GENERATION OF NATURAL LANGUAGE EXPLANATIONS FOR AMCC DERIVED COUNTERFACTUALS

by

Tarun Thomas Eapen

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Computer Science

at

The University of Wisconsin-Milwaukee

May 2025

ABSTRACT

GENERATION OF NATURAL LANGUAGE EXPLANATIONS FOR AMCC DERIVED COUNTERFACTUALS

by

Tarun Thomas Eapen

The University of Wisconsin-Milwaukee, 2025
Under the Supervision of Professor Susan McRoy

This thesis explores 2 different approaches to generate simple natural language sentences for counterfactual explanations to better help user understand a machine learning model's decision. The objective is to determine a domain independent natural language generation approach to generate effective and simple sentences to explain counterfactual explanations. Counterfactual explanations generated from the achievable minimally-contrastive counterfactual explanations (AMCC) algorithm is used to conduct experiments for the two approaches, a template-based approach, enhanced with WordNet synonyms and lexical substitution using a BERT model, and a template-guided approach using a fine-tuned T5 model. Evaluations are performed using an automatic metric, BLEURT, to identify the effectiveness of each approach. To test the effectiveness of domain independence of the proposed approaches, experiments were conducted on a dataset from a new domain to see the effectiveness of generated natural language explanations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

XAI	Explainable AI
AMCC	Achievable Minimally-Contrastive Counterfactual Explanations
NLG	Natural Language Generation
MLM	Masked Language Modeling
FAT	Feature Actionability Taxonomy
BERT	Bidirectional Encoder Representations from Transformers
RoBERTa	Robustly Optimized BERT
T5	Text-to-Text Transfer Transformer
PEFT	Parameter-Efficient Fine-Tuning
LoRA	Low-Rank Adaptation
NLTK	Natural Language Toolkit
CUDA	Compute Unified Device Architecture
BLEURT	Bilingual Evaluation Understudy with Representations from Transformers
HMEQ	Home Equity Loan
LM	Language Model

ACKNOWLEDGEMENTS

I would like to sincerely thank my mentor, Dr. Susan W. Mcroy, for her great advice, consistent support, and mentorship, all of which greatly impacted the outcome of my research. Her extraordinary endurance and unwavering support have shaped my research path and helped me finish my thesis.

Additionally, I would like to thank Professors Tian Zhao and Rohit Kate for agreeing to serve on my thesis committee and offering their professional knowledge.

I thank my wife for her constant support, love and encouragement, and telling me that I could do it. Lastly, I thank my family for their unwavering love, and inspiration throughout my academic journey. Their belief in me, their advice, and their kind contributions will always be my sincere gratitude.

1. INTRODUCTION

Explainable AI refers to methods and techniques in artificial intelligence that allow people to comprehend and trust the results and operations of AI models. As AI systems increasingly influence critical areas like healthcare, finance, and education, their decision-making processes must be transparent. XAI aims to provide insights into the "why" and "how" of AI decisions, making these systems interpretable to users while ensuring their fairness, reliability, and ethical compliance.

Counterfactual explanations are a specific approach within XAI, offering "what if" scenarios to explain AI decisions. Explanations that describe instances leading to an alternative outcome than the provided one are termed counterfactuals. For instance, a counterfactual explanation for a rejected loan application might state: *"Your loan application was rejected, but it would have been approved if you had fewer delinquent credit lines."* These explanations not only clarify why a certain decision was made but also suggest what actionable changes can be undertaken for a different outcome. Counterfactuals are particularly valuable for their simplicity and intuitive appeal, often the type of explanation that many people find most natural and intuitive. However, it is essential to evaluate the feasibility of realizing a counterfactual, indicating that the alteration would be attainable by a change made by an individual.

Achievable Minimally-Contrastive Counterfactual Explanations (AMCC) [1] aims to enhance explainable AI by providing actionable, feasible, and minimal changes that could lead to a different decision or outcome from a black-box model. Unlike traditional counterfactual explanations that may suggest impractical or unrealistic changes, AMCC incorporates domain-specific constraints to ensure the proposed adjustments are achievable and relevant to the user. By focusing on minimally invasive alterations that satisfy these constraints, AMCC seeks to answer both "why" and "what can be done" questions in real-time, making it suitable for dynamic, interactive decision-support systems. However, despite the effectiveness of AMCC in providing counterfactuals, its explanations are currently presented in a form that is not easy for non-technical audiences to grasp. To bridge the gap between technical explanations and easier human understanding, this research aims to augment the AMCC algorithm by generating simple natural language explanations.

Natural language explanations of counterfactuals make AI-generated statements more accessible and intuitive for users by presenting information in a conversational and contextually meaningful way. Instead of merely stating a feature and the required action (e.g., "Increase loan amount by $5,000"), natural language can frame the explanation in terms of cause and effect (e.g., "If your loan amount were $5,000 higher, your application would be approved"). This human-centered approach not only makes the reasoning process behind the AI's decision more transparent but also helps users better understand the practical steps they need to take, fostering trust and facilitating actionable change.

This thesis proposes and compares two natural language generation (NLG) solutions for counterfactual explanations, focusing on producing effective and contextually meaningful statements that are easy for users to understand. Designed to be domain-independent, such solutions can seamlessly adapt to a new domain while ensuring that the generated statements align with user needs. The approaches selected for comparison were picked for their potential computational efficiency, aiming for a generation time of under one or two seconds, aligning with human-computer interaction standards for responsiveness. This ensures that users receive timely, actionable explanations without compromising their experience.

The first approach uses a template-*based* method to generate natural language explanations. A template is a fixed span of text with variables that can be replaced by task-specific values. To achieve variety, paraphrases for these templates are generated using a new method that is domain-independent, with slot values augmented by synonyms from WordNet [2], a lexical database that organizes words into synonym sets, providing a rich resource for varied word choices. To further streamline and enhance the selection process, a sub-approach leveraging Masked Language Modeling (MLM) is introduced. MLM, a technique which uses pre-trained language models such as BERT [3], predicts the most likely replacements for masked tokens in a sentence based on their surrounding context. This integration of templates, WordNet, and MLM achieves a balance between structured generation and linguistic diversity, ensuring flexibility and efficiency in producing natural language explanations.

The second approach uses a template-*guided* method to generate natural language explanations augmented with a generative Large Language Model. It employs a fine-tuned T5 model [4] that takes template-generated ‘simple explanations’ as inputs, where each ‘simple explanation’ encapsulates a concise, actionable recourse (e.g., "*Decrease delinquent credit lines to 3*"). Leveraging the T5 model's learned generalization capabilities, this approach ensures that the generated explanations are coherent while introducing linguistic diversity. The reliance on templates simplifies the representation of structured inputs, while the model’s adaptability ensures seamless generalization to new domains.

Due to the absence of a dataset specifically designed for the task of generating natural language explanations, a small synthetic dataset was created. This dataset consists of reference gold-standard sentences against which the generated explanations can be evaluated. To generate the synthetic dataset, the capabilities of Large Language Models were leveraged. The generated sentences were subsequently manually reviewed to ensure both factual and grammatical correctness, providing a reliable benchmark for comparison and evaluation.

1.1 Problem Statement

This thesis investigates two generation approaches—template-based and template-guided—to evaluate which produces more effective real-time explanations while maintaining coherence and diversity. Furthermore, it explores how these methods can be adapted to new domains to ensure generalizability without compromising user experience or generation efficiency.

2. BACKGROUND

2.1 Achievable Minimally-Contrastive Counterfactual Algorithm

The Achievable Minimally-Contrastive Counterfactual (AMCC) algorithm [1] offers a novel approach to generating feasible, actionable counterfactual explanations that are domain-specific and implementable in real-time. Addressing a critical need in explainable AI, AMCC identifies the minimal changes required to alter a black-box AI model's outcome while ensuring these changes are practical and actionable. This enhances system interpretability and empowers users to evaluate strategies for achieving desired outcomes.

AMCC operates through a two-step process. First, it generates high-precision explanations for a given prediction by isolating key features influencing the outcome. Second, it searches for achievable modifications to these features, guided by domain-specific constraints, ensuring that the generated counterfactuals are meaningful and feasible. By incorporating domain knowledge, such as practical constraints on feature changes (e.g., *reducing debt-to-income ratio rather than increasing it*), the algorithm guarantees actionable recommendations.

To meet real-time application demands, AMCC employs a Breadth-First Search based approach to explore modifications efficiently. The algorithm is optimized to identify actionable counterfactuals typically within 200 milliseconds, making it well-suited for interactive systems. Empirical evaluations demonstrate a 47% success rate in generating counterfactuals for a representative dataset, with an average computation time of 80 milliseconds for successful cases. Its ability to focus on actionable features within strict time limits ensures practicality for decision-support applications in domains such as finance, healthcare, and education.

2.2 Template-based Natural Language Generation approach

The template-based NLG approach for generating counterfactual explanations builds on the methodologies proposed by Salmi et.al [5]. It employs taxonomy-guided templates (see Figure 1) to create clear and actionable explanations that are both domain-independent and easily adaptable to diverse contexts.

A key component of this approach is the Feature Actionability Taxonomy (FAT), which categorizes features based on their mutability and actionability. FAT defines four categories:

1. **Mutable Directly:** Features that users can change directly, such as reducing loan amounts.
2. **Mutable Indirectly:** Features influenced by changes to other features, such as increasing discretionary income by reducing expenses.
3. **Immutable Non-sensitive:** Features that cannot be changed but are not ethically sensitive, such as existing loan terms.
4. **Immutable Sensitive:** Features like race or gender that are ethically sensitive and must be presented with caution.

The actionable recourse suggested by AMCC primarily falls under the *Mutable Directly* and *Mutable Indirectly* categories, ensuring both feasibility and user applicability. This taxonomy-guided approach enables systematic selection of templates tailored to each feature's actionability, ensuring explanations are both meaningful and actionable.

<table>
<tr><th colspan="2">Template Variables with Synonym Examples</th></tr>
<tr><td colspan="2">VERB={Take|Initiate|Undertake|Pursue|Negotiate}
OBJECT={steps|measures|actions|}
ACTION={Pos: (increase|improve|raise)| Neg:(decrease|reduce)}
COMPARATIVE={Pos: (increase|higher|better) | Neg: (decrease|lower|worse)}
OUTCOME={undesired:(rejected|fail) | desired:(accepted|pass)}
FEATURE= feature name in dataset, QUERY_VALUE= feature value from query, CF_VALUE= feature value from counterfactual, POSSESSIVE={Your}</td></tr>
<tr><th>Actionability Category</th><th>Feature Sentence Template</th></tr>
<tr><td>Mutable Directly</td><td>1. {ACTION} {FEATURE} from {QUERY_VALUE} value to {CF_VALUE}
2. {ACTION} {FEATURE} to {CF_VALUE}</td></tr>
<tr><td>Mutable Indirectly</td><td>1.{VERB} {OBJECT} to {ACTION} {FEATURE} from {QUERY_VALUE} to {CF_VALUE}
2.{VERB} {OBJECT} to {ACTION} {FEATURE} to {CF_VALUE}</td></tr>
<tr><td>Immutable Non-sensitive</td><td>Having a value of {CF_VALUE} for {FEATURE} would provide a {COMPARATIVE} chance of {DESIRED_OUTCOME} compared to a value of {QUERY_VALUE}</td></tr>
<tr><td>Immutable Sensitive</td><td>{POSSESSIVE} {FEATURE} has contributed to {OUTCOME}</td></tr>
</table>

Figure 1. Templates and Feature Actionability categories proposed by Salmi et al. [6].

The templates were developed through comprehensive user study and thematic analysis, integrating content and structure-related themes to provide actionable guidance. These templates employ clear action verbs (e.g., “increase,” “reduce”), comparative adjectives (e.g., “higher,” “lower”), and structured constructs to enhance readability and comprehension. Including both query and counterfactual values ensures explanations remain contextually relevant.

The taxonomy-guided template approach is inherently domain-independent, leveraging the universality of the FAT framework. By aligning feature actionability categories with domain-specific requirements, the templates can seamlessly adapt to various contexts, such as finance, healthcare, and education. For instance, in finance, templates might suggest reducing credit utilization, while in healthcare, they might propose lifestyle changes to lower blood pressure. This modular design ensures scalability and transferability without significant redesign efforts.

By grounding the NLG process in structured templates and actionability taxonomy, this approach enhances the interpretability and practicality of AMCC explanations, making them more accessible and actionable for users across domains.

2.2.1 WordNet

WordNet [2] is a comprehensive lexical database of the English language, developed at Princeton University. It organizes words into sets of synonyms, called synsets, and connects them through semantic relationships such as hypernymy (generalization) and hyponymy (specialization). These connections provide a deeper understanding of word meanings and their relationships.

As of version 3.0, WordNet has a total of 117,659 synsets, covering 155,287 unique words/strings. This includes 82,115 synsets for nouns and verbs 13,767 synsets.

WordNet serves as a valuable resource for introducing lexical variety. By leveraging its extensive synsets, synonyms for specific template values can be identified, ensuring that generated explanations are varied and avoid rigidity. This variability enhances the naturalness and readability of the output, making it more engaging for users.

For instance, if a template includes the phrase "increase your income," WordNet can suggest alternatives like "raise your income" or "boost your income," providing diverse expressions of the same concept. This approach not only enriches the language but also helps in tailoring the explanations to different user preferences and contexts.

2.2.2 Relexicalization Approach Using Contextual Word Embeddings

The relexicalization approach proposed by Rämö et al. [6] introduces variations in generated explanations by replacing existing template words with synonyms identified through a combination of WordNet and contextual embeddings. This method leverages a high-resource setup to enhance language diversity while preserving the intended meaning of the explanations.

Contextual embeddings, particularly from models like BERT, play a pivotal role in identifying the most suitable synonym for a given context. BERT’s masked language modeling capability evaluates the contextual fit of candidate synonyms, scoring them based on how well they integrate into the sentence. This ensures that selected synonyms align semantically and syntactically with the surrounding text, enhancing lexical variety while maintaining grammatical correctness and avoiding mismatches.

2.2.3 Lexical Substitution

Lexical substitution is the task of generating plausible replacements for a target word within a given context. In their study, Arefyev et al. [21] conduct a comprehensive comparison of lexical substitution methods, focusing on neural language models (LMs) and masked language models (MLMs). Their research investigates the performance of these models in lexical substitution and explores whether injecting additional information about the target word can enhance substitution quality.

To improve lexical substitution, the authors experiment with various target word injection methods. One of the key approaches is the +embs method, which refines substitute ranking by combining contextual substitute probabilities with the proximity of candidate words to the target word in the embedding space. Another notable approach is duplicate input, where the sentence is duplicated while omitting the target word in one instance, enabling the model to infer substitutes based on repeated context. The study finds that both +embs and duplicate input significantly enhance lexical substitution, with duplicate input proving particularly effective for transformer-based models such as BERT, RoBERTa, and XLNet.

Given these findings, the duplicate input method aligns well with the template-based NLG approach. It offers performance comparable to +embs while being simpler and more straightforward to integrate. Transformer-based models naturally excel at leveraging contextual cues, and the duplicate input method capitalizes on this strength without requiring additional computational steps or modifications. This makes it a practical and efficient choice for seamless integration into the existing framework while ensuring high-quality lexical substitutions. Comparison of scores with and without duplicate input methods is provided in Appendix E.

2.2.4 BERT

BERT (Bidirectional Encoder Representations from Transformers) [3] is a state-of-the-art language model developed by Google Research. It uses a masked language model (MLM) training objective, allowing it to predict missing words in a sentence based on the surrounding context. This bidirectional approach enables BERT to capture rich semantic and syntactic relationships, making it ideal for tasks requiring deep contextual understanding, such as synonym selection.

Despite its power, BERT's large size and computational demands can pose challenges for real-time applications. Two notable variants address these limitations:

1. RoBERTa (Robustly Optimized BERT) [7] enhances BERT by training on larger datasets, removing the next-sentence prediction objective, and fine-tuning hyperparameters for improved performance. It is highly accurate and particularly suited for scenarios demanding precise contextual understanding. Variants include RoBERTa-base and RoBERTa-large, optimized for tasks requiring high accuracy.

2. DistilRoBERTa [23] is a lighter and faster variant of the RoBERTa-base model, trained using the same distillation process as DistilBERT [8]. While being 35% smaller and twice as fast on average, it retains most of RoBERTa’s language understanding capabilities. This makes it particularly effective for resource-constrained environments or applications requiring rapid inference.

2.3 Template-guided Neural NLG Approach

The template-guided neural natural language generation (NLG) approach draws inspiration from the methodology proposed by Kale et.al [9]. This method combines the precision of template-based generation with the flexibility of neural models, creating a robust system for generating contextually appropriate and natural utterances.

This approach operates in two steps:

1. Template-Based Generation: A small set of manually defined templates generates semantically correct but potentially unrefined utterances for specific system actions. These templates are simple to create, requiring one for every action, and ensure that all essential information is included in the output.
2. Neural Rewriting: The template-generated utterances are fed into a pre-trained T5 encoder-decoder model, which rewrites them into fluent and conversational responses. The T5 model excels at transforming fragmented template outputs into coherent sentences that sound natural to users.

