

Article

Geometry of Textual Data Augmentation: Insights from Large Language Models

Sherry J. H. Feng , Edmund M-K. Lai * and Weihua Li 

School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand; jiahui.feng@autuni.ac.nz (S.J.H.F.); weihua.li@aut.ac.nz (W.L.)

* Correspondence: edmund.lai@aut.ac.nz

Abstract: Data augmentation is crucial for enhancing the performance of text classification models when labelled training data are scarce. For natural language processing (NLP) tasks, large language models (LLMs) are able to generate high-quality augmented data. But a fundamental understanding of the reasons for their effectiveness remains limited. This paper presents a geometric and topological perspective on textual data augmentation using LLMs. We compare the augmentation data generated by GPT-J with those generated through cosine similarity from Word2Vec and GloVe embeddings. Topological data analysis reveals that GPT-J generated data maintains label coherence. Convex hull analysis of such data represented by their two principal components shows that they lie within the spatial boundaries of the original training data. Delaunay triangulation reveals that increasing the number of augmented data points that are connected within these boundaries correlates with improved classification accuracy. These findings provide insights into the superior performance of LLMs in data augmentation. A framework for predicting the usefulness of augmentation data based on geometric properties could be formed based on these techniques.

Keywords: data augmentation; large language models; text classification; topological data analysis



Citation: Feng, S.J.H.; Lai, E.M.-K.; Li, W. Geometry of Textual Data Augmentation: Insights from Large Language Models. *Electronics* **2024**, *13*, 3781. <https://doi.org/10.3390/electronics13183781>

Academic Editors: Jian Liu, Bo Xu and Linmei Hu

Received: 9 August 2024

Revised: 10 September 2024

Accepted: 18 September 2024

Published: 23 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Supervised machine learning for classification tasks involves training models with labelled training data. In order to achieve generalization for accurate classification of previously unseen data, the availability of a sufficiently large amount of training data is crucial. In practice, this is often hampered by high annotation and labelling costs [1]. Data augmentation (DA) is a way to expand the training dataset by generating additional data artificially through label-preserving transformations [2] without deviating from the original dataset's underlying distribution. Apart from increasing the volume of data, DA could also be used to enhance the diversity of training data, thereby enhancing model performance and robustness.

Generating augmentation data for computer vision tasks such as object detection and recognition is relatively easy. This is because of the fact that it is obvious what kinds of transformation of the original data would increase the diversity of the dataset [3]. For example, creating augmented images with the target objects scaled, translated, and rotated to various degrees could increase the robustness of object orientation, size, and location in the image. However, with natural languages, it is often not obvious what types of transformation of the original sentences would increase the diversity of the original dataset. Consequently, many text augmentation methods have been proposed, but none of them have been found to be generally effective [4].

More recently, large language models (LLMs) like BERT [5] and GPT [6] are able to provide unprecedented text generation capabilities. Making use of these models' text data augmentation often results in much improved performances for natural language

processing (NLP) tasks such as text classification [7–10]. Despite these empirical successes, a fundamental understanding of what constitutes useful text augmentation data is still lacking.

In this paper, we present our investigation into what constitutes useful text augmentation. Using text classification as the NLP task, we compare the augmentation data generated by GPT-J with those based on more traditional word embeddings—Word2Vec and GloVe. Since DA based on LLMs is much more effective than that based on word embedding, we can determine what kind of augmentation data are useful. For this comparison, a geometric and topological perspective is adopted where computational geometry and topological data analysis tools are utilized.

Our investigation yields the several following key insights:

1. Augmented data points generated by LLMs like GPT-J are closely aligned with the original training data in terms of spatial boundaries, maintaining semantic integrity and ensuring consistency of labels. This is in contrast to augmented data points generated by Word2Vec and GloVe embeddings, which often extend beyond the boundaries of the original training data.
2. The addition of meaningful augmented data points within the convex hull of the original training data significantly enhances the efficacy of text classification systems by providing richer training datasets. Increasing the number of augmented data points within these defined boundaries correlates with improved classification accuracy.
3. Techniques such as topological data analysis, convex hull, and Delaunay triangulation prove effective in analyzing the spatial distribution and connectivity of NLP data points, offering a novel approach to understanding textual DA and explaining the superior performance of LLMs in this task.
4. In terms of dimensionality reduction, using principal component analysis with two components is optimal for capturing the majority of variance in augmented datasets. This approach balances information preservation with computational efficiency across various augmentation techniques, without significant loss of model performance compared to using three components.

1.1. Use of Topological and Geometric Techniques

Advanced text augmentation techniques, particularly those utilizing large language models, outperform traditional word replacement methods across various NLP tasks [7–10]. Despite the empirical success of generative approaches, a fundamental understanding of their effectiveness remains limited.

Word embeddings represent words as dense vectors in a high-dimensional space, where semantic relationships between words are encoded as geometric relationships. Therefore, examining the topological structure of word embeddings before and after augmentation could reveal how an augmentation technique alters the semantic space of the training data. Topological data analysis (TDA) provides a powerful framework for capturing the global structure of the word embedding space. TDA techniques, such as persistent homology, can reveal intrinsic shape characteristics and connectivity patterns that persist across different scales. This global perspective is important in identifying the overarching structures in the semantic space, such as clusters of related words or higher-dimensional “holes” that might represent semantic gaps. By comparing the topological features of the original and augmented embedding spaces, we can assess how augmentation techniques affect the overall organization of semantic concepts. Such information could help explain the effectiveness of the technique concerned.

Complementing this global view, computational geometric analyses, such as nearest neighbor analysis, convex hull computation, and Delaunay triangulation, offer insights into the local relationships between word vectors. We can examine how augmentation alters the fine-grained spatial distribution of word embeddings in the neighborhood around individual words. This could reveal whether augmented words are inserted into semantically appropriate regions, and therefore positively contribute to the training of the model.

The combination of global topological analysis and local geometric examination provides a comprehensive framework for understanding the effects of different augmentation techniques on the semantic space represented by word embeddings. In this paper, we showed that this dual perspective enables us to bridge the gap between the mathematical properties of the embedding space and the empirical effectiveness of augmentation methods in improving text classification performance.

1.2. Research Objectives

Our study aims to explore the underlying mechanisms and properties of the generated data [11,12] using a combination of geometric and topological analysis techniques. By comparing traditional word replacement method using word embedding with a generative method (GPT-J), we aim to conduct the following:

1. Apply topological data analysis to examine the structural properties of the augmented data spaces.
2. Utilize computational geometry techniques such as convex hull analysis and Delaunay triangulation to investigate the spatial distribution of augmented data points.
3. Explore the relationship between these geometric and topological properties and the effectiveness of the augmentation methods in improving classification performance.

By bridging the gap between the empirical success of advanced augmentation techniques and their underlying mathematical properties, we aim to develop a more robust theoretical foundation for text data augmentation. This understanding could provide a way to evaluate new augmentation methods, and guide the development of more effective and less resource-intensive augmentation strategies.

2. Review of Textual Data Augmentation Techniques

Data augmentation is an important technique used to address the problem of limited labelled training data in machine learning. Textual data augmentation methods typically make use of text replacement at various levels—word, sentence, and document. There are other advanced techniques that could operate at any level.

2.1. Word-Level Augmentation

Word-level augmentation techniques focus on manipulating individual words within a text to create new, semantically similar examples. One of the simplest and most widely used methods is synonym replacement [13,14]. This technique involves substituting words with their synonyms, often utilizing lexical databases such as WordNet [15] as a source of synonyms. Various studies have employed this approach for text augmentation, including the work by Marivate and Sefara [16]. While straightforward, synonym replacement can sometimes lead to contextually inappropriate substitutions, as the chosen synonyms may not always fit the specific context of the original sentence. Word embedding models, such as Word2Vec [17] and GloVe [18], have been leveraged for more nuanced word-level augmentation. These models represent words as dense vectors in a continuous space, where semantically similar words are close to each other. Augmentation techniques using word embeddings typically involve replacing words with their nearest neighbors in the embedding space [16]. More recent approaches have utilized contextual word embeddings from models like BERT [5] for word-level augmentation. These models provide context-aware word representations, allowing for more accurate and context-appropriate word substitutions [19]. Random noise insertion is another technique used at the word level. This method involves randomly inserting, deleting, or swapping characters or words in the text [14]. While simple, it can help improve model robustness to spelling errors and minor text variations. Other word-level techniques include random word deletion, where words are randomly removed from the text, and random word insertion, where new words are added to the text based on the surrounding context [14]. These methods can help models learn to handle missing information and different sentence structures. However, word-level augmentation techniques, while simple to implement, often face challenges in preserving

semantic coherence and grammatical correctness [14]. Synonym replacement can lead to contextually inappropriate substitutions, potentially altering the intended meaning of the text [13]. Word embedding-based augmentation methods, while effective for generating diverse examples, may introduce rare words that can make the augmented text seem unnatural [12]. Random noise insertion techniques can create unrealistic or ungrammatical sentences, potentially introducing noise into the training data rather than meaningful variations [14]. Additionally, these methods struggle to capture higher-level semantic structures and relationships between words in a sentence [19].

2.2. Sentence-Level Augmentation

Sentence-level augmentation techniques aim to generate new sentences while preserving the original semantic meaning and label. Back-translation is a popular method in this category [20]. It involves translating a sentence to an intermediate language and then back to the original language, often resulting in paraphrases of the original sentence. While effective, this method can be computationally expensive and may introduce errors or alter the original meaning. Various approaches have been proposed for generating paraphrases of sentences. These include rule-based methods [21], statistical machine translation techniques [22], and more recently, neural network-based approaches [23]. Advanced language models like GPT [6] have shown promising results in generating high-quality paraphrases. Sentence mixing is another technique where new sentences are created by combining parts of existing sentences [1]. This method can help create diverse sentence structures while maintaining semantic coherence. Syntactic tree transformation is a more sophisticated approach that involves manipulating the syntactic structure of sentences to create new, grammatically correct variations [24]. This method can help models learn to handle different syntactic structures while preserving the original meaning. Despite their potential, sentence-level augmentation techniques face several challenges. Back-translation, while effective, can be computationally expensive and may introduce errors or subtle changes in meaning, especially for languages with significant structural differences [25]. Paraphrasing methods often struggle to generate diverse sentence structures while maintaining the exact original meaning [23]. Sentence mixing and syntactic tree transformation can sometimes produce unnatural or semantically inconsistent sentences [24]. Moreover, these techniques may not always preserve the nuanced sentiment or style of the original text, which can be crucial for certain NLP tasks like sentiment analysis or style transfer [26].

