SMOTExT: SMOTE meets Large Language Models

Mateusz Bystroński[1], Mikołaj Hołysz[1], Grzegorz Piotrowski[1],
Nitesh V. Chawla[2], Tomasz Kajdanowicz[1]
[1]Wrocław University of Science and Technology, [2]University of Notre Dame
Correspondence: mateusz.bystronski@pwr.edu.pl

Abstract

Data scarcity and class imbalance are persistent challenges in training robust NLP models, especially in specialized domains or low-resource settings. We propose a novel technique, **SMOTExT**, that adapts the idea of Synthetic Minority Over-sampling (SMOTE) to textual data. Our method generates new synthetic examples by interpolating between BERT-based embeddings of two existing examples and then decoding the resulting latent point into text with xRAG architecture. By leveraging xRAG's cross-modal retrieval-generation framework, we can effectively turn interpolated vectors into coherent text. While this is preliminary work supported by qualitative outputs only, the method shows strong potential for knowledge distillation and data augmentation in few-shot settings. Notably, our approach also shows promise for privacy-preserving machine learning: in early experiments, training models solely on generated data achieved comparable performance to models trained on the original dataset. This suggests a viable path toward safe and effective learning under data protection constraints.

1 Introduction

Building reliable NLP models in low-resource scenarios or for under-represented classes remains a major challenge due to the limited availability of annotated training data and ethical concerns. Data augmentation offers a promising strategy to mitigate this issue by generating additional synthetic examples. However, common textual augmentation techniques—such as synonym replacement, random deletion, or back-translation—often introduce semantic drift, grammatical errors, or distributional inconsistencies. In contrast, interpolation-based augmentation has proven effective in continuous feature spaces, especially in domains like vision or tabular data. The **S**ynthetic **M**inority **O**ver-sampling **TE**chnique (SMOTE) (Chawla et al., 2002) is a well-known example that creates new minority-class samples by linearly interpolating between feature vectors. Adapting such interpolation to text, however, is non-trivial due to the discrete and high-dimensional nature of natural language. We propose a novel method—**SMOTExT**—that performs SMOTE-style interpolation for text by operating in a continuous latent space. Specifically, we embed two textual examples using a pre-trained encoder and compute a linear interpolation between their embeddings. This intermediate vector is then decoded into natural language using xRAG (Cheng et al., 2024).

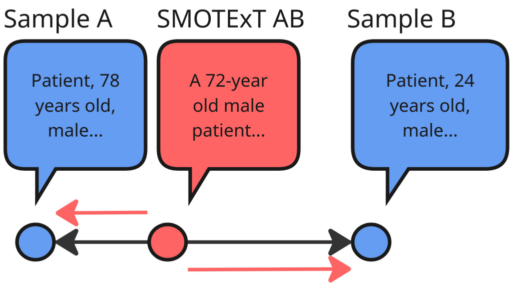
Sample A
SMOTExT AB
Sample B
Patient, 78 years old, male...
A 72-year old male patient...
Patient, 24 years old, male...

Figure 1: Two textual examples A and B are encoded into embeddings. An interpolated vector SMOTExT is fed into the xRAG model via its modality bridge (projector). Language model then generates a synthetic text that reflects combined features of the inputs.

This approach enables the generation of realistic, domain-consistent samples that can enrich training data in few-shot or imbalanced-class settings. Compared to perturbation-based augmentation, interpolation in latent space may better preserve semantic relationships between examples. However, interpolated vectors are not guaranteed to remain on the natural language manifold, and out-of-distribution artifacts may occur—particularly when inputs differ significantly. Nevertheless, by decoding these vectors using a strong generative model, we can often obtain fluent and contextually appropriate text, especially when source examples are from the same domain or class.

Our contributions are as follows: (1) We introduce a novel textual data augmentation technique based on embedding space interpolation, inspired by SMOTE and enabled by large-scale pre-trained models; (2) We provide a real-world proof-of-concept in the medical domain, showing that our method can synthesize medically plausible patient descriptions; (3) We conduct a preliminary quantitative evaluation on a general-purpose classification task, showing that the method improves metrics in low-resource settings and approximates real data distribution when used alone. and (4) We discuss limitations of the current approach and outline directions for future quantitative evaluation, including applications in privacy-preserving and generative NLP tasks.

