POLITECNICO DI TORINO

MASTER's Degree in Data Science & Engineering

Masters's Degree Thesis

Bidirectional Text Style Transfer via Reinforcement Learning for Professional Social Media Communication

Supervisors

Cantoro Riccardo

Bellarmino Nicolo

Davide Morra

Candidate

Maham Maham

Abstract

Effective professional communication on platforms such as LinkedIn increasingly demands the ability to adapt message tone and style to suit different audiences and contexts. Whether the goal is to convey approachability through a casual tone or to demonstrate professionalism through formality, the capacity to flexibly transform written communication is invaluable. However, manual rewriting is subjective, inconsistent, and impractical at scale. This thesis addresses these challenges by proposing a fully automated, reinforcement learning-based system for bidirectional text style transfer, transforming messages from formal to casual and vice versa. The core of this research is the application and extension of the RLPrompt framework, a state-of-the-art method that casts text style transfer as a discrete prompt optimization problem. RLPrompt employs a reinforcement learning agent to discover optimal sequences of prompt tokens that guide a pre-trained model (specifically, DistilGPT2) in generating text in the target style. Allowing both directions of style transfer, we propose a single framework to serve multiple communication objectives within professional networking. A custom dataset was compiled from authentic LinkedIn outreach messages, each annotated as formal or casual. After rigorous cleaning and pre-processing, a DistilBERT-based style classifier was trained to assess the style of any message This classifier played a dual role: it not only validated the effectiveness of the style transfer but also formed a crucial component of the reward function in the reinforcement learning loop. Central to the RLPrompt approach is its sophisticated reward design, which integrates the three principal metrics outlined in the original RLPrompt framework: content preservation, style accuracy, and fluency. Content preservation is measured by BERTScore, quantifying the semantic similarity between the original and generated messages to ensure that essential information and intent are maintained. Style accuracy is evaluated through the DistilBERT-based style classifier, providing a reliable measure of how well the output aligns with the desired formal or casual style. Fluency is assessed using language model perplexity, ensuring that generated texts are not only correct in content and style but also natural, coherent, and easily readable. Training proceeded iteratively: the agent proposed prompt sequences, the language model generated outputs, and the rewards informed prompt refinement. Experiments on a curated subset of ~500 messages showed that the system effectively automates style transfer while preserving content and fluency, outperforming traditional rule-based methods. Moreover, the framework is generalizable to other domains and styles with minimal adjustments. In summary, this thesis presents a scalable, flexible, and effective approach to automated text style transfer for professional communication. By leveraging reinforcement learning for discrete prompt optimization and integrating state-of-the-art evaluation metrics, the proposed system empowers users and organizations to personalize outreach at scale, adapt communication strategies dynamically, and improve the effectiveness of digital networking and business development. The methodology not only advances the state of the art in style transfer but also showcases the broader potential of prompt engineering and reinforcement learning in natural language generation.

Chapter 1: Introduction

1.1 Background and Motivation

The rapid advancement of Artificial Intelligence has transformed the field of Natural Language Processing (NLP). Today machines can understand, create and utilize human language at an increasingly accurate level. This growth over the past decade has been driven by advances in computational power, availability of large datasets, and improvements in machine learning methodologies which together have enabled the development of powerful language models that can perform complex linguistic functions. As a result of this progress, the utility of NLP has expanded across more domains, e.g., automated customer service, information retrieval, conversational agents, and content generation.

Recently, large language models (LLMs), have emerged as one of the most critical technologies in the area of NLP. They learn from vast amounts of textual data and can produce coherent and contextually relevant text. Similar to traditional systems based on rules or earlier machine learning methodologies that were trained on specific tasks, LLMs are capable of performing a variety of language tasks when provided with input prompts or instructions. Their flexibility has led to their adoption in various applications and fields, such as marketing, business communication, and automated message systems.

One reason for increasing interest in utilizing LLMs is their potential for automating communication-related tasks that traditionally required human effort. An increasing number of organizations are exploring the utilization of these models to generate personalized communications, enhance customer engagement, and facilitate business outreach. But even though they are good at what they do, it is still hard to write messages that are both high-quality and fit the situation. LLMs can give different results depending on how the prompts are worded and how the instructions are given. Consequently, comprehending the methodologies to adeptly direct these models towards generating reliable and superior text has emerged as a critical research concern.