2.3. Document-Level Augmentation

Document-level augmentation techniques aim to generate entirely new documents while maintaining the overall theme and label of the original text. One approach to document-level augmentation involves extracting key information from a document and then using abstractive summarization techniques to generate a new document [1]. This method can help create diverse yet topically consistent augmented data. Document expansion is a technique where additional relevant information is added to a document based on external knowledge sources or related documents in the corpus [27]. This can help enrich the content of documents and provide more context for classification tasks. However, document-level augmentation techniques face significant challenges. Those ones that make use of large language models face difficulties in maintaining consistent factual accuracy and coherence throughout long-form text [28]. Generated documents may contain invented factor inconsistencies that are difficult to detect automatically [29]. Topic modeling-based approaches can effectively capture high-level semantic structures, but they may produce text that is thematically related yet lacks the specific context or intricate details required for certain tasks [30]. Document expansion techniques may introduce irrelevant information, potentially diluting the key features necessary for accurate classification [27]. Furthermore, these methods often require significant computational resources and may be impractical in resource-constrained environments [31].

2.4. Recent Advanced Techniques

Recent advancements in NLP have led to more sophisticated augmentation techniques that often combine multiple levels of augmentation. Conditional text generation models, such as CTRL [32], allow for fine-grained control over various aspects of the generated text, including style, content, and sentiment. This enables the creation of highly tailored augmented data. Large language models like GPT-3 [31] and GPT-J [33] have shown promising results in generating high-quality augmented data for various NLP tasks [7–10]. These models can generate diverse, coherent, and label-consistent augmented data at various levels (word, sentence, and document). They can also be fine tuned or prompted to generate augmented data that closely align with the original dataset's distribution [28]. Adversarial training methods have also been applied to data augmentation, where a generator model creates challenging examples to improve the robustness of the classifier [34]. This approach can help models learn to handle more diverse and potentially adversarial inputs.

While powerful, these advanced augmentation techniques come with their own set of challenges. Conditional generation models and large language models like GPT-J can be computationally expensive and require significant amounts of data to fine tune effectively [33]. For instance, fine tuning LLMs on a specific task can require hundreds of gigabytes of GPU memory and several days of training time on high-performance hardware, making it impractical for many researchers and smaller organizations [31,33]. Furthermore, data requirements for effective fine tuning can be substantial, often necessitating tens of thousands of task-specific examples to achieve optimal performance [28]. Biases that are present may be amplified, leading to potentially unfair or biased augmentations [35]. Adversarial training methods can be difficult to balance, potentially creating examples that are too challenging or unrealistic [34]. Furthermore, the black-box nature of many of these advanced models makes it difficult to interpret or control the augmentation process precisely [36].

3. Experimental Design

We use text classification as the NLP task to study text data augmentation. In this section, details of the classifier model, datasets, and augmentation techniques methods used in our experiments are described. Details of our experiment code can be found here <https://colab.research.google.com/drive/1qESNNvnPc7H-1W1j7LLFdfpD1v5dvviN?usp=sharing> (accessed on 17 September 2024).

3.1. Data Augmentation Techniques

Two different augmentation techniques will be compared: word replacement based on word embeddings (Word2Vec and GloVe) and generation using a large language model. These techniques represent two distinct approaches to text data augmentation, allowing us to compare traditional methods with more recent advancements in natural language processing.

3.1.1. Word Replacement

The first technique is word replacement based on word embeddings, specifically using Word2Vec [17] and GloVe [18]. This method has been widely used in various NLP tasks due to its simplicity and effectiveness [13]. The algorithm we are using is the popular word replacement technique implemented in the Gensim library [37].

Word replacement using word embeddings operates on the principle that words with similar meanings tend to have similar vector representations in the embedding space. By replacing words in the original text with their nearest neighbors in the embedding space, this technique aims to create semantically similar but lexically diverse augmented samples [38]. This approach has been shown to be somewhat effective in text classification tasks, particularly when dealing with limited training data [16]. However, it can sometimes lead to semantic inconsistencies or grammatical errors, especially when replacing words without considering the broader context [14].

A pseudo-code of the algorithm is shown in Algorithm 1.

Algorithm 1 Word replacement algorithm

Require: *input_sentence*, *word_embedding_model*, *num_similar_words*
Ensure: *augmented_sentences*

```

tokens ← tokenize(input_sentence)
selected_words ← select_words_to_replace(tokens)
augmented_sentences ← []
for all word in selected_words do
  similar_words ← word_embedding_model.most_similar(positive = [word], topn =
num_similar_words)
  for all (similar_word, similarity_score) in similar_words do
    augmented_sentence ← replace(input_sentence, word, similar_word)
    augmented_sentences.append(augmented_sentence)
  end for
end for
return augmented_sentences

```

3.1.2. GPT-J

The second technique employs a GPT-based approach to generate entirely new sentences or paragraphs that are contextually relevant to the original text. Specifically, we use GPT-J, a large language model developed by EleutherAI [39]. It is an autoregressive language model based on the GPT-3 architecture [31], but with 6 billion parameters, making it more accessible for research purposes while still maintaining impressive language generation capabilities. GPT-J, like other models in the GPT family, has been pre-trained on a diverse corpus of internet text, allowing it to generate coherent and contextually appropriate text across a wide range of domains and styles [40]. It has also been proven to be the best-performing publicly available Transformer LM in terms of zero-shot performance [39]. For our data augmentation task, we use GPT-J in a few-shot learning setting [31]. We provide the model with a few examples of text samples and their corresponding labels, followed by a prompt for generating new samples. This approach leverages the model's ability to understand the pattern and context from the given examples and generate new, semantically similar text samples [28].

For our implementation, we utilized the following resources:

- **Model:** a RAM-reduced GPT-J-6B model, which is publicly available through the Hugging Face model hub [41].
- **Framework:** the model is implemented using the transformer library (version 4.18.0) from Hugging Face, which provides a high-level API for working with pre-trained language models.
- **Hardware:** Single NVIDIA A100 GPU with 40 GB of VRAM.

A template of the prompt format used to obtain augmentation data is shown below:

```

Each item in the following contains a
text sample and the respective label:
Label: <Original_Training_Data_Text_Sample>
Label: <Original_Training_Data_Text_Sample>
Label:

```

This prompt-based approach allows GPT-J to generate new text samples that are likely to preserve the semantic content and label of the original samples while introducing lexical and syntactic diversity [42]. Unlike the word replacement method, this technique has the potential to generate entirely new sentences or even paragraphs, potentially offering greater diversity in the augmented data [43]. By comparing these two distinct augmentation techniques, we aim to investigate how different approaches to preserving semantic information while introducing diversity affect the geometric and topological properties of the resulting text embeddings, through the performance of text classification models.

3.2. Classification Model and Dataset

The text classification model we used for the experiments in this paper is the convolutional neural network (CNN). CNNs have been used extensively for text classification tasks, particularly when combined with word embeddings [13]. Moreover, CNNs are computationally efficient and can handle variable-length input, which is advantageous for processing diverse text data [44]. The CNN architecture used in our study is based on the model in [45]. The model parameters are listed in Table 1.

Table 1. Architecture of the CNN model used for the experiments.

Configuration	Values
Feature maps per region size	2
Univariate vectors per region size	6
Concatenated vectors per region size	Single feature vector
Sentence matrix size	7×5
Region sizes	(2, 3, 4)
Filters per region size	2
Total filters	6
Convolution	Yes
Activation function	ReLU
1-Max pooling	Yes
Softmax function	Yes
Regularization	Yes

Two widely used benchmark datasets, SST2 [46] and TREC [47], are used. SST2 consists of movie reviews with binary sentiment labels—positive or negative. The TREC dataset contains questions that are classified into six categories. One classification model is trained on the original database to obtain the baseline results. Training data are selected as follows:

1. With each of these two datasets, one of the class labels is randomly chosen.
2. For this chosen label, five training samples are randomly selected.
3. For each of the remaining labels, 20 training samples are randomly chosen.

This setup simulates scenarios where annotated data are scarce and there is an imbalance in training samples among the classes.

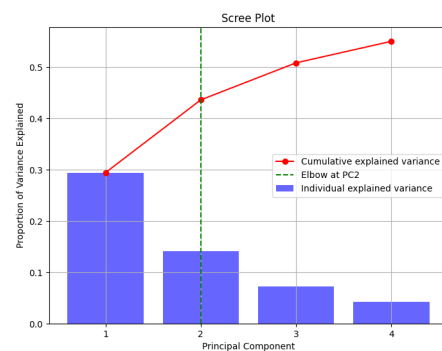
Then augmentation data are produced to augment the class with less training samples in order to maintain class balance. This is achieved by training on the 5 original training samples to generate 15 new augmentation samples. Lastly, another CNN model with the same architecture will be trained on the same set of training samples with the new augmentation additions. Classification performances are obtained using the default test sets provided with each dataset. That is, 1821 and 500 test samples, respectively, for SST2 and TREC.