The authors demonstrate that this approach achieves strong zero-shot generalization capabilities, with a BLEU score improvement of 7.3 points over baseline models for unseen domains. Human judges rated the model's responses highly in terms of informativeness and naturalness, often surpassing human-authored ground truth in naturalness. Additionally, the method drastically reduces the need for labeled data. For instance, the Template Guided Text Generation model in an 80-shot setting (80 dialogs per domain) outperformed baseline models trained on the full dataset, showcasing its data efficiency.

The template-guided approach is particularly well-suited for new, unseen domains. Since templates are minimal and easy to create, domain experts can quickly define them for new domains. The neural rewriting step ensures that even with basic templates, the output remains fluent and user-friendly. This combination of simplicity and adaptability makes the approach highly effective for scalable NLG systems.

2.3.1 T5

T5 (Text-to-Text Transfer Transformer) [4] is a versatile pre-trained sequence-to-sequence transformer model developed by Google Research. It reframes all natural language processing tasks into a unified text-to-text format, enabling consistent and flexible solutions for tasks such as translation, summarization, and question answering. With its encoder-decoder architecture, T5 processes input text into context-aware representations and generate high-quality outputs, making it an excellent choice for rewriting tasks, as required in the template-guided approach.

T5-Flan, a fine-tuned version of T5, [10] incorporates instruction tuning, which enhances its ability to follow complex instructions and generate precise, nuanced outputs. Pre-trained in a wide range of tasks and optimized for understanding task-specific instructions, T5-Flan is particularly well-suited for generating coherent and natural language responses, even with minimal templates or in unseen domains. This makes it a powerful tool for the neural rewriting step in template-guided natural language generation.

T5-Flan is available in 4 sizes, catering to different computational resources and performance requirements.

2.3.2 PEFT

Parameter-Efficient Fine-Tuning (PEFT) is a framework designed to adapt pre-trained models to specific tasks with minimal computational and memory overhead. Instead of updating all the parameters of a large model during fine-tuning, PEFT techniques focus on modifying a small subset of parameters or introducing lightweight auxiliary components. This approach is particularly advantageous in resource-constrained scenarios or when handling multiple tasks, as it reduces both training time and storage requirements.

Low-Rank Adaptation (LoRA) [11] is a parameter-efficient fine-tuning technique that enables the adaptation of large pre-trained models to specific tasks while significantly reducing computational and memory requirements. Rather than updating all model parameters during fine-tuning, LoRA introduces trainable low-rank matrices that approximate the necessary weight adjustments, substantially decreasing the number of parameters that need to be optimized. This approach preserves the original pre-trained weights in a frozen state while integrating the low-rank matrices into specific layers, capturing task-specific nuances without modifying the entire parameter set. By allowing different low-rank matrices to be learned for different tasks and merged with the base model during inference, LoRA enables the same pre-trained model to be efficiently adapted to multiple applications without requiring full re-training.

Its efficiency makes it feasible to fine-tune large models, even those with billions of parameters, on consumer-grade hardware, making it particularly beneficial in resource-constrained environments. Additionally, LoRA is especially advantageous when working with small datasets containing a limited number of samples, as it mitigates the risk of overfitting and enhances the model's ability to generalize effectively.

2.4 Python Packages

2.4.1 HuggingFace Transformers

HuggingFace Transformers [12] is a widely-used library for natural language processing tasks, providing pre-trained models and tools for tasks like text classification, translation, and summarization. It supports a variety of transformer architectures, including BERT, T5, and GPT, and simplifies the fine-tuning and deployment of state-of-the-art models for diverse applications.

2.4.2 PyTorch

PyTorch [13] is an open-source deep learning framework that provides flexibility and speed for building and training machine learning models. Its dynamic computation graph and support for GPU acceleration make it particularly well-suited for research and production applications.

2.4.3 NLTK

The Natural Language Toolkit (NLTK) [14] is a comprehensive library for natural language processing in Python. It includes tools for text preprocessing, tokenization, stemming, and more, making it a valuable resource for building language-based applications. Additionally, NLTK provides access to WordNet, a lexical database of English. Through NLTK, users can perform operations like finding synonyms, antonyms, hypernyms, and hyponyms, as well as exploring word relationships and meanings efficiently.

2.4.4 CUDA

CUDA (Compute Unified Device Architecture) [15] is a parallel computing platform and API developed by NVIDIA. It enables developers to use NVIDIA GPUs for general-purpose processing, significantly accelerating computational tasks in fields such as machine learning and scientific computing.

2.4.5 BLEURT

BLEURT (Bilingual Evaluation Understudy with Representations from Transformers) [18] is a learned evaluation metric for natural language generation (NLG) that leverages pre-trained transformer models, such as BERT, to assess the quality of generated text by comparing it to reference outputs. Unlike traditional metrics like BLEU, which rely on exact n-gram matches, BLEURT captures nuanced semantic similarities between sentences, offering closer alignment with human judgment. It is developed through multiple stages of transfer learning—starting with a pre-trained BERT model, followed by additional pre-training on synthetic data, and fine-tuning on human-annotated examples.

To enable evaluation in PyTorch-based workflows, a PyTorch port of BLEURT [22] is employed, given the original implementation's tight integration with TensorFlow. Alignment tests indicate that the average difference in scores between the two implementations is just 0.036%, with a maximum deviation of 0.592% for a single sample pair. These results suggest the PyTorch port is sufficiently reliable for evaluation tasks. Further details on the alignment tests are provided in Appendix C.

3. METHODS

3.1 Input Dataset and Counterfactual Generation

The dataset used for testing is the Home Equity Loan (HMEQ) dataset [16], as utilized by the authors in the AMCC paper. This dataset is from the domain of lending and includes 5960 instances of individual home equity loans, where approximately 20% of cases (1189 instances) are classified as "bad loans," representing defaults or delays in repayment. The dataset comprises 12 input variables, with 7 identified as actionable as shown in Table 1 in the AMCC paper. As in the AMCC paper, the recourse action for the features, MORTDUE, DELINQ, NINQ and DEBTINC, were restricted to only a decrease in value.

Table 1. Actionable features including recourse actions.

Feature	Description	Example Recourse Actions
LOAN	the amount of the loan request	increase (or reduce) the loan request
MORTDUE	the amount due on the existing mortgage	pay down the existing mortgage
VALUE	the current value of property	have the property reappraised
DELINQ	the number of delinquent credit lines	pay off delinquent credit
NINQ	the number of recent credit inquiries	reduce credit applications
CLNO	the total number of credit lines	increase (or reduce) open credit lines
DEBTINC	the client’s current debt-to-income ratio	pay off some debt

The dataset's continuous numeric features were discretized into four broad categories, Table 2. This means that the generated natural language counterfactual explanations proposed actions to change actionable feature value from one range to another acceptable range. The features DELINQ and NINQ are categorical, and their original values were used without modification.

Table 2. The value ranges for each feature

Feature	**Bins**	**Actual Ranges**
LOAN	['Q1', 'Q2', 'Q3', 'Q4'],	[(1100, 11100), (11101, 16300), (16301, 23300), (23301, 89900)]
MORTDUE	['Q1', 'Q2', 'Q3', 'Q4'],	[(2060, 46300), (46301, 65000), (65001, 91500), (91501, 241000)]
VALUE	['Q1', 'Q2', 'Q3', 'Q4'],	[(8000, 66100), (66101, 89200), (89201, 120000), (120001, 347000)]
CLNO	['Q1', 'Q2', 'Q3', 'Q4'],	[(0, 15), (16, 20), (21, 26), (27, 71)]
DEBTINC	['Q1', 'Q2', 'Q3', 'Q4']	[(0.52, 29.1), (29.11, 34.8), (34.81, 39), (39.01, 61)]

The output CSV file containing counterfactual explanations was modified to include high-precision features identified during the AMCC algorithm. These features were used in the natural language explanations to make it more accessible and diverse, Table 3.

Table 3. Sample modified output from the AMCC algorithm

Time Taken	Original Instances	Modified Instances	Counterfactual changes	Other Candidate Features
0.059430599	[0.0, 1.0, 1.0, 1.0, 2.0, 3.0, 0.0, 0.0, 1.0, 0.0, 1.0, 1.0]	[0.0, 1.0, 1.0, 1.0, 2.0, 3.0, 0.0, 0.0, 1.0, 0.0, 1.0, 0.0]	{'DEBTINC': (1.0, 0.0)}	['DEBTINC', 'LOAN', 'VALUE', 'CLNO']
0.062189341	[3.0, 3.0, 3.0, 0.0, 1.0, 2.0, 0.0, 0.0, 1.0, 10.0, 3.0, 1.0]	[3.0, 3.0, 3.0, 0.0, 1.0, 2.0, 0.0, 0.0, 1.0, 0.0, 3.0, 1.0]	{'NINQ': (10.0, 0.0)}	['NINQ', 'DEBTINC', 'CLNO', 'MORTDUE', 'LOAN', 'VALUE']
0.453100204	[3.0, 2.0, 3.0, 1.0, 5.0, 2.0, 1.0, 7.0, 0.0, 9.0, 0.0, 1.0]	[3.0, 2.0, 3.0, 1.0, 5.0, 2.0, 1.0, 0.0, 0.0, 9.0, 0.0, 0.0]	{'DELINQ': (7.0, 0.0), 'DEBTINC': (1.0, 0.0)}	['DELINQ', 'DEBTINC', 'NINQ']
0.525336504	[3.0, 1.0, 2.0, 0.0, 5.0, 0.0, 0.0, 1.0, 0.0, 12.0, 3.0, 1.0]	[3.0, 1.0, 2.0, 0.0, 5.0, 0.0, 0.0, 1.0, 0.0, 0.0, 3.0, 0.0]	{'NINQ': (12.0, 0.0), 'DEBTINC': (1.0, 0.0)}	['NINQ', 'DELINQ', 'DEBTINC']

3.2 Synthetic Dataset

Since no equivalent dataset exists for training, a small synthetic dataset was created. The ‘simple explanation’ templates (Table 6) were used to generate a unique set of simple explanations, which was then paired with OpenAI's GPT-4o model [17] to produce natural language explanations. The simple explanations generated using ‘simple explanation’ templates were programmatically generated, by choosing the counterfactual feature and values and other candidate features at random to ensure no duplicates were introduced. These generated explanations were manually verified to ensure they were correct, coherent, and diverse. Care was taken to ensure that all possible actionable recourse options were included in each configuration to ensure coverage of all actionable recourse options.

Synthetic samples were created to form a training set, validation set, and a test set. The training and validation sets were used to fine-tune the T5-Flan model for the template-guided approach, while the test set was used to compare the performance of the two approaches.

3.3 Template-based NLG approach

This approach employs taxonomy-guided templates developed by Salmi et al. [5], which were designed using the Feature Actionability Taxonomy (FAT) to ensure domain independence, while maintaining a focus on actionable recommendations. Figure 2 shows an overview of the Template-based approach.

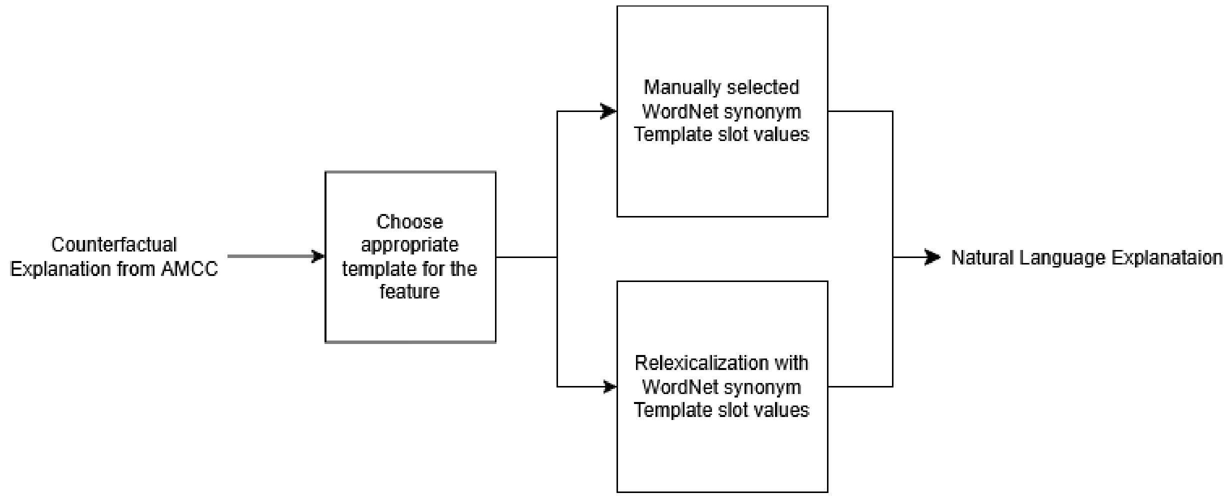


Figure 2. Overview of experiments for Template-based NLG

The taxonomy-guided templates categorize templates based on feature actionability, as Mutably Direct and Mutably Indirect. By mapping the actionable features of the loan approval dataset to these templates, domain-specific nuances in loan-related decision-making are effectively addressed (see Table 4).

Table 4. Feature Categorization

Mutably Direct Features	LOAN, CLNO, DELINQ, DEBTINC
Mutably Indirect Features	VALUE, MORTDUE, NINQ

To enhance template diversity in generated natural language explanations, OpenAI's ChatGPT 4o model [17] was utilized to generate and explore additional template examples. Templates for generating natural language explanations for 'Other Candidate Features' were also designed using this method. The design of these templates was done ensuring reuse of the same template slots from the taxonomy-guided templates. Slot values from the original taxonomy-guided templates are maintained, with additional logic added to accommodate feature specific slot values. For example, features like MORTDUE and VALUE can have specific action values 'pay down' and 'reappraise' respectively.

Like the taxonomy-guided templates, the new templates were designed in two variants: one containing only the counterfactual value and another including both the original (query) value and the counterfactual value. This approach promotes variability in the natural language explanations generated from a single template. For other candidate features, two templates were designed to ensure sufficient diversity, leveraging combinations of different features to enhance the variety of generated explanations.

The newly generated templates were manually reviewed to ensure they were domain-independent and checked for grammatical and syntactical accuracy. For example, templates containing present continuous verbs were excluded as they could interfere with the flexibility of filling other template slot values.

3.3.1 Expanding Slot values using Synonyms from WordNet

To enhance the diversity of explanations, slot values from the taxonomy-guided templates were augmented with synonyms obtained from WordNet. WordNet provides synsets for a given word, which may vary in sense or meaning depending on whether they belong to the Noun or Verb POS tags. Furthermore, a single POS tag can have multiple meanings. To ensure accuracy and alignment with the intended templates, each synonym was manually checked and verified before being marked as a valid template slot value.

This process increases the flexibility of template-based explanations by expanding the range of possible slot values. However, as the number of options grows significantly with the addition of synonyms, selecting or designing template slot values becomes more complex.

3.3.2 Lexical Substitution with BERT-like models

A pre-trained BERT model was employed to score and select synonyms returned by WordNet in real-time, eliminating the need for manual verification. This is achieved through Masked Language Modeling, a technique where pre-trained language models such as BERT predict the most suitable replacements for a masked token in a sentence based on its surrounding context.

This approach is like the relexicalization method proposed by Rämö et al. [6]. To control the suggestions generated by the MLM, WordNet is used to provide synonym candidates for a given "starter" word. This method prevents the erroneous suggestion of antonyms, which could otherwise distort the intended counterfactual explanation. Since the task of masked language modelling is to predict a token in place of the masked token, it isn't trained to predict multiple words for a multi-token input. Therefore, predictions for lexical substitution will only be performed for the action verb slots as this is the slot that conveys the meaning of the proposed recourse action.

To improve the selection by the BERT model, the 'duplicate input' method proposed by Arefyev et al. [21] was used. This involves appending the sentence with the masked token with a sentence where the masked token is replaced by the ideal candidate word. An example is shown below:

<MASK> the loan request amount from $16,301 - $23,300 to $11,101 - $16,300. Reduce the loan request amount from $16,301 - $23,300 to $11,101 - $16,300.

The final target synonym was chosen by randomly choosing from the top 5 words with the highest probability scores returned by the model. Additional filtering was applied by setting a minimum probability threshold, where candidates scoring below this threshold were discarded. Tests were done with 3 different threshold values, 0, 1e-3 and 1e-6.

Experiments were conducted using both RoBERTa, base and large variants, and DistilRoBERTa models. RoBERTa employs a byte-level Byte-Pair Encoding as a tokenizer and when encountering an out-of-vocabulary word, the tokenizer breaks it down into sub-word units. The character 'Ġ' denotes the start of a new word or word fragment. Table 5 illustrates a few examples of sub-word tokenization.

Table 5. Sub-word tokenization examples

Word	**RoBERTa tokenizer**
adjust	'Ġadjust'
valuate	'Ġval', 'uate'
appraise	'Ġappra', 'ise'
reexamine	'Ġre', 'ex', 'amine'
reevaluate	'Ġre', 'evaluate'

In Masked LM, the model returns probabilities at the token level. Therefore, to capture whether a word is a good replacement for a slot value, the same number of <MASK> tokens are required as the number of sub-word tokens. The probabilities returned for sub-word tokens for its corresponding <MASK> token indices were aggregated using SoftMax to compute a final score which could be compared with other words with different number of sub-word token lengths. For example, if we are trying to determine the score for 'appraise' as a slot value, the input for both would like,

To improve your chances of loan approval, <MASK> <MASK> the current property's value. To improve your chances of loan approval, appraise the current property's value.

