Optimizing NLP Classification Models: Vocabulary Reduction and Embedding Size Scaling

Anthony Martini
Bellini College of Artificial Intelligence, Cybersecurity, and Computing
University of South Florida
Tampa, Fl, USA
acmartini@usf.edu

Abstract

This study examines whether large vocabularies in large language models are necessary for successful downstream classification. By utilizing NLTK WordNet to map tokens to synsets, we implemented a hierarchical consolidation strategy that reduced vocabulary redundancy by 13.03%. Testing on the IMDb sentiment benchmark revealed that this filtered vocabulary not only maintained baseline accuracy in standard architectures but also mitigated performance loss when the model's hidden size was reduced by 50%. These results demonstrate that semantic consolidation creates an "information-dense" vocabulary, allowing for significant reductions in computational overhead without compromising model efficacy.

1. Introduction

The rapid evolution of Natural Language Processing has led to models with increasingly larger parameter counts and vocabulary sizes. While these expansive vocabularies capture subtle linguistic nuances, they also introduce significant redundancy. In many downstream classification tasks, the distinction between synonymous terms may be computationally expensive yet semantically negligible.

This paper examines the necessity of maintaining distinct embeddings for synonyms and logically similar tokens in fine-tuned classification models. Our central premise is that discerning the subtle differences between semantically equivalent words is not essential for successful sentiment classification.

To test this hypothesis, we propose a method of *de novo* model training where similar tokens are identified and consolidated before training. By mapping synonyms to a single representative token, we aim to reduce the vocabulary size and computational overhead while maintaining high performance on binary classification tasks.

To further probe the efficiency implications of vocabulary reduction, we extend our hypothesis to the model architecture itself: models with a smaller, filtered vocabulary may require a simpler embedding space to represent the remaining nuances of words, offering a secondary path to reduced computational overhead.

2. Experimental Setup & Methodology

2.1 Datasets

Our training pipeline utilizes two distinct datasets to separate general language acquisition from specific task performance:

1. **Pretraining (WikiText-103)**: To build general linguistic competence, we used the WikiText-103 dataset (Merity et al.). This collection, accessed via Hugging Face, contains over 100 million tokens sourced from verified Good and Featured Wikipedia articles. This collection provides a high-quality sample of formal English suitable for large-scale language modeling.
2. **Pretraining (IMDB Reviews)**: For downstream sentiment classification, we employed the IMDb Large Movie Review Dataset (Maas et al.), accessed from Hugging Face, which contains 25,000 labeled training reviews and 25,000 labeled test reviews. This benchmark supports binary sentiment prediction and is widely used to evaluate text classification models.

2.2 Model Architecture

All Base and Filtered models share an identical transformer architecture with set hyperparameters, including Flash Attention 2 (Dao) for faster attention, to ensure performance differences are solely attributable to vocabulary optimization. The experiment includes Original and Smaller model sizes, with the Smaller Size testing the secondary hypothesis that a reduced vocabulary allows for a commensurately smaller Hidden Size.

Original Size Models		**Smaller Size Models**	
Parameter	**Value**	**Parameter**	**Value**
Hidden Layers	8	Hidden Layers	8
Attention Heads	8	Attention Heads	8
Hidden Size	512	Hidden Size	256
Intermediate Size	2048	Intermediate Size	1024
Max Position Embeddings	256	Max Position Embeddings	256
Attention Implementation	Flash Attention 2	Attention Implementation	Flash Attention 2
Initial Vocab Size	30,000	Initial Vocab Size	30,000

2.3 Training Procedure

Our training pipeline utilizes two distinct datasets to separate general language acquisition from specific task performance:

1. **Base Model Training:** We first trained a control model using a standard 30,000-token WordPiece tokenizer on the Wikitext corpus. This model was subsequently fine-tuned on the IMDb dataset to establish a performance baseline.
2. **Vocabulary Filtration & Retraining:** We analyzed the base tokenizer's vocabulary to identify semantic redundancies (detailed in Section 3). Based on our filtration criteria (NLTK-Consolidated and Subset-Inclusion), we generated a mapping to replace redundant tokens with their semantic equivalents within the corpus. A new model was then trained *de novo* on this filtered corpus and fine-tuned on the IMDb dataset.