2 Related Work

Text Data Augmentation. Numerous augmentation methods address data scarcity in NLP. Lexical approaches like EDA (Wei and Zou, 2019) apply random synonym replacement, insertion, or deletion, but often risk semantic drift or ungrammaticality. Paraphrasing via back-translation (Sennrich et al., 2016) or prompt-based LLM generation offers more fluency, though may drift off-distribution without constraints.

Other techniques operate in continuous representation spaces. Mixup (Zhang et al., 2018; Verma et al., 2019), originally for vision, has been adapted to NLP by interpolating hidden states. TMix (Chen et al., 2020) and SSMix (Yoon et al., 2021) create intermediate representations by mixing hidden layers in transformer models, improving training. However, these methods are typically tied to specific architectures and do not generate standalone text.

In contrast, our method decouples augmentation from model training by decoding interpolated embeddings into human-readable text. This enables model-agnostic data augmentation and facilitates qualitative inspection of synthetic examples.

SMOTE and Interpolation. Our approach is inspired by SMOTE (Chawla et al., 2002), a classic oversampling technique that mitigates class imbalance by interpolating between a minority class instance and one of its neighbors in feature space. This generates synthetic samples within the minority region, avoiding duplication. While effective in continuous domains (e.g., numeric or image data), applying SMOTE to text requires a suitable latent representation.

DeepSMOTE (Dablain et al., 2023) extended this idea to high-dimensional data by performing interpolation in the latent space of a neural encoder. It maps images into a semantic embedding space, applies SMOTE-style interpolation, and reconstructs synthetic examples via a decoder—circumventing unrealistic blending in pixel space. We adopt a similar principle for text, operating directly in the embedding space of pre-trained language models.

xRAG. The recently proposed xRAG model (Cheng et al., 2024) enables language models to condition on a single dense vector, derived from a retrieved document, via a special token. A fixed retriever generates document embeddings, which are projected into the LM's input space using a small trainable *modality bridge*. Only this projection is trained, keeping both the retriever and the LM frozen. The LM attends to the resulting embedding and generates fluent, context-aware output.

3 Proposed Method

Our approach leverages xRAG's capability to integrate and decode an arbitrary fixed-length embedding into natural language. Essentially, we repurpose the xRAG architecture as a decoder for interpolated example embeddings. Prior work (Cai et al., 2021) has shown that BERT (Devlin et al., 2019) embeddings tend to form well-structured and concentrated clusters in the latent space, making linear interpolation between them a viable strategy for generating semantically meaningful intermediate representations. This structural property supports the idea that interpolated points in embedding space can correspond to plausible language outputs. xRAG provides an off-the-shelf, pre-trained mechanism to realize such vectors as text, without the need to train a new autoencoder or decoder architecture from scratch. By treating the interpolated vector as if it were a retrieved document embedding, we enable the language model to generate a coherent sentence that blends features of the source examples.

Our method, consists of three main steps: (1) obtaining vector representations of input texts; (2) interpolating between two such vectors to produce a new latent point; and (3) decoding this latent vector into text using xRAG.

Embedding Text Examples. We first convert each text example into a fixed-dimensional embedding. In this work we use the [CLS] representation from `SRF-Embedding-Mistral` (Meng et al., 2024). Let x_i and x_j be two text examples (e.g., two sentences or short paragraphs). We obtain $e_i = \text{BERT-Enc}(x_i)$ and $e_j = \text{BERT-Enc}(x_j)$, where $e_i \in \mathbb{R}^d$ (with $d = 4096$ in our case). These embeddings are intended to capture the major features of each text in a continuous vector form.