4. Dimensionality Reduction

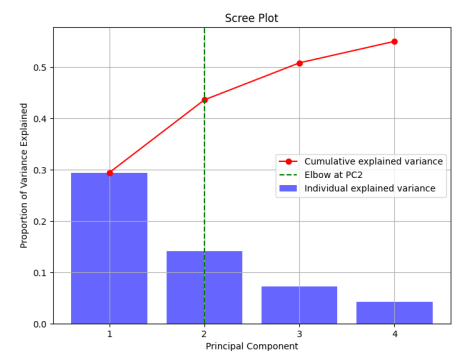
Previous attempts to apply convex hull analysis to text data encountered challenges due to the computational complexity [48]. This is because word embeddings are extremely high-dimensional vectors, with typical dimensions of 300. One way to overcome this problem is to perform dimensionality reduction. This can be achieved by using principal component analysis (PCA), which ensures that the most important patterns and variances in the data are captured. Studies have shown that PCA preserves the essential semantic relationships between words [49], especially for text classification [50].

Principal Component Selection Analysis

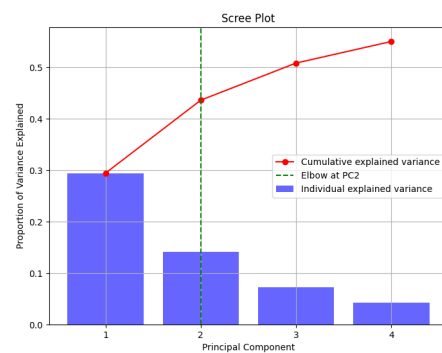
When applying PCA for dimensionality reduction in word embeddings, choosing the number of principal components (k) is crucial. This decision impacts the balance between information preservation and computational efficiency. While researchers often select k arbitrarily [51,52], the optimal value theoretically depends on the data's intrinsic dimensionality [53]. Scree plots, which visualize the variance explained by each principal component, can guide this decision [54]. The 'elbow point' in these plots indicates where additional components offer diminishing returns [55]. Our scree plots for SST2 and TREC datasets (Figure 1) show consistent patterns across augmentation techniques. The first two principal components (PC1 and PC2) account for 42–44% of total variance, with PC1 alone explaining about 30%. A clear elbow point at PC2 is evident in all plots. Based on this analysis, we justify using $\text{PCA} = 2$ in our experiments. This choice strikes a balance between dimensionality reduction and information retention. While $\text{PCA} = 3$ or $\text{PCA} = 4$ would capture more variance, the gains are marginal compared to the increased computational cost. PC3 and PC4 individually contribute less than 10% to the explained variance, whereas PC1 and PC2 together explain 42–44%. The elbow point at PC2 marks a clear transition where the rate of variance explained by each additional component begins to level off. This diminishing return in variance explanation beyond PC2 supports our decision to set $\text{PCA} = 2$. Including more components would incrementally increase the total variance explained, but at the cost of higher dimensionality and increased computational complexity, potentially compromising the efficiency gains of dimensionality reduction. This approach significantly reduces computational complexity while preserving key features of the original embeddings. The consistency across datasets and augmentation methods suggests that $\text{PCA} = 2$ offers a robust, efficient dimensionality reduction strategy for various NLP tasks, balancing information preservation with computational efficiency.



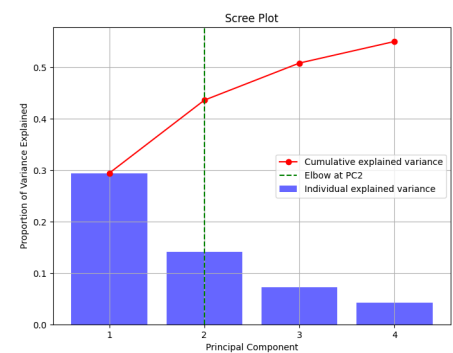
(a) SST2 Word2Vec.



(b) SST2 GloVe.



(c) SST2 GPT-J.



(d) TREC Word2Vec.

Figure 1. Cont.

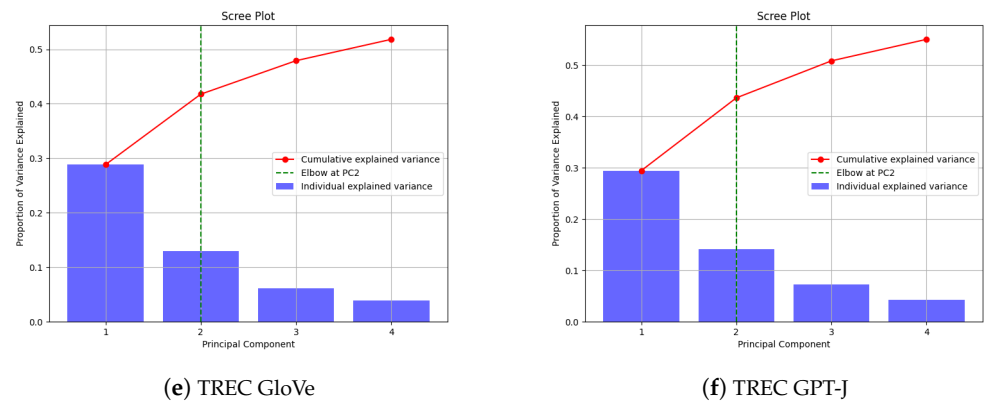


Figure 1. Scree plots of post-augmentation PCA components of SST2 and TREC datasets.

5. Techniques in Analyzing Geometric Properties

5.1. Topological Data Analysis

TDA is a field that has emerged from applied (algebraic) topology and computational geometry. It is motivated by the idea that topology and geometry can provide more insights and information about the structure of data; this is performed by computing abstract “shapes” of datasets [56]. There has been recent research into the use of TDA for NLP. In fact, there have also been some uses of topology in explaining various LLMs and deep learning phenomena [57,58]. Thus, we have chosen to use TDA in looking at the underlying structure of DA techniques.

Naturally, with the growing amount of complex data used in the field of ML, TDA has provided different perspectives to understand the structure and shapes of data. For instance, in the realm of unsupervised learning, the persistence diagram, a summary of topological features across scales, has been effectively utilized to enhance clustering algorithms by providing a metric that captures the shape of data clusters beyond mere proximity or density considerations [59]. Similarly, in supervised learning, topological features have been employed to enrich the feature space, improving the accuracy of classifiers when dealing with complex datasets where the relationship between features and labels is subtle and intertwined with the geometric structure of the data space [60].

At the core of TDA is the construction of a simplicial complex, which encapsulates the relationships between data points across various scales. This is achieved by defining a notion of proximity between data points and connecting them to form higher-dimensional simplices, thereby capturing the topological features of the data. The lifespan of these simplices is analyzed as the scale parameter changes [61].

Given a set of vertices V such that

$$V = \{1, \dots, |V|\}, \tag{1}$$

a simplex σ is defined as a subset of vertices $\sigma \subseteq V$. A simplicial complex K on V is then a collection of simplices

$$\{\sigma\}, \quad \sigma \subseteq V, \tag{2}$$

satisfying the condition that

$$\tau \subseteq \sigma \in K \Rightarrow \tau \in K. \tag{3}$$

The dimension of σ , denoted n , is given by

$$n = |\sigma| - 1, \tag{4}$$

representing the number of elements in σ minus one. A filtration of a simplicial complex is defined through a function

$$f : K \rightarrow \mathbb{R}, \quad (5)$$

which satisfies

$$f(\tau) \leq f(\sigma) \quad \text{whenever} \quad \tau \subseteq \sigma, \quad (6)$$

allowing for the visualization of the formation and dissolution of loops and voids within the data space [62]. These phenomena are typically represented in a persistence diagram, elucidating critical structures—such as clusters, holes, tunnels, and voids—that persist across varying scales [63].

We shall employ TDA to unveil any latent topological structures that may define the relationship between the original training data and the augmentation data. The emphasis is particularly placed on two primary homology groups: H_0 and H_1 . H_0 elucidates the connected components within the data, indicative of clusters or isolated data points, while H_1 reveals the presence of more complex topological features such as loops or voids. While TDA provides valuable insights into the topological features of our data, we complement this approach with computational geometry techniques to gain a more comprehensive understanding of the spatial relationships in our dataset.

5.2. Computational Geometry

Computational geometry provides powerful tools for analyzing the spatial relationships and structures within datasets [64]. In the context of textual data augmentation, these techniques can offer valuable insights into the distribution and characteristics of the augmented data points in the high-dimensional embedding space [65]. By applying computational geometry concepts to our word embeddings, we can visualize and quantify how different augmentation methods affect the spatial arrangement of data points [66]. This analysis can help us understand why certain augmentation techniques, particularly those using LLMs, perform better than others [67]. In this study, we focus on two key concepts from computational geometry: convex hulls and Delaunay triangulation.

A convex hull is the smallest convex set that encloses a given set of points [64]. It can be viewed as the smallest possible shape that encompasses all the data points without any concave regions or indentations. Convex hull has been applied in machine learning, including image classification [68]. Identifying the convex hull of a set of data can provide insights into its overall geometric structure and boundaries. In our study, we apply this concept to analyze the spatial distribution of original and augmented data points, helping us understand how different augmentation techniques affect the geometry of the dataset.

Delaunay triangulation connects a set of points by forming triangles in such a way that no point is inside the circumcircle of any triangle [64]. In our study, we directly applied Delaunay triangulation in the word embedding space of the data points. This technique allows us to quantify the connectivity and relationships between data points, providing additional insights into the structure of our augmented datasets.

These techniques, when used in conjunction with TDA, provide a multi-faceted view of the spatial and topological characteristics of the augmentation data.

6. Results and Analyses

6.1. Classification Results

Text classification is performed in the setting described in Section 3, both with and without text augmentation. Table 2 shows the classification accuracies. The baseline classification accuracies using CNN are low. This is expected as the amount of training data is very small. After including the augmentation data, there is a significant decrease in performance for the TREC dataset, where the accuracy dropped by 21.2%. There is also a slight decrease (−0.8%) in accuracy after augmentation for SST2. This shows that augmentation using word replacement has an adverse effect on the trained models. The situation is similar when GloVe embedding is used.

Table 2. Classification accuracies with and without augmentation.

