

**SC4001 Neural Network & Deep Learning**

**Project Report**Enhancing Sentiment Analysis through Domain Adaptation:   
Transitioning from General to Financial Contexts

| **Name** | **Matriculation Number** |
| --- | --- |
| Chew You Chun | U2123251E |
| Leong Hong Yi | U2120932C |
| Lim Jun Hern | U2120981B |

\* Each group member contributed equally to the project.

**College of Computing and Data Science**

**Nanyang Technological University, Singapore**

**Table of Contents**

[**1 Introduction 3**](#_rpw8mwyuxg1)

[1.1 Background 3](#_po2lkw59f6uj)

[1.2 Objectives and Motivations 3](#_mvgaw5agh3fa)

[**2 Methodology and Review of Existing Techniques 3**](#_er2capumlfds)

[2.1 Zero-Shot Learning via Prompt Engineering 3](#_n063ehui3zim)

[2.2 Pretrained BERT 4](#_k4bbm0fdds8z)

[2.2.1 BERT-Base Architecture 4](#_oamco72dhwb9)

[2.2.2 Transfer Learning and Fine-Tuning 4](#_u4yiuoict70s)

[2.3 Multi Domain Adversarial Adaptation 5](#_oi1equb386il)

[2.3.1 Domain Adversarial Neural Network (DANN) 5](#_yh5qt4umolb3)

[2.3.2 Semi-Supervised Domain Adversarial Neural Network 5](#_e6hkv23ehqhh)

[2.3.3 Feature-Level Fusion For Multi-Domain Sentiment Classification 5](#_8c8uxa6vofp8)

[2.3.4 Fully Supervised Domain Adversarial Neural Network 5](#_j53rd8pgq9bi)

[2.4 Sequential Fine-Tuning 6](#_y5xp2my69o9n)

[**3 Datasets 6**](#_1toqwsl4t2wv)

[**4 Experiments 6**](#_a26mh0b1yc9a)

[4.1 Data Preprocessing and Tokenization 6](#_5cp97mjdcdd7)

[4.2 Experiment Setup 7](#_k2qgmwy8ag5g)

[4.2.1 Zero Shot Learning using GPT-4o-mini 7](#_2n5s7slr2qkf)

[4.2.2 Supervised Domain Adversarial Neural Network with BERT 8](#_ae13rbmqldf9)

[4.2.2.1 Feature Extractor 8](#_4bzcaefledd5)

[4.2.2.2 Label Classifier 8](#_gy381e8bopmz)

[4.2.2.3 Discriminator 8](#_vhug0z2hcd6k)

[4.2.2.4 Gradient Reversal Layer 8](#_annhen4cq1ob)

[4.2.2.5 Multi-Domain Learning 9](#_ttrrcn2gd3to)

[4.2.3 Sequential Fine-Tuning with BERT 9](#_vmkiozivvsm8)

[4.2.3.1 Classifier Layer 9](#_5wb60qcvbn0q)

[4.2.3.2 Baseline Model 9](#_9cdf9jdu4v25)

[4.2.3.3 Initial Fine-Tuning with Source Domain 9](#_bg5jj5wku4jc)

[4.2.3.4 Adaptation to Target Domain 10](#_jnyqsk5hz3z)

[4.3 Performance Analysis 10](#_xa0i7m2j9jhm)

[4.3.1 Accuracy and Loss Plots 10](#_xzxhc2x3xqnq)

[4.3.1.1 Supervised Domain Adversarial Neural Network with BERT 10](#_nscq63fpwuco)

[4.3.1.2 Sequential Fine-Tuning with BERT 11](#_csjbjuptchc8)

[4.3.2 Confusion Matrix and Scores 11](#_q8e2oqev645p)

[**5 Discussion 12**](#_z5cf1guq50a1)

[5.1 Advantages and Disadvantages 12](#_pbt29ohw8j3)

[5.2 Conclusion 12](#_2yjsbvxs56yv)

[**6 References 13**](#_pymdk13o261x)

[6.1 Datasets 13](#_qrpvkxdawzdl)

[6.2 Pretrained model 13](#_ywf61g1y34pq)

[6.3 Research Papers 13](#_4jnm3h7hglz1)

# 1 Introduction

## 1.1 Background

Sentiment analysis is a crucial area of study in natural language processing (NLP) which focuses on identifying and interpreting emotions, opinions, or sentiments expressed in text documents. This field has broad applications across domains, such as social media monitoring, customer feedback analysis, and public opinion tracking, where understanding sentiment is valuable for data-driven decision-making. Traditionally, sentiment analysis relied on lexicon-based methods or classical machine learning models, which often required extensive feature engineering and had limited generalization across domains.