Latent Space Interpolation. Next, we generate a new latent point between e_i and e_j. We sample a mixing coefficient $\lambda \in [0, 1]$. We then compute the interpolated vector $e_{\text{new}} = \lambda e_i + (1 - \lambda)e_j$. This linear interpolation in the embedding space is analogous to the DeepSMOTE procedure in image data, which assumes that the space is locally linear such that points between two same-class examples are likely to remain within the same class region. In practice, e_{new} should represent a blend of the semantic content from x_i and x_j. One key assumption here is that the embedding space of the encoder is smooth enough that this linear combination yields a meaningful intermediate representation, rather than an out-of-distribution vector. We also note that to maintain label consistency (in a supervised setting), x_i and x_j would typically be chosen from the same class or category.

Decoding via xRAG. The core novelty of our approach lies in using a xRAG architecture to decode e_{new} into a synthetic text y_{new}. In our work we use xRAG-7B model (Cheng et al., 2024) that was trained to accept an embedded context vector through a special token and generate an output conditioned on it. We denote the language model component as LM_{xRAG} and the modality bridge (projector) as W_p. W_p is a linear projection that takes as input a context embedding (originally, a document embedding from the retriever) and outputs a vector in the same space as the LM's token embeddings. During xRAG's training, $W_p(e)$ is inserted into the LM's input sequence as the embedding for a designated token (let's call it [X]). This way, the LM can attend to [X] as if it were a compressed representation of a document relevant to the task (e.g., answering a question or providing knowledge).

To use xRAG for our purpose, we feed the interpolated embedding through the same projector and prompt the LM to paraphrase the decoded text. Specifically, we compute $h_X = W_p(e_{\text{new}})$. Because LM_{xRAG} was trained with an autoencoding objective on documents—particularly in the context of paraphrasing—it has learned to generate coherent text that reflects the semantic content of h_X..

Example Workflow. Pseudocode for generating one synthetic example is as follows:

```
Input: $x_i$, $x_j$ (text examples)
$e_i \leftarrow$ BERT-Enc($x_i$)
$e_j \leftarrow$ BERT-Enc($x_j$)
Sample $\lambda \in [0, 1]$
$e_{\text{new}} \leftarrow \lambda e_i + (1 - \lambda)e_j$
$h_X \leftarrow W_p(e_{\text{new}})$    // project to LM space
Feed $h_X$ as [X] into $\text{LM}_{\text{xRAG}}$, generate $y_{\text{new}}$
```

In practice, multiple synthetic samples can be generated by choosing different pairs (x_i, x_j) and/or different λ values. These y_{new} outputs can then be included as additional training data for downstream models or further analyzed.

4 Experiments

To provide a proof of concept for the proposed interpolation-based text augmentation method, we conduct a set of small-scale experiments designed to test its viability in low-resource classification settings. While these evaluations are preliminary, they offer insights into the effectiveness and limitations of the approach and help establish a foundation for future, more comprehensive benchmarking.

We include in Appendix A.1 a clinical case study illustrating how our method generates semantically coherent synthetic samples by interpolating between real examples in the latent space. The resulting text combines features from both inputs while maintaining domain plausibility.

To further evaluate the proposed method in a general-purpose classification setting, we use the `20 Newsgroups` dataset.[1] This corpus contains discussion posts from 20 topical newsgroups that form several well-known semantic clusters. Following `scikit-learn`'s recommendations, we remove email headers, signatures, and quoted content to reduce information leakage and simplify the classification task.

We select a subset of classes that are semantically distant from each other, as prior work has shown that some groups (e.g., computer-related or religion-related) are challenging to distinguish—even for human annotators. As a baseline,

[1] https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html

Training Setup	Accuracy	Macro F1	Weighted F1
Real only (1,000 samples)	0.8361	0.8382	0.8382
Real + SMOTExT (1,000 + 2,600)	**0.8394**	**0.8404**	**0.8408**
SMOTExT only (2,600 samples)	0.8037	0.8038	0.8036

Table 1: Classification results on a low-resource 20 Newsgroups subset. Augmenting real training data with 2,600 interpolated SMOTExT examples improves macro and weighted F1 scores. Training on synthetic data alone approximates the performance of the real-data baseline, suggesting potential for privacy-preserving dataset generation.

we fine-tune a ModerBERT (Warner et al., 2024) model, freezing all encoder layers and training only the classification head.

