Specializing Small Language Models into Business and Industry Idea Reviewer Experts with Supervised Fine-Tuning

Gabriel Braga Ladislau[1,2], Guilherme Ratti Moraes[1,2], Guilherme Goes Zanetti[1,2], Abner Grahan Jacobsen[2], Claudine Badue[1], Alberto Ferreira De Souza[1,2], Thiago Oliveira-Santos[1]

[1]Department of Informatics – Universidade Federal do Espírito Santo (UFES)
Vitória – Espírito Santo – Brazil

[2]Research Department – Aumo S.A.
Vitória – Espírito Santo – Brazil

{gabriel.ladislau,guilherme.r.moraes}@edu.ufes.br

{guilherme.zanetti, claudine, alberto}@lcad.inf.ufes.br

abner@aumo.ai, todsantos@inf.ufes.br

Abstract. ***Research Context:*** *The application of Natural Language Models in industrial and business environments is rapidly expanding. While powerful, these models often require specialization to match the performance of human experts.* ***Practical Problem:*** *Large Language Models (LLMs) face two major barriers for enterprise adoption: 1) the lack of specific, private knowledge required for nuanced tasks, such as classifying internal company innovations, and 2) the operational costs are prohibitively high for long-term, large-scale use.* ***Proposed Solution:*** *We propose a cost-effective alternative by fine-tuning Small Language Models (SLMs) and encoder models (BERTs) in business ideas classification, transforming them into expert systems tailored to a company's unique context.* ***Related IS Theory:*** *This research is grounded in Task-Technology Fit (TTF) theory, examining the alignment between the task's characteristics (classifying specialized ideas) and the technology's attributes (general-purpose LLMs vs. fine-tuned SLMs and BERTs) to determine the optimal fit.* ***Research Method:*** *The research involves developing and evaluating a training method for SLMs and BERTs, with real-world data augmented by an artificial dataset. Additionally, the artificial dataset creation pipeline is showcased by the research. The performance of the resulting SLMs and BERTs are then compared against that of larger, general-purpose LLMs.* ***Results:*** *The findings indicate that the fine-tuned SLMs and BERTs achieve superior performance on the specialized classification task compared to larger, non-fine-tuned LLMs, while significantly reducing operational costs. The results also highlight that augmenting scarce real-world data with diverse artificial data can lead to a more robust, generalizable and rich model.* ***Contributions:*** *This work contributes to a practical and economically viable method for specialized AI agents creation and augmentation of scarce real-world data through synthetically made datasets. Its impact lies in enabling businesses to deploy tailored, high-performing AI solutions for specific and knowledge-based tasks without the high costs of large-scale, general-purpose models.*

1. Introduction

Currently, the use of natural language models across diverse applications for solving industrial and business problems is widespread. Recent years have witnessed a growing utilization of these models, with continuous, ever-growing demand across multiple domains. This increase of interest is justified when looked at the strikingly high performance of intelligent systems on well-defined problems (see for example [Wang et al. 2024] and [Chinnalagu 2024]). Nevertheless, a common barrier met by generalist models resides in tasks that require very specific knowledge, often having poor results that are far from what would be expected of a human specialist [Arvidsson et al. 2024]. This problem becomes even more pronounced when considering the subjectivity with which the same task may be interpreted across different domains, such as distinct corporate environments [Javaid et al. 2023]. These are important factors to be taken in consideration when developing tailored intelligent systems for specific applications [Javaid et al. 2023].

One of the many domains of interest for corporate management is the proper classification of innovative ideas. This process is of utmost importance for companies that aim to efficiently improve their infrastructure and services over the time, determining which ideas are feasible [López and Oliver 2023]. However, this process is extremely specific, to the point where two different companies that provide similar services may greatly diverge in approach. While Large Language Models (LLMs), such as GPT4-o, may be able to reasonably automatize the process of idea pruning, as we show in Section 5, this level of specificity often restricts GPT's performance due to its scarce knowledge of each company's specific preferences. In addition to that, the usage of LLMs for large-scale, long-term applications of automatization are prohibitively high for real-worlds business applications [Aryan et al. 2023].

In the lights of these insights, a viable alternative is to train Small Language Models (SLMs) and small encoder models, such as BERTs, into specialists for the idea classification task. While LLMs often require an extensive budget to operate, SLMs and BERTs are a more cost-effective alternative that can perform better than LLMs with a much lower cost [Irugalbandara et al. 2024]. With this, we developed an intelligent system with low budget that efficiently automatizes the process of idea classification with high performance.

The developed intelligent system falls within the domain of Information Systems, as it integrates technological and intelligent processing components to support management and decision-making in the classification of ideas. By leveraging fine-tuned language models (SLM and BERT), the system can automatically interpret and categorize textual data, converting it into organized and meaningful information. In this regard, the system functions as a decision-support subsystem, enhancing the efficiency of information flows and contributing to knowledge management within the organization.

In addition to the proposed system, a data generation method was developed to enhance the model's training process. While the method itself is grounded in natural language processing techniques, its integration contributes to the overall efficiency and robustness of the proposed intelligent system. We show in Section 3 that our synthetically developed dataset not only successfully retains insightful behavior from real-world data, but also expands its knowledge to different domains that are otherwise not covered by the real-world dataset. Consequently, we show in Section 5 that augmenting real-world data with our artificial dataset significantly enhances the intelligent system robustness and richness with very little trade-off, achieving near peak performance in a larger domain than what could otherwise be accomplished with the scarce real world dataset. All in all, the main contributions of this study are summarized as follows:

- **Development of an intelligent idea classification system:** Based on fine-tuned language models (LLM and BERT), the system automatically interprets and categorizes unstructured textual data.
- **Integration into the Information Systems domain:** The proposed system functions as a decision-support mechanism, enhancing the efficiency of information flows and contributing to organizational knowledge management.
- **Proposal of a synthetic data generation method:** A data augmentation approach was developed to expand and balance the training dataset, improving model performance and robustness.
- **Evaluation, validation, and data availability:** The effectiveness of the proposed approach was demonstrated through experiments with both real and artificially generated datasets, achieving accurate idea classification across multiple domains. The resulting synthetic dataset is publicly available alongside models trained on synthetic data[1] to support future research in this area.

2. Related Works

Research on language models for text classification has explored different strategies to balance accuracy, generalization, and efficiency. Chae and Davidson [Chae and Davidson 2025] investigate the application of LLMs to the supervised text classification task of stance detection. They compare zero-shot prompting, few-shot prompting, fine-tuning, and instruction-tuning across a range of model sizes. Their results show that larger models generally achieve better performance with minimal supervision, while smaller models benefit from fine-tuning due to lower cost and competitive accuracy.

Sun et al. [Sun et al. 2023] propose Clue and Reasoning Prompting (CARP), a method designed to improve LLM performance on classification tasks involving complex linguistic phenomena such as irony and contrast. CARP introduces a progressive reasoning process, combining shallow clue detection with deeper diagnostic reasoning. It also integrates kNN retrieval to enhance context selection. The method achieves state-of-the-art results on several benchmarks. Nevertheless, the approach requires additional components and big capable pre-trained models, which increases complexity and cost.