With the advent of deep learning, large language models (LLMs) such as BERT, T5, and GPT have revolutionized sentiment analysis. These models leverage attention mechanisms, allowing them to capture context and semantic nuances far beyond traditional models like traditional LSTMs and GRUs. However, these models contain millions or even billions of parameters and typically rely heavily on substantial labeled datasets for optimal performance. Most publicly available datasets cover general topics, leaving niche domains like finance reviews underserved.

In niche market segments or specialized fields with unique language patterns, the scarcity of labeled data poses a critical challenge, including the possibility of overfitting, or low model performance due to the limited representation of domain-specific language. Thus, developing robust techniques to enable sentiment analysis in these niche domains is essential for advancing both research and industry applications.

## 1.2 Objectives and Motivations

The primary goal of this project is to explore and implement effective strategies in sentiment analysis to address the challenges associated with domain-specific data scarcity, utilizing both supervised and unsupervised domain adaptation techniques. A central focus is on supervised domain adaptation, which distinguishes itself from unsupervised methods by leveraging a limited amount of labeled data from the target domain to guide the adaptation process, thereby enhancing model performance by effectively identifying and mitigating domain shifts.

In this project, we employ three advanced techniques, with an emphasis on the latter two for supervised domain adaptation: Zero-Shot Learning via Prompt Engineering, Domain-Adversarial Neural Networks (DANN), and Sequential Fine-Tuning. As an approach to unsupervised domain adaptation, we first utilize Zero-Shot Learning with the latest GPT model by crafting targeted prompts that facilitate generalization across domains without the need for domain-specific training data. Subsequently, we focus on how DANN can generate domain-invariant features in sentiment analysis through supervised adversarial training. Additionally, we delve into sequential fine-tuning techniques by utilizing pre-trained large language models to tailor them for domain-specific sentiment analysis tasks.

Through these efforts, we aim to advance the capabilities of sentiment analysis models through effective domain adaptation, enabling them to perform effectively in niche and specialized domains, even when faced with the challenge of limited labeled data in the target domain.

# 2 Methodology and Review of Existing Techniques

## 2.1 Zero-Shot Learning via Prompt Engineering

*Prompt Engineering* is the process of structuring instructions that can be effectively interpreted by a GAI model. By strategically adding context or specific vocabulary in a prompt, we can guide the model to produce responses that better align with the target domain.

Specifically, we leveraged *Prompt Engineering* to enable *Zero-Shot Learning*, which allows models to make accurate predictions on data instances on tasks they have never seen before. This can be considered as a type of unsupervised domain adaptation. For example, in sentiment analysis, a model not specifically trained on financial news could still infer sentiment from key words or descriptions in a headline.

Unlike traditional training through backpropagation, we "fine-tuned" the model by designing prompts that trigger *Zero-Shot Learning*, helping it adapt to the target domain and thereby improving prediction accuracy. While this method is relatively straightforward and simpler than the other methods mentioned in this report, it remains an interesting approach to be explored, especially in light of the rapid growth and trends associated with advanced language models.

## 2.2 Pretrained BERT

BERT (Bidirectional Encoder Representations from Transformers) is a deep learning model developed by Google in 2018. Built on the Transformer architecture, BERT processes words bidirectionally, simultaneously considering both preceding and following words (Devlin & Chang, 2018). This capability enables it to capture word meanings in context, making it highly effective for NLP tasks like sentiment analysis.

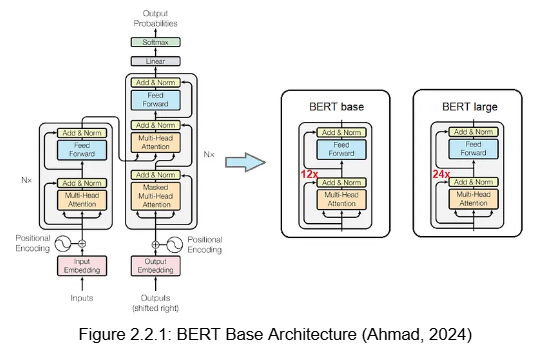
BERT learns the structure and meaning of language through unsupervised learning on large text corpora like Wikipedia and BookCorpus. Its pretraining involves two key tasks:

* **Masked Language Model (MLM)**: In this task, BERT randomly masks 15% of words in a sentence and predicts the masked words based on the surrounding context. This enables BERT to capture word meanings and relationships in diverse contexts.
* **Next Sentence Prediction (NSP)**: BERT is given sentence pairs and must predict if the second sentence logically follows the first. The sentences are selected randomly from the corpus or the actual sentences (Devlin et al., 2018). This task improves BERT’s ability to understand sentence relationships, essential for tasks like question answering.