To simulate a low-resource scenario, we restrict the training set to 1,000 real examples. We then generate 2,600 SMOTExT examples using our latent interpolation pipeline and add them to the training data. We compare three conditions: (1) training on 1,000 real examples (baseline), (2) training on 1,000 real + 2,600 SMOTExT examples, and (3) training on 2,600 SMOTExT examples alone. Results are reported in Table 1.

The addition of SMOTExT samples yields modest but consistent improvements in both macro and weighted F1 scores. This demonstrates the usefulness of interpolation-based augmentation in low-resource settings.

Notably, training on SMOTExT data alone results in competitive performance, with only a small drop relative to the real-data baseline. This suggests that the SMOTExT samples capture meaningful features of the original distribution—an encouraging result for privacy-sensitive scenarios. The slight performance gap also highlights a known limitation of generative approaches based on latent interpolation: without constraints, some outputs may drift slightly out of distribution, introducing minor inconsistencies or irrelevant details. Addressing this challenge will be an important focus of future work.

5 Discussion and Conclusions

We presented a method to synthesize new textual data points by interpolating between sentence embeddings and decoding the result using a large language model. This approach adapts the core idea of SMOTE to the NLP setting.

Our preliminary results show that the method can generate fluent, coherent texts that plausibly combine attributes of their source inputs. This suggests potential utility in data augmentation for low-resource or imbalanced classification tasks, as well as in generative scenarios where realism and contextual richness matter more than label precision. Importantly, since the method operates purely on latent representations and decodes them into novel outputs, it offers a promising direction for privacy-preserving data synthesis—particularly in sensitive domains such as healthcare, where access to real patient records is often restricted.

Beyond augmentation, our approach also opens up the possibility of generating fully synthetic datasets that approximate the original data distribution. This is particularly promising for training generative models, where access to large, high-quality datasets is essential, yet often constrained by privacy, legal, or ethical considerations. By distilling the structure and diversity of real data into novel synthetic samples, our method may enable pretraining or fine-tuning of generative language models without direct exposure to protected content. This has the potential to mitigate risks of data leakage while preserving downstream performance.

Importantly, our method does not rely on task-specific training: synthetic data can be generated once and reused across applications. This makes it particularly attractive for semi-structured tasks such as generating medical histories, populating patient profiles, or augmenting dialogue datasets. We believe this approach has wide applicability across domains and task types, especially where labeled data is scarce.

Future work will focus on scaling the evaluation to more domains and tasks, exploring interpolation beyond pairs (e.g., multi-point or manifold-based mixing), and developing mechanisms for better control over semantic consistency and out-of-distribution risks. We also plan to investigate strategies for automatic filtering or calibration of generated examples.

6 Limitations

This work is in an early exploratory stage, but initial experiments provide encouraging evidence.

Nonetheless, several limitations remain.

First, while we now present preliminary quantitative results, a more comprehensive evaluation across a wider range of tasks and domains is needed to fully assess the generality and impact of our approach. Our current benchmarks are limited to a subset of the 20 Newsgroups dataset, and further analysis is required to determine how well the method performs in truly low-resource or high-stakes settings such as medicine or law.

Secondly, although we assume that interpolating between two examples from the same class yields a valid sample from that class, this assumption is not formally guaranteed and may result in out-of-distribution samples. Nevertheless, our experiments demonstrate that a model trained exclusively on SMOTExT data achieves performance comparable to that of a model trained on real examples, suggesting that SMOTExT effectively approximates the underlying data distribution.

A further limitation of our current setup is the use of a relatively small language model (7B parameters). While this choice enables efficient experimentation, it may limit the expressiveness and fidelity of the generated outputs. Nevertheless, the method proves effective even in this setting, and we expect that applying it with larger models would lead to improved results.

Lastly, while the method shows promise for privacy-preserving synthetic data generation—particularly by avoiding direct exposure to sensitive inputs—it currently lacks explicit mechanisms to enforce semantic constraints or prevent subtle inconsistencies. In generative applications, this may be acceptable; but in settings where factual accuracy or clinical safety is paramount, stricter control mechanisms will be necessary.