Dataset	Model	Baseline	Augmented	
			Algorithm 1	GPT-J
TREC	Word2Vec	51.0%	−21.2%	+5.0%
	GloVe	32.8%	−18.4%	+3.2%
	GPT-J	74.0%	–	+6.0%
SST2	Word2Vec	51.2%	−0.8%	+11.4%
	GloVe	50.2%	−5.1%	+12.6%
	GPT-J	62.0%	–	+13.2%

The GPT-J model, on the other hand, gives much higher baseline classification accuracies without augmentation. With augmentation, accuracies improved by 6.0% and 13.2% for TREC and SST2, respectively. Such results are expected as GPT-J has undergone extensive pre-training and is able to generalize better.

Interestingly, the augmented data generated by GPT-J can enhance the performances of the CNN model. The improvements for TREC are 5% and 3.2% with Word2Vec and GloVe, respectively. The improvement is even more pronounced for SST2—11.4% and 12.6%. The reasons why these augmentation data are effective while word replacement using Word2Vec and GloVe are not are provided by our topological and geometric analyses.

6.2. TDA of Embedding Vectors

TDA is performed on the embedding vectors of the training data selected as described in Section 3. Their H_0 and H_1 homologies are shown in Figures 2 and 3 for SST2 and TREC, respectively. A point (x, y) on these diagrams represents a topological feature that comes into existence at scale x and its existence lasts until scale y , when it becomes indistinguishable from other clusters.

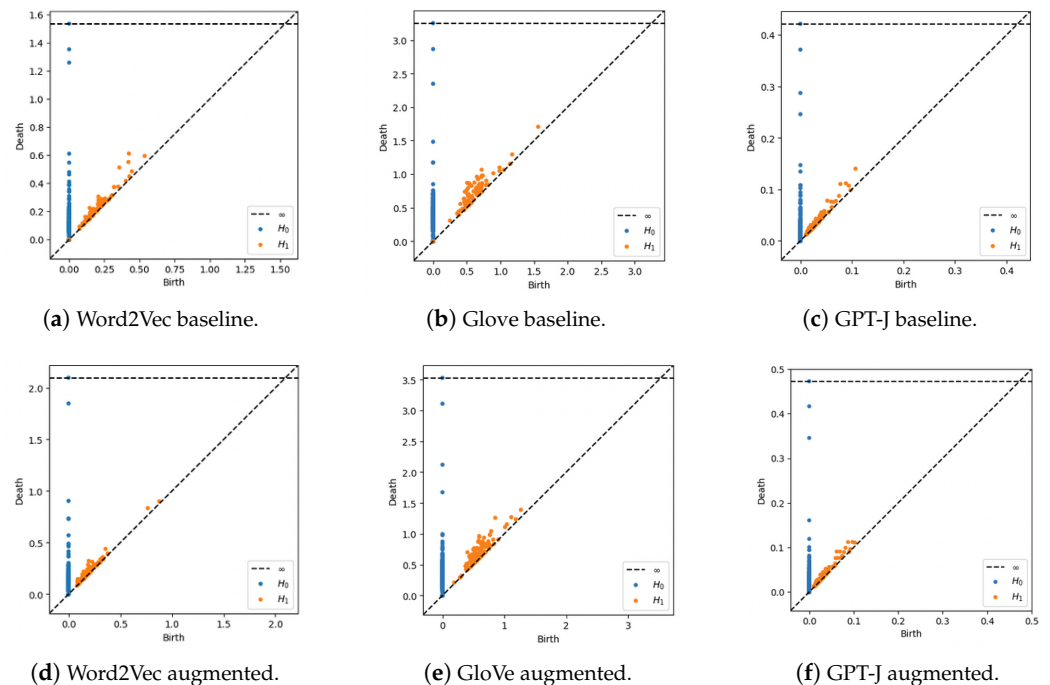


Figure 2. Persistence diagrams of SST2: (a) base model using Word2Vec embeddings; (b) base model using GloVe embeddings; (c) base model using GPT-J embeddings; (d) augmented data using Word2Vec embeddings; (e) augmented data using GloVe embeddings; and (f) augmented data visualization using GPT-J embeddings.

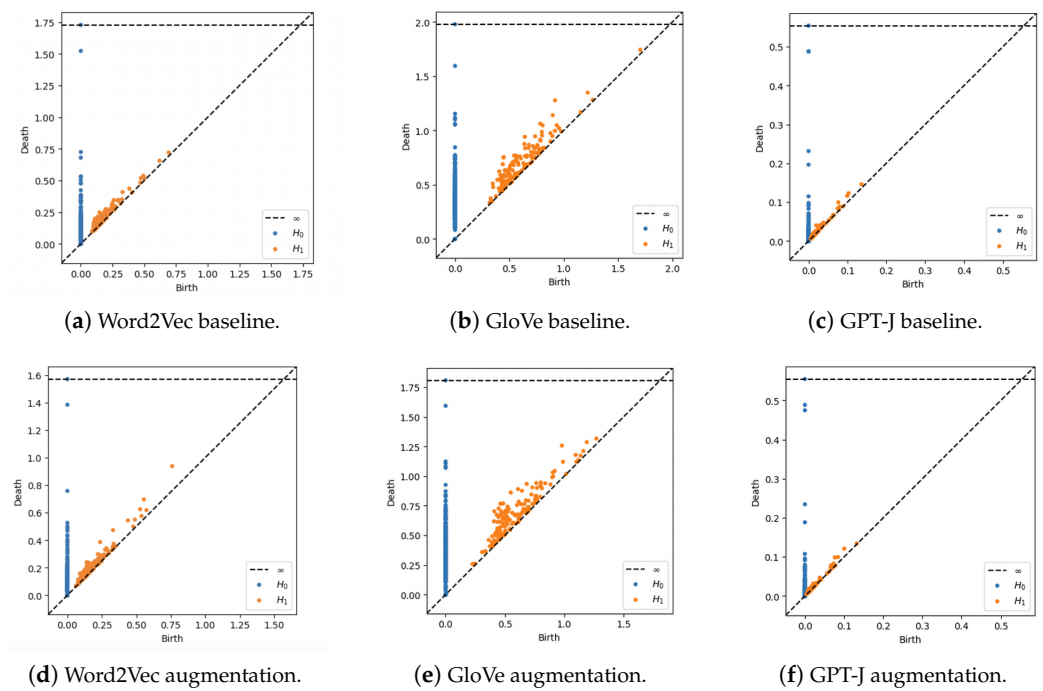


Figure 3. Persistence diagrams of TREC: (a) base model using Word2Vec embeddings; (b) base model using GloVe embeddings; (c) base model using GPT-J embeddings; (d) augmented data using Word2Vec embeddings. (e) augmented data using GloVe embeddings; and (f) augmented data using GPT-J embeddings.

The persistence diagrams of the SST-2 dataset both before and after augmentation are shown in Figure 2. In Figure 2d,e, the H_1 points (in orange) show increased dispersion. This suggests that Word2Vec and GloVe augmentation may reduce model performance by disrupting the coherence of the topological structure. This is reflected in Table 2, where both accuracies drop. For GPT-J, the clusters for both H_0 and H_1 points post-augmentation are tighter. This indicates that GPT-J is more resilient to augmentation and better preserves label coherence compared to Word2Vec and GloVe. The accuracy improvement of 13.2% for the GPT-J-augmented SST2 data shown in Table 2 underscores the effectiveness of GPT-J in maintaining topological consistency, leading to better model performance.

There are six class labels for the TREC data. Therefore, in the persistence diagrams for GloVe in Figure 3, one would expect to see six H_0 (in blue) dots representing six distinct clusters points as the “Death” value increases. While the six clusters for the augmented GPT-J data in Figure 3f is still barely observable, the corresponding ones for Word2Vec and GloVe (Figures 3d,e) only show three and four clusters, respectively. This tells us that the augmented data from GPT-J are able to preserve the six output labels. However, those from GloVe and Word2Vec could not.

Figure 4a shows the H_1 values against classification accuracies for SST2. For this dataset, there is a correlation between connections between data points and improved augmentation results. This observation is consistent with the structure of word embeddings, where embeddings of higher dimensionality generally exhibit better performance. However, for the TREC dataset in Figure 4b, such a correlation does not hold for non-GPT-J augmented data. A closer examination of Figures 2 and 3 reveals that there is a distinct spatial boundary within the word embedding spaces. For example, the augmentations resulting in decreased performance; specifically, Figures 2d and 3b,d, all exhibit H_1 data points that have become more dispersed post-augmentation. Conversely, augmentations showing improvement from the baseline, namely Figures 2e,f and 3f demonstrate that the H_1 data points remain closely clustered around their original baseline positions. This means that the augmented words are in close proximity to the baseline words. Thus, there appears to be a spatial

boundary limit to the connections beyond which augmented data points begin to lose label coherence.

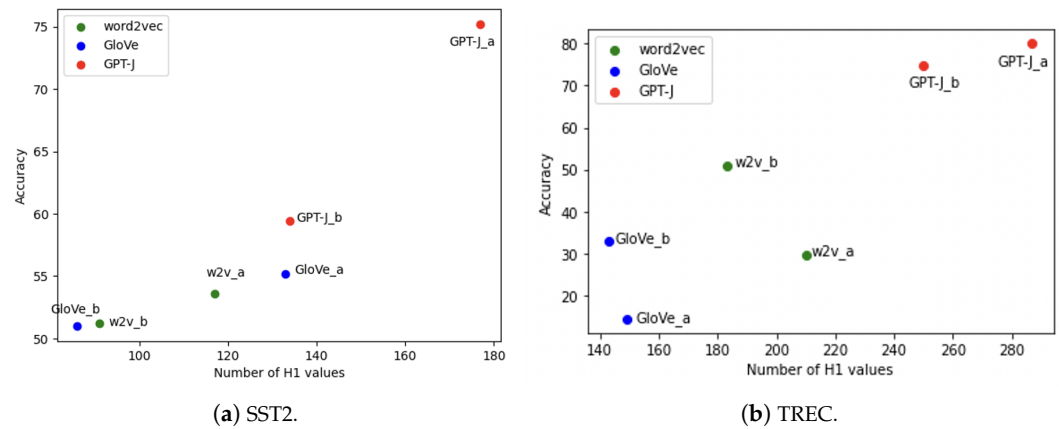


Figure 4. H1 values for SST2 and TREC datasets.