This dissertation examines methodologies for enhancing the quality of automatically generated messages by concentrating on strategies to optimize prompts.

. It specifically looks at how reinforcement learning can be used to improve the performance of large language models when they are writing professional outreach messages by optimizing prompts. This research seeks to enhance the reliability and controllability of language generation systems by investigating prompt optimization within the framework of automated communication.

1.1.1 Evolution of Natural Language Processing

Most of the time, early NLP systems were based on rules and used rules that people made up by hand. These systems tried to model language by using dictionaries and grammar rules that were already set up. Rule-based methods worked well for some tasks, but they couldn't handle the complexity, ambiguity, and variability of natural language very well. Building and keeping rule-based systems up and running also took a lot of manual work and knowledge of the field.

The next big change in NLP happened in the 1990s and early 2000s when statistical methods were added. Statistical NLP methods used big sets of text data to find patterns in language using models that worked with probabilities. For tasks like machine translation, speech recognition, and part-of-speech tagging, techniques like n-gram language models, Hidden Markov Models, and conditional random fields became very popular. These methods worked better than rule-based systems, but they still needed carefully crafted features and were based on hand-crafted features.

Deep learning is considered to be a big step forward in the history of NLP. Neural network-based methods started to take the place of traditional statistical methods by automatically learning how to represent language from big datasets. Long Short-Term Memory (LSTM) networks and Recurrent Neural Networks (RNNs) were two of the first deep learning architectures used for language modeling and predicting sequences. These models showed that they could better capture contextual dependencies in text than earlier methods.

The transformer architecture proved to be a big step forward in natural language processing (NLP) because it made it easier to process long text sequences. Self-attention mechanisms are the core mechanisms under transformers. The attention mechanisms are designed in a way that they let the models see how words relate to each other in a whole sentence or document. This new idea made language models much more scalable and faster, and it was the start of modern large-scale pretrained models. Because of this, NLP has moved from rule-based and statistical systems to strong neural architectures that can understand and create complex language patterns.

1.1.2 Emergence of Large Language Models (LLMs)

Most large language models are constructed on deep neural networks that have billions of parameters. The model learns to guess the next word in a sequence based on the words that came before it during training. The model learns about grammar, syntax, facts, and how things are related in the training data over time. These models can then be employed for many different NLP tasks once they have been trained, and they don't need to be trained for every specific task.

One of the most important things about large language models is that they can do things by prompting. Users can give the model instructions or prompts that tell it how to get the desired output instead of having to retrain it for each task. This feature lets LLMs do things like generate text, summarize it, answer questions, translate it, and create dialogue all in one framework.

The GPT family, BERT-based architectures, and other transformer-based models made by research groups and tech companies are all examples of large language models that are well-known. These models have shown amazing performance on a variety of NLP benchmarks and have been used in many real-world applications.

In spite of their remarkable abilities, the LLMs are fraught with a number of issues as well. The accuracy, hallucinations or inconsistencies in their outputs may also appear occasionally, particularly when prompts are not well configured or do not have enough context. Therefore, enhancing the reliability and controllability of the outputs of LLM has become one of the major research studies. Prompt engineering, prompt optimization and methods based on reinforcement learning techniques have also been suggested to solve these issues and improve the performance of language models in practice.

1.1.3 Applications of LLMs in automated communication

The fact that big language models are able to produce coherent and contextually relevant text has introduced new opportunities to automated communication systems. The world is witnessing the rise in the use of the LLM-based solutions by organizations in various sectors to make it easy to carry out tasks related to written communication. Such applications are automated customer services to outreach marketing and professional networking.

Among the applications of LLMs is personalized business outreach messages. Businesses usually use the services of digital platforms like email or professional networking to build a relationship with potential customers, collaborators, or partners. Someone may take time to create individual messages to each of them especially when handling enormous amounts of leads. LLMs offer a chance to mechanize this procedure generating a personal message depending on the accessible information concerning the recipient and the organization.