Through unsupervised pretraining, BERT learns syntax and semantics from vast unannotated data.

### 2.2.1 BERT-Base Architecture

In this project, we use the ‘**bert-base-uncased**’ model. We chose the uncased model because, although case differences can imply semantic nuances, the performance improvement from using a cased model is often marginal. The uncased model results in reduced computational complexity and memory usage, making it a more efficient choice for our sentiment analysis tasks without significantly impacting accuracy.

BERT has a self-attention mechanism to learn contextual word relationships. BERT uses only the encoder, aiming to create a robust language representation. The BERT-base model consists of 12 identical layers, each with two main components:

* **Multi-Head Attention:** This mechanism helps BERT assess the importance of each word and understand relationships within the sentence context with the calculation of attention scores, with 12 attention heads and an input dimension of 768. (Rogers et al., 2020).
* **Feed-Forward Neural Network (FFN)**: After multi-head attention, each layer has an FFN to refine each token’s representation from the attention mechanism individually. In BERT-base, the FFN’s hidden size is 4×768=3072 (Rogers et al., 2020).

This architecture allows BERT to build rich, contextualized embeddings that are highly effective for a wide range of language tasks.

### 2.2.2 Transfer Learning and Fine-Tuning

Transfer learning is a technique that involves adapting a pre-trained model to a new but related task, using knowledge gained from training on large datasets to enhance performance on the target task, even with smaller datasets. In NLP, transfer learning has been especially impactful. Models like BERT, pre-trained on extensive text corpora, gain a broad understanding of language structures, vocabulary, and context (Devlin et al., 2018). Instead of building a model from scratch, transfer learning enables us to use the model’s pre-existing language knowledge by transferring its weights to a new domain.

Within this framework, fine-tuning serves as a strategic approach to further adapt the pre-trained model to the specific linguistic nuances of a target task. It involves refining the model on a specific dataset, adjusting its representations to handle distinct grammar, vocabulary, and context of the target data.

## 2.3 Multi Domain Adversarial Adaptation

### 2.3.1 Domain Adversarial Neural Network (DANN)

Domain Adversarial Neural Network (DANN) transfers knowledge from a source domain with labeled data to a target domain with limited labels, making it valuable for cases where target domain labels are scarce but related source data exists (Ganin et al., 2016). The DANN architecture includes three components:

1. **Feature Extractor:** Encodes input data into a shared representation.
2. **Classifier:** Predicts sentiment based on encoded features.
3. **Domain Discriminator:** Distinguishes between source and target domain data.

Through adversarial training, the encoder learns domain-invariant features, making source and target data appear similar in the shared feature space. A Gradient Reversal Layer helps by reversing the gradient from the discriminator, encouraging the encoder to produce indistinguishable features across domains. However, performance may still degrade due to domain-specific vocabulary or variations, especially in models like BERT, which are sensitive to domain shifts.

### 2.3.2 Semi-Supervised Domain Adversarial Neural Network

To address these challenges, (Ma et al.,2019) introduced a semi-supervised domain adversarial network (SDANN) for domain adaptation when limited labeled target data is available. SDANN combines classification losses from both domains with a domain loss to reduce domain-specific feature discrepancies, improving target domain classification performance even with sparsely labeled data.

### 2.3.3 Feature-Level Fusion For Multi-Domain Sentiment Classification

In fully supervised settings, which typically yield superior results, Li & Zong (2008) proposed a multi-domain sentiment classification approach that uses labeled data from multiple domains to train a single model. This method extracts shared features, or "global sentiment information," allowing the model to learn consistent sentiment patterns across domains without needing separate models for each.