The trade-off between large and small models is examined by Lepagnol et al. [Lepagnol et al. 2024], who evaluate SLMs in zero-shot text classification across 15 datasets. Their findings demonstrate that smaller models, even without fine-tuning, can match or surpass LLMs in some cases. This challenges the assumption that larger models are always superior. However, some datasets still show clear advantages for LLMs.

Focusing on specialized domains, Fatemi et al. [Fatemi et al. 2025] study instruction fine-tuning of smaller LLMs for financial text classification. They show that models such as Mistral-7B and Llama3-8B can be effectively adapted through instruction fine-tuning, and that model merging further improves zero-shot performance. While the

[1] https://github.com/LCAD-UFES/Specializing-Small-Language-Models-with-SFT

approach delivers robust results in financial contexts, it also reveals that base model fine-tuning can degrade generalization if not combined with instruction-tuning, limiting its portability across tasks.

Another line of work explores synthetic data generation. Li et al. [Li et al. 2023] analyze the effectiveness of LLM-generated synthetic datasets for training classifiers. They find that while synthetic data can reduce data collection costs, its utility varies by task. Specifically, tasks with high subjectivity suffer from significant performance degradation when trained solely on synthetic data. This highlights the limitations of synthetic generation as a replacement for curated datasets, though it remains useful for augmentation.

The present work builds on these directions by proposing a cost-effective methodology to specialize SLMs into domain-specific classifiers for business idea evaluation. Unlike Chae and Davidson [Chae and Davidson 2025] and Sun et al. [Sun et al. 2023], who emphasize large models and complex prompting strategies, this work shows that smaller fine-tuned models can achieve superior performance on specialized tasks. It also differs from Lepagnol et al. [Lepagnol et al. 2024] by demonstrating the advantages of fine-tuning over zero-shot use of SLMs. Similar to Fatemi et al. [Fatemi et al. 2025], it focuses on domain adaptation, but in the business and innovation context rather than finance. Finally, consistent with Li et al. [Li et al. 2023], this work leverages synthetic data, while applying refinement mechanisms to address diversity and domain-gap limitations. In doing so, it contributes a practical and economically viable solution for industry applications where LLMs are costly and less adaptable.

3. Idea Classification System

This section details a systematic approach for developing an intelligent idea classification system, customizable for specific corporate requirements. In parallel, a pipeline was developed for data augmentation. This LLM-backed approach enhances the dataset's robustness, generalization, and richness through synthetic data generation. The methodology is segmented into three primary stages: data pre-processing, the fine-tuning of language models, such as the encoder-based BERT and the prompt-oriented SmolLM, and finally, classification inference. The subsequent subsections will elaborate on each stage.

3.1. Data Pre-Processing

This subsection covers a comprehensive range of data processing tasks, from handling real-world data characteristics to generating and augmenting synthetic data.

3.1.1. Proprietary Dataset - Real World Ideas

The initial phase of this work utilized a data collection provided by a private company. This proprietary dataset consisted of 5,000 suggestions (henceforth referred to as 'ideas') concerning improvements in business, logistics, and infrastructure. A subset of these were classified as high-quality, innovative proposals, while the remainder lacked critical features that diminished their value. Consequently, the dataset contains two balanced classes: 'ideas' and 'not ideas'. The classification was performed by domain experts who thoroughly assessed the strengths and weaknesses of each idea, resulting in a balanced distribution of 2,500 examples for each class.

Each idea was structured into three core fields: a title, a description, and the expected benefits. These suggestions were manually authored by employees of the partner company. Although complete submissions were encouraged, the 'benefits' field was frequently left blank, with its content either omitted or implicitly included within the description. Nevertheless, the absence of an explicit benefits statement did not lead to automatic rejection; other important aspects, such as pertinence and innovation capacity, were still carefully considered in the quality assessment.

The primary value of this dataset lies in its encapsulation of key information from real-world business scenarios, possessing a richness and depth that the models can leverage to learn domain-specific patterns. While this is a significant feature, it introduces copyright and confidentiality issues, as the dataset contains sensitive information. For this reason, the proprietary dataset cannot be publicly disclosed. To overcome this limitation, a new artificial dataset was constructed, and designed, to mimic real-world data characteristics without exposing sensitive information.

3.1.2. Artificial Dataset - LLM Created Ideas

The inherent complexity of human writing presents a significant challenge when mimicking natural language. To address this, the methodology leverages powerful, pre-trained LLMs, such as GPT-4o and Qwen3-14B, to create a synthetic dataset of ideas and not ideas. However, using a generic LLM would compromise the objective of tailoring the model to specific corporate needs. Furthermore, LLM-based data generation often suffers from low diversity and informational richness, frequently generating repetitive content across different samples. This paper's approach systematically addresses these core issues by implementing an efficient, robust, and systematic pipeline for the mass production of synthetic data that emulates real-world datasets.

To specialize the data generation to specific domains while simultaneously introducing variability and richness, it was utilized a small subset of the proprietary dataset to serve as a reference for the generation process. Guidelines were also carefully established to constrain idea generation to one of ten distinct corporate domains, such as fintech, healthcare, e-commerce, etc... (see in the Artificial Dataset Creation Prompt the company_type variable). In total, 500 examples from the proprietary dataset were reserved as references. The generation process created 10 new synthetic examples for each combination of a company domain and a set of 5 reference examples (see in the Artificial Dataset Creation Prompt the real_world_example_ variables). By systematically varying the domains and ensuring that the same set of 5 references was not reused, it was introduced high variability and robustness into the generation pipeline, successfully mimicking the richness and depth of real-world data. The precise prompt structure used to guide this generative process is detailed below for full transparency and reproducibility.

This method yielded 1,000 artificial ideas for each of the 10 domains, totaling 10,000 new, synthetic examples. The prompt used to generate the ideas, and not ideas, is detailed in the frame named Artificial Dataset Creation Prompt below.

Artificial Dataset Creation Prompt

You are tasked with generating synthetic data samples for a binary classification problem. The goal is to create structured suggestions that would be submitted to the management team of a company.

# Important Instructions

- Do not include any hidden reasoning, intermediate thoughts, or explanations in your response.
- Only provide the final dataset in valid JSON format.
- The JSON must be properly structured and machine-readable (no comments, no trailing commas).
- Each record should follow the schema below:

{ ”Title”: ”A short title summarizing the idea”, ”Description”: ”An idea that directly contributes to the company”, ”Benefit”: ”The advantages the company would gain if the idea is applied”, ”Classification”: 0 or 1 }

# Classification Guidelines

## Ideas (1):

- Must propose a concrete action or solution.
- Clearly describes the expected impact or area of change.
- Contains a purpose (explicit or implicit) with a goal, benefit, or problem resolution.
- Written in a way that is easy to understand and interpret.

## Not Ideas (0):

- Only describes an issue without proposing an action.
- Personal requests that lack collective benefit.
- Neutral comments or compliments without intent of improvement.
- Vague, ambiguous, or generic statements without direction.
- Reports of situations without suggesting what should be done.
- May mention potential benefits, but they are not substantial enough to be classified as an idea.

