dataset	Model	Performance metrics					
		Accuracy	F1-Score	Precision	Recall	MRR	Top-5 Acc
Amazon Reviews	Base SBERT	0.3499	0.2624	0.7854	0.3499	0.4079	0.4779
	Tuned SBERT	0.2859	0.2205	0.7492	0.2859	0.3943	0.5260
	Base DistilBERT	0.2994	0.2087	0.7325	0.2994	0.3519	0.4128
	Tuned DistilBERT	**0.3779**	**0.3058**	**0.8265**	**0.3779**	**0.4324**	**0.4984**
20 Newsgroups	Base SBERT	0.3920	0.3140	0.5438	0.3920	0.4802	0.5601
	Tuned SBERT	**0.4410**	**0.3786**	**0.6224**	**0.4410**	**0.5242**	**0.6020**
	Base DistilBERT	0.3264	0.2494	0.2119	0.3264	0.3913	0.4705
	Tuned DistilBERT	0.4058	0.3415	0.5935	0.4058	0.4514	0.5045
TREC	Base SBERT	0.4427	0.3925	0.4639	0.4427	0.5594	0.7059
	Tuned SBERT	0.8089	0.7977	0.8535	0.8089	0.8634	0.9106
	Base DistilBERT	0.2973	0.2449	0.2679	0.2973	0.4099	0.5412
	Tuned DistilBERT	**0.8153**	**0.8053**	**0.8531**	**0.8153**	**0.8756**	**0.9266**

seen them before.

3.2.3 TREC

The hold-out set for TREC comprised of 26 subclasses with about 615 records.

Results

As seen in the Table 3, The base SBERT model yielded 44.27% accuracy. After tuning, performance jumped to 80.89% accuracy showcasing a substantial boost in both classification precision and retrieval relevance. The base DistilBERT model performed significantly lower with only 29.73%. After tuning, accuracy significantly increased to 81.53%. This makes it not only competitive with but slightly better than the tuned SBERT model.

Discussion

These results are largely shaped by both the nature of the dataset and the models' pretraining regimes. TREC questions are short (averaging 10 words) and semantically dense, but the label overlap and class imbalance makes classification non-trivial. The baseline failures can be attributed to a lack of task-specific fine-tuning: pretrained models like SBERT and DistilBERT, though powerful, do not inherently know how to differentiate between TREC categories without adaptation. The few-shot setup exacerbates this, as models must generalize from limited exposure. Tuning with cosine similarity was critical in aligning embeddings with label semantics, allowing both models to generalize effectively to unseen categories.

Between models, base SBERT exhibited better performance, likely due to more effective pretraining for semantic tasks. However, DistilBERT showed the most dramatic improvement post-tuning (from 29.73% to 81.53% accuracy). This suggests that while DistilBERT starts weaker, it is highly responsive to targeted training. After tuning, both models converge closely in performance, with DistilBERT even slightly surpassing SBERT. This convergence indicates that architectural differences matter less once models are aligned with the task through tuning.

3.2.4 Cross-Dataset Discussion

When comparing the few-shot learning performance across datasets it is apparent that the nature of each dataset significantly affects how SBERT and DistilBERT are able to generalize and handle new classes in the embedding space.

In Amazon Reviews, SBERT demonstrated a clear advantage due to its ability to generalize using semantic embeddings and dynamically incorporate new classes via FAISS indexing. In 20 Newsgroups, while SBERT still led in performance, the overlap among the held-out subclasses (politics-related topics) reduced separability in the embedding space, making fine-grained classification more difficult. However, using the TREC dataset, tuning resulted in major improvements for both SBERT and DistilBERT. This highlights that while SBERT is generally more adaptable in few-shot settings, DistilBERT can become competitive when tuned appropriately.

4. Future Work

Following these findings, we identify key directions to improve embedding-based classification. Probing and visualization methods can assess how well the embedding space aligns with label semantics, particularly when class boundaries are ambiguous or overlapping. While our current FAISS index includes all training points, future work can

explore sampling strategies such as hard negative mining, prototype averaging, or clustering-based selection to improve robustness under class imbalance. Finally, re-ranking retrieved candidates using lightweight lexical models like BM25 may help correct semantic drift and improve top-k classification accuracy.

5. References

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[2] Kishaloy Halder, Josip Krapac, Alan Akbik, Anthony Brew, and Matti Lyra. Task-Specific Embeddings for Ante-Hoc Explainable Text Classification, 2022.

[3] Jeff Johnson, Matthijs Douze, and Herve Jegou. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, July 2021.

[4] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[5] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2019.

[6] Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. Efficient Few-Shot Learning Without Prompts, September 2022. arXiv:2209.11055.

[7] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-Art Natural Language Processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

A. Work Division

Table 4. Contributions of each team member.

Student Name	Contributed Aspects	Details
Jainil Modi	Amazon Reviews Subset Curation; fine-tuning SBERT/DistilBERT Models	Downloaded large samples of Amazon Data and created subset using random sampling technique. Additionally, evaluated efficacy of SBERT and DistilBERT on pre-trained and fine-tuned versions on Amazon Reviews Subset Dataset.
Aaron Rodrigues	Project planning and organization. Embedding search, evaluation, and visualization framework	Led project planning and coordinated experimental design. Developed reusable modules for dataset preprocessing, SBERT/DistilBERT evaluation, and visualization pipelines across the three datasets.
Joshua Belot	Fine-tuning SBERT/DistilBERT Models for 20 Newsgroups Dataset	Evaluated efficacy of SBERT and DistilBERT on pre-trained and fine-tuned versions on the 20 Newsgroups Dataset.
Arun K Tipingiri	Fine-tuning SBERT/DistilBERT Models for TREC Dataset	Evaluated efficacy of SBERT and DistilBERT on pre-trained and fine-tuned versions on the TREC Dataset.