### 2.3.4 Fully Supervised Domain Adversarial Neural Network

Inspired by the concept of feature-level fusion, which maps both the source and target domains into a unified feature space for simultaneous training with all labeled data, and the principles of the semi-supervised domain adversarial network (SDANN), that uses discriminator regularization to train the feature extractor to represent both source and target domain data, even with limited labeled target domain data, we developed a fully supervised domain adversarial neural network (DANN). This approach is specifically designed for scenarios where the labeled target domain data is sufficient for supervised learning to some extent, but not plentiful enough to train a model independently, while still maintaining the model's accuracy.

This approach leverages both labeled source and target domain data, integrating classification losses from both domains alongside domain losses. The goal is to harness the rich labeled data from the source domain while effectively utilizing the limited labeled data from the target domain, enabling the model to learn a unified feature space that generalizes across both domains.

By integrating the classification loss from the labeled source and target data, we effectively guide the training process, ensuring that the model captures rich, discriminative features that are prevalent in both domains. The domain loss on the other hand supervises and encourages the model to learn domain-invariant features.This dual supervised loss approach not only facilitates better feature representation but also improves the overall classification performance in the target domain, even when the labeled data is scarce.

## 2.4 Sequential Fine-Tuning

Sequential fine-tuning is a process in which a pre-trained model is fine-tuned progressively in a sequence of steps, with each step adapting the model to a specific domain before moving on to another. During this process, the model's foundational feature space and predictive functions are maintained, enabling the model to align with the task-specific requirements of new datasets and capture the nuances of domain-specific language. While the distribution of sentiment labels may vary between datasets, the label categories remain consistent, allowing the model to benefit from its pre-existing label structures.

In this approach, we begin with initial training, or initial fine-tuning in the context of the BERT pre-trained model, on a large dataset from the Source Domain, followed by layer freezing. This technique preserves the essential language knowledge captured during pre-training. For domain adaptation, the model then undergoes a final fine-tuning phase using target domain data. During this phase, the unfrozen layers are refined to enhance the model's ability to assimilate the specific language patterns, terminology, and sentiment nuances relevant to the target task (Howard & Ruder, 2018). This strategy enables the model to effectively blend its broad linguistic understanding with specialized performance tailored for sentiment analysis in the target domain.

# 3 Datasets

*Note: The Reddit Comments dataset below is not applicable to our Zero-Shot Learning via Prompt Engineering approach. Instead, the domain adaptation considers the pre-trained ChatGPT model as the Source Domain.*

In our sentiment analysis domain adaptation project, we utilized two datasets as follow:

1. **Source Domain Dataset (Reddit Comments):** This extensive dataset contains 37,249 Reddit comments reflecting a broad spectrum of sentiments and expressions. It is heavily utilized in sentiment analysis research and is our primary source domain.  
Distribution: Positive=42.6%, Negative=35.1%, Neutral=22.3%

2. **Target Domain Dataset (Financial Headlines):** Based on recommendations from our TAs, we chose the Financial Headlines dataset as our target domain. It consists of 4,846 samples focused on financial topics such as market regulations, fiscal policies and geopolitical events.   
Distribution: Positive=28.1%, Negative=12.5%, Neutral=59.4%

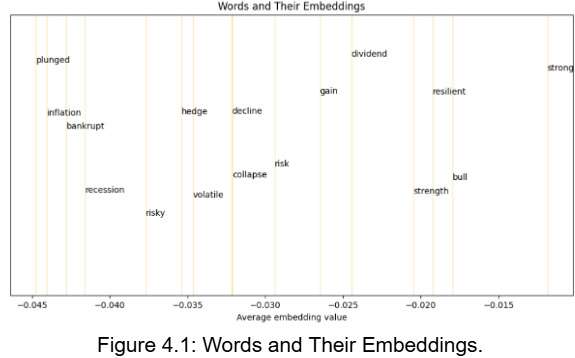
Both datasets feature a consistent structure with two key columns: text content and associated sentiment. Do note that the domain dataset’s distribution is more skewed. Links to the datasets are included under References.

# 4 Experiments

## 4.1 Data Preprocessing and Tokenization

*Note: These steps are not applicable to our Zero-Shot Learning via Prompt Engineering approach. The only preprocessing involves categorizing sentiment as -1 for 'negative', 0 for 'neutral', and 1 for 'positive'.*

The sentiment labels associated with the text are originally represented either as strings such as 'negative', 'neutral', and 'positive', or as corresponding numerical values. To ensure consistency and reflect their categorical nature and inherent ordinal relationship, we standardize these representations by converting them into integers: 0 for 'negative', 1 for 'neutral', and 2 for 'positive'.