Language models can be used in an automated outreach system to generate messages based on structured inputs like company names, job titles or contextual information about the recipient. With prompts including this kind of information, LLMs can generate messages that look personal and relevant. The business development and marketing operations can be greatly enhanced with a personalized communication capacity where the company can make it more engaging and the response rates will be enhanced.

In addition to the outreach messaging, LLMs are also common in other types of automated communications. As an example, language model-driven conversational agents can help customers to solve queries, reserve services, or access information. Likewise, report, summary or marketing writing can be done by automated content generation software with minimum human

effort. Such systems assist the organizations to lower the cost of operation and still have effective communication channels.

Nonetheless, the quality of the automatized communication systems strongly relies upon the quality of the text produced. The messages should be correct, congruent and based on the human communication norms. Messages that are created poorly can be unnatural, erroneous in fact, or they can be incapable of communicating the right tone. Thus, the key to successful implementation of the LLM-based communication systems is the creation of methods to enhance the quality and reliability of the generated messages.

The study conducted in this thesis aims at enhancing the quality of AI-generated outreach messages by investigating the methods of prompt optimization. The present research will integrate the concept of the reinforcement learning and text style transfer to improve the capability of the language models to produce messages that are contextually relevant and meet the human expectations of communication.

1.2 Internship Project Context

The study offered in this thesis is based on the internship project that examined artificial intelligence utilization to create automated outreach messages in a professional communicative context. As large language models become more and more accessible and natural language processing is being developed, companies are looking into the ways in which they can use these tools to assist in communication activities that involve considerable human input.

Business communications and networking may involve the use of individualized messages to contact potential clients, partners, or other stakeholders. The messages need to be well developed to make sure that it is relevant, professional and in line with the standards of the communication that is expected in a business setup. Nevertheless, in cases where the organizations have many people to communicate with, it is inefficient and time consuming to write the personalized messages separately to each lead. The internship project was a venture to understand how the large language models could be employed to automatize such a process and still achieve high quality of the message.

The project used the potential of language models to create coherent and contextually relevant text to explore whether AI-generated messages could help professionals to conduct outreach work more effectively. The internship was a chance to study the real-world problems related to the process of automated message generation such as timely design, message testing, and inconsistency of the output produced by language models. All these issues led to the research orientation of this thesis that aims at enhancing the quality of produced messages based on timely optimizations methods. The project of the internship was aimed at finding the possibility of generating personalized outreach messages using large language models and using them to communicate with professionals.

1.2.1 Overview of the internship project

The primary approach was to assess whether the AI-generated messages can facilitate business networking operations through the generation of messages that can appear natural human messages. The system that was created in the course of the internship was based on structured input data, which was company and lead information. This information acted as context to the language model. This information was included in the form of prompts, allowing the system to create outreach messages that targeted individual people. This data pipeline has the following appearance:

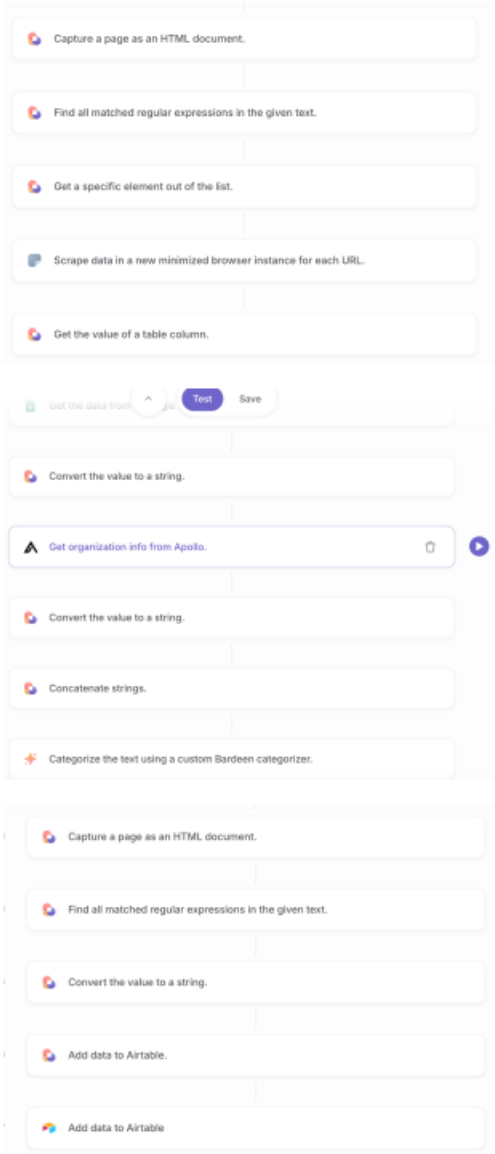

Figure 1: Bardeen Pipeline

The analysis of difference in performance in message generation was conducted using multiple language models during the experimentation phase. The project sought to compare the performance of various models to understand more of the impact of selection of model and timely design on the quality of communication generated.

The assessment of produced messages was also an important point in the project. As the generation tasks of texts do not have a single perfect answer, the messages were rated in terms of the various quality dimensions. These assessment measures measured the aspects like fluency, coherence, factual accuracy and concurrence with human communication criteria.

The assessment process gave an idea of the development of language models in professional messages and allowed finding aspects to be improved. Specifically, it was demonstrated that prompt design is an important factor in the quality and relevance of the generated messaging.

Table 1.1: *Main components of the automated message generation system developed during the internship project.*

Component	**Description**
Input data	Company name and lead information
Prompt template	Structured prompt used to guide the language model
Language model	Model used to generate outreach messages
Generated message	AI-generated outreach communication
Evaluation metrics	Metrics used to assess message quality

1.2.2 AI-generated outreach messages for business communication

Traditionally, outreach messages are written manually by professionals to ensure personalization and contextual relevance. However, as businesses increasingly adopt large-scale lead generation strategies, manually crafting individualized messages for every potential contact becomes inefficient and difficult to scale. This challenge has led to growing interest in the use of artificial intelligence to assist in generating personalized communication messages.

Large Language Models (LLMs) have demonstrated strong capabilities in generating human-like text and can be applied to automate outreach communication. By incorporating contextual information such as the recipient's name, company, or professional role, these models can generate messages that appear personalized and relevant to the intended recipient. This capability makes LLMs particularly suitable for applications in professional networking and business outreach.

Key Characteristics of AI-Generated Outreach Messages

AI-generated outreach messages are designed to replicate several essential characteristics of effective professional communication. These include:

- Personalization – Messages incorporate contextual information about the recipient, such as their name or organization.
- Professional Tone – Messages maintain appropriate formality for business communication.
- Clarity of Purpose – The message clearly communicates the intention of establishing a professional connection.
- Conciseness – Messages remain brief while conveying relevant information.
- Scalability – Automated systems allow organizations to generate messages for a large number of leads efficiently.

These characteristics help ensure that automated messages remain aligned with the expectations of professional communication environments.

Advantages of AI-Generated Outreach Communication

The adoption of AI-generated messaging systems offers several advantages for organizations. These benefits include:

- Time Efficiency
  AI systems can generate messages within seconds, reducing the time required for manual message composition.
- Scalability
  Automated systems enable organizations to contact large numbers of potential leads simultaneously.
- Consistency in Communication
  Language models can maintain consistent tone and messaging across multiple communications.
- Generation of Multiple Variations
  AI systems can produce different versions of a message, allowing users to select the most appropriate one.

Table 2.1: Comparison between manually written outreach messages and AI-generated outreach communication.

Feature	**Manual Outreach Messages**	**AI-Generated Outreach Messages**
Time required	High	Low
Scalability	Limited	High
Personalization	High	Moderate to High
Consistency	Varies between individuals	Consistent
Automation	Not automated	Fully automated

Challenges of AI-Generated Outreach Messages

Despite their advantages, AI-generated outreach messages also present several challenges that must be addressed to ensure their effectiveness in professional communication.

Key challenges include:

- Dependence on Prompt Design
  The quality of generated messages strongly depends on how prompts are constructed.
- Risk of Inaccuracies or Hallucinations
  Language models may generate information that is factually incorrect or misleading.
- Maintaining Professional Tone
  Messages must remain aligned with professional communication standards.
- Evaluation Complexity
  Assessing the quality of generated text requires multiple evaluation metrics.