# Text Generation Guidelines

- Suggestions must be consistent with the context of a {company_type}.
- The Description field must present a practical and relevant proposal to improve services, products, operations, or infrastructure.
- The Title field should summarize the suggestion concisely.
- Real company or product names should not be mentioned.
- Do not reuse text from the examples provided — use them only as inspiration.
- Ensure originality and semantic diversity across generated records.

# Examples for Reference (DO NOT COPY)

- {real_world_example_1}
- {real_world_example_2}

- {real_world_example_3}
- {real_world_example_4}
- {real_world_example_5}

# Task

Generate 10 synthetic records according to the schema above. Exactly 5 must be classified as `0` and 5 as `1`.

3.1.3. De-duplicating Artificial Data

It is imperative to validate the LLM's output, as it could produce similar or identical examples or inadvertently replicate content from the proprietary reference data. To mitigate this risk, it was developed an evaluation mechanism to validate and refine the synthetic dataset generated.

A primary concern in LLM-based synthetic data generation is ensuring originality. This issue is exacerbated by the risk of leaking sensitive information from the proprietary reference data. With this in mind, a multi-stage cosine similarity analysis was employed as a mechanism to detect and eliminate duplicate or near-duplicate content. This mechanism was used not only to prevent potential leakage of sensitive data from the proprietary dataset but was also instrumental in maximizing the diversity of the synthetic dataset. By identifying and removing redundant information, it was ensured that the final dataset remained of high quality. A diagram of this process is shown in Figure 1.

As illustrated in the workflow, the first step is to generate the vector embeddings for each data sample. Subsequently, pairwise cosine similarity calculations are performed across two distinct configurations: (i) a comparison of synthetic data against itself (intra-dataset evaluation) and (ii) a comparison of synthetic data against the proprietary data (inter-dataset evaluation). This allows establishing a robust score that quantifies the similarity between different ideas. The resulting similarity score alone is insufficient for a definitive conclusion. Therefore, the idea pairs were sorted by their scores in descending order and conducted a meticulous qualitative review of those exhibiting high similarity. It was observed that a similarity score of 0.90 served as a critical threshold for the artificial intra-dataset comparison; below this value, the ideas demonstrated significant diversity. The inter-dataset evaluation revealed no instances of replicated information from the proprietary dataset, even at the highest similarity scores. Consequently, all synthetic ideas that scored 0.90 or higher in similarity were removed, randomly retaining only one unique example from each cluster of similar items, resulting in 8,315 synthetic examples from the starting 10,000.

3.1.4. Enhancing Dataset Complexity via Class Boundary Refinement

To further refine the synthetic dataset and ensure it mirrored the nuanced complexity of the proprietary data, a detailed analysis of class separability using pairwise cosine similarity method was conducted. This involved comparing vector embeddings both within the same class (intra-class), between opposing classes (inter-class) for both the real-world and artificial datasets, as well as a similarity between the opposing datasets (inter-dataset). The

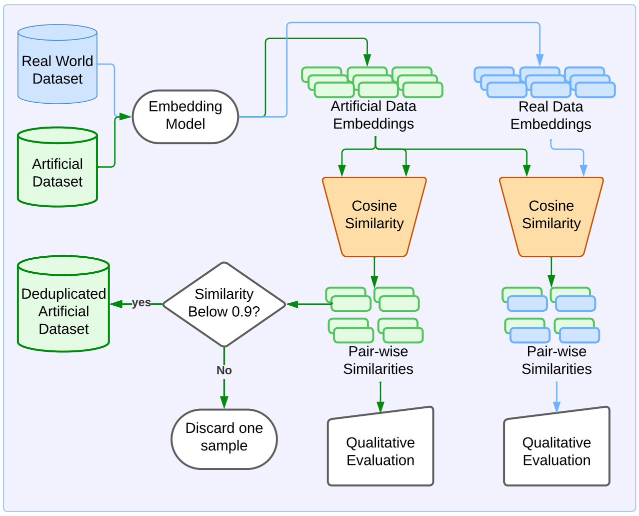

Figure 1. Deduplication Workflow.

analysis revealed a significant discrepancy: the inter-class similarity within the artificial dataset was substantially lower than that observed in the real-world dataset. This indicated that the synthetic classes were overly distinct, potentially simplifying the classification task and failing to capture the subtle ambiguities present in genuine data. To address this and increase the dataset's difficulty, it was implemented an additional data enhancement step. This process involved randomly selecting 2,000 examples from each class in the artificial dataset and reversing their labels. Subsequently, an LLM was prompted to perform minor textual modifications to these selected examples. The objective was to subtly alter the content to justify the new classification, effectively creating ambiguous samples that reside near the decision boundary between the 'idea' and 'not idea' classes. By introducing these challenging, borderline cases, the method developed aimed to compel the model to learn more granular and robust features, thereby improving its generalization performance on complex, real-world examples.

3.1.5. Evaluating Final Artificial Dataset Quality

Finally, the synthetic dataset was evaluated against the original real-world dataset to assess its suitability for data augmentation. The final evaluation consisted of the same cosine similarity method described in the previous subsection, in addition to a Uniform Manifold Approximation and Projection (UMAP) [McInnes et al. 2020]. The Table 1 presents a comparative summary of key statistical and semantic metrics for both datasets.

Table 1. Comparative Metrics: Real vs. Synthetic Datasets

Metric	Real Dataset	Synthetic Dataset
Number of Samples	5,000	8,315
Label Distribution	'Not Idea': 2,500; 'Idea': 2,500	'Not Idea': 4,248; 'Idea': 4,067
Avg. Pairwise Similarity (Inter-Dataset)	0.2692	0.2313
Avg. Pairwise Similarity (intra-class 'Not Idea')	0.2890	0.2222
Avg. Pairwise Similarity (intra-class 'Idea')	0.2940	0.2930
Avg. Pairwise Similarity (inter-class)	0.2470	0.2064

From a semantic perspective, the synthetic dataset reveals two notable characteristics. First, it successfully replicates the internal cohesion of the Idea class: its average intra-group similarity (0.2930) is almost identical to that of the real dataset (0.2940). This demonstrates that the generative model was able to produce examples that are thematically consistent and representative of this class. In contrast, the Not Idea class shows a slight decline in intra-group similarity, from 0.2890 to 0.2222. The drop in average similarity indicates that the synthetic dataset is more diverse than the original real dataset, suggesting that our generation process produced a wider range of examples rather than repeating the same patterns.

Additionally, for the purpose of data augmentation, the synthetic dataset reduced the excessive separation between classes: the cross-group similarity shifted from 0.2064 to 0.2470. Although the synthetic data still presents lower inter-class similarity, this difference may stem from the inclusion of 10 distinct company domains in the artificial dataset, compared to a single domain in the real data. Ultimately, the generator appears to have learned to create more "prototypical" or archetypal examples of each class, which is highly advantageous for training a classifier to establish a more robust decision boundary. To further explore the geometric structure of the embedding space, UMAP was employed to visualize the datasets, with Figure 2 providing clear visual confirmation of the similarity metric findings.