For text preprocessing, we utilize the BertTokenizer, which effectively tokenizes the text by breaking each sentence into smaller units known as tokens. These tokens are then further processed into their respective token IDs. It is important to highlight that when employing BERT for NLP tasks, extensive text preprocessing is generally unnecessary. BERT takes full advantage of the context within a sentence, including punctuation and stop-words, by employing the sophisticated multi-head self-attention mechanism.

For exploratory purposes, we randomly selected a set of common finance-related words, tokenized them, and individually passed them through the model’s embedding layer without any context to extract their embeddings. Generally, the embeddings successfully represent the semantic meanings of the words. Negative terms such as ‘bankrupt’, ‘plunged’, and ‘decline’ are positioned on the left side, using ‘risk’ as a reference point. Conversely, positive words like ‘bull’, ‘strong’, and ‘resilient’ are clustered on the right side.

Next, we examined the token length of each piece of text within our two datasets. This step is crucial to ensure compatibility with the 512-token limit imposed by the BERT-base-uncased model. The distribution of token lengths for each dataset is detailed in Table 4.1 below.

Table 4.1 Token Length Density Across Different Datasets.

| Source Domain (Reddit) Distribution | Target Domain (Financial) Distribution |
| --- | --- |

From the two histograms above, it is evident that most pieces of text can be adequately represented with 75 tokens. Therefore, we set the maximum token limit to 75. Any tokens beyond this threshold will be truncated, while texts shorter than 75 tokens will be padded to meet the fixed length. This choice strikes a balance between preserving semantic content and computational efficiency for our domain adaptation task.

Notably, the similar token count distributions between source and target domains suggests minimal distribution shift in terms of text length, which is favorable for domain adaptation.

## 4.2 Experiment Setup

### 4.2.1 Zero Shot Learning using GPT-4o-mini

To perform domain adaptation using *Zero Shot Learning*, we leveraged OpenAI API’s Chat Completion model with hyperparameters as set below:

1. **model = gpt-4o-mini** - Selected for its optimal balance between computational efficiency and performance, while maintaining cost-effectiveness.

2. **temperature = 0** - Chosen to maximize deterministic output and consistency in sentiment predictions, as financial sentiment analysis requires high precision and reproducibility rather than creative variations.

We then performed *Prompt Engineering* as such:

1. **system\_msg** = 'You are a macroeconomic researcher in the finance industry, specializing in financial sentiment analysis to support investment decisions. Given your background in Finance and Sentiment Analysis, you must classify the sentiment for the provided financial sentence using one of the following: 1 for positive, 0 for neutral and -1 for negative.

You should only provide the numeric score in your response.'

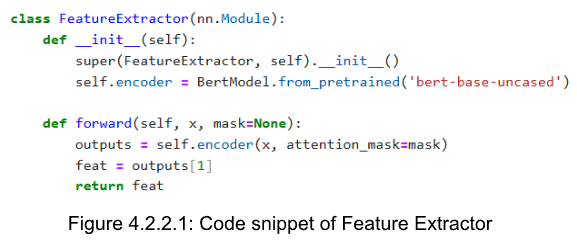
**system\_msg** allows us to instruct the model to behave like a macroeconomic researcher with specific criteria, effectively triggering *Zero Shot Learning* to adapt to our target domain for sentiment analysis. We also specified that only a numeric score (-1, 0, or 1) should be provided to facilitate easier parsing and reduce costs, as costs are calculated per token.

2. **user\_msg** = 'Classify the sentence provided below:\n*<sentence>*'   
**user\_msg** is simply the actual task that the model needs to perform.

The predictions were then parsed and stored into our dataframe. Overall, this method cost us 0.52 USD.

### 4.2.2 Supervised Domain Adversarial Neural Network with BERT

*Note: In this experiment, we utilized Optuna for hyperparameter tuning with cross-validation to determine the optimal hyperparameters.*



#### 4.2.2.1 Feature Extractor

In this domain adversarial framework for sentiment analysis, we leverage the BERT architecture as a feature extractor to encode sentence representations. The BERT model is fine-tuned during Domain-Adversarial Neural Network (DANN) training, enabling it to learn effective representations for both the source and target domains.