6.3. Bottleneck Distance Analysis

To further quantify the spatial relationships between the baseline and augmented data, we perform bottleneck distance analysis, a key metric in topological data analysis [69,70]. This metric provides a measure of the similarity between two persistence diagrams, offering insights into how the topological features of the data change after augmentation [71].

Bottleneck distance is defined as the smallest maximum distance required to match the points in one persistence diagram to the points in another. Formally, for two persistence diagrams X and Y , it is expressed as follows:

$$d_B(X, Y) = \inf_{\gamma} \sup_{x \in X} |x - \gamma(x)|_{\infty} \quad (7)$$

where γ ranges over all bijections from X to Y , and $|\cdot|_{\infty}$ denotes the L-infinity norm. Intuitively, it represents the smallest maximum distance that the points in one diagram need to be moved to transform them into the other diagram. A smaller bottleneck distance indicates greater similarity between the topological features of two datasets. In our context, a smaller distance between the baseline and augmented persistence diagrams suggests that the augmentation method preserves the topological structure of the original data more closely.

Figure 5 shows the persistence diagrams and the corresponding bottleneck distances. Each subfigure represents a specific combination of dataset and embedding method (Word2Vec, GloVe, and GPT-J). These diagrams show that GPT-J consistently produces augmented data with the smallest topological deviation from the baseline, as evidenced by the lowest bottleneck distances. This correlates with its superior performance improvements across both datasets. Word2Vec and GloVe show larger bottleneck distances, indicating more significant topological changes in the augmented data. This corresponds to their more variable performance impacts, sometimes leading to improvements (as in SST2) and sometimes to degradation (as in TREC for GloVe). The relationship between bottleneck distance and performance is not strictly linear. While GPT-J's small distances consistently correspond to improvements, GloVe's larger distances lead to different outcomes in TREC and SST2. The persistence diagrams visually confirm these observations, with GPT-J's diagrams showing the closest alignment between baseline and augmented points, especially for the H_1 features (orange points).

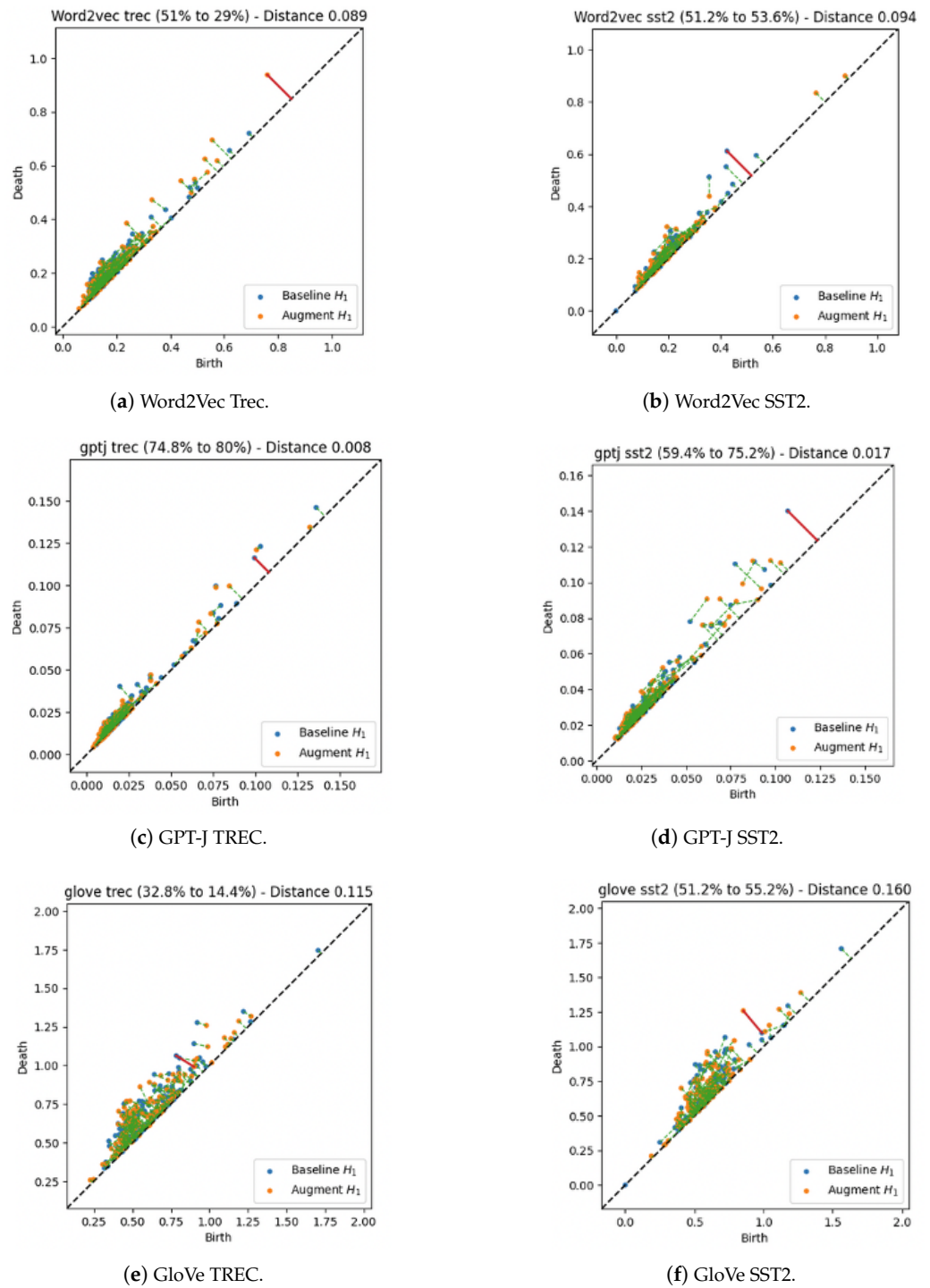


Figure 5. Bottleneck distance analysis for different models and datasets. Red lines indicate the largest bottleneck distance between pairs of matched points, suggesting more significant topological changes due to augmentation. Green indicates that the distance between matched points is smaller than the bottleneck.

7. Geometric Analyses

Our TDA and bottleneck distances have provided valuable information on the structural changes induced by different augmentation methods. They indicate the existence of a spatial boundary that augmented words must respect so that their meanings are retained. In order to more precisely define these spatial boundaries of effective augmen-

tation, we now turn to classic geometric analysis techniques—convex hull analysis and Delaunay triangulation.

7.1. Convex Hull Analysis

We compute the convex hulls of the two-dimensional PCA of the embeddings of the augmented and original training datasets. In [48], the convex hulls of NLP data points were computed. However, the feasibility of their approach was hindered by the time complexity of the computation. Here, we utilize the QuickHull (Qhull) algorithm that can efficiently handle complex geometries within datasets [72].

Figure 6 plots the computed convex hulls of the Word2Vec versus GPT-J augmentation. The corresponding plots for GloVe embedding are found in Figure 7. These figures show that the augmented data points for Word2Vec and GloVe both extend beyond the boundaries of the original training data for both TREC and SST2. This is the result of the use of cosine similarity to select replacement words. Some of the augmented data points are therefore outside the boundary of the original training data.

On the other hand, GPT-J produces augmented words that tend to be within the the convex hull of the baseline training words, as shown in Figures 6b,d and 7b,d. This indicates that more effective augmentation data should remain within the convex hull of the original training data in order to preserve their meaning and label coherence.

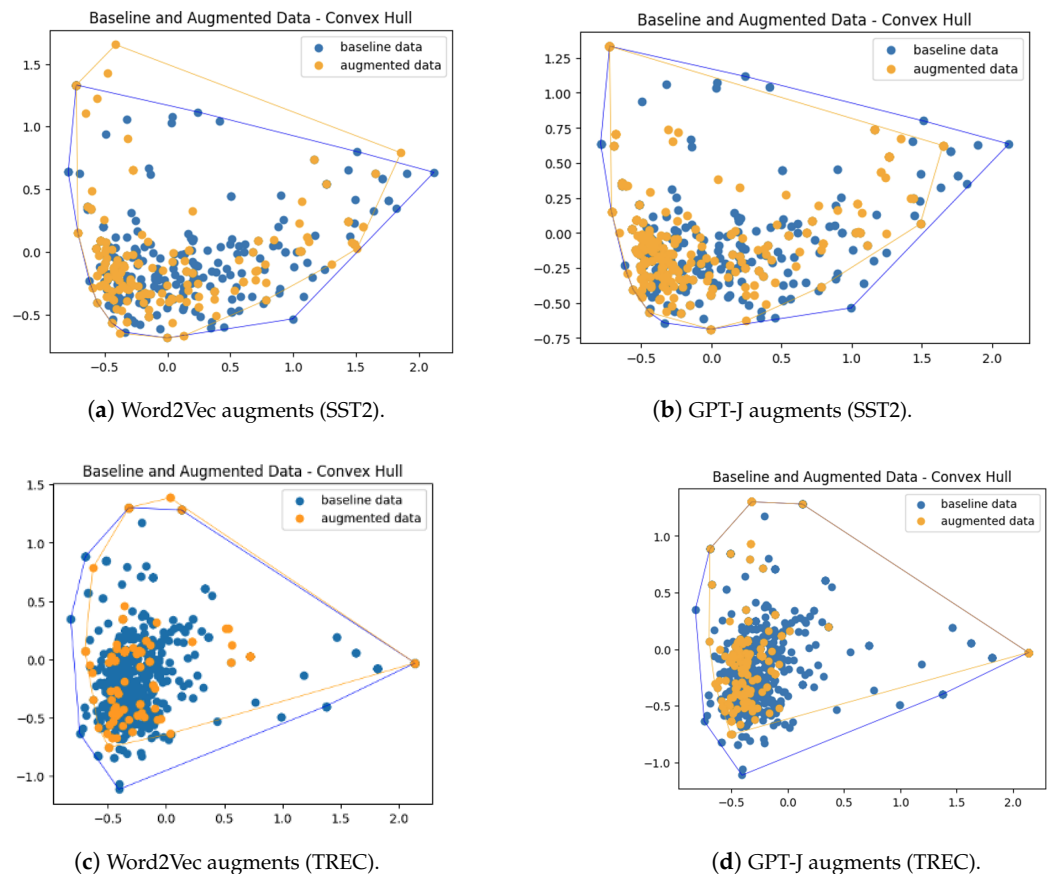


Figure 6. Comparison of augmented word shapes used for Word2Vec CNN embedding. The lines indicate the convex hull.