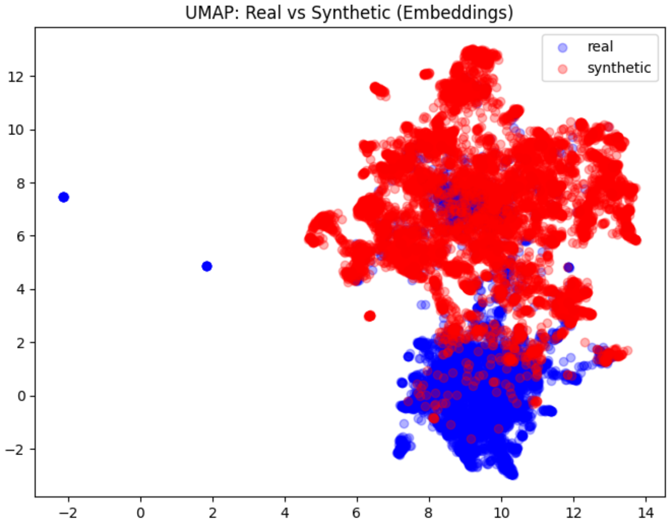

Figure 2. UMAP projection of real (left) and synthetic (right) data embeddings.

The visualization contrasts the real dataset (blue dots) with the synthetic dataset (red dots), clearly revealing the company domains represented in each. Although there are minor overlaps, the two clusters remain strongly distinct, showcasing different behaviors. The real data is more concentrated and localized, whereas the synthetic data spreads more broadly across domains. This wider dispersion offers a direct and compelling visual of the multiple company domains captured within the single synthetic dataset.

The synthetic dataset stands out as a powerful augmentation tool. While it does not fully mirror the linguistic richness of the real data, its ability to produce a large volume of samples with sharper and broader separation across domains is a decisive strength. Models trained on this data are better equipped to capture the fundamental features that distinguish Ideas from Not Ideas in a more generalist way. When paired with the original, more localized real-world data, the synthetic dataset amplifies learning and enables the development of a classification model that is not only more robust, but also more adaptable and generalizable.

3.2. Language Model Fine-Tuning Methodology

Fine-tuning is a transfer learning technique where a pre-trained model's parameters are further trained on a specific, downstream task. This process allows leveraging the powerful, generalized representations learned by large models while adapting them to the specialized idea classification task. While various fine-tuning methods exist, Low-Rank Adaptation (LoRA) [Hu et al. 2021] was selected as the primary strategy for the developed method. LoRA is particularly well-suited for this work's scenario as it offers significant computational efficiency and is highly effective for datasets of limited size. It operates by freezing the original model weights and injecting trainable, low-rank matrices into specific layers. This approach fundamentally constrains the scope of weight updates, which not only reduces the number of trainable parameters but also acts as a form of implicit regularization, mitigating the risk of catastrophic forgetting and overfitting.

The selection of model architectures, for the method's fine-tuning step, was guided by the HuggingFace Open LLM Leaderboard [2]. At the time this work was developed SmolLM-3B and mmBERT offered the most favorable cost-to-performance ratio. Their relatively small size (3 billion and 307 million parameters, respectively), combined with strong leaderboard scores and open-source availability, made them ideal candidates for the experiments conducted. The specific configurations for each model are detailed in the following subsections within this section.

3.2.1. SmolLM3-3B Fine-Tuning Configuration

For the fine-tuning of the SmolLM3-3B model, a decoder-style architecture, it was implemented a comprehensive LoRA configuration designed to adapt its core attention and feed-forward network components. The rank of the decomposition, r was set to 64. This hyperparameter determines the dimensionality of the trainable matrices, where a higher rank allows for more expressive adaptations. It was selected a relatively high value to empower the model to capture the nuanced patterns specific to the idea classification task chosen. The scaling factor, α, was set to 32. This parameter scales the learned weight updates, with the effective learning contribution being proportional to $\frac{\alpha}{r}$. This configuration results in a scaling factor of 0.5, which tempers the magnitude of the updates to prevent destabilizing the model's pre-trained knowledge. To provide regularization and enhance generalization, a dropout rate of 0.2 was applied to the LoRA adapter layers. Also, it was targeted a broad set of modules within the model's transformer attention blocks to ensure a thorough adaptation. This included the query, key, value, and output projections of the attention mechanism, as well as the gating, downward, and upward projections of the feed-forward layers. This comprehensive strategy allows for a complete adjustment of both how the model weighs tokens and how it processes information.

[2] https://huggingface.co/open-llm-leaderboard

3.2.2. mmBERT Fine-Tuning Configuration

The mmBERT model, being an encoder-based architecture, does not naturally perform classification tasks. A classification head, a simple feed-forward neural network to project a multidimensional output into 2 class output, was attached on top of the pre-trained model output. This configuration was fine-tuned with a strategic LoRA application to adapt its core, attention blocks, while keeping the pre-trained model's weights frozen. However, the newly added classification remained unfrozen and prone to adaptation during the training process. This is necessary to align the weights of the new layer with the rest of the model. A consistent adaptation capacity was maintained by setting the LoRA rank to 64 and the alpha to 32, preserving the same stable update ratio of 0.5 used for SmolLM. A conservative LoRA dropout of 0.1 was selected to provide sufficient regularization for this more targeted adaptation. For this model, the adaptation was focused on key components within each transformer layer by targeting the adapter modules to attention layers W_{qkv}, W_o and W_i. This choice allows for a comprehensive adaptation of the model's core mechanisms: W_{qkv} is the combined query, key, and value projection matrix in the self-attention head; W_o is the output projection of the attention block; and Wi is the input projection to the feed-forward network. By targeting these modules alongside tuning the newly added classification head, both how the model computes contextual attention and how it processes the resulting representations were efficiently refined, ensuring a thorough adjustment for the classification task.

3.3. Classification Inference Pipeline

The final stage of the system is the inference pipeline presented in the Figure 3, which test the trained models to classify new, unseen ideas. This pipeline is designed for efficiency and reliability, integrating the complementary strengths of the fine-tuned models to deliver a robust final verdict. The entire process, from input to output, is automated to provide a seamless classification experience.

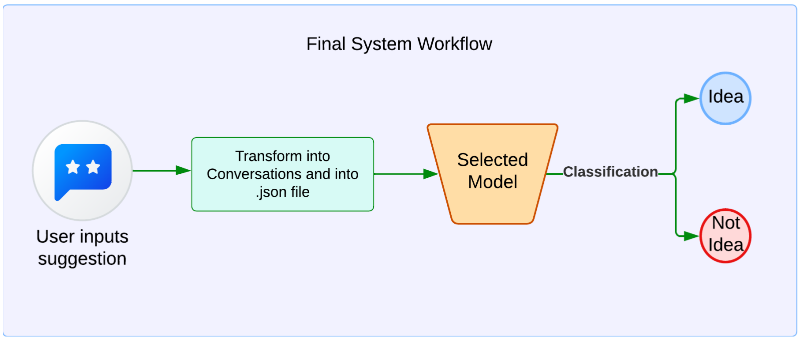

Figure 3. Automatic idea classification system workflow.