#### 4.2.2.2 Label Classifier

For sentiment analysis, we extend the label classifier module by using BERT-extracted features. To prevent overfitting, we add a dropout layer, followed by a linear layer to adapt embeddings for sentiment classification across three classes. We skip an explicit softmax layer, instead using PyTorch's CrossEntropyLoss, which combines softmax and log loss, allowing us to use raw logits directly. For consistency, we initialize the linear layer's weights with a normal distribution (mean=0, std=0.02) and set biases to zero.

#### 4.2.2.3 Discriminator

Similarly, we define a domain classifier module to distinguish the target domain from the source domain. The implementation of the DomainClassifier is similar to that of the label classifier, including dropout layers to prevent overfitting and a linear layer with two nodes for binary classification to differentiate between the source and target domains. The softmax function is also omitted here, as it is already integrated within the CrossEntropyLoss function in PyTorch. The primary distinction between the domain classifier and the label classifier is the inclusion of the gradient reversal layer, which reverses gradients during the backpropagation phase to maximize the discrimination loss.

#### 4.2.2.4 Gradient Reversal Layer

The gradient reversal layer enables adversarial training by reversing gradients during backpropagation, promoting domain-invariant feature learning. In the forward pass, it acts as an identity function, passing input unchanged. In the backward pass, it negates and scales gradients by a factor of alpha, encouraging the model to avoid domain-specific features. This setup returns modified gradients for the input but none for alpha, as alpha does not require a gradient.

#### 4.2.2.5 Multi-Domain Learning

To train the supervised DANN, the limited labeled target domain data is evenly distributed to match the batch size of the more abundant labeled source domain data. This is achieved by dynamically adjusting the target batch size to align with the source batch size as much as possible to ensure both target and source data are included in each iteration, even though their distributions are imbalanced.

During training with early stopping, the labeled target and source data compute their respective label classifier losses independently in a fully supervised manner to enhance sentiment prediction accuracy. On the other hand, the source and target domain are concatenated to compute the domain classifier loss to differentiate the 2 domains. The sum of all loss terms is used to update the model weights during backpropagation as shown as figure below.

This form of adversarial training, encourages the feature extractor to generate representations that minimize domain-specific discrepancies, facilitated by the gradient reversal layer. Ultimately, this contributes to the final objective function, which aims to minimize the classification loss while maximizing the domain classification loss.

### 4.2.3 Sequential Fine-Tuning with BERT

#### 4.2.3.1 Classifier Layer

In the SentimentClassifier class, we extend BERT by adding a classifier layer specifically for sentiment classification. To help prevent overfitting, we set the dropout rate to 0.2. Following the dropout, a linear layer adjusts the embeddings for the classification task, with an output dimension of 3 to match the three sentiment classes in our datasets. Finally, we use a softmax activation layer to transform the output into probabilities, enabling the model to predict the label with the highest probability across all classes.

#### 4.2.3.2 Baseline Model

To evaluate the performance of our enhanced model, we prepare a baseline model by directly applying the pre-trained model to the financial dataset without further adaptation. For adaptation, we fine-tune the classifier over up to 50 epochs, employing early stopping with a patience of 5 based on test loss. If test loss does not improve for 5 consecutive epochs, the training process will be stopped. The learning rate is set to 2e-5, Adam optimizer with weight decay as the optimizer, and weighted cross-entropy loss function due to class imbalance. To speed up training while preserving essential pre-trained features, the top 7 layers are frozen, keeping their weights fixed throughout training.

#### 4.2.3.3 Initial Fine-Tuning with Source Domain

To adapt the classifier from the source to the target domain, we start by fine-tuning it on the Reddit dataset, which serves as a more general source dataset. This fine-tuning step is essential as it helps the classifier learn domain-specific knowledge and gain familiarity with the unique characters and tokens of the source domain. During this stage, we unfreeze all layers, allowing the model to adapt features across all layers for enhanced sentence representation. The model is trained for 20 epochs using the same optimizer and loss function, retaining the model state with the highest accuracy for further fine-tuning on the target dataset.

#### 4.2.3.4 Adaptation to Target Domain

After training on the source dataset, we load the best model state from initial fine-tuning and freeze the first 7 layers to retain general language patterns. We then fine-tune only the lower layers to capture financial domain-specific linguistic patterns, using the same optimizer and loss function. Early stopping is applied if test loss does not improve for 5 epochs to prevent overfitting. This approach ensures effective domain adaptation while preserving model robustness.