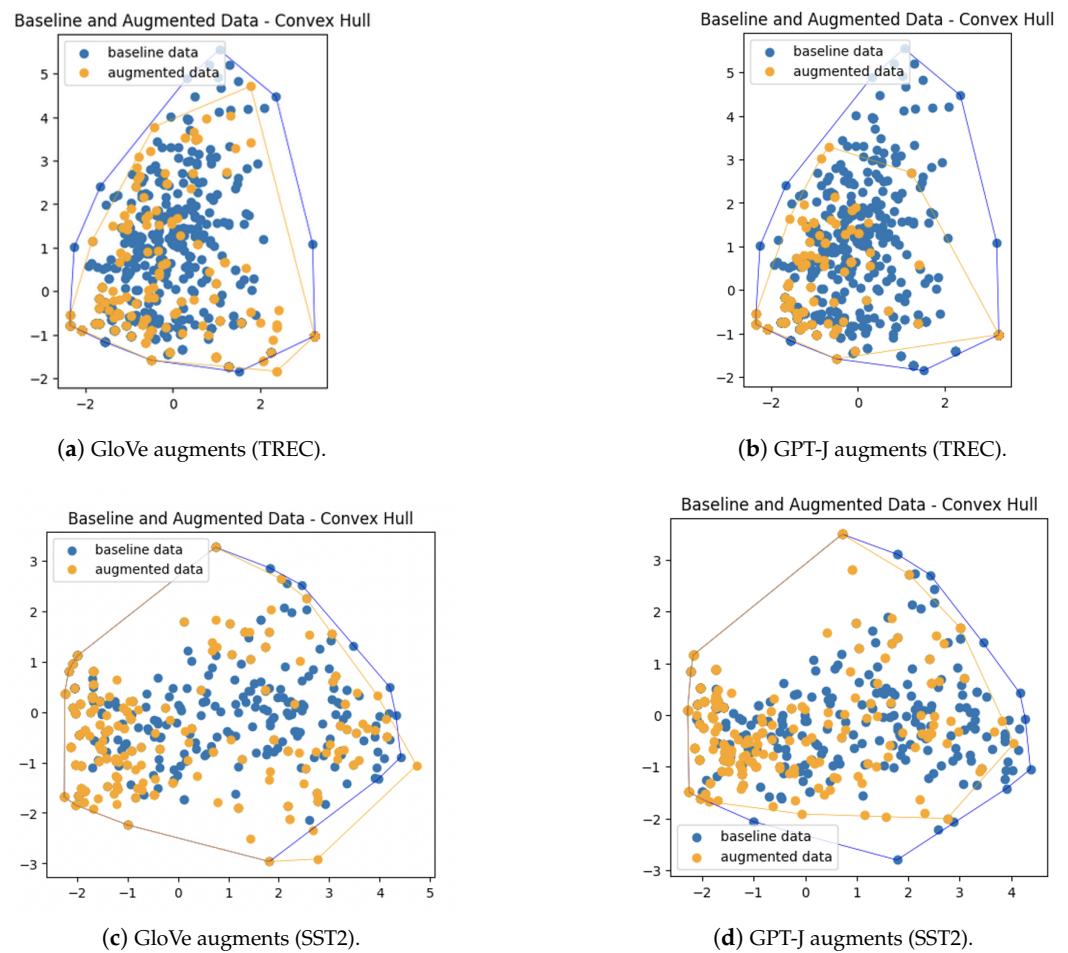


Figure 7. Comparison of augmented word shapes used for GloVe embedding. The lines indicate the convex hull.

7.2. Delaunay Triangulation Analysis

The above observation, along with Figure 4, raises the question of whether more augmentation data within the convex hull will lead to more accurate classifiers. This question could be answered by employing Delaunay triangulation. Delaunay triangulation connects a set of points by forming triangles such that no point is inside the circumcircle of any triangle [73]. This technique provides insights into the connectivity and distribution of points within the convex hull.

We perform Delaunay triangulation on the word embedding space of the augmented data points. Figure 8 shows the triangulation of augmented data points for both TREC and SST2. Figure 9 plots the number of edges of the triangulation versus the improvement in classification accuracy after augmentation. A higher number of edges represents increased connectivity within the convex hull. It shows a positive correlation between the number of edges and classification accuracy improvement. This relationship suggests that effective augmentation not only respects the boundaries of the original data but also increases the density of connections within that space.

These findings complement our convex hull analysis by showing that it is not just about staying within the boundaries of the original data, but also about how the augmented points are distributed and connected within those boundaries. Effective augmentation appears to create a denser, more interconnected representation of the data space while respecting its original geometric structure.

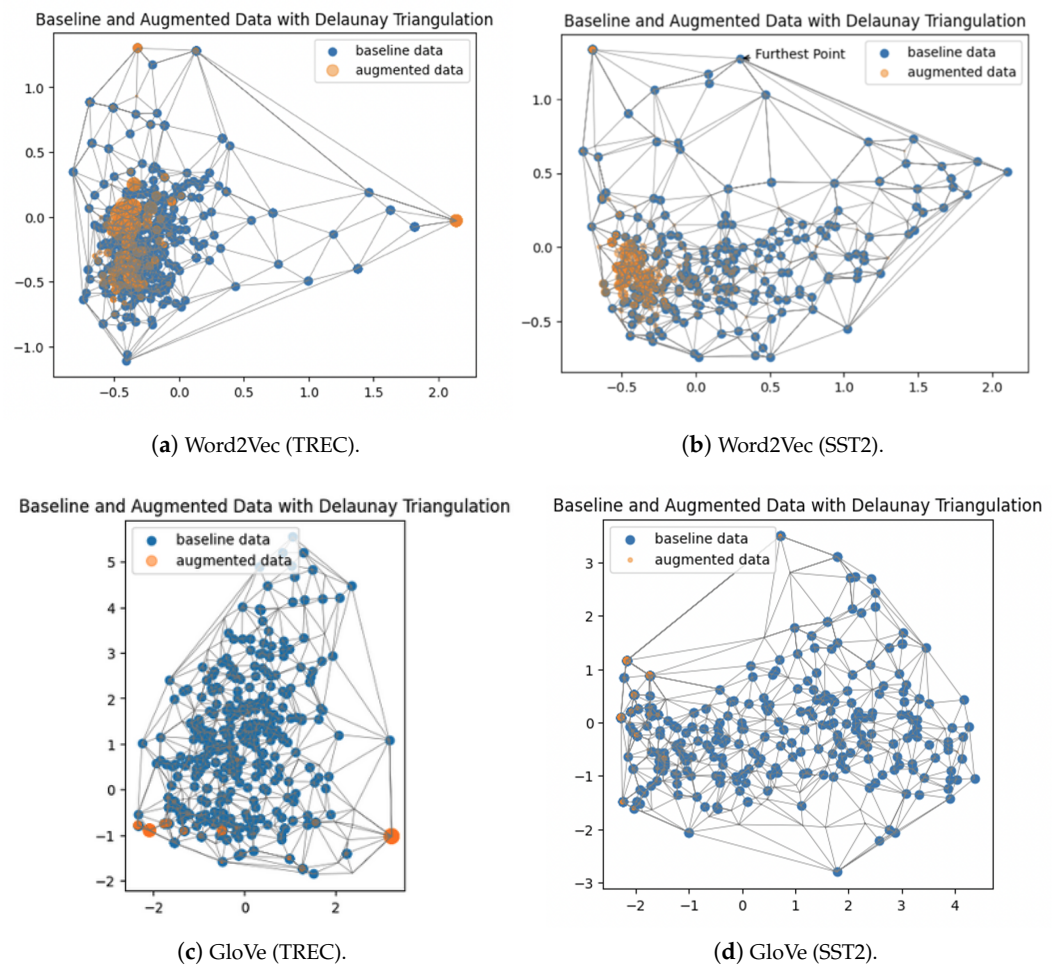


Figure 8. DT visualization of word embeddings' data points with GPT-J augmented data.

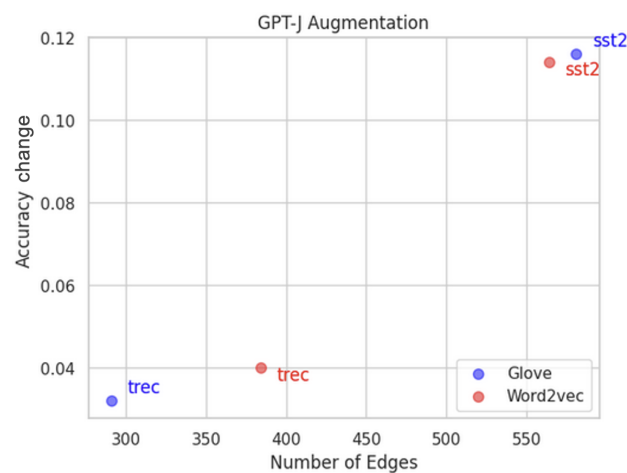


Figure 9. Relation between number of edges in the triangulation and the classifier accuracy.

This geometric perspective offers a new way to evaluate augmentation techniques. By combining convex hull analysis with Delaunay triangulation, we can assess both the global boundaries and the internal structure of augmented datasets. This approach provides an understanding of how augmentation techniques alter the geometry of the data space and how these alterations relate to improved classification performance.