3.4 Template-guided Neural NLG Approach

The second approach, inspired by the template-guided text generation methodology proposed by Kale et al. [9], utilizes a fine-tuned T5 model to generate natural language explanations. This approach takes template-generated ‘simple explanations’ as inputs, where each simple explanation conveys a concise and actionable recourse. Figure 3 shows an overview of the task for the Template-guided approach.

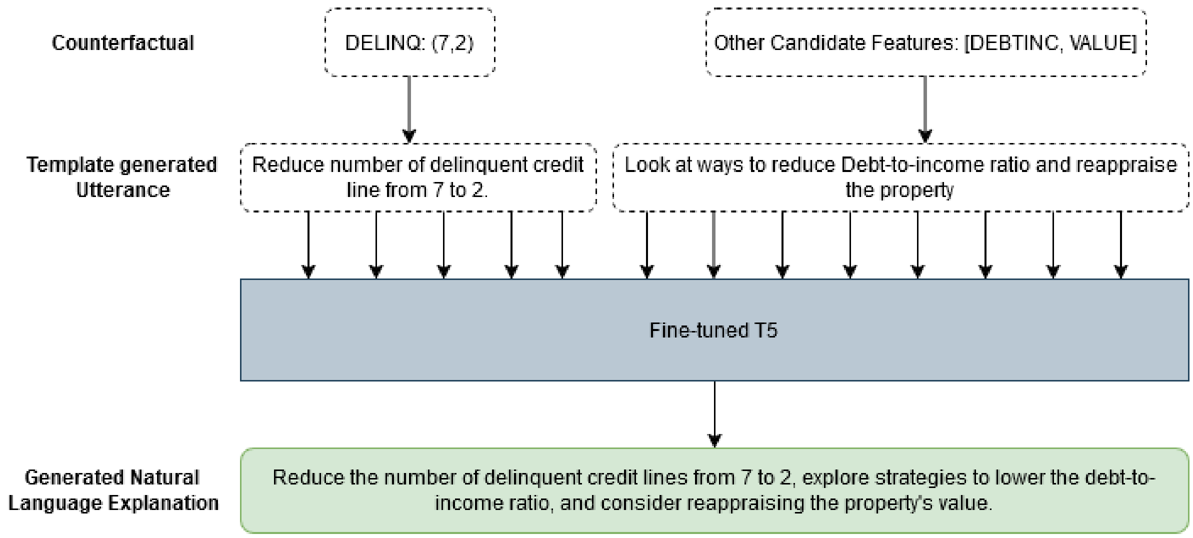


Figure 3. Overview of Template-guided NLG

These simple explanations are generated using a minimal set of manually defined templates (e.g., "Decrease delinquent credit lines to 3"). Two templates were created for each recourse action: one that includes only the counterfactual value, and another that includes both the original (query) value and the counterfactual value. This dual-template strategy was employed to introduce variability in the generated natural language explanations.

The input to the T5 model is obtained by concatenating the templatized representations corresponding to each recourse action. The goal here was not to generate fully formed and grammatically correct natural language explanations but rather provide a simplified representation of the recourse actions, which can be rewritten by the model into a natural and fluent response. As a result, not all edge cases were covered which is typically required in a general template based approaches such as handling of plurals, subject-verb agreement, morphological inflection etc. - and only need to define a small number of templates.

‘Simple explanation’ templates for the features LOAN and DELINQ are shown in Table 6 and Appendix A lists all simple explanation templates.

Table 6. Simple Explanation Templates

Feature	**Simple Explanation Template**
LOAN - Increase	
INC!!LOAN!!@	Increase the loan request amount by $@.
INC!!LOAN!!@@	Increase the loan request amount from $@ to $@.
LOAN - Decrease	
DEC!!LOAN!!@	Decrease the loan request amount by $@.
DEC!!LOAN!!@@	Decrease the loan request amount from $@ to $@.
DELINQ - Decrease	
DEC!!DELINQ!!@	Reduce the number of delinquent credit lines by @.
DEC!!DELINQ!!@@	Reduce the number of delinquent credit lines from @ to @.

The T5-Flan [10] model, pre-trained on a wide range of text-to-text tasks, is then fine-tuned to rewrite these structured inputs into fluent, natural, and contextually diverse explanations. To fine-tune the T5 model, the synthetic training set, validation set, and test set were used for the 3 different variants – small, base and large. LoRA [11], a parameter-efficient fine-tuning technique, was utilized during training. This approach preserves the original parameters and knowledge of the model by freezing them, ensuring they remain unaltered. This method significantly reduces computational, and storage demands, making it feasible to fine-tune large models using a small dataset in resource-constrained environments.

Table 7 lists the different run configurations and the corresponding hyperparameters used to fine-tune the T5-Flan variants. A lower learning rate was used for the large variant considering that it is a large model, and the smaller dataset used for training to mitigate overfitting. All training was done for 500 epochs with early stopping in place if validation loss doesn't decrease for 10 evaluation steps, 1 evaluation step is 5 epochs.

Table 7. Training Run Names and Hyperparameters used

Run Name	**Hyperparameters**
Small	LORA_dropout = 0.05 LORA_layers = 8 learning_rate = 1e-4 batch_size = 16 weight_decay = 0.01
Base I	LORA_dropout = 0.05 LORA_layers = 8 learning_rate = 1e-4 batch_size = 16 weight_decay = 0.01
Base II	LORA_dropout = 0.1 LORA_layers = 8 learning_rate = 1e-4

	batch_size = 16 weight_decay = 0.05
Large I	LORA_dropout = 0.05 LORA_layers = 8 learning_rate = 5e-5 batch_size = 4 weight_decay = 0.01
Large II	LORA_dropout = 0.1 LORA_layers = 16 learning_rate = 2.5e-5 batch_size = 4 weight_decay = 0.05

To evaluate the fine-tuned model's ability to generate varied explanations for the same concatenated counterfactual simple explanation, Top-p (nucleus) sampling was employed. The top_p parameter controls the diversity of generated text by sampling from the smallest set of top probable tokens whose cumulative probability exceeds p. Lower values prioritize more likely words, enhancing coherence, while higher values promote more diverse and creative outputs. Experiments were conducted with top_p values of 0.9, 1.0 (default), and 1.1 to identify the setting that produces the most natural and varied language explanations.

3.5 Evaluation

The synthetic test set was used to evaluate both approaches for performance and average inference time. Performance was measured with BLEURT [18], an automatic evaluation metric. BLEURT goes beyond traditional metrics like BLEU by leveraging pre-trained transformer models to capture semantic nuances and align generated text with reference explanations based on human-like judgment. This enables more accurate assessments of linguistic quality and contextual relevance.

Average inference time was measured to compare the efficiency of the approaches, providing insights into their suitability for real-time applications. Considering the varied nature of the generated natural language explanations, the results were averaged by performing the tests against the test set 5 times.

3.5.1 Domain Adaptability

To test the adaptability of the proposed methods to a different domain, the Students' Academic Performance dataset [20] was utilized. This dataset contains five actionable features, such as class attendance and study resources, that are critical for academic improvement. The AMCC model was applied to this dataset to generate counterfactual explanations tailored to improve the overall grade of a student. This involved training both a Random Forest and an XGBoost classifier to predict whether a student will achieve a high or low overall grade.

Both the template-based and template-guided approaches were adapted, with minimal modifications and a smaller dataset, to produce natural language explanations for these counterfactuals. For the template-guided approach, rather than performing full fine-tuning, the checkpoint with the best prior performance was further trained using a lower learning rate on a synthetically generated small dataset to prevent overfitting. This synthetic dataset, created using the same prompt as detailed in Appendix F, is relatively small and was augmented using backtranslation techniques. Manual verification was conducted to ensure that the generated outputs were both grammatically and factually accurate, with particular attention paid to the quality of back translated samples. The smaller dataset was chosen to test to see how much of the learned knowledge of the fine-tuned checkpoint can be transferred compared to performing full fine-tuning. Backtranslation was performed using MarianMT models [19] for German and French to introduce lexical and syntactic variation.

The effectiveness of the generated explanations was evaluated against synthetically generated test set to assess the performance of each approach.

4. RESULTS

All experiments were performed on a system with 16GB RAM and NVIDIA GT GPU with 6 GB dedicated RAM.

4.1 Synthetic Dataset

The following prompt was used to generate natural language explanations using OpenAI's 4.o model. The programmatically generated simple explanations, created using the 'simple explanation' templates, Table 6, were used as examples to generate references that follow the previously established constraints.

Prompt:

'*I want to generate simple natural language explanation utilizing the counterfactual feature changes suggested in these sentences and the other features that can influence a positive change to improve loan applications.*

Give me an example of how such explanations will look like for these examples.

- *Look at ways to increase the property value from $8,000 - $66,100 to $120,001 - $347,000. Look at ways to reduce the number of recent credit inquiries from 8 to 2. Pay off delinquent credit lines and review the number of open credit lines, to improve the chances of loan approval.*
- *Reduce the number of delinquent credit lines from 7 to 6. Explore ways to pay down the amount due on the existing mortgage from $65,001 - $91,500 to $2,060 - $46,300. Reappraise the value of the property or review the number of open credit lines, to improve the chances of loan approval.*
- *Increase the number of credit lines to 16 - 20. Review the loan request amount and review recent credit inquiries, to improve the chances of loan approval.*'

The generated simple explanations and reference explanations were then paired to create the dataset.

Table 8. Synthetic dataset samples

Template Input	**Reference Sentence**
Look at ways to increase the property value from $8,000 - $66,100 to $120,001 - $347,000. Look at ways to reduce the number of recent credit inquiries from 8 to 2. Pay off delinquent credit lines and review the number of open credit lines, to improve the chances of loan approval.	To increase the chances of loan approval, consider significantly increasing your property value from between $8,000 - $66,100 to around $120,001 - $347,000. Additionally, reduce the number of recent credit inquiries from 8 to about 2. Paying off any delinquent credit lines and reviewing your total number of open credit lines can also help positively influence the loan decision.
Reduce the number of delinquent credit lines from 7 to 6. Explore ways to pay down the amount due on the existing mortgage from $65,001 - $91,500 to $2,060 - $46,300. Reappraise the value of the property or review the number of open credit lines, to improve the chances of loan approval.	Improving your likelihood of loan approval may be achieved by reducing delinquent credit lines from 7 down to 6 and substantially paying down your existing mortgage from a current range of $65,001 - $91,500 to approximately $2,060 - $46,300. Additionally, reappraising your property's value or reviewing your open credit lines can further enhance your application's strength.
Increase the number of credit lines to 16 - 20. Review the loan request amount and review recent credit inquiries, to improve the chances of loan approval.	You can enhance your loan application by increasing your total number of credit lines to between 16 and 20. It's also important to review your requested loan amount and manage recent credit inquiries effectively, as these factors can positively influence the loan approval decision.

Explore ways to pay down the amount due on the existing mortgage from $65,001 - $91,500 to $46,301 - $65,000. Look at ways to increase the property value to $66,101 - $89,200. Review recent credit inquiries and review the loan request amount, to improve the chances of loan approval.	To enhance your chances of loan approval, try to reduce your existing mortgage amount from the current range of $65,001 - $91,500 down to between $46,301 - $65,000. Additionally, consider increasing your property's value to between $66,101 - $89,200. It's also beneficial to review recent credit inquiries and adjust your loan request amount accordingly.
Increase the loan request amount to $11,101 - $16,300. Reappraise the value of the property or reduce the debt-to-income ratio, to improve the chances of loan approval.	Increasing your loan request amount to approximately $11,101 - $16,300 may improve your application's strength. Additionally, consider reappraising the value of your property or reducing your debt-to-income ratio, as these changes can positively influence your loan approval outcome.

Table 9. Number of samples per split

Split	**Number of Samples**
Train	500
Validation	62
Test	63

Table 8 and Table 9 shows some samples from the synthetic dataset and the number of samples per spit respectively. The test set was used to evaluate both performance and inference time, allowing for a comparison between the template-based and template-guided approaches. Appendix F details the results of prompting used to create the synthetic dataset.

4.2 Template-based NLG approach

4.2.1 Template Design

The taxonomy-guided templates are enhanced to include one new pair of templates for Mutually Direct and Mutually Indirect features and a pair of templates for other candidate features. Since other candidate features generally consist of at the max 2 features, the different combinations of features will ensure that generated explanations for the other candidate features will be varied for different samples. Table 10 shows the new templates along with the taxonomy-guided templates. The prompt and the generated examples are provided in Appendix D.

Table 10. Templates

Taxonomy-guided Templates	
Mutably Direct	"{action} {feature_desc} from {query_range} to {cf_range}" "{action} {feature_desc} to {cf_range}"
Mutably Indirect	"{verb} {obj} to {action} {feature_desc} from {query_range} to {cf_range}" "{verb} {obj} to {action} {feature_desc} to {cf_range}"
Newly generated and manually verified Templates	
Mutably Direct	"{feature_desc} needs to {action} to {cf_range}" "{feature_desc} need to {action} from {query_range} to {cf_range}"
Mutably Indirect	"You can {verb} {obj} to {action} {feature_desc} to {cf_range}." "You can {verb} {obj} to {action} {feature_desc} from {query_range} to {cf_range}."

Other Candidate Features	"{Outcome}, {Feat_1}{Connective}{Feat_2}" "{Feat_1}{Connective}{Feat_2}, {Outcome}"
OCF Feature (Feat_1 & Feat_2)	"{OCF_action} {feature_desc}"
OCF Outcome	"to {outcome_verb} loan approval chances"

The templates for Other Candidate Features are nested templates, consisting of the templates OCF Feature and Outcome. The design of these templates was done to ensure reuse of the slots from the taxonomy-guided templates.

Table 11 and Table 12 list the templates values and feature descriptions respectively.

Table 11. Template Slot Values

Template Slot	**Values**
Verb	take, initiate, undertake, pursue, negotiate
Object	steps, measures, actions
Action - positive	increase, improve, raise
Action - negative	decrease, reduce
Action-MORTDUE-negative	pay down
Action-VALUE-positive	reappraise
Action-VALUE-negative	reappraise
OCF Action - general	adjust, review, assess, reassess, address
OCF Action - decrease	reduce, decrease, lower
Outcome Verb	boost, improve, increase, strengthen
Connective	and, or

Table 12. Feature Descriptions for feature_desc slot

Feature	**Description**
LOAN	the loan request amount
MORTDUE	the amount due on existing mortgage
VALUE	the value of current property
DELINQ	the number of delinquent credit lines
NINQ	the number of recent credit inquiries
CLNO	the number of credit lines
DEBTINC	the debt-to-income ratio

4.2.2 Synonym expanded slot values

Synonyms for the slot values from Table 11 were used to enhance the number of options for each slot. WordNet was queried to retrieve all possible synsets for a word, filtered by its POS tag. The final step was to then manually filter out words that don't fit into the templates. Table 13 shows the synonyms retrieved from WordNet for the words 'reduce' and 'improve', Appendix B lists more examples of synonyms suggested by Wordnet. Table 14 shows the expanded slot values after manual selection, words in bold represent newly added synonym slot values, included after manual verification.

Table 13. Synonyms for 'reduce' and 'improve'

Word	**Synonyms**
reduce	**reduce**, cut down, cut back, trim, trim down, trim back, cut, **bring down**, come down, boil down, shrink, scale down, deoxidize, deoxidise, tighten, repress, quash, keep down, subdue, subjugate, decoct, concentrate, dilute, thin, thin out, melt off, lose weight, slim, slenderize, slim down

improve	**better**, **improve**, amend, ameliorate, meliorate

Table 14. Synonyms enhanced slot values

Template Slot	**Values**
Verb	take, **choose**, **select**, **pick out**, **consider**, **look at**, initiate, **start**, undertake, **attempt**, pursue, negotiate
Object	steps, measures, actions
Action - positive	increase, improve, **better**, raise, **change**
Action - negative	decrease, **diminish**, reduce, **cut down**, **cut back**, **cut**, **bring down**, **lower**. **change**
Action-MORTDUE-negative	pay down
Action-VALUE-positive	reappraise
Action-VALUE-negative	reappraise
OCF Action - general	adjust, review, **reexamine**, assess, **evaluate**, reassess, **reevaluate**, address, **change**
OCF Action - decrease	reduce, **cut down**, **cut back**, **cut**, **bring down**, **shrink**, decrease, **diminish**, lower
OCF Action-VALUE	reappraise
Outcome Verb	boost, **advance**, improve, **better**, increase, strengthen
Connective	and, or

Table 15. lists some examples of the natural language explanations generated with the expanded template slot values and Table 16. shows results for evaluation metrics.