## 4.3 Performance Analysis

### 4.3.1 Accuracy and Loss Plots

In this section, we evaluate the performance of our approaches using accuracy and loss plots. The Zero-Shot Learning via Prompt Engineering approach is excluded due to its lack of a traditional training phase.

#### 4.3.1.1 Supervised Domain Adversarial Neural Network with BERT

| Accuracy Plot (Source Domain) | Loss Plot (Source Domain) |
| --- | --- |
|  |  |
| Accuracy Plot (Target Domain) | Loss Plot (Target Domain) |
|  |  |

The plots for both domains are presented together since they were trained simultaneously. The plots above show that test and train accuracy for both the source (Reddit) and target (Financial) domains improve significantly in the first epoch. Afterward, the model appears to converge, with minimal growth beyond epoch 1. This indicates fast convergence, likely due to the large volume of training data. The train and test accuracy for the Reddit domain align closely, reaching up to 97%, which suggests the model is well-adapted to the Reddit data distribution. For the Financial domain, the test accuracy improves alongside train accuracy, indicating effective learning, although some accuracy differences persist.

Looking at the loss plots, both training and testing losses drop sharply during the first epoch and quickly converge, especially for the Reddit domain where train and test loss curves align well. This alignment suggests strong model learning for Reddit, likely facilitated by the larger dataset size. For the Financial domain, however, there is an increase in test loss after the first epoch, while train loss continues to decrease. This may indicate slight overfitting due to the scarcity of data, though the rise in test accuracy.

Comparing the test accuracy and loss curves between domains, we see that the model performs slightly worse on the Financial domain, which is expected given the complexity and sparsity of financial data. Nevertheless, both domains achieve high accuracy overall for sentiment analysis.

#### 4.3.1.2 Sequential Fine-Tuning with BERT

| Accuracy Plot | Loss Plot |
| --- | --- |
|  |  |

The plots above show the accuracy and loss for the target model. As shown in the plots, the model exhibits significant improvement within the first few epochs before beginning to converge. With train and test accuracy reaching above 0.90 from the source domain model, after fine-tuning, the train and test accuracy of the target model also improves significantly in the first 5 epochs and eventually reaches a test accuracy of 0.85. This suggests that the model is adapting well to the target domain.

In examining the loss plot, target train and test losses decrease consistently across epochs, with only some humps. The simultaneous decrease of test and train loss suggests that the model is well-fitted to the data. In the target domain, early stopping was triggered at epoch 11, as the test loss had not improved since epoch 6.

These results demonstrate that extensive training on the source domain enables the model to generalize effectively to data in the target domain. Although the target domain shows slightly lower test accuracy than the source domain, fine-tuning and adapting from the source domain increased test accuracy over the baseline by approximately 0.07.

### 4.3.2 Confusion Matrix and Scores

To compare the performance of each approach, we present their confusion matrices. We also computed and compared their test accuracy and test F1-scores against the baseline model (trained on the Financial Dataset without Domain Adaptation).

| Approach 1 (Zero-Shot Learning via Prompt Engineering) | Approach 2 (DANN with BERT) | Approach 3 (Sequential Fine-Tuning with BERT) |
| --- | --- | --- |

| **Approach** | **Test Accuracy** | **Test F1-Score** |
| --- | --- | --- |
| *Baseline* | *0.7855* | *0.5436* |
| 1 | 0.8382 | 0.8279 |
| 2 | 0.8449 | 0.8425 |
| 3 | 0.8573 | 0.8411 |

The results demonstrate a clear improvement in both test accuracy and test F1-score across approaches. The baseline model, which was trained on the financial dataset without Domain Adaptation using Reddit Comments as the source domain, exhibited the lowest performance. These findings suggest that each of the three proposed approaches effectively facilitates Domain Adaptation. Notably, Approaches 2 and 3 yield comparable overall outcomes; however, Approach 2 achieves a marginally superior F1-score, suggesting a more effective management of class distribution. Approach 2 appears to be the most balanced choice, while Approach 3 maximizes overall accuracy.