7.3. Inspecting Generated Samples

Table 3 below presents examples of text generated by GPT-J, Word2Vec, and GloVe for both the SST2 and TREC datasets. These samples provide concrete illustrations of the qualitative differences between these augmentation techniques, which align with our geometric and topological analyses.

Table 3. Generated text using different techniques.

Dataset (Label)	Technique	Generated Text
SST2 (0)	GPT-J	the only pleasure this film has to offer lies in the first twenty minutes when the protagonist is a normal guy
	Word2Vec	it is a visual rorschach test and im should have failed
	GloVe	this a visual barcode test also think can still failed
TREC (5)	GPT-J	What is the name of the river which carries the water from a large lake to the Atlantic Ocean?
	Word2Vec	what river in scots is said to hold one or more zombies?
	GloVe	how lakes this scotland has adding could give another same more beast why

7.3.1. GPT-J-Generated Text

As shown from the Table, the samples generated by GPT-J demonstrate high coherence, grammatical correctness, and semantic relevance to the original labels. For SST2, the generated text maintains a negative sentiment while presenting a complete, logical sentence. For TREC, GPT-J produces a well-formed, semantically appropriate question that fits the expected label. These observations align with our earlier findings from the persistence diagrams in Figures 2 and 3, where GPT-J maintained tighter clusters in both H0 and H1 homologies, indicating preservation of the topological structure. The coherence of these samples also corresponds to our convex hull analysis (Figures 6 and 7), which showed GPT-J augmentations remaining within the boundaries of the original data space.

7.3.2. Word2Vec-Generated Text

The Word2Vec samples show a decline in coherence and semantic relevance compared to GPT-J. While they maintain some thematic connection to the original labels (e.g., “visual test” for SST2, “river” for TREC), the overall sentences are less grammatical and semantically clear. This aligns with our observations from the persistence diagrams, where Word2Vec augmentations showed increased dispersion of H1 points, suggesting a disruption of the original data’s topological structure. The introduced augment of somewhat related but contextually inappropriate words (e.g., “rorschach” for SST2, “zombies” for TREC) corresponds to our convex hull analysis, which showed Word2Vec augmentations often extending beyond the boundaries of the original data space.

7.3.3. GloVe-Generated Text

The GloVe-generated samples exhibit the lowest level of coherence and grammaticality among the three methods. While some relevant words are present (e.g., “visual” for SST2, “lakes” and “scotland” for TREC), the overall sentences lack proper structure and clear meaning. This corresponds to our earlier findings where GloVe augmentations showed significant topological disruption in the persistence diagrams and extended furthest beyond the original data boundaries in the convex hull analysis. The lack of syntactic awareness in GloVe’s word-level replacements is evident in these samples, resulting in semantically incoherent augmentations. This aligns with our Delaunay triangulation analysis in Figure 8, which showed that less effective augmentation methods created fewer meaningful connections within the data space.

These examples vividly illustrate how the geometric and topological properties we analyzed translate into qualitative differences in the generated text. GPT-J’s ability to

maintain topological consistency and operate within the established semantic boundaries of the training data results in coherent, contextually appropriate augmentations. In contrast, the word replacement methods of Word2Vec and GloVe, which we found to disrupt topological structure and violate data space boundaries, produce less coherent and contextually appropriate text.

8. Limitations and Future Research

While our study provides valuable insights into the geometric and topological properties of effective text data augmentation, it is important to acknowledge its limitations and identify areas for future research.

- **Dataset diversity:** Our study focused on two datasets (SST2 and TREC). Future work should extend this analysis to a broader range of datasets across different domains and languages to validate the generalizability of our findings.
- **Dimensionality reduction:** Our analysis relied heavily on PCA for dimensionality reduction, with most datasets showing that two principal components captured the majority of the variance. However, the SNIPS dataset proved to be an exception, requiring higher-dimensional analysis. This limitation highlights the need for future research to cover the following:
 - Explore alternative dimensionality reduction techniques that might better capture the complexity of diverse datasets.
 - Develop methods to determine the optimal number of dimensions for analysis for different types of datasets.
- **Linguistic structure analysis:** Our current study looks at word-level analysis, examining the geometric and topological properties of individual word embeddings. While this approach has provided valuable insights, it does not fully capture the complexity of higher-level linguistic structures. Future research could extend our framework to analyze sentence-level properties, word order, and syntactic relationships.
- **Multimodal data:** As many real-world applications involve multimodal data, future research could extend our geometric framework to analyze augmentation techniques for combined text and image data.
- **Optimization framework:** Building on our findings, future research could develop an optimization framework that uses geometric and topological properties to automatically select or generate the most effective augmentation data for a given task and dataset.
- **New augmentation strategies:** Our geometric framework may prove useful in developing new augmentation strategies that explicitly consider the spatial distribution and connectivity of data points.

By addressing these limitations and pursuing these future research directions, we can further refine our understanding of effective text data augmentation and develop more robust and efficient augmentation techniques for various NLP tasks.

9. Conclusions

In this paper, we explored the topological and geometric properties of effective augmentation data for text classification, comparing data generated through Word2Vec, GloVe embeddings, and GPT-J. Our analysis revealed critical insights into successful augmentation strategies. We found that the augmented data points that improved classification accuracy consistently fell within the boundaries of the original data space, thus preserving meaning and relevance. Furthermore, we observed that the interconnectedness of the augmented points within these original data boundaries was positively correlated with the improved classifier performance.

To reach these conclusions, we employed a diverse set of analytical techniques. We utilized TDA to compare the three sets of augmented data, applied PCA for dimensionality reduction, and conducted convex hull and Delaunay triangulation analyses to examine

spatial relationships. In particular, our PCA findings revealed that for most datasets, the first two principal components captured the majority of the variance in the data. This allowed us to effectively visualize and analyze the geometric properties of the augmented data in a two-dimensional space.

Our findings demonstrate that effective text data augmentation goes beyond simply increasing dataset size. Rather, it requires maintaining the original data structure and preserving relationships between data points. This insight has far-reaching implications that extend beyond text classification. We have developed a framework that potentially allows for the evaluation of augmentation techniques without extensive training and testing, which could revolutionize the development of new augmentation strategies. Moreover, our approach offers a novel perspective on text data augmentation in machine learning more broadly, opening up possibilities for developing comprehensive metrics that combine topological persistence with geometric measures to evaluate augmentation effectiveness.

Our research provides both a theoretical framework for understanding the properties of effective augmented data and practical insights for improving augmentation techniques. By viewing text data augmentation through a geometric and topological lens, we have uncovered fundamental principles that govern its effectiveness. As we continue to explore these geometric relationships, we can develop more sophisticated and effective augmentation strategies, ultimately improving the performance and robustness of machine learning models across a wide range of natural language processing tasks. We encourage others to further explore this, potentially with other embeddings and LLMs. This work not only advances our understanding of text data augmentation but also paves the way for more efficient and powerful NLP models in the future.

Author Contributions: Conceptualization, E.M.-K.L.; Methodology, S.J.H.F. and W.L.; Software, S.J.H.F.; Investigation, S.J.H.F.; Writing—original draft, S.J.H.F.; Writing—review & editing, E.M.-K.L.; Supervision, E.M.-K.L. and W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used are publicly available as cited. Program code for the experiments are available from the URL in the first paragraph of Section 3.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhang, S.; Jafari, O.; Nagarkar, P. A survey on machine learning techniques for auto labeling of video, audio, and text data. *arXiv* **2021**, arXiv:2109.03784.
2. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
3. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 1–48. [[CrossRef](#)]
4. Bayer, M.; Kaufhold, M.A.; Reuter, C. A Survey on Data Augmentation for Text Classification. *ACM Comput. Surv.* **2022**, *55*, 146:1–146:39. [[CrossRef](#)]
5. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *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)*; Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186. [[CrossRef](#)]
6. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. *Improving Language Understanding by Generative Pre-Training*; Technical Report; Open AI: San Francisco, CA, USA, 2018.
7. Sahu, G.; Rodriguez, P.; Laradji, I.H.; Atighehchian, P.; Vazquez, D.; Bahdanau, D. Data augmentation for intent classification with off-the-shelf large language models. *arXiv* **2022**, arXiv:2204.01959.
8. Edwards, A.; Ushio, A.; Camacho-Collados, J.; de Ribaupierre, H.; Preece, A. Guiding generative language models for data augmentation in few-shot text classification. *arXiv* **2021**, arXiv:2111.09064.
9. Dai, H.; Liu, Z.; Liao, W.; Huang, X.; Cao, Y.; Wu, Z.; Zhao, L.; Xu, S.; Liu, W.; Liu, N.; et al. Auggpt: Leveraging chatgpt for text data augmentation. *arXiv* **2023**, arXiv:2302.13007.
10. Møller, A.G.; Aarup Dalsgaard, J.; Pera, A.; Aiello, L.M. Is a prompt and a few samples all you need? Using GPT-4 for data augmentation in low-resource classification tasks. *arXiv* **2023**, arXiv:2304.13861. [[CrossRef](#)]