References

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. 2021. Isotropy in the contextual embedding space: Clusters and manifolds. In *International Conference on Learning Representations*.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. xRAG: Extreme context compression for retrieval-augmented generation with one token. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Damien Dablain, Bartosz Krawczyk, and Nitesh V. Chawla. 2023. Deepsmote: Fusing deep learning and smote for imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9):6390–6404.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. Sfr-embedding-mistral:enhance text retrieval with transfer learning. Salesforce AI Research Blog.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Y. Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *Preprint*, arXiv:2412.13663.

Jason Wei and Kai Zou. 2019. 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)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Soyoung Yoon, Gyuwan Kim, and Kyumin Park. 2021. SSMix: Saliency-based span mixup for text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages

3225–3234, Online. Association for Computational Linguistics.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

A Appendix

A.1 Clinical Text Interpolation Case Study

To illustrate the generative capabilities of our method in a structured domain, we present a proof-of-concept example in the medical setting. Due to privacy concerns and annotation costs, medical text datasets are typically small, making them well-suited for data augmentation.

We constructed two clinical cases (A and B) using GPT-4o to simulate realistic outpatient scenarios. These were embedded using a BERT-based encoder, linearly interpolated in latent space, and decoded using xRAG-7B to produce a SMOTExT case A-B.

Example A (Original):
"Patient, 24 years old, male. No significant chronic illnesses. Presents with mild sore throat for the past 2 days. No fever, no shortness of breath. No known allergies. Non-smoker, not on any regular medications."

Example B (Original):
"Patient, 78 years old, male. Type 2 diabetes, hypertension, history of myocardial infarction 5 years ago. Complains of exertional dyspnea and dry cough for the past week. Temperature 37.8°C. Takes metformin, bisoprolol, and ramipril."

SMOTExT A-B (Synthetic):
"A 72-year-old male patient presented to the clinic with a chief complaint of shortness of breath and chest pain for the past 2 days. He described the symptoms as a feeling of tightness in the chest, which worsened with exertion and improved with rest. The patient also reported a non-productive cough and mild fever. He denied any recent travel, contact with sick individuals, or exposure to allergens. The patient's past medical history was significant for hypertension, hyperlipidemia, and type 2 diabetes mellitus. He was a former smoker, having quit 10 years ago. The patient's vital signs were within normal limits, and his physical examination was unremarkable. His laboratory results showed an elevated white blood cell count, normal electrolytes, and a slightly elevated creatinine level. A chest X-ray was ordered, which revealed a possible infiltrate in the right lower lobe. The patient was diagnosed with a possible respiratory infection and was prescribed antibiotics and a bronchodilator. He was advised to follow up with his primary care physician for further evaluation and management."

Table 2: Proof-of-concept generation via latent interpolation in the medical domain. The SMOTExT case plausibly integrates features from both source cases (A and B) while maintaining clinical coherence.

This comparison highlights how the generated sample semantically resides between the two source cases, blending demographic, symptomatic, and comorbidity information. The resulting case maintains medical plausibility and exhibits coherent narrative structure.

The example was reviewed by a physician, who confirmed its consistency with medical knowledge. Minor suggestions included the possible addition of elevated temperature and auscultatory findings in the lungs to fully match the described clinical picture. Nonetheless, the overall scenario was deemed realistic and clinically valid.

Attribute	**Example A**	**Example B**	**SMOTExT A-B**
Age	24 years old	78 years old	72 years old (closer to B)
Chronic conditions	None	Type 2 diabetes, hypertension, myocardial infarction	Hypertension, type 2 diabetes, hyperlipidemia (mix of B + novel)
Symptom profile	Mild sore throat, no fever or dyspnea	Exertional dyspnea, dry cough, fever (37.8°C)	Shortness of breath, chest pain, dry cough, mild fever (onset similar to A, symptoms closer to B)
Medications	None	Metformin, bisoprolol, ramipril	Not specified
Lifestyle factors	Non-smoker	Not mentioned	Ex-smoker (new plausible feature)

Table 3: Structured comparison of two original clinical descriptions (A and B) and a SMOTExT casE generated via latent space interpolation. The SMOTExT case plausibly combines features from both inputs while maintaining medical realism and narrative fluency.