The pipeline is initiated when a new idea is submitted. The system first performs a pre-processing step where the textual content from the title, description, and benefits fields is concatenated into a single, unified text input. This consolidated string is then tokenized according to the specific format required by each of the fine-tuned models. The system's output is the final classification label (either 'idea' or 'not idea') which provides a rapid, data-driven assessment to help efficiently triage and prioritize employee suggestions.

4. Experimental Methodology

This section details the experimental methodology used to evaluate the proposed method. The process involves two main stages: data preparation and the cross-validation experiment, which are outlined below. This section also explains the statistical metrics employed to evaluate the models.

4.1. Data Processing and Division

The evaluation process begins with data preparation. Each raw dataset (in .csv format) is first transformed into a structured .json format. Following this, each dataset is partitioned into five folds using a stratified k-fold strategy. Stratification ensures that the class distribution (i.e., 'accepted' and 'rejected' ideas) within each fold is representative of the overall dataset, resulting in five balanced subsets.

4.2. Cross-Validation Experiments

To rigorously evaluate our approach, five different trained models were benchmarked across the three distinct datasets (Artificial, Real, and Combined). The core evaluation relies on a 5-fold cross-validation scheme for each model-dataset combination. See Figure 4 for an overview of this configuration.

In this procedure, one fold is held out as the validation set, while the remaining four folds are used to train the model. This process is repeated five times, with each fold serving as the validation set exactly once. By the end of the process, predictions have been generated for every sample in the dataset by a model that was never trained on it.

This cross-validation methodology is important for obtaining a reliable and unbiased estimate of each model's generalization performance. It ensures our results are robust by making efficient use of the entire dataset for both training and validation, thereby reducing the risk of performance bias from a single, arbitrary data split.

Also during this cross-validation evaluation a cross-dataset test was added, meaning all models have been tested by the different corresponding folds of the three different datasets. This enhances our performance evaluation and making our results robust.

4.3. Evaluation Metrics

To quantitatively evaluate the performance of the classification models, two standard statistical metrics were employed: accuracy and the confusion matrix. These were selected to provide both a high-level summary of performance and a detailed breakdown of the model's predictive behavior.

4.3.1. Accuracy

Accuracy was primarily utilized as a straightforward and effective measure of overall model performance. As defined in Equation 1, accuracy expresses the ratio of correct predictions to the total number of predictions, where TP represents True Positives, TN True Negatives, FP False Positives, and FN False Negatives.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

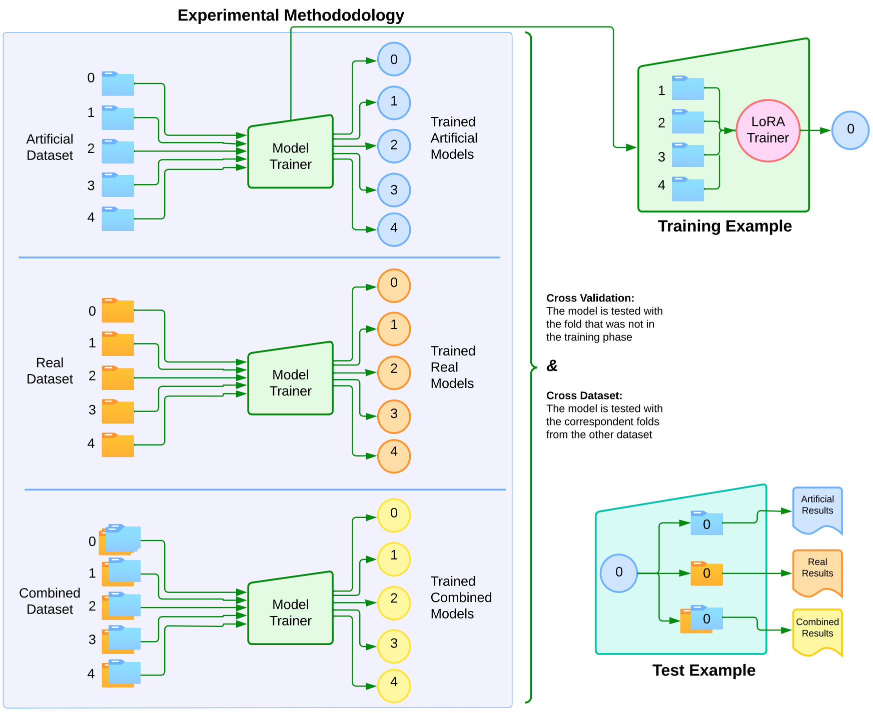

Figure 4. Experimental Methodology Overview. For each of the three datasets, a 5-fold cross-validation is performed. A model is trained using four folds and tested on the remaining hold-out fold.

Since both the proprietary and synthetic datasets used in the experiments were intentionally balanced between the ’idea’ and ’not idea’ classes, accuracy serves as a reliable and unbiased indicator of the model’s effectiveness. The absence of class imbalance negates the need for more complex metrics like weighted F1-scores or balanced accuracy, making this direct approach suitable for evaluation [Naidu et al. 2023].

4.3.2. Confusion Matrix

While accuracy provides a single score for overall performance, the confusion matrix offers a more granular analysis of the model’s classification decisions. For the binary classification task, the confusion matrix is a 2x2 table that visualizes the interplay between the actual and predicted labels. It breaks down the results into four cardinal outcomes: True Positives (TP), where ideas are correctly identified as feasible ideas; True Negatives (TN), where not ideas are correctly identified as not ideas; False Positives (FP), where not ideas are incorrectly classified as ideas in a Type I Error; and False Negatives (FN), where ideas are incorrectly classified as not ideas in a Type II Error. By examining this matrix, it is possible to move beyond asking ”How accurate is the model?” to answer more specific questions, such as ”What type of errors is the model most prone to making?”. This detailed view is of importance for a qualitative assessment of the model’s reliability and for understanding its potential biases in a real-world deployment scenario.

4.4. Experimental Setup

The environment of the experiments was Ubuntu 24.04.3 LTS. The experiments were all carried out using bash and Python scripts, and Python's open sources libraries such as Oumi, Unsloth, Numpy, Pytorch and Pandas. The computer used composed by an AMD Ryzen Threadripper 7960X 24-Cores CPU and 2 Nvidia 4090 RTX GPUs, each with 24GB of VRAM.

5. Results

This section will be divided into 3 different subsections, each displaying their group experiments results. Firstly, the SLMs results are displayed, then the BERT models results, and finally the LLMs.

Let's first establish a pattern to make the figure reading process easy. The models (excluding LLMs) will be named as the following pattern: $model\_name^{trained\_dataset}_{tested\_dataset}$. For the LLMs it was chosen the following simple pattern, since they are not fine-tuned: $model\_name_{tested\_dataset}$.