Table 15. Examples of natural language explanations generated

Counterfactual	Generated Natural Language Explanations
VALUE: $8,000 - $66,100 -> $120,001 - $347,000 NINQ: 8 -> 2 OCF: DELINQ, CLNO	You can pursue steps to change the value of current property from $8,000 - $66,100 to $120,001 - $347,000 and you can start steps to lower the number of recent credit inquiries from 8 to 2. Reassess the number of delinquent credit lines or reevaluate the number of credit lines, to boost loan approval chances.
DELINQ: 7 -> 6 MORTDUE: $65,001 - $91,500 -> $2,060 - $46,300 OCF: VALUE, CLNO	The number of delinquent credit lines needs to cut back from 7 to 6 and undertake measures to diminish the amount due on existing mortgage from $65,001 - $91,500 to $2,060 - $46,300. To strengthen loan approval chances, adjust the value of current property or address the number of credit lines.
CLNO: 0 - 15 -> 16 - 20 OCF: LOAN, NINQ	The number of credit lines needs to improve to 16 - 20. Adjust the loan request amount or adjust the number of recent credit inquiries, to better loan approval chances.
MORTDUE: $65,001 - $91,500 -> $46,301 - $65,000 VALUE: $8,000 - $66,100 -> $66,101 - $89,200 OCF: NINQ, LOAN	Negotiate steps to bring down the amount due on existing mortgage from $65,001 - $91,500 to $46,301 - $65,000 and you can choose steps to better the value of current property to $66,101 - $89,200. To strengthen loan approval chances, adjust the number of recent credit inquiries and reevaluate the loan request amount.
LOAN: $1,100 - $11,100 -> $11,101 - $16,300 OCF: VALUE, DEBTINC	Improve the loan request amount to $11,101 - $16,300. Reassess the value of current property or review the debt-to-income ratio, to improve loan approval chances.

Table 16. BLEURT Score and Time taken

BLEURT Scores	Time Taken (seconds)
0.69041	0.00002

4.2.3 Lexical Substitution using Masked Language Modelling

Except for the action verbs, all other slot values remain unchanged. A representative set of starter slot values for action verbs is presented in Table 17, from which synonyms will be retrieved and scored.

Table 17. Action Verb slot values

Template Slot	Values
Action - positive	increase, improve, raise, change
Action - negative	decrease, reduce, lower. Change
OCF Action - general	adjust, review, assess, evaluate, reassess, reevaluate, address, change
OCF Action - decrease	reduce, decrease, lower

Table 18. shows the results of the evaluation metrics across different thresholds and Table 19. lists some examples generated by the DistilRoBERTa model with a threshold of 1e-6.

Table 18. BLEURT Scores and Average Time taken (in seconds)

	Threshold=0		**Threshold = 1e-6**		**Threshold = 1e-3**	
Model	**BLEURT Scores**	**Time Taken**	**BLEURT Scores**	**Time Taken**	**BLEURT Scores**	**Time Taken**
RoBERTa-base	0.69143	0.18195	0.69006	0.18375	0.70538	0.18322
RoBERTa-large	0.69174	0.31965	0.69278	0.30978	0.7043	0.3076
DistilRoBERTa	0.69337	0.12797	**0.69393**	**0.12107**	0.69642	0.12396

Table 19. Examples of natural language explanations generated using Lexical Substitution

Counterfactual	**Generated Natural Language Explanations**
VALUE: $8,000 - $66,100 -> $120,001 - $347,000 NINQ: 8 -> 2 OCF: DELINQ, CLNO	You can negotiate measures to raise the value of current property from $8,000 - $66,100 to $120,001 - $347,000 and you can attempt actions to diminish the number of recent credit inquiries from 8 to 2. Cut the number of delinquent credit lines or assess the number of credit lines, to better loan approval chances.
DELINQ: 7 -> 6 MORTDUE: $65,001 - $91,500 -> $2,060 - $46,300 OCF: VALUE, CLNO	Decrease the number of delinquent credit lines from 7 to 6 and attempt actions to decrease the amount due on existing mortgage from $65,001 - $91,500 to $2,060 - $46,300. To strengthen loan approval chances, review the value of current property or evaluate the number of credit lines.
CLNO: 0 - 15 -> 16 - 20 OCF: LOAN, NINQ	The number of credit lines needs to get up to 16 - 20. To boost loan approval chances, review the loan request amount or assess the number of recent credit inquiries.
MORTDUE: $65,001 - $91,500 -> $46,301 - $65,000 VALUE: $8,000 - $66,100 -> $66,101 - $89,200 OCF: NINQ, LOAN	You can choose actions to pay down the amount due on existing mortgage from $65,001 - $91,500 to $46,301 - $65,000 and you can look at steps to increase the value of current property to $66,101 - $89,200. Decrease the number of recent credit inquiries or evaluate the loan request amount, to increase loan approval chances.
LOAN: $1,100 - $11,100 -> $11,101 - $16,300 OCF: VALUE, DEBTINC	The loan request amount needs to increase to $11,101 - $16,300. To advance loan approval chances, review the value of current property or decrease the debt-to-income ratio.

4.3 Template-guided approach

The following graphs show the training and validation loss and BLEURT scores across epochs for all 5 runs.

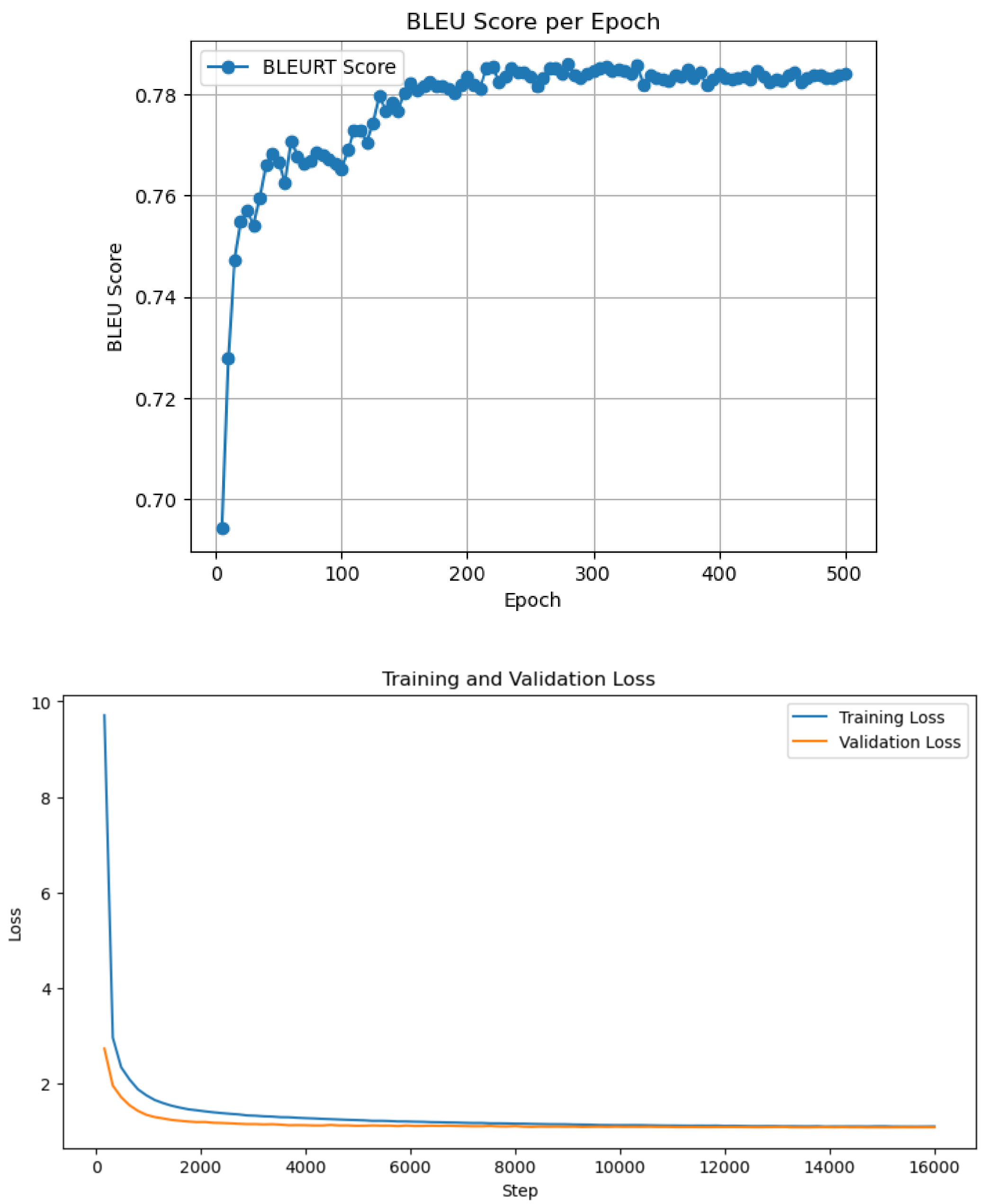
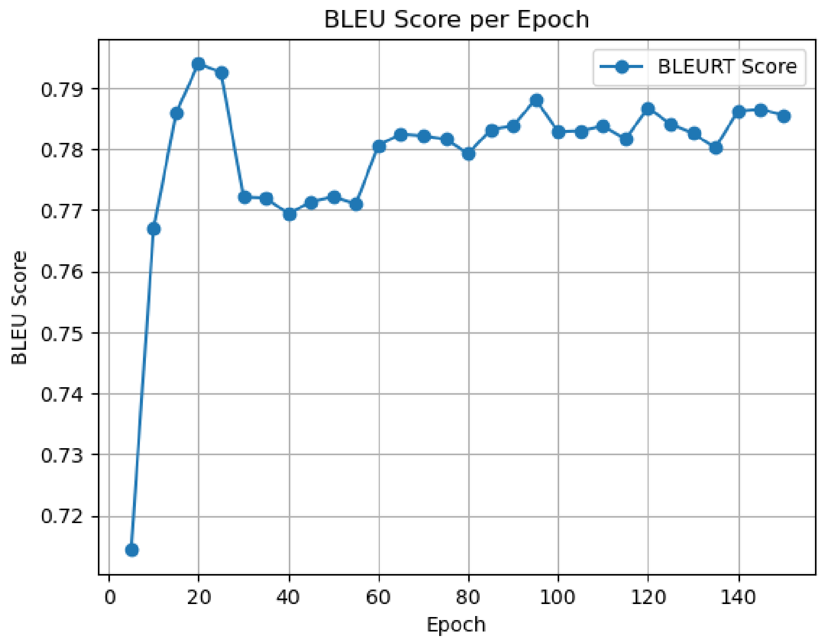


Figure 4. BLEURT Graphs and Training and Validation Loss for the Run ‘Small’

Figure 4 shows the graphs for the training metrics across epochs for Run ‘Small’.



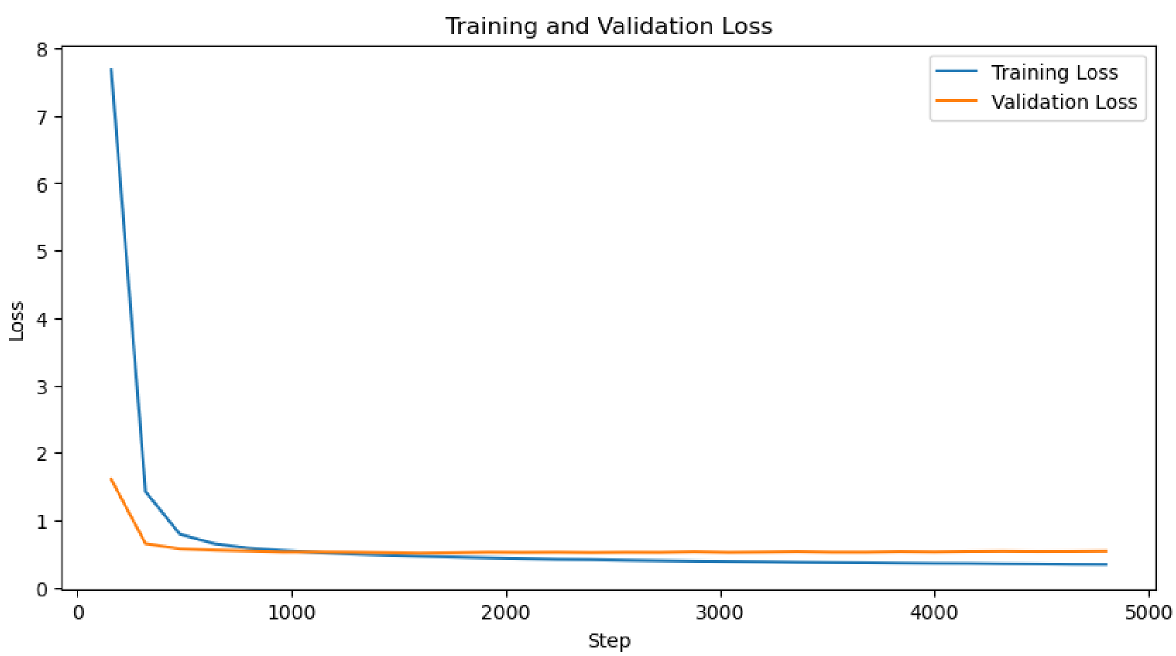


Figure 5. BLEURT Graphs and Training and Validation Loss for the Run ‘Base I’.

Training stopped after 5000 steps when the early stopping condition was met. Figure 5 shows the graphs for the training metrics across epochs for Run ‘Base I’.

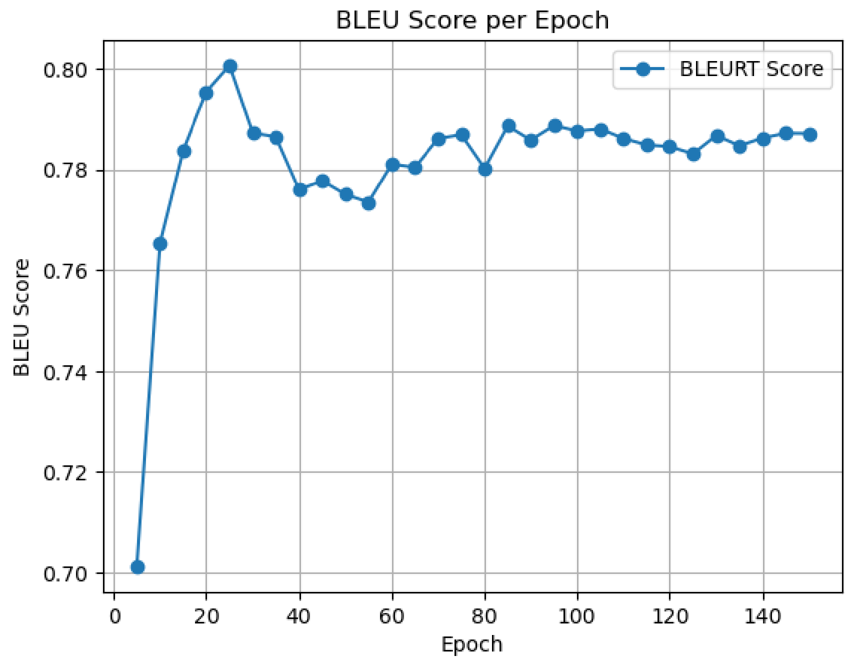
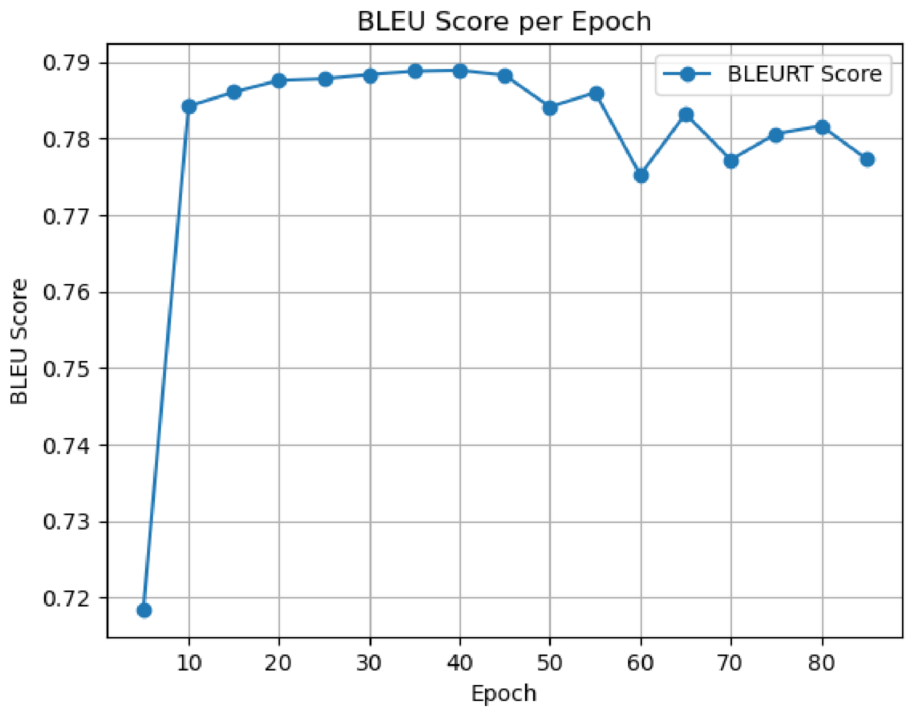


Figure 6. BLEURT Graphs and Training and Validation Loss for the Run 'Base II'.

Training stopped after 5000 steps when the early stopping condition was met. Figure 6 shows the graphs for the training metrics across epochs for Run 'Base II'.