These challenges highlight the need for systematic approaches to improve the reliability and quality of AI-generated communication systems.

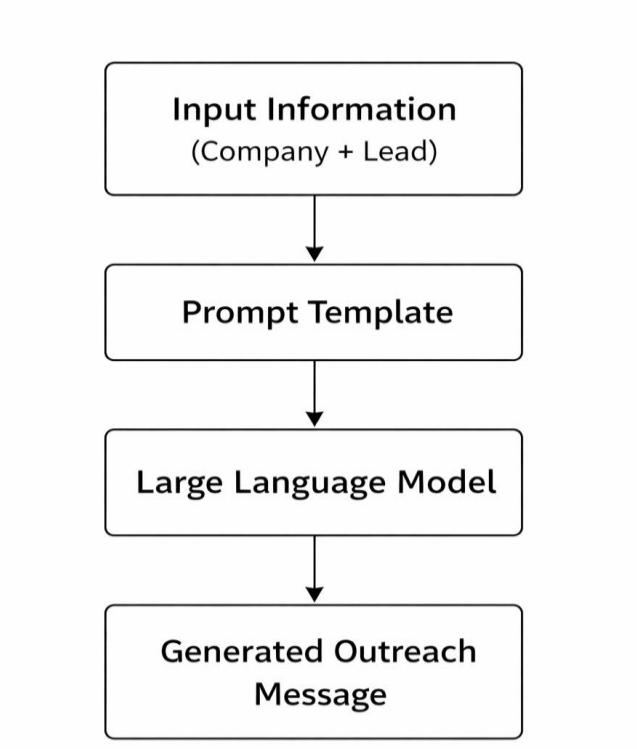

Figure 2.2: AI-Based Outreach Message Generation Workflow

1.2.3 Message generation workflow

The internship project adopted standardized workflow process of creating AI-generated outreach messages with large language models. The workflow incorporates the combination of various steps, which are the preparation of the input, immediate construction, message production, and the assessment of the message. All the stages are a part of the general process of converting structured information about companies and culminates in custom-made messages of communication. This workflow was used to simulate a realistic system that can automatically produce outreach messages without losing contextual relevance and professional tone. The workflow allows conducting the analysis systematically by dividing the process into several steps in order to determine the influence of various elements on the quality of the generated message, including prompts, models, and metrics of evaluation.

Overview of the Workflow

The message generation process consists of several sequential steps that convert structured input data into AI-generated outreach messages.

The key stages of the workflow include:

- **Input Data Preparation**
  Collecting contextual information about companies and leads that will be used for message personalization.
- **Prompt Construction**
  Creating prompt templates that incorporate contextual information to guide the language model.
- **Message Generation**
  Providing prompts to large language models to generate candidate outreach messages.
- **Message Evaluation**
  Assessing the generated messages using predefined evaluation metrics.
- **Performance Analysis**
  Analyzing evaluation results to understand how prompts and models influence message quality.

Step-by-Step Workflow Description

1. Input Data Preparation

The first stage of the workflow involves collecting and organizing the input data required for message generation. The dataset used during the internship project contained information about companies and leads that served as contextual input for the language model.

Each record in the dataset typically included:

- company name
- lead name
- model used for generation
- generated message

This information enables the system to create messages that are tailored to specific individuals and organizations.

Table 3 Example structure of the dataset used for AI-based outreach message generation.

Company	Lead	Model	Message
Namirial	Alessandro Mandolini	Gemini	Generated outreach message
Example company	Example Lead	Gemini	Generated outreach message

2. Prompt Construction

After preparing the input data, the next step involves constructing prompts that guide the language model in generating messages. A prompt acts as an instruction that defines how the model should interpret the input information and what type of output it should produce.

Prompt templates typically include contextual information about the company and lead. For example, a prompt may instruct the language model to generate a professional outreach message that introduces a service or expresses interest in collaboration.

The design of the prompt is a critical factor influencing the quality of the generated output. Different prompt structures can lead to significant variations in message tone, content, and clarity.

3. Message Generation Using Language Models

Once the prompts are constructed, they are provided as input to a large language model. The model processes the prompt and generates a candidate outreach message based on the contextual information and instructions provided.