5.1. LLMs Without Fine-Tuning

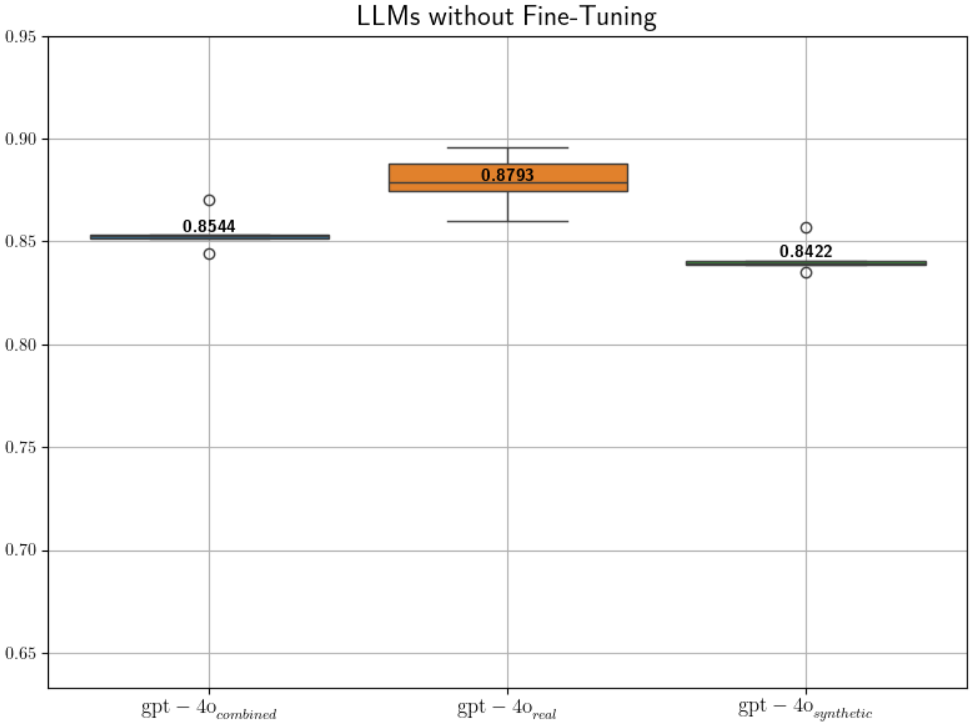

Figure 5. LLMs accuracy boxplot

The boxplot in Figure 5 clearly demonstrates that LLMs are capable of addressing the task. However, their performance remains consistently below the 90% accuracy threshold. Consequently, to achieve a better performance level, the task-specific fine-tuning comes as the most effective strategy. The fine-tuning adapts the models directly to the new task, resulting in a substantial performance improvement, as evidenced by the results shown in Figures 6 and 9.

Beyond performance, the long-term, large-scale deployment of huge proprietary models (like GPT-4o used here) presents significant practical challenges. The reliance on models hosted by large tech companies involves substantial per-query costs. Over a large time span, this expenditure becomes financially unsustainable, making the approach unrealistic for a scaled, continuous operational environment. This further necessitates the adoption of a more cost-effective and self-contained solution, such as fine-tuning smaller, open-source models for the specific task.

5.2. Fine-Tuned SLMs

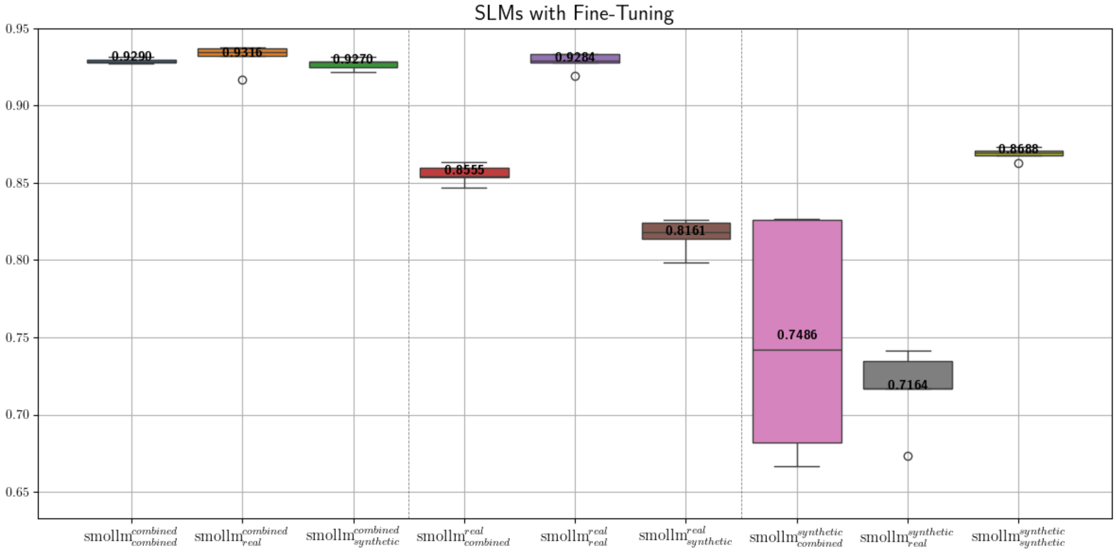

Figure 6. SLMs accuracy boxplot.

Figure 6 summarize the results for the fine-tuned SLMs. Here the dataset nuances are shown. Having been trained in a real dataset helps the SLM react better in the combined and in the real datasets. A drop is perceived at the real to synthetic test when compared to the synthetic to synthetic test. This can be attributed to the fact that every model trained on its own dataset domain will always perform better overall.

The analysis of Figure 6 reveals that the most effective strategy for fine-tuning of SLMs is to use a combined dataset of real and synthetic data. This approach yields exceptionally high-performing and stable models, as evidenced by the consistently high mean scores across different evaluation sets: 92% for the combined test set ($smollm^{combined}_{combined}$), 93% for the real test set ($smollm^{real}_{combined}$), and 92% for the synthetic test set ($smollm^{synthetic}_{combined}$). In contrast, models fine-tuned on specialized datasets exhibit some generalization issues. For instance, a model tuned only on real data performs well on real data ($smollm^{real}_{real}$ at 92%) but struggles when evaluated on synthetic data ($smollm^{synthetic}_{real}$ at 81%). Therefore, combining real and synthetic data is crucial for developing robust models that can generalize effectively across different data distributions.

A comparison of the confusion matrices in Figures 7 and 8 highlights a significant shift in the $smollm^{synthetic}$ model's predictive behavior. While the model performs well on the synthetic data, Figure 8 reveals that it struggles when tested on the real-world dataset, particularly in correctly identifying 'accepted' ideas. This specific diffi-

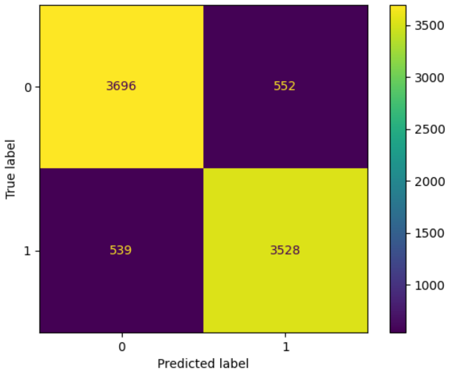

Figure 7. SmolLM3-3B synthetic finetuned confusion matrix for the synthetic test. 0 is a not idea and 1 is an idea.