11. Feng, S.Y.; Gangal, V.; Wei, J.; Chandar, S.; Vosoughi, S.; Mitamura, T.; Hovy, E. A survey of data augmentation approaches for NLP. *arXiv* **2021**, arXiv:2105.03075.
12. Shorten, C.; Khoshgoftaar, T.M.; Furrht, B. Text data augmentation for deep learning. *J. Big Data* **2021**, *8*, 101. [[CrossRef](#)]
13. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28.
14. Wei, J.; Zou, K. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*; Inui, K., Jiang, J., Ng, V., Wan, X., Eds.; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 6382–6388. [[CrossRef](#)]
15. Miller, G.A. WordNet: A lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [[CrossRef](#)]
16. Marivate, V.; Sefara, T. Improving short text classification through global augmentation methods. In Proceedings of the Machine Learning and Knowledge Extraction: 4th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2020, Dublin, Ireland, 25–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 385–399.
17. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the 2013 International Conference on Learning Representations, Scottsdale, AZ, USA, 2–4 May 2013.
18. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543. [[CrossRef](#)]
19. Kobayashi, S. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*; Walker, M., Ji, H., Stent, A., Eds.; Association for Computational Linguistics: Minneapolis, MN, USA, 2018; pp. 452–457. [[CrossRef](#)]
20. Sennrich, R.; Haddow, B.; Birch, A. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2016.
21. McKeown, K. Paraphrasing questions using given and new information. *Am. J. Comput. Linguist.* **1983**, *9*, 1–10.
22. QUIRK, C. Monolingual machine translation for paraphrase generation. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP), Barcelona, Spain, 25–26 July 2004.
23. Iyyer, M.; Wieting, J.; Gimpel, K.; Zettlemoyer, L. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv* **2018**, arXiv:1804.06059.
24. Coulombe, C. Text data augmentation made simple by leveraging nlp cloud apis. *arXiv* **2018**, arXiv:1812.04718.
25. Fadaee, M.; Bisazza, A.; Monz, C. Data augmentation for low-resource neural machine translation. *arXiv* **2017**, arXiv:1705.00440.
26. Fu, Z.; Tan, X.; Peng, N.; Zhao, D.; Yan, R. Style transfer in text: Exploration and evaluation. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
27. Xie, Q.; Dai, Z.; Hovy, E.; Luong, T.; Le, Q. Unsupervised data augmentation for consistency training. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6256–6268.
28. Kumar, V.; Choudhary, A.; Cho, E. Data augmentation using pre-trained transformer models. *arXiv* **2020**, arXiv:2003.02245.
29. Maynez, J.; Narayan, S.; Bohnet, B.; McDonald, R. On Faithfulness and Factuality in Abstractive Summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 1906–1919.
30. Yang, S.; Wang, Y.; Chu, X. A survey of deep learning techniques for neural machine translation. *arXiv* **2020**, arXiv:2002.07526.
31. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
32. Keskar, N.S.; McCann, B.; Varshney, L.R.; Xiong, C.; Socher, R. Ctrl: A conditional transformer language model for controllable generation. *arXiv* **2019**, arXiv:1909.05858.
33. Wang, B.; Komatsuzaki, A. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. 2021. Available online: <https://github.com/kingoflolz/mesh-transformer-jax> (accessed on 17 September 2024).
34. Zhu, C.; Cheng, Y.; Gan, Z.; Sun, S.; Goldstein, T.; Liu, J. Freelib: Enhanced adversarial training for natural language understanding. *arXiv* **2019**, arXiv:1909.11764.
35. Bender, E.M.; Gebru, T.; McMillan-Major, A.; Shmitchell, S. On the dangers of stochastic parrots: Can language models be too big? In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual, 3–10 March 2021; pp. 610–623.
36. Howard, J.; Ruder, S. Universal language model fine-tuning for text classification. *arXiv* **2018**, arXiv:1801.06146.
37. Řehůřek, R.; Sojka, P. Gensim—Statistical Semantics in Python. Available online: <https://radimrehurek.com/gensim/models/word2vec.html> (accessed on 17 September 2024).
38. Wang, W.Y.; Yang, D. That’s so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2557–2563.
39. Wang, B. Mesh-Transformer-JAX: Model-Parallel Implementation of Transformer Language Model with JAX. 2021. Available online: <https://github.com/kingoflolz/mesh-transformer-jax> (accessed on 17 September 2024).

40. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; pp. 6000–6010.
41. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, 16–20 November 2020; pp. 38–45.
42. Yang, Y.; Malaviya, C.; Fernandez, J.; Swayamdipta, S.; Le Bras, R.; Wang, J.P.; Bhagavatula, C.; Choi, Y.; Downey, D. Generative Data Augmentation for Commonsense Reasoning. In *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP*, Online, 16–20 November 2020; Association for Computational Linguistics: Minneapolis, MN, USA, 2020; pp. 1008–1025. [\[CrossRef\]](#)
43. Anaby-Tavor, A.; Carmeli, B.; Goldbraich, E.; Kantor, A.; Kour, G.; Shlomov, S.; Tepper, N.; Zwerdling, N. Do not have enough data? Deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 7383–7390.
44. Johnson, R.; Zhang, T. Effective use of word order for text categorization with convolutional neural networks. *arXiv* **2014**, arXiv:1412.1058.
45. Zhang, Y.; Wallace, B. A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification. *arXiv* **2015**, arXiv:1510.03820.
46. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.; Potts, C. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, WA, USA, 18–21 October 2013; pp. 1631–1642.
47. Hovy, E.; Gerber, L.; Hermjakob, U.; Lin, C.Y.; Ravichandran, D. Toward Semantics-Based Answer Pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*, San Diego, CA, USA, 18–21 March 2001.
48. Casadio, M.; Komendantskaya, E.; Rieser, V.; Daggitt, M.L.; Kienitz, D.; Arnaboldi, L.; Kokke, W. Why Robust Natural Language Understanding is a Challenge. *arXiv* **2022**, arXiv:2206.14575.
49. Ning-min, S.; Jing, L. A Literature Survey on High-Dimensional Sparse Principal Component Analysis. *Int. J. Database Theory Appl.* **2015**, *8*, 57–74. [\[CrossRef\]](#)
50. Taloba, A.I.; Eisa, D.A.; Ismail, S.S.I. A Comparative Study on using Principle Component Analysis with Different Text Classifiers. *Int. J. Comput. Appl.* **2018**, *180*, 1–6. [\[CrossRef\]](#)
51. Heimerl, F.; Gleicher, M. Interactive analysis of word vector embeddings. *Comput. Graph. Forum* **2018**, *37*, 253–265. [\[CrossRef\]](#)
52. Raunak, V.; Gupta, V.; Metze, F. Effective Dimensionality Reduction for Word Embeddings. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, Florence, Italy, 2 August 2019; pp. 235–243. [\[CrossRef\]](#)
53. Camastra, F.; Staiano, A. Intrinsic dimension estimation: Advances and open problems. *Inf. Sci.* **2016**, *328*, 26–41. [\[CrossRef\]](#)
54. Jolliffe, I.T.; Cadima, J. Principal component analysis: A review and recent developments. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2016**, *374*, 20150202. [\[CrossRef\]](#)
55. Cattell, R.B. The scree test for the number of factors. *Multivar. Behav. Res.* **1966**, *1*, 245–276. [\[CrossRef\]](#) [\[PubMed\]](#)
56. Leykam, D.; Angelakis, D.G. Topological data analysis and machine learning. *Adv. Phys. X* **2023**, *8*, 2202331. [\[CrossRef\]](#)
57. Rathore, A.; Zhou, Y.; Srikumar, V.; Wang, B. TopoBERT: Exploring the topology of fine-tuned word representations. *Inf. Vis.* **2023**, *22*, 186–208. [\[CrossRef\]](#)
58. Jakubowski, A.; Gašić, M.; Zibrowius, M. Topology of word embeddings: Singularities reflect polysemy. *arXiv* **2020**, arXiv:2011.09413.
59. Niyogi, P.; Smale, S.; Weinberger, S. A topological view of unsupervised learning from noisy data. *SIAM J. Comput.* **2011**, *40*, 646–663. [\[CrossRef\]](#)
60. Hensel, F.; Moor, M.; Rieck, B. A survey of topological machine learning methods. *Front. Artif. Intell.* **2021**, *4*, 681108. [\[CrossRef\]](#)
61. Munch, E. A user’s guide to topological data analysis. *J. Learn. Anal.* **2017**, *4*, 47–61. [\[CrossRef\]](#)
62. Maria, C. Persistent Cohomology. In *GUDHI User and Reference Manual*, 3.9.0 ed. Available online: https://gudhi.inria.fr/python/latest/persistent_cohomology_user.html (accessed on 17 September 2024).
63. Wasserman, L. Topological data analysis. *Annu. Rev. Stat. Its Appl.* **2018**, *5*, 501–532. [\[CrossRef\]](#)
64. de Berg, M.; Cheong, O.; van Kreveld, M.; Overmars, M. Convex Hulls. In *Computational Geometry: Algorithms and Applications*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 243–258. [\[CrossRef\]](#)
65. Edelsbrunner, H.; Kobbelt, L.; Polthier, K.; Boissonnat, J.D.; Carlsson, G.; Chazelle, B.; Gao, X.S.; Gotsman, C.; Guibas, L.; Kim, M.S.; et al. *Geometry and Computing*; Springer: Berlin/Heidelberg, Germany, 2010.
66. Chazelle, B. An optimal convex hull algorithm in any fixed dimension. *Discret. Comput. Geom.* **1993**, *10*, 377–409. [\[CrossRef\]](#)
67. Toussaint, G.T. *Computational Geometry: Recent Developments*. In *New Advances in Computer Graphics: Proceedings of CG International’89*; Springer: Berlin/Heidelberg, Germany, 1989; pp. 23–51.
68. Yousefzadeh, R. Deep learning generalization and the convex hull of training sets. *arXiv* **2021**, arXiv:2101.09849.
69. Cohen-Steiner, D.; Edelsbrunner, H.; Harer, J. Stability of persistence diagrams. In *Proceedings of the Twenty-First Annual Symposium on Computational Geometry*, Pisa, Italy, 6–8 June 2005; pp. 263–271.
70. Edelsbrunner, H.; Harer, J. Persistent homology—A survey. *Contemp. Math.* **2008**, *453*, 257–282.
71. Carlsson, G. Topological pattern recognition for point cloud data. *Acta Numer.* **2014**, *23*, 289–368. [\[CrossRef\]](#)

-
72. Barber, C.B.; Dobkin, D.P.; Huhdanpaa, H. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* **1996**, *22*, 469–483. [[CrossRef](#)]
 73. Musin, O.R. Properties of the Delaunay triangulation. In Proceedings of the Thirteenth Annual Symposium on Computational Geometry, Nice, France, 4–6 June 1997; pp. 424–426.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.