2.4 Hardware and Environment

All computational tasks, including tokenization, pretraining, and fine-tuning, were executed in a Google Colab environment utilizing a single NVIDIA H100 GPU.

2.5 Performance Metrics

Model efficacy is assessed on the IMDb test set (25,000 examples) using Accuracy, Precision, Recall, and F1 Score.

3. Tokenizer Generation

3.1 Base Model

The Base Model's tokenizer serves as our primary benchmark and the starting point for the following models. Its vocabulary consists of 30,000 unique tokens generated via a WordPiece tokenizer algorithm trained on the Wikitext corpus. To maintain consistency, we normalized all text to lowercase, removed accents, and added spacing around Chinese characters.

3.2 NLTK-Consolidated Model (Version 1: Strict Equivalence)

The first iteration of our optimization strategy focused exclusively on strict semantic equivalence. We utilized the NLTK WordNet database (Loper and Bird) to map tokens to their synsets (Miller), sets of synonyms sharing a common meaning, and filtered specifically for pairs that required the set of definitions for both words to be identical.

Limitations

While this approach successfully consolidated morphological variations (e.g., "generating" vs. "generated") and spelling variants ("adviser" vs. "advisor"), it failed to capture true synonyms where one term is polysemous, such as “movie” and “film”. “Movie” has a singular meaning referring to cinema, but because film contains additional definitions (e.g., a thin layer of material), they are not interchangeable.

Result

Despite these limitations, this method successfully identified and removed 2,186 redundant tokens–7.29%–from the base vocabulary.

3.3 Subset-Inclusion Model (Version 2: Hierarchical Reduction)

To address the limitations of Version 1, which required definitions to be identical, Version 2 adopts a hierarchical approach. We recognized that semantic relationships are often vertical rather than horizontal: a specific term is often fully encompassed by a general term, even if they are not exact synonyms.

Methodology

To identify semantic relationships, we analyzed the tokenizer vocabulary for pairs satisfying the condition S_A ⊆ S_B, where S denotes the set of WordNet synsets for a given token. These pairs were then processed into a JSON mapping from specific subset words to their broader supersets, refined by the following logic:

1. Disambiguation: Any word associated with multiple potential supersets is removed from the mapping to ensure each specific term maps to only one broader term.
2. Transitive Flattening: In cases where a word acts as both a subset and a superset (e.g., word A maps to B, and word B maps to C), we collapse the chain so that word A maps directly to the most distal superset, word C.

Result

This approach successfully collapses pairs like "movie" → "film". This strategy resulted in a significantly more aggressive reduction, removing 3,910 tokens–13.03%–from the vocabulary

4. Results

4.1 Training Results

This section presents the training loss results for the two model sizes investigated: Original Size and Smaller Size. Each size includes the Base Model configuration and two Wordnet token filter variants (V1 and V2), with results separated into Pretraining and Fine-Tuning stages.

Original Size Models

Model Configuration	Stage	Training Loss	Epoch	Train Time (s)
Base Model	Pretraining	3.55956	4	1310.7072
	Fine-Tuning	0.122611	10	113.0061
Wordnet V1	Pretraining	3.538649	4	1414.063
	Fine-Tuning	0.122788	10	113.4786
Wordnet V2	Pretraining	3.521829	4	1382.7368
	Fine-Tuning	0.120737	10	109.9172

Smaller Size Models

Model Configuration	Stage	Training Loss	Epoch	Train Time (s)
Base Model	Pretraining	4.701199	3	668.4157
	Fine-tuning	0.176365	10	108.3085
Wordnet V1	Pretraining	4.636131	3	767.8154
	Fine-tuning	0.210016	10	105.8329
Wordnet V2	Pretraining	4.617355	3	736.8979
	Fine-tuning	0.21247	10	97.1544

4.2 Evaluation Results

Original Size Models

Metric	BaseModel	Wordnet V1	Wordnet V2	Best Model	Improvement vs Base
Accuracy	0.8739	0.8753	0.8757	Wordnet V2	0.0017
Precision	0.868	0.8717	0.8715	Wordnet V1	0.0037
F1 Score	0.8749	0.8759	0.8763	Wordnet V2	0.0014
Recall	0.882	0.8802	0.8813	BaseModel	N/A