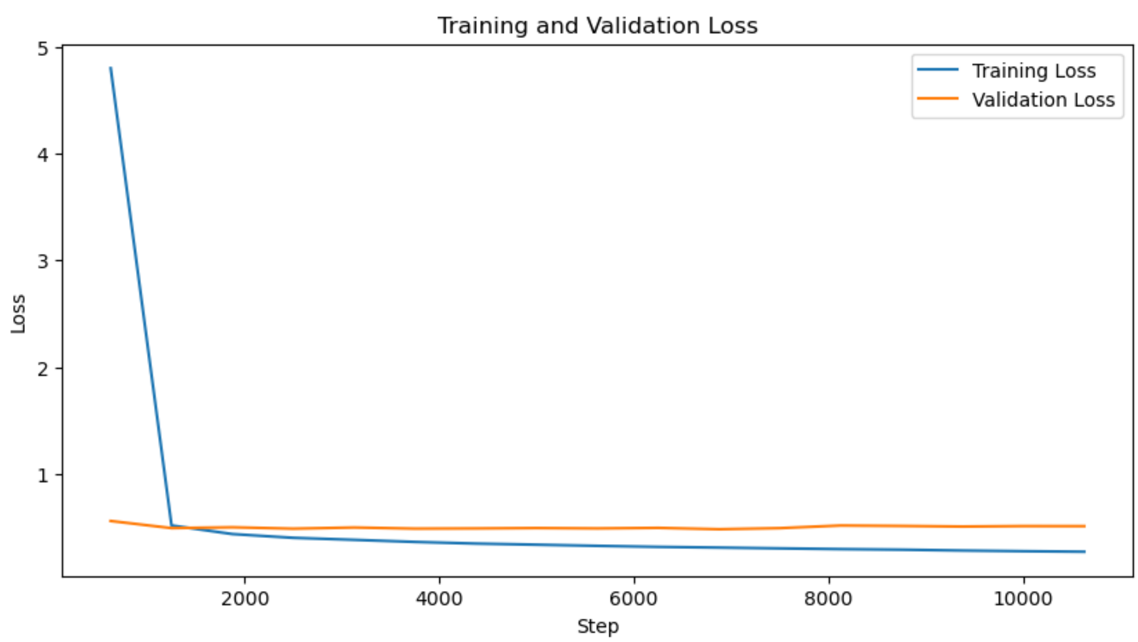


Figure 7.BLEURT Graphs and Training and Validation Loss for the Run ‘Large I’.

Training stopped after 10,000 steps when the early stopping condition was met. Figure 7 shows the graphs for the training metrics across epochs for Run ‘Large I’.

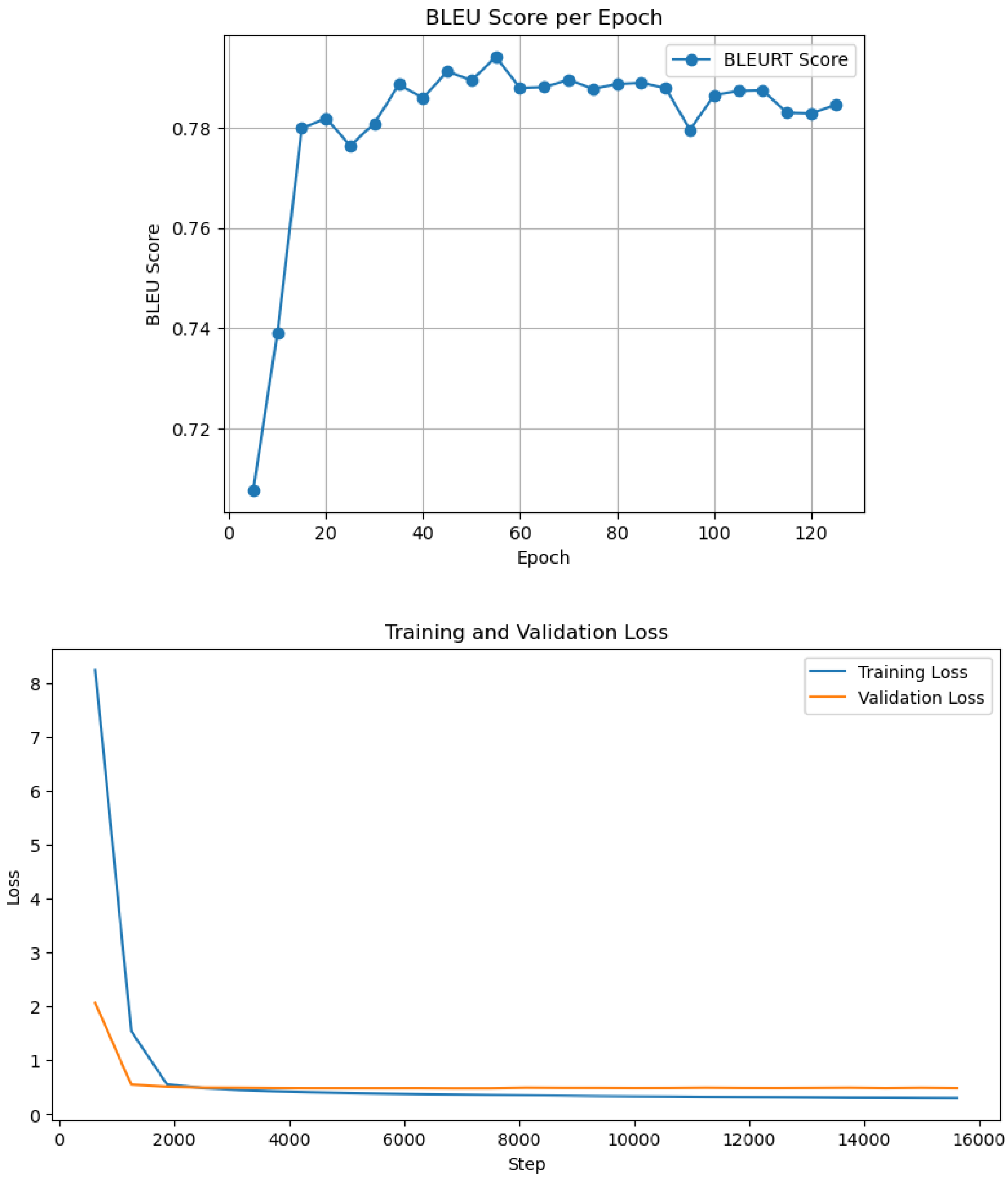


Figure 8. BLEURT Graphs and Training and Validation Loss for the Run 'Large II'.

Training stopped after 15,000 steps when the early stopping condition was met. Figure 8 shows the graphs for the training metrics across epochs for Run 'Large II'.

Table 20. shows the results of the evaluation metrics using the best checkpoints from all the runs and Table 21. lists some examples generated by the best checkpoint from ‘Base Run I’.

Table 20. BLEURT scores and average time taken (in seconds)

Runs	BLEURT Scores	Time Taken
Small	0.74737	1.16761
Base Run I	**0.76972**	**1.58625**
Base Run II	0.76705	1.60761
Large Run I	0.74785	3.08085
Large Run II	0.75249	2.98956

Table 21. Examples of natural language explanations generated using the best checkpoint from Base Run I.

Counterfactual	Generated Natural Language Explanations
VALUE: $8,000 - $66,100 -> $120,001 - $347,000 NINQ: 8 -> 2 OCF: DELINQ, CLNO	Increase property value from $8,000 - $66,100 to $120,001 - $347,000. Reduce recent credit inquiries from 8 to 2, while reducing delinquent credit lines and reviewing open credit lines, will improve loan approval chances.
DELINQ: 7 -> 6 MORTDUE: $65,001 - $91,500 -> $2,060 - $46,300 OCF: VALUE, CLNO	Reducing delinquent credit lines from 7 to 6, while paying down the mortgage balance from $65,001 - $91,500 to $2,060 - $46,300, will improve loan approval chances. Reappraising the property or reviewing open credit lines will further enhance loan approval chances.
CLNO: 0 - 15 -> 16 - 20 OCF: LOAN, NINQ	Increasing credit lines to 16 - 20 while reviewing the loan request amount and reviewing recent credit inquiries will improve your loan approval chances.

MORTDUE: $65,001 - $91,500 -> $46,301 - $65,000 VALUE: $8,000 - $66,100 -> $66,101 - $89,200 OCF: NINQ, LOAN	Paying down the mortgage balance from $65,001 - $91,500 to $46,301 - $65,000, while increasing property value to $66,101 - $89,200, will improve your loan approval chances. Reviewing recent credit inquiries and reviewing the loan request amount will further enhance your application.
LOAN: $1,100 - $11,100 -> $11,101 - $16,300 OCF: VALUE, DEBTINC	Reducing the loan request amount to $11,101 - $16,300, while reassessing the property value or reducing the debt-to-income ratio, will improve loan approval chances.

The evaluation metrics for testing the generation of varied explanations using the best checkpoint from 'Base Run I' is shown in Table 22. Tests were performed with 2 different top_p values, 0.9 and 1.1. Table 23. lists some examples generated with top_p value 0.9.

Table 22. BLEURT scores and time taken (in seconds) for different top p values

Top_p value	**BLEURT Scores**	**Time Taken**
0.9	**0.73130**	**1.64182**
1	0.76972	1.58625
1.1	0.71176	1.67785

Table 23. Examples of generated natural language explanations with top p = 0.9

Counterfactual: VALUE: $8,000 - $66,100 -> $120,001 - $347,000 NINQ: 8 -> 2 OCF: DELINQ, CLNO
Increasing property value from $8,000 - $66,100 to $120,001 - $347,000. Additionally, lowering recent credit inquiries from 8 to 2 and maintaining open credit lines significantly improves creditworthiness.
Increase the property value from $8,000 - $66,100 to $120,001 - $347,000. Additionally, lowering recent credit inquiries from 8 to 2 and reviewing delinquent credit lines and reviewing open credit lines can improve loan approval chances.
Increasing the property value from $8,000 - $66,100 to $120,001 - $347,000. Keeping recent credit inquiries from 8 to 2, and paying off delinquent credit lines, will help your financial profile improve. Additionally, balancing open credit lines and reassessing open credit lines can further improve loan approval chances.
Counterfactual: CLNO: 0 - 15 -> 16 - 20 OCF: LOAN, NINQ
Increase the number of credit lines to 16 - 20 while reviewing the loan request amount and recent credit inquiries to improve your loan application.
Increasing credit lines to 16 - 20, while reducing the loan request amount and reviewing recent credit inquiries, can boost loan approval chances.
Reorganizing credit lines to 16 - 20 and evaluating the loan request amount and recent credit inquiries will help optimize financial eligibility.
Counterfactual: MORTDUE: $65,001 - $91,500 -> $46,301 - $65,000 VALUE: $8,000 - $66,100 -> $66,101 - $89,200 OCF: NINQ, LOAN
Paying down the mortgage balance from $65,001 - $91,500 to $46,301 - $65,000 and increasing the property value to $66,101 - $89,200 will improve loan approval chances. Additionally, reviewing recent credit inquiries and reviewing the loan request amount will further reinforce financial stability.

Lowering the mortgage balance from $65,001 - $91,500 to $46,301 - $65,000, while boosting property value to $66,101 - $89,200, can enhance financial stability. Reviewing recent credit inquiries and examining the loan request amount will further enhance approval prospects.
Removing the mortgage from $65,001 - $91,500 to $46,301 - $65,000, along with increasing property value to $66,101 - $89,200, can improve loan approval chances. Additionally, reviewing recent credit inquiries and the loan request amount will further strengthen loan eligibility.

Table 24. shows the evaluation metrics for all approaches.

Table 24. Comparison of BLEURT scores and time taken for all 3 approaches

Approach	**BLEURT Scores**	**Time Taken**
Template-based without Lexical Substitution	0.69041	0.00002
Template-based with Lexical Substitution (DistilRoBERTa threshold=1e-6)	0.69393	0.12107
Template-guided (Base Run I)	0.76972	1.58625

4.4 Domain Adaptability

The Students' Academic Performance had 478 samples after removing duplicates and missing values. The target feature has three classes for overall grade, lower, middle and high. The middle and high overall grade target class was combined to a high overall grade target class, to change the task to a binary classification task. Figure 9 shows the distribution of the final 2 classes.

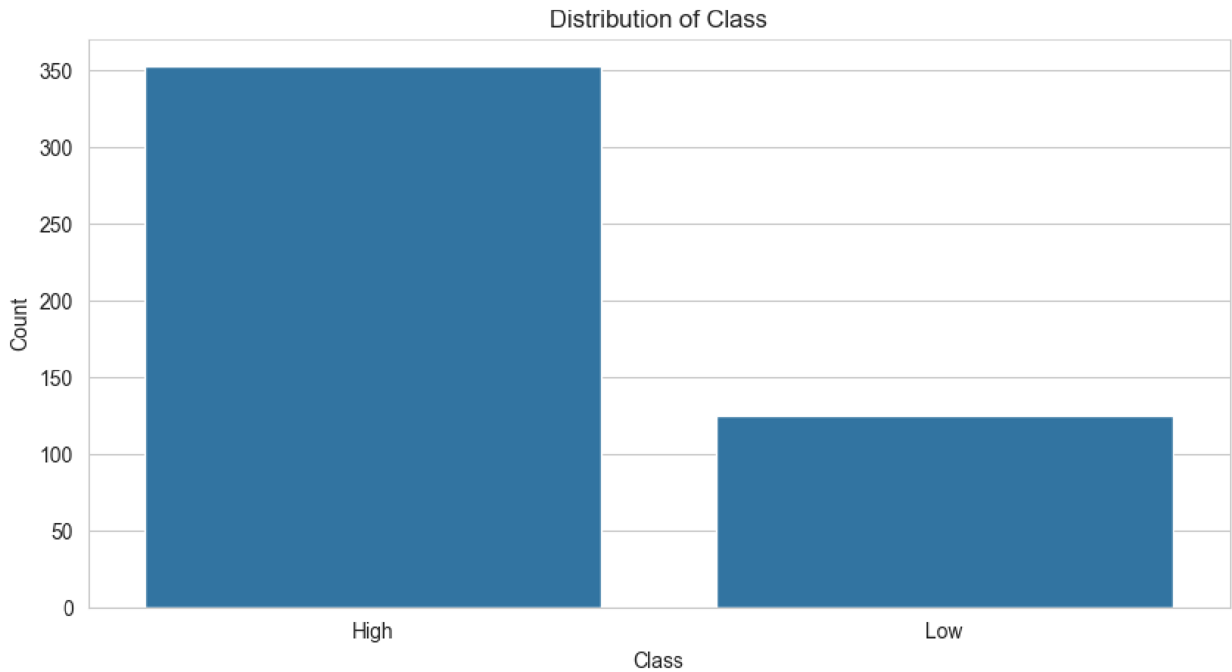


Figure 9. Distribution of the target classes

Random Forest and XGBoost models were trained on 3 different data splits, following the similar approach proposed to train the AMCC model, to determine the best model. Table 25. shows results for the same and Table 26. shows the parameters used during training. The model with the highest F1 score, Random Forest 60-20-20 split, was used in the AMCC model to generate counterfactuals.

Table 25. Results of Random Forest and XGBoost training

Data Split	**Random Forest Parameter**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
80-10-10	‘n_estimator’: 10	0.9583	0.8571	0.9474	0.9
70-15-15	‘n_estimator’: 68	0.9583	0.9032	0.9032	0.9032
60-20-20	‘n_estimator’: 35	0.9688	0.9362	0.9362	**0.9362**

Data Split	XGBoost Parameter	Accuracy	Precision	Recall	F1 Score
80-10-10	'colsample_bytree': 0.8, 'max_depth': 4, 'n_estimators': 26, 'subsample': 0.8	0.9271	0.7727	0.8947	0.8293
70-15-15	'colsample_bytree': 0.9, 'max_depth': 8, 'n_estimators': 28, 'subsample': 0.9	0.9306	0.8387	0.8387	0.8387
60-20-20	'colsample_bytree': 0.8, 'max_depth': 4, 'n_estimators': 16, 'subsample': 0.9	0.9375	0.8723	0.8723	**0.8723**

Table 26. Parameters for Random Forest and XGBoost models

Model	Parameters
Random Forest	n_estimators, n_jobs = 8, criterion = ’log_loss’, max_features = ’log2’
XGBoost	eval_metric = ’logloss’, max_depth, subsample, colsample_bytree, n_estimators

Out of the 16 features, 5 features were identified as actionable features, Table 27., and Table 28. details their FAT categorization. ABSENT is categorized as a binary feature that can only decrease. The two values are ‘absence above 7 days’ and ‘absence under 7 days’.

Table 27. Actionable Features

Feature	Description	Example Recourse Actions
VISITRES	the number of times the student views the resource center	advise the student to visit the resource center more
VIEWANC	the number of times the student views the announcements	pay closer attention to announcements
RAISE	the number of times the student raises their hand	encourage the student to raise their hand more

DISCUSS	the number of times the student participates in discussions	encourage the student to participate more during discussions
ABSENT	the number of absences days	reduce the number of absences to under 7 days

Table 28. Feature Categorization

Mutably Direct Features	VISITRES, VIEWANC
Mutably Indirect Features	RAISE, DISCUSS, ABSENT

The counterfactual generated by AMCC model was used as the test set with synthetically generated natural language explanations. The train set was augmented with backtranslation, and Table 30 shows some samples. Of the set of 3 samples, the first sample is the original sample, the next two samples are the French and German back translated samples. Table 29. shows the number of samples per set.