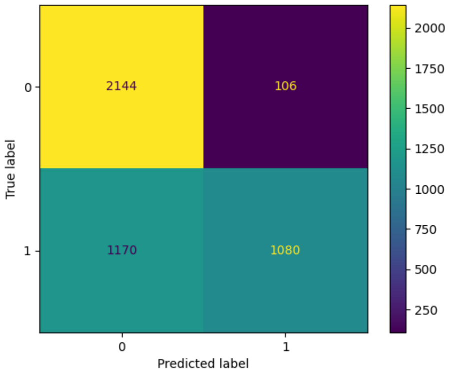

Figure 8. SmolLM3-3B synthetic finetuned confusion matrix for the real test. 0 is a not idea and 1 is an idea.

culty in classifying positive instances is the primary driver for the overall performance decay graphically represented in Figure 6.

This discrepancy is likely attributed to the real-world dataset being more complex or nuanced than the synthetic data used for training. The model, being an SLM with a smaller architecture and more limited context than the better-performing LLM (see Figure 5), is less equipped to generalize to these subtle, real-world variations that were not present in its training set.

However, the most important finding here is that both combined and real trained models showcase excellent results for the real world scenario of the real test set with over 90% accuracy. Thus, showcasing the success of the fine-tuning as a method to increase performance over the LLMs in the specialized task defined.

5.3. Fine-Tuned mmBERT

The empirical results for the fine-tuned mmBERT models across various training and evaluation configurations are summarized in Figure 9. The boxplots illustrate the distribution of accuracy scores over multiple training runs, providing insight into both the performance and stability of each approach.

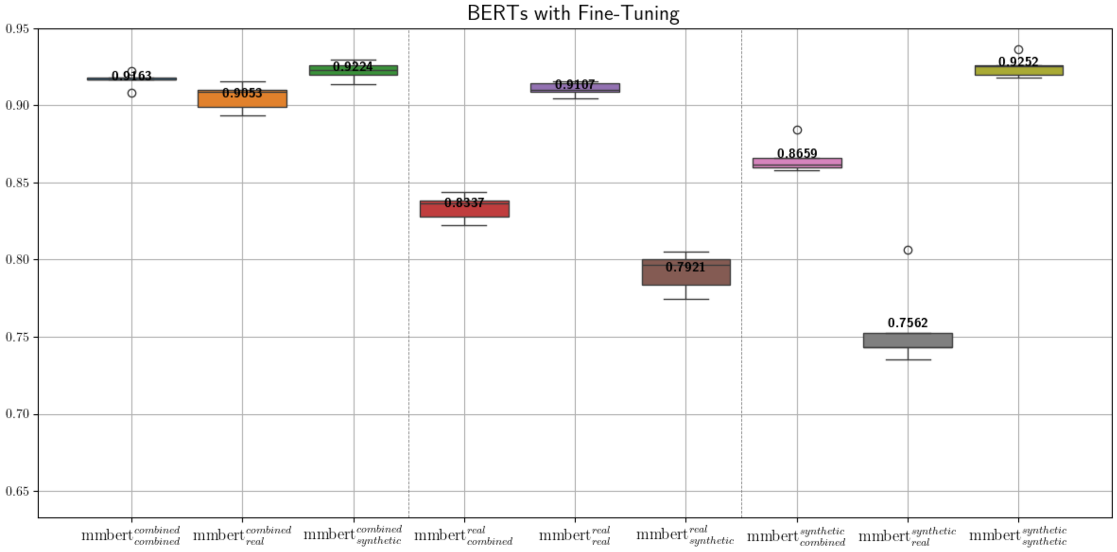

Figure 9. BERTs accuracy boxplot.

The results clearly delineate three distinct performance tiers. The top tier, with accuracies between 90% and 92.5%, consists of models evaluated on data distributions they were trained on (e.g., real-trained model on real data, synthetic-trained model on synthetic data). In contrast, a small performance loss is observed in the cross-domain tier, where accuracies fall to between 75.6% and 79.2%. This occurs when a model is trained on one data type and tested on the other, confirming the existence of a ”domain gap” between the real and synthetic datasets. While the synthetic data successfully captures the core task characteristics, it is not an identical substitute for the proprietary data.

The most significant finding, however, is the performance of the model trained on a combined dataset. When evaluated on the real-world test set, this model achieves a median accuracy of 90.5%. This is only marginally lower than the 91.9% accuracy of the model trained exclusively on the limited real-world data. This demonstrates that our synthetic data serves as a powerful augmentation tool. By enriching the real training data, we can develop a model that maintains near-peak performance on the primary task while being less reliant on scarce, proprietary examples.

Furthermore, this combined model also achieves top-tier accuracy (92.2%) on the synthetic test set, proving it has learned a more generalized and robust representation of the classification task. In conclusion, the results validate our data generation mechanism not merely as a method for mimicking data, but as a highly effective strategy for augmentation. It enables the development of robust classifiers with a negligible performance trade-off. The narrow error ranges across all experiments also underscore the stability and consistency of our training methodology.

5.4. Comparison and Discussion

As illustrated by the boxplots in Figure 6 both the SLM and BERT models (Figure 9) trained on the combined dataset outperforms the LLM (Figure 5) when evaluated on the same test folds.