B. Appendix

Code Repository

The full implementation for all models, preprocessing scripts, and experiments can be found at our public GitHub repository:
`https://github.com/jmodi23/CS7643_Final_Project`

Supplemental Images

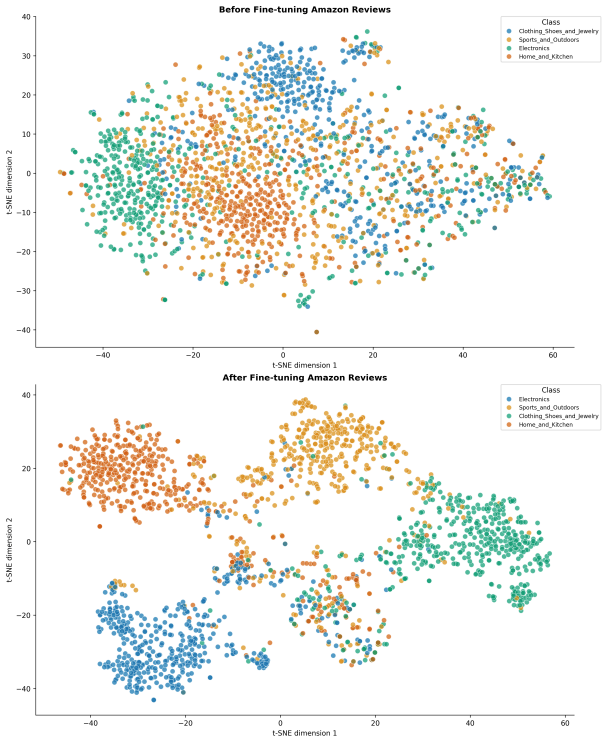

Figure 2. Amazon Reviews with DistilBERT

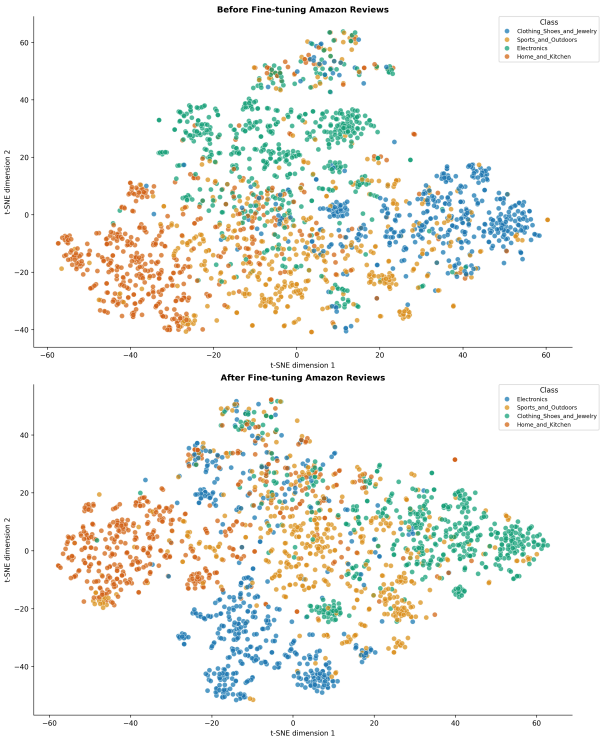

Figure 3. Amazon Reviews with SBERT

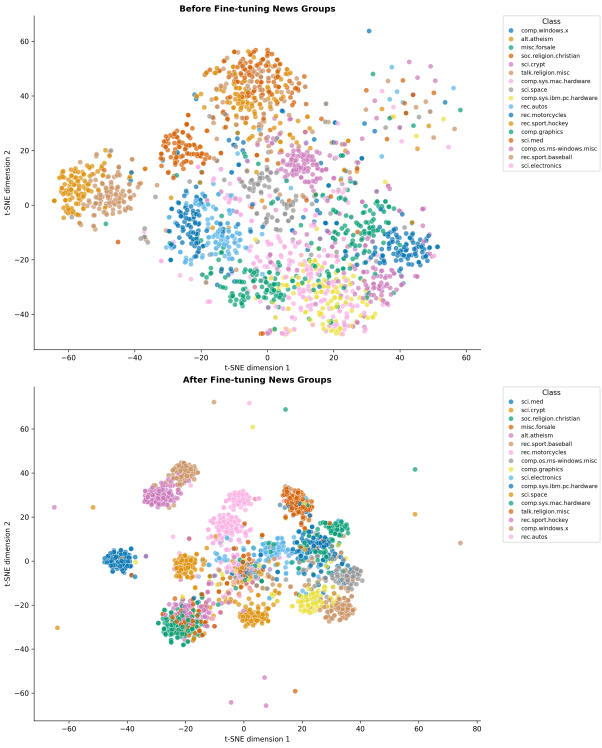

Figure 4. 20 Newsgroups with SBERT

Figure 5. TREC with DistilBERT

Figure 6. TREC with SBERT