Table 29. Sample per set

Set	**Number of Samples**
Test	38
Train	200
Train - with backtranslation	600

Table 30. Examples of Generated Natural Language explanations and back translated explanations

Counterfactual	**Generated Natural Language Explanations**
RAISE: 17 - 50 -> 0 - 16 DISCUSS: 0 - 20 -> 71 - 99 OCF: VISITRES	Reducing the number of times the student raises their hand to between 0 and 16, while increasing participation in discussions from 0 - 20 to 71 - 99, may contribute to improved overall grades. It is also recommended to review how often the student visits the resource center.
RAISE: 17 - 50 -> 0 - 16 DISCUSS: 0 - 20 -> 71 - 99 OCF: VISITRES	Reducing the number of times the student holds a hand between 0 and 16, while increasing participation in the discussions from 0 to 20 to 71 to 99, can help improve the overall grades. It is also recommended to examine how often the student visits the resource centre.
RAISE: 17 - 50 -> 0 - 16 DISCUSS: 0 - 20 -> 71 - 99 OCF: VISITRES	Reducing the number of students raises their hands between 0 and 16, while participating in discussions from 0 - 20 to 71 - 99, can contribute to improved overall scores. It is also recommended to check how often the student visits the resource center.
DISCUSS: 21 - 39 -> 40 - 70 ABSENT OCF:	Encouraging participation in discussions within the range of 40 - 70 and reducing the number of absences could positively impact the student's overall grade.
DISCUSS: 21 - 39 -> 40 - 70 ABSENT OCF:	Encouraging participation in discussions in the 40-70 age range and reducing the number of absences could have a positive impact on the student's overall score.

DISCUSS: 21 - 39 -> 40 - 70 ABSENT OCF:	Promoting participation in discussions between 40 and 70 and reducing the number of absences could have a positive impact on the overall grade of students.

Table 31 details the feature descriptions used in the template-based approach. For the feature ‘ABSENT’, separate descriptions were designed for counterfactual explanations and other contributing explanations. As a binary feature, this distinction helps ensure the most effective construction of template-generated explanations. In addition, the minimal changes performed on the template values were

1. ‘encourage’ was added to the verb slot values
2. ‘adjust’ was removed from the OCF-General slot values

These changes were made keeping in mind the domain of the task.

Table 31. Feature Description for Students' Academic Performance

Feature	**Description**
RAISE	the number of times the student raises their hand
VISITRES	the number of times the student views the resource center
VIEWANC	the number of times the student views the announcements
DISCUSS	the number of times the student participates in discussions
ABSENT For OCF	the number of absences to under 7 days the number of absences

Table 32. shows examples of the new ‘simple explanation’ templates created for the template-guided approach. Appendix A lists all the ‘simple explanation’ templates for the Students’ Academic Performance task.

Table 32. Examples of ‘simple explanation’ templates for Students' Academic Performance

Feature	**Simple Explanation Template**
RAISE - Increase	
INC!!RAISE!!@	Encourage the student to increase the number of times they raise their hand to @
INC!!RAISE!!@@	Encourage the student to increase the number of times they raise their hand from @ to @
RAISE - Decrease	
DEC!!RAISE!!@	Encourage the student to decrease the number of times they raise their hand to @
DEC!!RAISE!!@@	Encourage the student to decrease the number of times they raise their hand from @ to @
ABSENT	
DEC!!ABSENT	Decrease the number of absences to under 7 days
OCF!!ABSENT	reduce the number of absences

Table 33. shows some samples of natural language explanations generated using each of the approaches. For the template-based, with Lexical Substitution, approach the DistilRoBERTa model was used with threshold value 1e-6. For the template-guided approach, the best checkpoint from the Base Run I run was used. Explanations generated using the model which was trained using the synthetic dataset of Students’ Academic Performance dataset and without. The results of the training are shown in Figure 10.

Table 33. Examples of Generated Explanations with each approach

Approach	**Generated Natural Language Explanations**
RAISE: 0 - 16 -> 17 - 50 OCF: VISITRES, VIEWANC	
Template-based without Lexical Substitution	Look at measures to improve the number of times the student raises their hand to 17 - 50. Reexamine the number of times the student views the resource center or review the number of times the student views the announcements, to improve the overall grade.
Template-based with Lexical Substitution	You can encourage measures to change the number of times the student raises their hand to 17 - 50. To improve the overall grade, assess the number of times the student views the resource center and evaluate the number of times the student views the announcements.
Template-guided - before training	Increase the number of times the student raises their hand from 0 - 16 to 17 - 50 and review the number of times the student views the announcements to improve the overall grade.
Template-guided - after training	Encourage the student to increase their hand raise from 17 - 50, and review how frequently the student visits the resource center or reviews the announcements, to increase the overall grade.
VISITRES: 21 - 65 -> 0 - 20 OCF: RAISE, VIEWANC	
Template-based without Lexical Substitution	Cut back the number of times the student views the resource center to 0 - 20. To better the overall grade, change the number of times the student raises their hand or review the number of times the student views the announcements.

Template-based with Lexical Substitution	The number of times the student views the resource center needs to reduce from 21 - 65 to 0 - 20. Review the number of times the student raises their hand and assess the number of times the student views the announcements, to better the overall grade.
Template-guided - before training	Decrease the number of times the student visits the resource center to 0 - 20. Review the number of times the student raises their hand or review the number of times the student views the announcements to improve the overall grade.
Template-guided - after training	Reducing resource center visits to 0 - 20 and reviewing how often the student raises their hand or views announcements can help improve overall grades.
VISITRES: 0 - 20 -> 66 - 84 OCF: RAISE, VIEWANC	
Template-based without Lexical Substitution	The number of times the student views the resource center needs to better from 0 - 20 to 66 - 84. To strengthen the overall grade, address the number of times the student raises their hand and evaluate the number of times the student views the announcements.
Template-based with Lexical Substitution	The number of times the student views the resource center needs to grow to 66 - 84. Evaluate the number of times the student raises their hand and evaluate the number of times the student views the announcements, to improve the overall grade.
Template-guided - before training	Increasing the number of times the student visits the resource center from 0 - 20 to 66 - 84, while reviewing the number of times the student raises their hand or reviews the announcements, will improve the overall grade.

Template-guided - after training	Increasing visits to the resource center from 0 - 20 to 66 - 84 and reviewing how often the student raises their hand or views announcements can help improve overall grades.

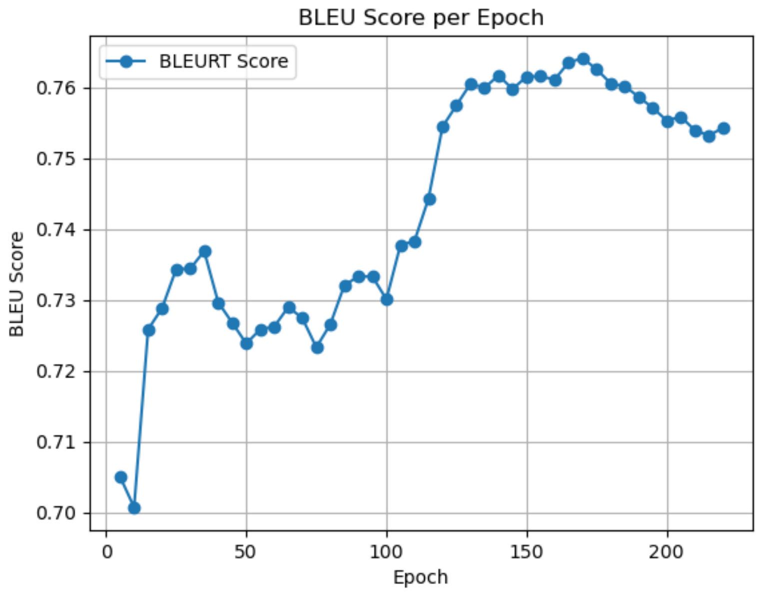


Figure 10. Training Results on the Students' Academic Performance synthetic dataset

Training was performed with a lower learning rate of 1e-5 and the other hyperparameters being the same. The training stopped at epoch 220 when the early stopping logic was satisfied, Figure 10. Table 34. details the evaluation metrics for each approach.

Table 34. Evaluation metrics for the Students' Academic Performance

Approach	**BLEURT Scores**	**Time Taken**
Template-based without Lexical Substitution	0.651	0.00003
Template-based with Lexical Substitution	0.6568	0.11771
Template-guided - before training	0.70086	1.35626
Template-guided - after training	0.73688	1.14283

5. DISCUSSION

5.1 Synthetic Dataset

Due to the absence of a real-world dataset for this specific task, a synthetic dataset was created to evaluate the two approaches proposed in this thesis. A smaller dataset was intentionally chosen to maintain the quality of generated samples through manual verification, while still enabling effective fine-tuning of the T5 model using LoRA. The dataset size could be further augmented using regularization techniques or backtranslation. Notably, backtranslation was employed in the creation of the dataset for the Students' Academic Performance task. However, back-translated samples require additional verification, as the meaning of certain concepts may shift during the translation process. For instance, the term *"announcements"* was incorrectly translated back from French to English as *"advertisements"* or *"ad"*, highlighting the need to carefully screen for such semantic errors.

While large language models (LLMs) offer powerful capabilities for generating synthetic data, they are not ideally suited for creating large-scale datasets. Over time, generated samples tend to exhibit repetition, either in sentence structure or vocabulary. Moreover, generating large datasets using LLMs demands significant computational resources if running locally or constrained by model prompt limits if accessing through a subscription. An additional limitation is the number of samples that can be reliably produced per prompt. With OpenAI's GPT-4o model, generating 10 to 15 samples per prompt yielded optimal results. Attempts to generate more than 20 samples at once often led to reduced adherence to instructions, including inconsistent counterfactual values or repetitive outputs with minimal structural variation.

Despite these limitations, the use of LLMs for synthetic data generation highlights their potential to produce natural language explanations through prompting, especially when compared to the two approaches explored in this thesis. As context lengths increase, it may become feasible to incorporate richer user-specific details or more elaborate recourse strategies within prompts, allowing for the generation of more complex and personalized counterfactual explanations.

5.2 Template-based Approach

Unlike FAT-based templates, the newly created templates, slot values, and associated synonyms have not been validated by domain experts. While prompting OpenAI's GPT-4o model offers useful examples and serves as a helpful starting point for designing new templates, these outputs cannot be used directly without adaptation.

Expanding slot values using WordNet increases the range of options available and supports the generation of more varied explanations. However, additional manual verification is essential to ensure the selected synonyms are appropriate and contextually accurate. Since each synonym must be manually checked, a balanced strategy is required, one that weighs the effort involved against the level of variability desired in the generated explanations.

For lexical substitution, the action verb slot was chosen as it conveys the most meaning on the change required to achieve a favored outcome. Synonyms for this slot do not require manual verification instead, only a few seed actions need to be determined for which synonyms will be retrieved and scored by the BERT model.

The threshold serves as a mechanism to control variability. A higher threshold typically reduces the number of viable candidates among the top five to just one or two, leading to more static and repetitive outputs. This trade-off can be useful when consistency is preferred, but it comes at the cost of diversity.

Among the BERT-based models tested, RoBERTa-large shows slightly longer inference times due to its larger architecture. Nonetheless, this does not significantly affect performance in real-time settings. BLEURT scores remain consistent across RoBERTa-large, RoBERTa-base, and DistilRoBERTa at various thresholds, suggesting that the smaller DistilRoBERTa model can be reliably used without compromising explanation quality.

A limitation of masked language modeling (MLM) is its inability to work with multi-tokens. Therefore, selecting a single template slot that maximizes impact, such as the action verb, is needed. As shown in Table 24, the comparable results between approaches with and without lexical substitution are promising. However, expanding the MLM task to include additional or multiple slots will require further validation and testing.

5.3 Template-guided Approach

Due to the limited size of the synthetic dataset, the larger model variants (base and large) exhibited signs of overfitting. To address this, five different run configurations were designed to mitigate overfitting. Hyperparameter adjustments such as reducing the learning rate, lowering batch sizes, and increasing weight decay, did not significantly improve training performance. However, the larger capacity of these models allowed them to learn the task quickly, as reflected in the evaluation metrics shown in Table 20. Notably, BLEURT scores were similar across configurations, with 'Base Run I' achieving the highest score.

Training for both the base and large variants was stopped early upon meeting the early stopping criteria, which is expected given the small dataset size. While the small, base and large variants performed comparably in terms of output quality, the large variant incurred longer inference times due to its increased complexity. In contrast, the small and base variants demonstrated similar average inference times of approximately one second, making them better suited for real-time applications. As a result, Base Variant – Run I was selected for further testing, balancing performance and efficiency.

Although the fine-tuned model demonstrated strong performance, it frequently generated identical explanations for the same input. To introduce variation, experiments were conducted using two different top_p values, 0.9 and 1.1. The value of top_p = 0.9 offered the best trade-off between diversity and performance. While this setting enabled more varied explanation generation, it resulted in a slight decrease in BLEURT score, 0.03 points, indicating a minor drop in output quality. This suggests that while increasing top-p enhances variability, it may slightly reduce the semantic alignment or fluency of the generated explanations.

A key drawback of the template-guided approach is the introduction of a black-box model into the generation pipeline. Although the use of ‘simple explanation’ templates offers a degree of transparency and control over the input structure, the final explanations cannot be fully constrained as they can in the template-based approach. This introduces the risk of slight deviations from the intended or correct natural language explanations.

5.4 Domain Adaptability

The aim of this experiment was to evaluate how well the proposed approaches generalize to a new domain. To assess their adaptability, modifications to the original methods were kept minimal. A small synthetic dataset was created for the new domain and augmented using backtranslation, which allowed for a reduction in the total number of synthetic samples required.

Although the generated explanations are generally correct and straightforward, their BLEURT scores are lower in comparison to previous results. Nevertheless, the outcomes offer a strong starting point for further fine-tuning to improve explanation quality.

The feature descriptions in template-based approach and ‘simple explanation’ templates for the template-guided approach, were made slightly more detailed than typically recommended. This decision was based on the nature of the new domain, providing recourse actions to help students improve their overall academic performance. The rationale was that adding contextual detail might produce more appropriate and nuanced natural language explanations, rather than overly direct.

While the initial results are promising, full fine-tuning remains the more effective strategy at this stage. The original paper trained across multiple domains, and a similar approach could be revisited in the future if datasets from multiple domains or subdomains become available.

5.5 Conclusion

Both approaches demonstrate that it is possible to generate effective and simple natural language counterfactual explanations. However, the template-guided approach consistently outperforms the template-based method in generating more nuanced and fluent explanations.

The template-based approach, while simpler, effectively conveys the counterfactual by generating straightforward and interpretable explanations. One limitation of this approach is that its generations are constrained to simple verb and noun forms. This restricts linguistic diversity and contributes to its lower BLEURT scores, especially when evaluated against a varied test set. In contrast, the template-guided approach leverages the full natural language generation capabilities of the T5 model, resulting in more structurally complex and expressive outputs.

While both versions of the template-based approach (with and without lexical substitution) yield comparable results, incorporating lexical substitution where possible offers improved flexibility and expressiveness. This allows for slightly more varied outputs while still maintaining simplicity and clarity that make the template-based method effective.

Both approaches demonstrate that it is possible to adapt to a new domain with minimal changes and a small dataset. However, full fine-tuning proves to be more effective for the template-guided approach, particularly when the approach is not intended for use across multiple domains or subdomains.

5.6 Future Work

Several avenues remain for extending and improving the current approaches. First, the template-based lexical substitution method, which is presently limited to the action_verb slot, could be expanded to cover additional slots. This would significantly reduce the time spent on manual verification of slot value expansions while maintaining output quality. Similarly, the template-guided approach, which utilizes a T5 model, has the capacity to scale across multiple domains or subdomains. While the original study demonstrates the promise of this approach, further experimentation is needed to evaluate its effectiveness when applied in multi-domain settings.

Another important direction is incorporating human evaluation to more accurately assess the quality and clarity of the generated natural language explanations. Human judgment can offer insights into nuances that automated metrics may overlook, particularly in context-sensitive explanations.

Developing a supporting tool for subject matter experts could enhance the interpretability and usability of generated outputs. Such a tool would allow experts to map dataset value ranges (e.g., "17 to 50 occurrences per term") to more meaningful, context-specific units (e.g., "1 to 3 times per class session"), improving the relevance and accessibility of explanations for end users.

Together, these enhancements aim to improve scalability, usability, and reliability, paving the way for broader adoption of template-based and template-guided natural language generation methods in real-world applications.

