**Context-Aware Perturbations for Data Augmentation**

(<https://github.com/Kahl-d/Data-Augmentation-Project>)

**Khalid Mehtab Khan**

SFSU ID: 923673423

Professor: Isabel Hyo Jung Song

**Project Description**

Limited data availability often impacts the performance of machine learning models, especially in niche domains.

Machine learning models **often require large**, diverse datasets to perform effectively. However, in many real-world scenarios—such as healthcare, finance, or niche research areas—data is either limited, sensitive, or imbalanced. This constraint leads to models that lack robustness and generalization, especially in scenarios where critical decisions depend on their predictions.

**Technical Problem (solved)**

**Random perturbations**, often used for data augmentation or model interpretability, introduce their own set of challenges:

• **Unrealistic Variations**:

Randomly modifying input values can result in data that is nonsensical or impossible in the real world. For example, changing a patient’s age from 35 to 150 makes no practical sense in a healthcare dataset.

• **Loss of Context**:

Random changes fail to consider the relationships between features. A perturbation in one feature might not align with changes in another, leading to unrealistic or irrelevant outputs.

• **Reduced Stability**:

Models trained on or explained through such perturbations often produce inconsistent results, making them unreliable for sensitive applications.

**Proposed Solution**

We propose a novel framework for generating **context-aware perturbations** that improve data augmentation and model explanation.

**Key Solutions -**

1. **Sensible Perturbations**: Instead of relying on random changes, we generate perturbations that are contextually and structurally meaningful.

2. **Domain Understanding**: By fine-tuning a language model (DeBERTa) on the training data, we ensure that the model learns the specific relationships and patterns within the dataset.

3. **Simulation of Data World**: The perturbations simulate plausible variations within the data’s real-world context, maintaining coherence and preserving feature relationships.

**Approach and Methodology**

1. **Generating Sensible Perturbations**

The core idea is to simulate the **“world”** represented by the dataset and generate values that fit naturally into this simulation.

1. **Data Conversion to Text**:
   1. Each row in the dataset is converted into a structured textual description, with each feature described in natural language.

*{'Gender': 'Male’ | 'Age': 25 | ‘BMI’: 54.6 | ‘Smoking Status’: ‘Non-Smoker’ | ‘Restaurant Visits’: 3’}*

1. **Feature Masking**:  
   1. Randomly mask one or more features, ensuring that certain fields (like critical identifiers) remain fixed.

*{'Gender': 'Male’ | 'Age': [MASK] | ‘BMI’: [MASK] | ‘Smoking Status’: ‘Non-Smoker’ | ‘Restaurant Visits’: 3'}*

1. **Context-Aware Predictions**:
   1. Fine-tune DeBERTa on the dataset using Masked Language Modeling (MLM) to predict realistic replacements for masked fields.

*Prediction:*

*{'Gender': 'Male’ | 'Age': 28 | ‘BMI’: 58.3 | ‘Smoking Status’: ‘Non-Smoker’ | ‘Restaurant Visits’: 3'}*

1. **Reconstruction**:
   1. Replace the masked values in the original row with the predicted values to create a new perturbed row.
2. **Repeat for Multiple Perturbations**:
   1. Iterate over multiple combinations of masked features to generate a diverse yet realistic set of perturbations.

A screenshot of a computer

Description automatically generated

1. **Performance Testing**

The process was tested on a classical machine learning task to validate its effectiveness.

**Baseline Accuracy**:

• Train a classical machine learning model (e.g., Random Forest) on the original dataset and record its accuracy.

A table with numbers and text

Description automatically generated

**Fine-Tuning**:

• Fine-tune DeBERTa on the training data to understand feature relationships and generate context-aware perturbations.

A screen shot of a computer program

Description automatically generated

**Data Augmentation**:

• Use the perturbations to augment the training data.A person holding a sign

Description automatically generated

**Improved Model Performance**:

• Retrain the same classical model on the augmented dataset, achieving a performance improvement of 3-3.5%.

A screenshot of a graph

Description automatically generated

**Metrics**

**Efficiency**:

• I demonstrated that the model’s performance improved by using the generated data, showing that the perturbations enhanced its ability to generalize and handle diverse scenarios effectively.

A table with numbers and text

Description automatically generated

**Stability**:

• I ran experiments using different random seeds to ensure that the generated data and the model’s performance remained consistent across multiple trials.

A table with numbers and text

Description automatically generated

A table with numbers and text

Description automatically generated

**Fidelity**:

• By testing the fine-tuned model on this task, we’ll see if it can match or even outperform the classical approach.

• Tested the fine-tuned DeBERTa model on the same classification task to evaluate if it understands the relationships in the data.

• The fine-tuned model showed improved classification accuracy, proving it aligns well with the original task.

**Solution**

By combining context-aware perturbation generation with fine-tuned models, we addressed the key challenges of limited data and random perturbations. The approach involved:

1. Fine-tuning DeBERTa to capture the dataset’s structure and relationships.

2. Generating perturbations that simulate realistic scenarios.

3. Validating the fidelity and stability of the fine-tuned model through classification tasks.

**Use Cases**

**1. Data Augmentation**

• Augment datasets in low-resource or imbalanced domains, improving model training and performance.

• Example: Enhance a small healthcare dataset by generating plausible patient records for training disease prediction models.

**2. Industry-Specific Applications**

• **Healthcare**: Doctors can simulate patient scenarios to train and evaluate models for diagnostics or treatment recommendations.

• **Finance**: Financial analysts can generate specific variations in credit or risk data for better decision-making models.

• **Education**: Adaptive learning systems can create simulated student data to personalize learning experiences.

**3. Simulation Fields**

• Use the perturbation algorithm to create realistic scenarios in simulation-heavy domains like autonomous driving, robotics, or policy modeling.

**Future Directions**

**Scaling Up:**

• Use **heavier models** (e.g., GPT-4 or larger domain-specific transformers) for richer and more nuanced perturbations.

• Explore scaling the solution across multiple datasets and industries, ensuring adaptability to various domains.

**Advanced Integration:**

• Integrate the solution with **explainability tools** like LIME and SHAP to explain the impact of augmented data on model decisions.

• Leverage **causal reasoning** for better augmentation strategies, ensuring generated data reflects causal relationships in the domain.

**Research and Development:**

• Investigate **multi-modal augmentation** by including text, images, or tabular data together.

• Compare the performance of augmented data against traditional augmentation techniques to highlight the strengths of your approach.

**Conclusion**

This project demonstrates how context-aware perturbations can address key challenges in data augmentation and model explanation.

By leveraging fine-tuned models, we successfully increased model performance, validated fidelity and stability, and showcased a framework that can be applied across industries. This approach not only improves training data quality but also offers actionable insights for interpretability, making it a valuable tool for machine learning practitioners and researchers.