Smaller Size Models

Metric	BaseModel	Wordnet V1	Wordnet V2	Best Model	Improvement vs Base
Accuracy	0.8529	0.8547	0.8559	Wordnet V2	0.0031

Precision	0.862	0.8574	0.8663	Wordnet V2	0.0042
F1 Score	0.8509	0.8541	0.8539	Wordnet V1	0.0032
Recall	0.8405	0.8508	0.8418	Wordnet V1	0.0104

Performance Comparison Between Model Sizes

Metric	BaseModel	Wordnet V1	Wordnet V2	Most Robust
Accuracy	2.11%	2.06%	1.97%	Wordnet V2
Precision	0.60%	1.43%	0.52%	Wordnet V2
F1 Score	2.40%	2.18%	2.25%	Wordnet V1
Recall	4.16%	2.93%	3.95%	Wordnet V1

5. Discussion of Results

The experimental results validate the hypothesis that semantic redundancy in vocabularies introduces computational overhead without contributing to classification efficacy.

5.1 Effectiveness of Vocabulary Filtration

The Subset-Inclusion strategy (WordNet V2) proved significantly more robust than strict equivalence (V1). By shifting from a 1:1 mapping to a hierarchical approach, V2 addressed the limitations of polysemy, capturing relationships like movie→ film, resulting in a 13.03% reduction in vocabulary. This confirms that a hierarchical understanding of synsets is necessary to effectively collapse redundant lexical space.

5.2 Performance of Original Size Models

Data from the 512-hidden-size models demonstrate that vocabulary reduction does not merely preserve performance; it can enhance it. WordNet V2 achieved the highest Accuracy and F1 Score, suggesting that "denoising" the vocabulary helps the model focus on core sentiment features. The lower pretraining loss in filtered models further indicates that a consolidated vocabulary provides a more efficient starting point for language acquisition.

5.3 Efficiency Implications of Smaller Size Models

The results from the 256-hidden-size models support the secondary hypothesis: a compact vocabulary facilitates a smaller embedding space. While reducing the hidden size naturally lowers performance, the WordNet-filtered models consistently outperformed the Base Model at this scale. Specifically, WordNet V2 showed the lowest performance degradation, with only a 1.97% drop in accuracy despite a 50% reduction in hidden size. This indicates that a curated vocabulary is more "information-dense," allowing a smaller architecture to capture the same semantic nuances that would otherwise require a larger parameter count.

6. Conclusion

This study demonstrates that reducing semantic redundancy in a model's vocabulary is an effective strategy for optimizing efficiency in NLP classification. The Subset-Inclusion (Wordnet V2) approach proved the most effective consolidation method, reducing the vocabulary by 13.03%. When applied to the Original Size architecture, this filtration yielded a modest performance improvement, achieving the highest Accuracy and F1 Score.

The results confirm that vocabulary reduction enables smaller embedding dimensions with minimal performance loss. Filtered "Smaller Size" models consistently outperformed their base counterparts, with WordNet V2 proving most robust. This demonstrates a strategic trade-off: optimizing vocabulary allows for a 50% reduction in Hidden Size (512 to 256) while maintaining high classification efficacy. Future work will quantify the resulting gains in memory efficiency and inference speed.

References

Dao, Tri. "FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning." arXiv, 2023, https://arxiv.org/abs/2307.08691.

Loper, Edward, and Steven Bird. "NLTK: The Natural Language Toolkit." Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, 2002, https://www.nltk.org/.

Maas, Andrew L., et al. "Learning Word Vectors for Sentiment Analysis." Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL), 2011, pp. 142-50.

Merity, Stephen, et al. "WikiText-103." Hugging Face, Salesforce, 2022, https://huggingface.co/datasets/Salesforce/wikitext.

Miller, George A. "WordNet: A Lexical Database for English." Communications of the ACM, vol. 38, no. 11, 1995, pp. 39-41, https://wordnet.princeton.edu/.

Pietrolesci, Roberto. "IMDb Dataset." Hugging Face, 2023, https://huggingface.co/datasets/pietrolesci/imdb.