6. REFERENCES

1. Barzekar, H., & McRoy, S. (2023). Achievable minimally-contrastive counterfactual explanations. *Machine Learning and Knowledge Extraction*, *5*(3), 922–936. https://doi.org/10.3390/make5030048
2. Miller, G. A. (1997). *WordNet - a lexical database for English*. Princeton University, Cognitive Science Laboratory.
3. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *Proceedings of the 2019 Conference of the North*. https://doi.org/10.18653/v1/n19-1423
4. Raffel, C. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*(140).
5. Salimi, P., Wiratunga, N., Corsar, D., & Wijekoon, A. (2023). Towards feasible counterfactual explanations: A taxonomy guided template-based NLG method. *Frontiers in Artificial Intelligence and Applications*. https://doi.org/10.3233/faia230499
6. Rämö, M., & Leppänen, L. (2021). Using contextual and cross-lingual word embeddings to improve variety in template-based {NLG} for automated journalism. *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*.
7. Liu, Y. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv Preprint arXiv:1907.11692*, *364*.
8. Sanh, V. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *NeurIPS EMC^2 Workshop*.
9. Kale, M., & Rastogi, A. (2020). Template guided text generation for task-oriented dialogue. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6505–6520. https://doi.org/10.18653/v1/2020.emnlp-main.527
10. Chung, H. W., & Hou, L. (2024). Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, *25*.
11. J. Hu, E., Shen, Y., & Wallis, P. (2021). LoRA: Low-Rank Adaptation of Large Language Models. CoRR, abs/2106.09685. https://doi.org/https://doi.org/10.48550/arXiv.2106.09685
12. Transformers Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., … Rush, A. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. https://doi.org/10.18653/v1/2020.emnlp-demos.6
13. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*.
14. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".
15. *Cuda Zone - Library of resources*. NVIDIA Developer. (n.d.). https://developer.nvidia.com/cuda-zone
16. Vallala, A. (2018, March 25). *HMEQ_DATA*. Kaggle. https://www.kaggle.com/datasets/ajay1735/hmeq-data
17. Hello gpt-4o. (n.d.). https://openai.com/index/hello-gpt-4o

18. Pu, A., Chung, H. W., Parikh, A., Gehrmann, S., & Sellam, T. (2021). Learning compact metrics for Mt. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 751–762. https://doi.org/10.18653/v1/2021.emnlp-main.58
19. Z Tiedemann, J., Aulamo, M., Bakshandaeva, D., Boggia, M., Grönroos, S.-A., Nieminen, T., Raganato, A., Scherrer, Y., Vázquez, R., & Virpioja, S. (2023). Democratizing Neural Machine translation with opus-mt. *Language Resources and Evaluation*, *58*(2), 713–755. https://doi.org/10.1007/s10579-023-09704-w
20. Amrieh, E. A., Hamtini, T., & Aljarah, I. (2016). Mining educational data to predict student's academic performance using Ensemble Methods. *International Journal of Database Theory and Application*, 9(8), 119–136. https://doi.org/10.14257/ijdta.2016.9.8.13
21. Arefyev, N., Sheludko, B., Podolskiy, A., & Panchenko, A. (2020). Always keep your target in mind: Studying semantics and improving performance of neural lexical substitution. Proceedings of the 28th International Conference on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.107
22. Liello, L. D. (2023). Lucadiliello/Bleurt-pytorch: BLEURT implementation in Pytorch. GitHub. https://github.com/lucadiliello/bleurt-pytorch
23. Distilbert/Distilroberta-base · hugging face. distilbert/distilroberta-base · Hugging Face. (n.d.). https://huggingface.co/distilbert/distilroberta-base

Appendix A: Simple Explanation Templates

The following is the simple explanation template used for template-guided NLG approach for all testing. The entries are tab separated.

INC!!LOAN!!@	Increase the loan request amount by $@.
INC!!LOAN!!@@	Increase the loan request amount from $@ to $@.
DEC!!LOAN!!@	Decrease the loan request amount by $@.
DEC!!LOAN!!@@	Decrease the loan request amount from $@ to $@.
INC!!VALUE!!@	Increase the property value by $@.
INC!!VALUE!!@@	Increase the property value from $@ to $@.
DEC!!VALUE!!@	Decrease the property value by $@.
DEC!!VALUE!!@@	Decrease the property value from $@ to $@.
INC!!CLNO!!@	Increase the number of credit lines by @.
INC!!CLNO!!@@	Increase the number of credit lines from @ to @.
DEC!!CLNO!!@	Decrease the number of credit lines by @.
DEC!!CLNO!!@@	Decrease the number of credit lines from @ to @.
DEC!!MORTDUE!!@	Decrease the amount due on the mortgage by $@.
DEC!!MORTDUE!!@@	Decrease the amount due on the mortgage from $@ to $@.
DEC!!DELINQ!!@	Reduce the number of delinquent credit lines by @.
DEC!!DELINQ!!@@	Reduce the number of delinquent credit lines from @ to @.
DEC!!NINQ!!@	Decrease the number of recent credit inquiries by @.
DEC!!NINQ!!@@	Decrease the number of recent credit inquiries from @ to @.
DEC!!DEBTINC!!@	Decrease the debt-to-income ratio by @%.
DEC!!DEBTINC!!@@	Decrease the debt-to-income ratio from @% to @%.
OCF!!LOAN	review the loan request amount
OCF!!VALUE	reappraise the value of the property
OCF!!CLNO	review the number of open credit lines
OCF!!MORTDUE	reassess the amount due on the existing mortgage
OCF!!DELINQ	pay off delinquent credit lines
OCF!!NINQ	review recent credit inquiries
OCF!!DEBTINC	reduce the debt-to-income ratio

Students' Academic Performance
INC!!RAISE!!@	Encourage the student to increase the number of times they raise their hand to @
INC!!RAISE!!@@	Encourage the student to increase the number of times they raise their hand from @ to @
DEC!!RAISE!!@	Encourage the student to decrease the number of times they raise their hand to @
DEC!!RAISE!!@@	Encourage the student to decrease the number of times they raise their hand from @ to @
INC!!VISITRES!!@	Increase the number of times the student visits the resource center to @
INC!!VISITRES!!@@	Increase the number of times the student visits the resource center from @ to @

DEC!!VISITRES!!@ Decrease the number of times the student visits the resource center to @
DEC!!VISITRES!!@@ Decrease the number of times the student visits the resource center from @ to @
INC!!VIEWANC!!@ Increase the number of times the student views the announcements to @
INC!!VIEWANC!!@@ Increase the number of times the student views the announcements from @ to @
DEC!!VIEWANC!!@ Decrease the number of times the student views the announcements to @
DEC!!VIEWANC!!@@ Decrease the number of times the student views the announcements from @ to @
INC!!DISCUSS!!@ Encourage the student to increase the number of times they participate in discussions to @
INC!!DISCUSS!!@@ Encourage the student to increase the number of times they participate in discussions from @ to @
DEC!!DISCUSS!!@ Encourage the student to decrease the number of times they participate in discussions to @
DEC!!DISCUSS!!@@ Encourage the student to decrease the number of times they participate in discussions from @ to @
DEC!!ABSENT Decrease the number of absences to under 7 days
OCF!!RAISE review the number of times the student raises their hand
OCF!!VISITRES review the number of times the student visits the resource center
OCF!!VIEWANC review the number of times the student views the announcements
OCF!!DISCUSS review the number of times the student participates in discussions
OCF!!ABSENT reduce the number of absences

Appendix B: Synsets from WordNet for template values

Synonyms for a sample of slot values retrieved from WordNet are shown below. The synonyms in bold are the ones manually chosen to be added to the slot values.

Word	Synonyms
reduce	**reduce**, cut down, cut back, trim, trim down, trim back, cut, **bring down**, come down, boil down, shrink, scale down, deoxidize, deoxidise, tighten, repress, quash, keep down, subdue, subjugate, decoct, concentrate, dilute, thin, thin out, melt off, lose weight, slim, slenderize, slim down
improve	better, **improve**, amend, ameliorate, meliorate
increase	**increase**
raise	raise, lift, elevate, get up, bring up, grow, farm, produce, rear, nurture, parent, conjure, conjure up, invoke, evoke, stir, call down, arouse, put forward, call forth, erect, set up, put up, elicit, enkindle, kindle, fire, provoke, **enhance**, heighten, promote, upgrade, advance, kick upstairs, leaven, prove, recruit, levy, resurrect, upraise
decrease	**decrease**, **diminish**, **lessen**, fall, minify
lower	**lower**, take down, let down, get down, bring down, lour, turn down, depress, frown, glower
measures	**measure**, step, quantity, amount, bill, measurement, measuring, standard, criterion, touchstone, meter, metre, beat, cadence, bar, measuring stick, measuring rod

Sample of synsets for 'reduce' obtained from WordNet.

Word: reduce
Index: 0
Part of Speech: v
Definition: cut down on; make a reduction in
Synonyms: ['reduce', 'cut_down', 'cut_back', 'trim', 'trim_down', 'trim_back', 'cut', 'bring_down']

Index: 1
Part of Speech: v
Definition: make less complex
Synonyms: ['reduce']

Index: 2
Part of Speech: v
Definition: bring to humbler or weaker state or condition
Synonyms: ['reduce']

Index: 3
Part of Speech: v
Definition: simplify the form of a mathematical equation of expression by substituting one term for another
Synonyms: ['reduce']

Index: 4
Part of Speech: v
Definition: lower in grade or rank or force somebody into an undignified situation
Synonyms: ['reduce']

Index: 5
Part of Speech: v
Definition: be the essential element
Synonyms: ['reduce', 'come_down', 'boil_down']

Index: 6
Part of Speech: v
Definition: reduce in size; reduce physically
Synonyms: ['shrink', 'reduce']

Index: 7
Part of Speech: v
Definition: lessen and make more modest
Synonyms: ['reduce']

Index: 8
Part of Speech: v
Definition: make smaller
Synonyms: ['reduce', 'scale_down']

Index: 9
Part of Speech: v
Definition: to remove oxygen from a compound, or cause to react with hydrogen or form a hydride, or to undergo an increase in the number of electrons
Synonyms: ['deoxidize', 'deoxidise', 'reduce']

Index: 10
Part of Speech: v
Definition: narrow or limit
Synonyms: ['reduce', 'tighten']

The document containing the synset retrieved for all slot values can be found in the following GitHub repository: https://github.com/TarunThomasEapen/NLG-System-for-Counterfactuals

The following snippet code can be used to retrieve synonyms from WordNet for a list of words for the noun and verb POS tags.

```
from nltk.corpus import wordnet

# Function to get synonyms and part of speech (POS) for a word
def get_synonyms_with_pos(word, pos=None):
    result = []
    for idx, synset in enumerate(wordnet.synsets(word)):
        # Check if the word is a verb
        if synset.pos() != pos:
            continue
        # Get part of speech
        pos = synset.pos()
        # Fetch synonyms from the synset

        synonyms = [lemma.name() for lemma in synset.lemmas()]
        result.append({"index": idx, "synonyms": synonyms, "part_of_speech": pos, "definition":
synset.definition()})
    return result

# Collect the words from the template values
verbs = ["take", "initiate", "undertake", "pursue", "negotiate", "increase", "improve", "raise"]

nouns = ["steps", "measures", "actions"]

words = [[verbs, "v"], [nouns, "n"]]
index_count = 0
synonym_set = set()
# Process each word in the template
for words, pos in words:
    for word in words:
        synonyms_with_pos = get_synonyms_with_pos(word, pos)
        print(f"Word: {word}")
        for entry in synonyms_with_pos:
            print(f"  Index: {entry['index']}")
            index_count += 1
            print(f"  Part of Speech: {entry['part_of_speech']}")
            print(f"  Definition: {entry['definition']}")
            print(f"  Synonyms: {entry['synonyms']}")
            synonym_set.update(entry['synonyms'])
            print("  " + "-" * 50)

print(f"Total number of unique synonyms: {len(synonym_set)}")
print(f"Total number of indexes: {index_count}")
print(synonym_set)
```

Appendix C: Alignment Tests for BLEURT

The following table shows the difference in scores using the same examples as provided in the original paper.

Reference	**Candidate**	**BleuRT Scores**	**BleuRT PyTorch Scores**	**Difference (in %)**
Bud Powell was a legendary pianist.	Bud Powell was a legendary pianist.	0.9778	0.97773	0.00603
Bud Powell was a legendary pianist.	Bud Powell was a historical piano player.	0.769	0.79420	2.52048
Bud Powell was a legendary pianist.	Bud Powell was a new yorker.	0.45	0.48986	4.97645
Bud Powell was a legendary pianist.	Bud a day keeps the doctor away.	0.16677	0.21265	4.58767

The following table shows the scores from for the test set and candidates generated using the finetuned T5-flan base model.

Reference	**Prediction**	**BLEURT-20_TensorFlow**	**BLEURT-20_PyTorch**	**Difference (in %)**
To increase the chances of loan approval, consider significantly increasing your property value from between $8,000 - $66,100 to around $120,001 - $347,000. Additionally, reduce the number of recent credit inquiries from 8 to about 2. Paying off any delinquent credit lines and reviewing your total number of open credit lines can also help positively influence the loan decision.	Increase property value from $8,000 - $66,100 to $120,001 - $347,000. Reduce recent credit inquiries from 8 to 2, while reducing delinquent credit lines and reviewing open credit lines, will improve loan approval chances.	0.71882	0.71851	0.03065
Improving your likelihood of loan approval may be achieved by reducing delinquent credit lines from 7 down to 6 and substantially paying down your existing	Reducing delinquent credit lines from 7 to 6, while paying down the mortgage balance from $65,001 - $91,500 to $2,060 - $46,300, will improve loan approval chances. Reappraising	0.76532	0.76517	0.01497

mortgage from a current range of $65,001 - $91,500 to approximately $2,060 - $46,300. Additionally, reappraising your property's value or reviewing your open credit lines can further enhance your application's strength.	the property or reviewing open credit lines will further enhance loan approval chances.			
You can enhance your loan application by increasing your total number of credit lines to between 16 and 20. It's also important to review your requested loan amount and manage recent credit inquiries effectively, as these factors can positively influence the loan approval decision.	Increasing credit lines to 16 - 20 while reviewing the loan request amount and reviewing recent credit inquiries will improve your loan approval chances.	0.71405	0.71402	0.00275
To enhance your chances of loan approval, try to reduce your existing mortgage amount from the current range of $65,001 - $91,500 down to between $46,301 - $65,000. Additionally, consider increasing your property's value to between $66,101 - $89,200. It's also beneficial to review recent credit inquiries and adjust your loan request amount accordingly.	Paying down the mortgage balance from $65,001 - $91,500 to $46,301 - $65,000, while increasing property value to $66,101 - $89,200, will improve your loan approval chances. Reviewing recent credit inquiries and reviewing the loan request amount will further enhance your application.	0.73716	0.73664	0.05230
Increasing your loan request amount to approximately $11,101 - $16,300 may improve your application's strength. Additionally, consider reappraising the value of your property or reducing	Reducing the loan request amount to $11,101 - $16,300, while reassessing the property value or reducing the debt-to-income ratio, will improve loan approval chances.	0.72284	0.72275	0.00890

your debt-to-income ratio, as these changes can positively influence your loan approval outcome.				

The full results of the alignment tests are present in the following GitHub repository: https://github.com/TarunThomasEapen/NLG-System-for-Counterfactuals

Appendix D: Prompt to generate template examples

OpenAI's GPT-4o model was used to generate the template examples. The responses have been edited for clarity and the full transcript for the chat is available here: https://github.com/TarunThomasEapen/NLG-System-for-Counterfactuals

The following is the chat to generate templates examples for Mutually Direct and Indirect features.

The pdf of the paper 'Towards Feasible Counterfactual Explanations: A Taxonomy Guided Template-Based NLG Method.' by Salmi et. al., was provided as additional context to the prompt.

Prompt: I would like a few new templates for mutably direct and indirect features

Response:

The document introduces **template-based natural language generation (NLG)** for counterfactual explanations, guided by a **Feature Actionability Taxonomy (FAT)**. These templates are organized according to the mutability and sensitivity of features and mapped to different types of content and structure themes identified in a user study.

Here are some new **template suggestions** for **Mutable Directly** and **Mutable Indirectly** features:

Templates for Mutable Directly Features (M.D.)

1. **Change with Explicit Instruction**
   - "To achieve the desired outcome, {ACTION} your {FEATURE} from {QUERY_VALUE} to {CF_VALUE}." *(Example: "To achieve the desired outcome, increase your income from $35,000 to $50,000.")*
2. **Goal-Oriented Change (A Variation of this was finally used)**
   - "Your {FEATURE} must be adjusted to {CF_VALUE} from the current value of {QUERY_VALUE} to meet the criteria." *(Example: "Your savings must be adjusted to $10,000 from the current value of $5,000 to meet the criteria.")*
3. **Step-by-Step Guidance**
   - "Start by {ACTION} your {FEATURE}. Aim to reach a value of {CF_VALUE}, progressing from your current value of {QUERY_VALUE}." *(Example: "Start by reducing your monthly expenses. Aim to reach a value of $1,200, progressing from your current value of $1,500.")*
4. **Positive Framing**

- "Increasing your {FEATURE} to {CF_VALUE} from {QUERY_VALUE} will significantly enhance your chances of success." *(Example: "Increasing your credit score to 750 from 680 will significantly enhance your chances of approval.")*

5. **Conditional Framing**
    - "If you {ACTION} your {FEATURE} to {CF_VALUE}, instead of {QUERY_VALUE}, it will positively impact the outcome." *(Example: "If you raise your GPA to 3.5, instead of 3.0, it will positively impact your application.")*

Templates for Mutable Indirectly Features (M.I.)