Moreover, Figure 10 displays the models with the highest average score for the

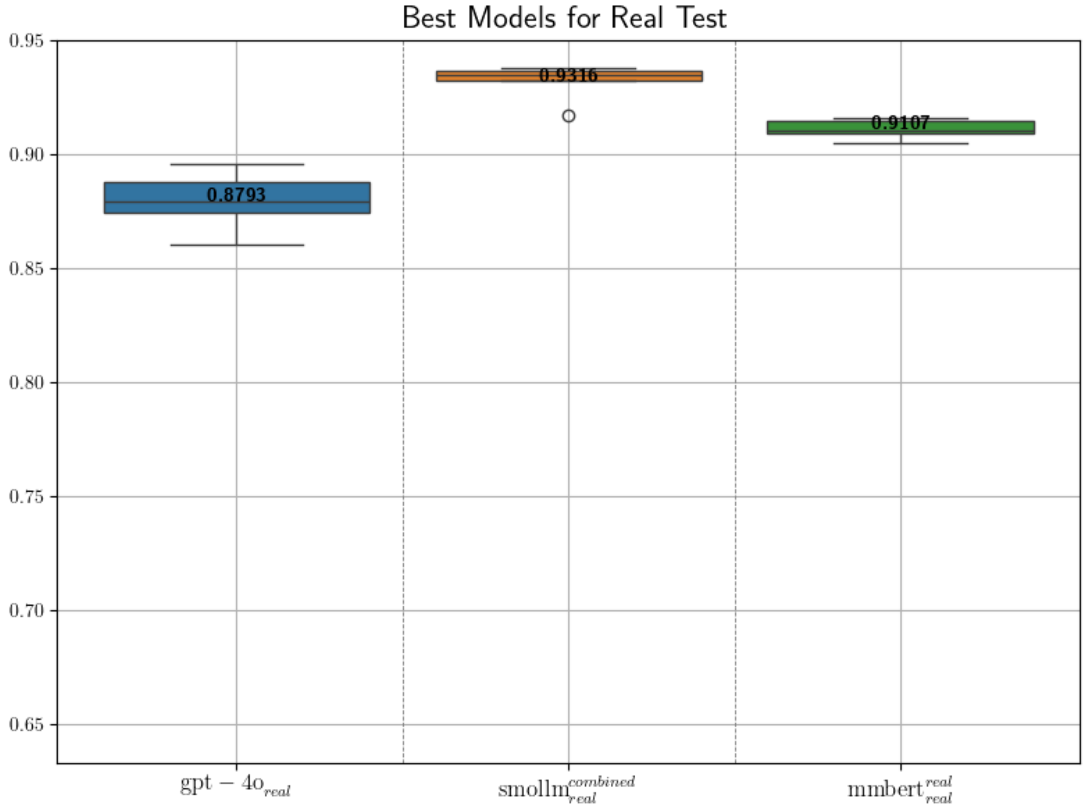

Figure 10. Best performing models accuracy boxplot.

real test evaluation. It's clear that the LLMs lost for both different models architectures. Once more, showcasing that smaller models with fine-tuning are capable of overcoming huge pre-trained LLMs.

It's also worth stating that the significantly higher querying costs associated with the gpt-4o model make it impractical for long-term, large-scale deployment in a business environment, highlighting a key advantage of the more cost-effective SLM and BERT approach.

Thus, it's clear from all results presented that performing SFT into a SLM or BERT model will result in a strong performance boost overall. That outcome showcases a viable, affordable way of creating task specific specialists. The presented results just show a glance of the possible outcomes with SFT in smaller language models.

6. Conclusion

This work successfully validated a methodology for creating specialized, high-performing, and cost-effective idea classifiers. We developed a diverse suite of fine-tuned models, encompassing both BERT architectures and Small Language Models, that are highly capable of classifying business ideas and can serve as a powerful toolkit for human specialists in corporate environments. The strong performance achieved confirms that smaller, specialized models can be expertly adapted for niche tasks, proving it's feasible to develop expert systems without relying on large-scale, general-purpose architectures.

Furthermore, the end-to-end workflow, from synthetic dataset creation and efficient LoRA-based fine-tuning to the final evaluation was proven to be both effective and economical, providing a clear roadmap for developing similar solutions for more challenging problems.

A critical finding of this research is that our specialized models outperform massive commercial LLMs on this specific task. Through a robust evaluation protocol involving rigorous cross-validation and cross-dataset testing, our fine-tuned BERT and SLM models achieved higher accuracy than the costly gpt-4o model (a state-of-the-art LLM). This superior performance, combined with significantly lower operational costs and faster inference times, makes them a more practical and scalable solution for long-term deployment.

Ultimately, this paper not only showcases an approach to using SLM and BERT models to solve industry and business problems, highlighting the power these cost-effective models can have on SI overall, but also proposes a robust synthetic dataset creation mechanism. Extensive tests proved our public, artificial dataset carries fundamental features of the proprietary dataset, while expanding the knowledge domain to unknown scenarios. The combination of the synthetic and real-world datasets for training resulted in a more generalizable, powerful model, incurring in an insignificant performance loss in real-world examples. Consequently, this proved our data generation method to be a reliable way of data augmentation with very little trade-off, enabling the development of robust, generalized classifiers.

Acknowledgments

This study was financed in part by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – (CAPES, Brazil) – Finance Code 001; Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq; and Fundação de Amparo à Pesquisa do Espírito Santo (FAPES, Brazil) - grant 2021-07KJ2. The authors also would like to thank Financiadora de Estudos e Projetos (FINEP) for supporting the company who provided the case and the data.

References

Arvidsson, R., Gunnarsson, R., Entezarjou, A., Sundemo, D., and Wikberg, C. (2024). Chatgpt (gpt-4) versus doctors on complex cases of the swedish family medicine specialist examination: an observational comparative study. *BMJ Open*, 14(12).

Aryan, A., Nain, A. K., McMahon, A., Meyer, L. A., and Sahota, H. S. (2023). The costly dilemma: Generalization, evaluation and cost-optimal deployment of large language models.

Chae, Y. and Davidson, T. R. (2025). Large language models for text classification: From zero-shot learning to instruction-tuning. *Sociological Methods & Research*.

Chinnalagu, A. (2024). Comparative analysis of fine-tuned llm, bert and dl models for customer sentiment analysis. pages 255–259.

Fatemi, S., Hu, Y., and Mousavi, M. (2025). A comparative analysis of instruction fine-tuning large language models for financial text classification. *ACM Transactions on Management Information Systems*, 16:1 – 30.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models.

Irugalbandara, C., Mahendra, A., Daynauth, R., Arachchige, T. K., Dantanarayana, J., Flautner, K., Tang, L., Kang, Y., and Mars, J. (2024). Scaling down to scale up: A cost-benefit analysis of replacing openai's llm with open source slms in production. In *2024 IEEE ISPASS*, pages 280–291.

Javaid, M., Haleem, A., and Singh, R. P. (2023). A study on chatgpt for industry 4.0: Background, potentials, challenges, and eventualities. *Journal of Economy and Technology*, 1:127–143.

Lepagnol, P., Gerald, T., Ghannay, S., Servan, C., and Rosset, S. (2024). Small language models are good too: An empirical study of zero-shot classification. In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N., editors, *LREC-COLING 2024*, pages 14923–14936, Torino, Italia. ELRA and ICCL.

Li, Z., Zhu, H., Lu, Z., and Yin, M. (2023). Synthetic data generation with large language models for text classification: Potential and limitations. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

López, D. and Oliver, M. (2023). Integrating innovation into business strategy: Perspectives from innovation managers. *Sustainability*, 15(8).

McInnes, L., Healy, J., and Melville, J. (2020). Umap: Uniform manifold approximation and projection for dimension reduction.

Naidu, G., Zuva, T., and Sibanda, E. M. (2023). A review of evaluation metrics in machine learning algorithms. In Silhavy, R. and Silhavy, P., editors, *Artificial Intelligence Application in Networks and Systems*, pages 15–25, Cham. Springer International Publishing.

Sun, X., Li, X., Li, J., Wu, F., Guo, S., Zhang, T., and Wang, G. (2023). Text classification via large language models. In *2023 Conference on Empirical Methods in Natural Language Processing*.

Wang, L., Shi, C., Du, S., Tao, Y., Shen, Y., Zheng, H., and Qiu, X. (2024). Performance review on llm for solving leetcode problems. *2024 4th AIIM*, pages 1050–1054.