# 5 Discussion

## 5.1 Advantages and Disadvantages

| **Approach** | **Advantages** | **Disadvantages** |
| --- | --- | --- |
| 1 | * Requires no training time or labeled data * Simple and intuitive * Holds huge potential for improvement given the rapid advancement in LLM technology | * Limited control over model architecture and training process * Security and privacy concerns when passing data through external APIs |
| 2 | * Facilitates the simultaneous training on datasets across different domains * The trained model exhibits strong performance on both domains and can generalize well to intermediate domains * Achieves the highest F1 score | * May underperform compared to other transfer learning approaches due to the use of DANN * Difficult to train as adversarial training introduces instability and requires careful hyperparameter tuning |
| 3 | * Retains general language features while specializing in target domains * Only the top few layers need to be unfrozen while fine-tuning to target domain * Achieves the highest accuracy | * Very time-consuming as it requires training on two separate datasets and models * May suffer from "catastrophic forgetting," where previously learned information is lost during fine-tuning |

## 5.2 Conclusion

In conclusion, this project effectively demonstrates how various domain adaptation techniques can enhance sentiment analysis models, particularly under constraints of limited domain-specific data. Our findings indicate that Zero-Shot Learning via Prompt Engineering performs decently, signifying its potential to generalize across domains without the need for domain-specific training data. This approach highlights the utility of leveraging large language models like GPT for scenarios where labeled data is scarce or unavailable.

Among the supervised domain adaptation strategies, Domain-Adversarial Neural Networks (DANN) achieved the highest F1-score, showcasing its strength in generating domain-invariant features and improving model robustness against domain shifts. Meanwhile, Sequential Fine-Tuning emerged as the method with the highest accuracy, demonstrating its efficacy in adapting pre-trained models for domain-specific sentiment tasks. Each approach has its pros and cons, and selecting the optimal strategy demands careful consideration of various factors, including time, performance, requirements for intermediate domains, and financial cost.

We hope these findings will inspire further research and practical implementations in sentiment analysis, ultimately advancing the capabilities of sentiment analysis models and optimizing their performance across various specialized domains.

# 6 References

## 6.1 Datasets

**Source Domain:** Twitter and Reddit Sentiment Analysis Dataset *(Note: Twitter data is excluded from this project)* <https://www.kaggle.com/datasets/cosmos98/twitter-and-reddit-sentimental-analysis-dataset>

**Target Domain:** Sentiment Analysis from Financial News Dataset <https://www.kaggle.com/datasets/ankurzing/sentiment-analysis-for-financial-news>

## 6.2 Pretrained model

Bert-base-uncased model <https://huggingface.co/google-bert/bert-base-uncased>

## 6.3 Research Papers

Ahmad, I. (2024, October 8). BERT ArchitectureBERT: Bidirectional Encoder Representations from Transformers A Deep Dive. *Medium*. <https://irfanahmad00.medium.com/bert-bidirectional-encoder-representations-from-transformers-a-deep-dive-cf93d8d4fe94>

Devlin, J., & Chang, M.-W. (2018, November 2). *Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing*. Google Research. <https://research.google/blog/open-sourcing-bert-state-of-the-art-pre-training-for-natural-language-processing/>

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.1810.04805>

Ebker, R., & Ebker, R. (2023, June 11). *Transfer Learning vs Domain Adaptation | Baeldung on Computer Science*. Baeldung on Computer Science. <https://www.baeldung.com/cs/transfer-learning-vs-domain-adaptation>

Ganin, Y., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V., Dogan, U., Kloft, M., Orabona, F., Tommasi, T., Ustinova, E., Ganin, Ustinova, Ajakan, & Germain, L. (2016). Domain-Adversarial Training of Neural Networks. Journal of Machine Learning Research, 17, 1–35. <https://arxiv.org/pdf/1505.07818>

Howard, J., & Ruder, S. (2018, January 18). *Universal Language model fine-tuning for text classification*. arXiv.org. <https://arxiv.org/abs/1801.06146>

Li, S., & Zong, C. (2008). Multi-domain sentiment classification. 257–257. <https://doi.org/10.3115/1557690.1557765>

Ma, X., Xu, P., Wang, Z., Nallapati, R., & Xiang, B. (2019, November 1). Domain Adaptation with BERT-based Domain Classification and Data Selection. ACLWeb; Association for Computational Linguistics. <https://doi.org/10.18653/v1/D19-6109>

Rogers, A., Kovaleva, O., & Rumshisky, A. (2020, February 27). *A Primer in BERTology: What we know about how BERT works*. arXiv.org. <https://arxiv.org/abs/2002.12327>

‌