1. **Causal Change Instruction**
    - "To improve your {FEATURE}, consider {ACTION} your related {RELATED_FEATURE} from {QUERY_VALUE} to {CF_VALUE}." *(Example: "To improve your savings, consider decreasing your monthly expenses from $2,000 to $1,800.")*
2. **Strategic Suggestion**
    - "You can achieve a better {FEATURE} by {VERB} {OBJECT} that indirectly affect it, such as {ACTION} your {RELATED_FEATURE} to {CF_VALUE}." *(Example: "You can achieve a better debt-to-income ratio by taking steps to increase your income to $4,000.")*
3. **Process-Oriented Guidance**
    - "To reach the target {FEATURE} value of {CF_VALUE}, focus on modifying related factors like {RELATED_FEATURE}." *(Example: "To reach the target savings value of $10,000, focus on modifying related factors like reducing discretionary expenses.")*
4. **Collaborative Framing**
    - "Work towards {ACTION} your {RELATED_FEATURE} to {CF_VALUE} to indirectly improve your {FEATURE}." *(Example: "Work towards lowering your rent to $1,200 to indirectly improve your savings.")*
5. **Outcome-Based Recommendation**
    - "Achieving a {FEATURE} of {CF_VALUE} may require changes in your {RELATED_FEATURE}, currently valued at {QUERY_VALUE}." *(Example: "Achieving a monthly savings of $500 may require changes in your discretionary spending, currently at $800.")*

Follow up Prompt: Let's reduce the amount of fixed text in the template. Creating new template variables if necessary

Response:
Here are some **revised templates** with reduced fixed text and additional template variables for **Mutable Directly** and **Mutable Indirectly** features. This approach focuses on flexibility and introduces new variables to capture a variety of sentence structures.

New Template Variables

- **TARGET**: Desired value or outcome for the feature (e.g., higher, lower, specific numeric values).
- **FOCUS**: Aspect or feature being modified (e.g., income, savings, credit score).
- **ADJUST**: Action verb specific to change (e.g., adjust, modify, alter).
- **REASON**: The purpose or reason for making the change (e.g., to qualify, to meet requirements).
- **METHOD**: Actionable approach (e.g., save more, spend less, negotiate).
- **CONTEXT**: Supporting context or information about the feature change.

Templates for Mutable Directly (M.D.)

1. **Focus on the Adjustment**
   - "{ADJUST} your {FOCUS} to {TARGET}."
     (Example: "Adjust your income to $50,000.")
2. **Emphasize the Outcome**
   - "{ADJUST} {FOCUS} from {QUERY_VALUE} to {TARGET} {REASON}."
     (Example: "Increase income from $35,000 to $50,000 to qualify for the loan.")
3. **Include Context**
   - "{FOCUS}: {QUERY_VALUE} → {TARGET}. {REASON}."
     (Example: "Income: $35,000 → $50,000. This is required to meet the approval criteria.")
4. **Highlight the Action**
   - "{FOCUS} needs {ADJUSTMENT} to {TARGET}."
     (Example: "Credit score needs improvement to 750.")
5. **Encourage the Change**

- "Bring your {FOCUS} to {TARGET}."
  (Example: "Bring your savings to $10,000.")

Templates for Mutable Indirectly (M.I.)

1. **Target and Method**
   - "{METHOD} to {ADJUST} your {FOCUS} to {TARGET}."
     (Example: "Reduce expenses to increase savings to $10,000.")
2. **Highlight the Indirect Feature**
   - "{FOCUS} depends on {RELATED_FEATURE}. {ADJUST} it to {TARGET}."
     (Example: "Savings depend on discretionary spending. Reduce it to $500.")
3. **Flexible Contextual Presentation**
   - "{FOCUS}: {CONTEXT}. {METHOD} to {ADJUST} to {TARGET}."
     (Example: "Debt-to-income ratio: High due to current debt. Pay off debt to reduce it.")
4. **Direct Call to Action**
   - "To {ADJUST} {FOCUS}, {METHOD}. Aim for {TARGET}."
     (Example: "To improve your credit score, pay outstanding balances. Aim for 750.")
5. **Outcome-Oriented Guidance**
   - "{TARGET} for {FOCUS} requires {METHOD} with {RELATED_FEATURE}."
     (Example: "A $10,000 savings goal requires reducing rent expenses.")

The following is the chat to generate examples for Other Candidate Feature templates.

The pdf of the paper ‘Towards Feasible Counterfactual Explanations: A Taxonomy Guided Template-Based NLG Method.’ by Salmi et. al., was provided as additional context to the prompt.

Prompt:

I am trying to create templates like the taxonomy guided templates. I have features called other candidate features, which are 2 features that can be explored to improve loan approval chances. These features don't have any value, the generated template sentence will be more sort like a suggestion to explore other features that could potentially be changed to improve loan approval chances

Response:

Thanks for the detailed context! Based on the paper and your goal of crafting a **template sentence** for “other candidate features” (i.e., features that were not modified in the counterfactual but could be explored for improving loan approval chances), here's a structured approach:

Context:

- These "other candidate features" are not changed in the current explanation.
- They don't have values attached to them.
- The goal is to **suggest** these features as **potential areas to explore** — not direct actions.

Proposed Template Sentences

You can model these sentences similarly to how the taxonomy-based templates are structured — but using **soft suggestion language** to reflect lower certainty or indirect actionability:

Base Template for Other Candidate Features

"Additionally, you may wish to explore whether modifying your {FEATURE_1} or {FEATURE_2} could improve your chances of loan approval."

Variants to match tone and use-case:

1. **Soft Advisory**

"You could also consider reviewing aspects like {FEATURE_1} or {FEATURE_2} as potential areas for improvement."

2. **Exploratory/Diagnostic**

"It might also be helpful to examine your {FEATURE_1} and {FEATURE_2}, as they could influence the outcome."

3. **Encouraging Self-Reflection**

"Think about whether changes to {FEATURE_1} or {FEATURE_2} might positively impact your application."

4. **Action-Prompted Suggestion**

"Consider seeking advice on whether adjusting {FEATURE_1} or {FEATURE_2} could improve your approval prospects."

5. **Professional Advice Framing**

"Consulting a loan advisor about potential improvements in {FEATURE_1} or {FEATURE_2} might also be beneficial."

A file containing the feature explanations used to template slot values is provided as additional context to the follow up prompt.

Follow up Prompt:

The attached document contains all the possible feature explanations. Enhance the base template from the previous response to have a verb slot associated with each other candidate feature slot.

Response:

Thanks! Based on your feature_exp.yaml file and your earlier request, I've enhanced the **base template** for *other candidate features* to include **verb slots** associated with each feature. These verbs are written in a way that aligns with natural language expectations for counterfactual explanations — i.e., what a user might plausibly *do* or *influence* in relation to the feature.

Enhanced Template (With Verbs per Feature)

*"Additionally, you may wish to explore whether **[VERB_1]** your **[FEATURE_1]** or **[VERB_2]** your **[FEATURE_2]** could improve your chances of loan approval."*

Feature–Verb Suggestions (from your file)

Feature	Description	Suggested Verb
LOAN	the loan request amount	reducing
MORTDUE	the amount due on existing mortgage	lowering
VALUE	the value of current property	reassessing
DELINQ	the number of delinquent credit lines	decreasing
NINQ	the number of recent credit inquiries	minimizing
CLNO	the number of credit lines	optimizing
DEBTINC	the debt-to-income ratio	improving

Follow up Prompt:

Provide more variations to the template

Response:

Absolutely! Here are **multiple variations** of the template sentence for “other candidate features,” incorporating both a **verb** and **feature description** slot for each feature. These templates vary by tone (neutral, supportive, advisory, professional) and structure (question form, suggestion form, conditional, etc.), giving you more flexibility for Natural-XAI generation.

Neutral/Supportive Suggestions

1. **Base Form (enhanced)**

"Additionally, you may wish to explore whether **[VERB_1]** your **[FEATURE_1]** or **[VERB_2]** your **[FEATURE_2]** could improve your chances of loan approval."

2. "You could consider taking steps to **[VERB_1]** your **[FEATURE_1]** or **[VERB_2]** your **[FEATURE_2]** as these may influence your application outcome."
3. "There may be potential in **[VERB_1]-ing** your **[FEATURE_1]** or **[VERB_2]-ing** your **[FEATURE_2]** to support a more favorable decision."
4. "Modifying aspects like **[FEATURE_1]** (by **[VERB_1]-ing**) or **[FEATURE_2]** (by **[VERB_2]-ing**) may also help strengthen your application."

Question/Reflective Framing

5. "Have you considered whether **[VERB_1]** your **[FEATURE_1]** or **[VERB_2]** your **[FEATURE_2]** could improve your approval prospects?"
6. "Might changes to your **[FEATURE_1]** (e.g., **[VERB_1]**) or **[FEATURE_2]** (e.g., **[VERB_2]**) help move your application toward approval?"

Professional/Advisory Tone

7. "It may be advisable to review areas such as **[FEATURE_1]** and **[FEATURE_2]**, where **[VERB_1]** and **[VERB_2]**, respectively, could support a better outcome."
8. "Consulting with a loan specialist on whether to **[VERB_1]** your **[FEATURE_1]** or **[VERB_2]** your **[FEATURE_2]** could provide additional insights."
9. "For additional improvement, consider seeking advice on **[VERB_1]-ing** your **[FEATURE_1]** or **[VERB_2]-ing** your **[FEATURE_2]**."

Conditional / Scenario-Based

10.

"If possible, try to **[VERB_1]** your **[FEATURE_1]** or **[VERB_2]** your **[FEATURE_2]**, as these may be contributing factors."

11.

"Should circumstances allow, changes such as **[VERB_1]** your **[FEATURE_1]** or **[VERB_2]** your **[FEATURE_2]** may positively impact your application."

Encouraging/Empowering Language

12.

"You have the opportunity to further improve your application by **[VERB_1]-ing** your **[FEATURE_1]** or **[VERB_2]-ing** your **[FEATURE_2]**."

13.

"Taking initiative to **[VERB_1]** your **[FEATURE_1]** and possibly **[VERB_2]** your **[FEATURE_2]** might enhance your approval likelihood."

Appendix E: Duplicate Input Results

The following table shows the results of scores returned with and without duplicate input Lexical Substitution method is used to pass the masked sentence to the RoBERTa-base model.
Input Sentence: Reduce the loan request amount from $5,000 to $1,000.

Without LS	**With LS**
reduce: 0.000666121719405055	reduce: 0.00032950853346847
cut: 0.0004971809103153646	**decrease**: 1.4687726434203796e-05
decrease: 0.00018212561553809792	cut: 1.2547124015327427e-06
trim: 4.829671979678096e-06	trim: 1.58399103611373e-07
shrink: 3.1156339446170023e-06	shrink: 4.683206000777318e-08
lessen: 5.782601988357783e-07	diminish: 4.195293712427883e-08
shorten: 3.424092369641585e-07	lessen: 3.5631288142212725e-08
diminish: 1.8611301300097693e-07	shorten: 1.2231002877172159e-08
tighten: 1.0011306450508073e-08	cut down: 3.259447013642706e-09
thin: 3.5604277304202014e-09	trim down: 2.1776136015665914e-09
contract: 1.663213122782281e-09	dilute: 1.6999802073100455e-09
concentrate: 3.06806163807849e-10	tighten: 2.4306789914163573e-10
cut down: 1.7577258107724726e-10	contract: 2.41662162503431e-10
cut back: 3.3271903968128827e-12	abbreviate: 1.5506636216500782e-10
scale down: 2.9479604504044563e-12	thin: 1.0963035340649796e-10
trim down: 1.7997746583730133e-12	scale down: 1.0225573940641567e-10
abbreviate: 7.389261602862051e-15	minify: 7.653730818202373e-11
quash: 8.780964110217189e-16	concentrate: 7.525815387543489e-11
dilute: 8.592633358494358e-16	cut back: 2.8361457286398208e-11
abridge: 1.5586241877049529e-16	subjugate: 1.8269241869483146e-12
minify: 1.3677841952714979e-16	subdue: 1.569006383608921e-12
subdue: 1.1996058618020728e-16	quash: 3.9082000879473265e-13
subjugate: 4.461279902387065e-17	abridge: 3.044674268709056e-14
lose weight: 6.226210456113419e-19	lose weight: 3.337877858859668e-17
foreshorten: 2.4281120748838267e-21	foreshorten: 1.0294963736253275e-19
deoxidize: 2.32365578033243e-22	deoxidize: 6.445848351399182e-20

With duplicate input lexical input substitution method, we notice that ‘decrease’, one of the original slot values, is ranked higher than the other synonyms. Synonyms that aren’t good fit are still scored lower when compared to without duplicate input model scores.

Appendix F: Synthetic Data Generation

Prompt:
I want to generate simple natural language explanations utilizing the counterfactual feature changes suggested in these sentences and the other features that can influence a positive change to improve loan applications.
Give me an example of how such explanations will look like for these examples.

Look at ways to increase the property value from $8,000 - $66,100 to $120,001 - $347,000. Look at ways to reduce the number of recent credit inquiries from 8 to 2. Pay off delinquent credit lines and review the number of open credit lines, to improve the chances of loan approval.
Reduce the number of delinquent credit lines from 7 to 6. Explore ways to pay down the amount due on the existing mortgage from $65,001 - $91,500 to $2,060 - $46,300. Reappraise the value of the property or review the number of open credit lines, to improve the chances of loan approval.
Increase the number of credit lines to 16 - 20. Review the loan request amount and review recent credit inquiries, to improve the chances of loan approval.

Note: Additional instructions can be given to the above prompt like,
1. Try to keep original values, where possible.
2. Also, do not use first person or second person language. Basically, avoid using 'you' or 'your'

These or new instructions maybe added depending on the generated explanations.

Response:
Here's how simple and clear natural language explanations for the provided examples might look:
Example Explanations:
1. To increase the chances of loan approval, consider significantly increasing your property value from between $8,000 - $66,100 to around $120,001 - $347,000. Additionally, reduce the number of recent credit inquiries from 8 to about 2. Paying off any delinquent credit lines and reviewing your total number of open credit lines can also help positively influence the loan decision.
2. Improving your likelihood of loan approval may be achieved by reducing delinquent credit lines from 7 down to 6 and substantially paying down your existing mortgage from a current range of $65,001 - $91,500 to approximately $2,060 - $46,300. Additionally, reappraising your property's value or reviewing your open credit lines can further enhance your application's strength.
3. You can enhance your loan application by increasing your total number of credit lines to between 16 and 20. It's also important to review your requested loan amount and manage recent credit inquiries effectively, as these factors can positively influence the loan approval decision.

<If the generated responses aren't copying the values correctly, the following prompt worked well.>
Follow up Prompt:

make sure to use the original values in the explanations.

Look at ways to reduce the number of recent credit inquiries from 10 to 4. Look at ways to increase the property value from $66,101 - $89,200 to $89,201 - $120,000. Review the loan request amount and pay off delinquent credit lines, to improve the chances of loan approval. Look at ways to reduce the number of recent credit inquiries to 0. Explore ways to pay down the amount due on the existing mortgage from $65,001 - $91,500 to $46,301 - $65,000. Reduce the debt-to-income ratio and pay off delinquent credit lines, to improve the chances of loan approval. Explore ways to pay down the amount due on the existing mortgage to $2,060 - $46,300. Decrease the loan request amount from $23,301 - $89,900 to $16,301 - $23,300. Reduce the debt-to-income ratio or pay off delinquent credit lines, to improve the chances of loan approval.

Response:
Here are clear, concise, natural-language explanations explicitly utilizing the original numerical values for each of your sentences:

1. Reducing recent credit inquiries from 10 to 4, increasing property value from $66,101–$89,200 up to $89,201–$120,000, carefully reviewing the loan request amount, and paying off delinquent credit lines can greatly improve your loan approval likelihood.
2. Completely eliminating recent credit inquiries (down to 0), significantly reducing the existing mortgage balance from $65,001–$91,500 down to $46,301–$65,000, lowering your debt-to-income ratio, and paying off delinquent credit lines will positively impact your application.
3. Paying down your mortgage to between $2,060–$46,300, decreasing your loan request from $23,301–$89,900 down to $16,301–$23,300, and either reducing your debt-to-income ratio or paying off delinquent credit lines can significantly enhance loan approval chances.

<If greater variation in the generated natural language explanations or adjustments to sentence structure are desired, the following prompt can be used. *Please note* with repeated prompting, the model may begin to settle into a fixed pattern, and additional instructions might be required to adjust its output accordingly.>

Prompt:
Here is the next batch. Make sure to use the original values, explanations are varied, a mix of single or two or three sentences wherever it makes sense.

Look at ways to decrease the property value from $66,101 - $89,200 to $8,000 - $66,100. Increase the loan request amount from $11,101 - $16,300 to $23,301 - $89,900. Pay off delinquent credit lines or review recent credit inquiries, to improve the chances of loan approval.

Reduce the debt-to-income ratio from 35% - 39% to 30% - 34.8%. Look at ways to reduce the number of recent credit inquiries from 8 to 7. Reappraise the value of the property and review the loan request amount, to improve the chances of loan approval.

Increase the number of credit lines from 21 - 26 to 27 - 71. Look at ways to reduce the number of recent credit inquiries from 2 to 0. Review the loan request amount or reduce the debt-to-income ratio, to improve the chances of loan approval.

Response:

1. Consider lowering your property value from the range of $66,101 - $89,200 down to $8,000 - $66,100, while simultaneously increasing the loan request from $11,101 - $16,300 up to $23,301 - $89,900. Additionally, clearing delinquent credit lines or thoroughly reviewing recent credit inquiries will enhance your chances of loan approval.
2. To strengthen your loan application, reduce your debt-to-income ratio from 35% - 39% to 30% - 34.8% and lower the number of recent credit inquiries from 8 to 7. Also, consider reappraising your property's current valuation and reviewing your requested loan amount to further increase your approval prospects.
3. Improve your loan application by increasing your number of credit lines from 21 - 26 to 27 - 71 and eliminating recent credit inquiries entirely, reducing from 2 to 0. Also, review your requested loan amount or consider lowering your debt-to-income ratio to boost approval odds.