During the internship project, multiple language models were used in order to evaluate differences in message generation performance. Each model generated messages that were later analyzed and compared based on predefined evaluation criteria.

The generated messages represent the primary output of the system and form the basis for further evaluation and analysis.

4. Message Evaluation

The generated messages are evaluated using a set of predefined metrics designed to assess the quality and reliability of the generated communication. Since text generation tasks do not have a single correct answer, multiple evaluation metrics are used to measure different aspects of message quality.

The evaluation metrics used in the dataset include:

- Accuracy
- Hallucination
- Efficacy
- Fluency
- Coherence
- Transparency
- Safety
- Human-alignment

Each metric is scored on a scale from **1 to 3**, where higher scores indicate better performance.

Table 1.4 Evaluation metrics used to assess the quality of AI-generated outreach messages.

Metric	**Description**
Accuracy	Measures correctness of information in the message
Hallucination	Measures whether the model generates fabricated information
Efficacy	Measures effectiveness of the message in achieving communication goals
Fluency	Measures grammatical correctness and readability
Coherence	Measures logical flow and clarity
Transparency	Measures clarity and interpretability of the message
Safety	Measures whether the message avoids harmful or inappropriate content
Human-alighnment	Measures similarity to human-written communication

1.3 Problem Statement

Large Language Models (LLMs) have demonstrated remarkable capabilities in generating human-like text and performing a wide range of Natural Language Processing tasks. Through prompting, these models can perform tasks such as text generation, summarization, translation, and conversational dialogue without requiring task-specific training. However, despite their flexibility and power, controlling the behavior and output quality of these models remains a significant challenge.

One of the primary factors influencing the performance of LLMs is the design of prompts used to guide the model. A prompt acts as an instruction that provides context and defines the expected output. In many practical applications, prompts are manually designed through a process known as prompt engineering. While manual prompt engineering can produce useful results, it is often inefficient and unreliable when applied to large-scale systems.

The internship project discussed in the previous section highlighted several challenges related to prompt design and message generation. In particular, it was observed that small variations in prompt wording could significantly influence the quality, tone, and accuracy of generated outreach messages. Additionally, different language models may respond differently to the same prompt, leading to variability in outputs.

These observations reveal an important research problem: **how to systematically improve prompt design in order to generate higher-quality outputs from large language models**. Addressing this problem requires understanding the limitations of manual prompt engineering, the variability of language model outputs, and the need for automated optimization techniques.

1.3.1 Limitations of manual prompt engineering

Prompt engineering indicates to the phenomenon of designing and refining prompts that guide a language model toward producing the desired output. In most real-world applications, prompts are created manually by developers or researchers who experiment with different prompt formulations until satisfactory results are obtained.

Although manual prompt engineering has enabled many successful applications of language models, it presents several important limitations.

Key Limitations

The major challenges associated with manual prompt engineering include:

- **Trial-and-Error Process**
  Prompt design often requires repeated experimentation, making the process time-consuming and inefficient.
- **Dependence on Human Expertise**
  The effectiveness of prompts often depends on the experience and intuition of the person designing them.
- **Lack of Scalability**
  Manually designing prompts becomes difficult when systems need to handle multiple tasks or large datasets.

- **Model-Specific Behavior**
  A prompt that works well for one language model may not perform well with another model.
- **Limited Exploration of Prompt Space**
  Humans can only test a small number of prompt variations, leaving many potential prompt configurations unexplored.

Table 4: Major limitations associated with manual prompt engineering.

Limitation	Description
Time-consuming	Requires repeated experimentation to find effective prompts
Human dependency	Success depends on the designer's intuition and expertise
Poor scalability	Difficult to design prompts for large-scale systems
Model sensitivity	Different models respond differently to the same prompt
Limited exploration	Only a small subset of possible prompts can be tested

1.3.2 Variability of LLM outputs

Another major challenge associated with large language models is the variability of their outputs. Even when provided with the same input prompt, language models can generate different outputs due to factors such as model architecture, sampling strategies, and prompt wording.

This variability presents a significant challenge for applications that require consistent and reliable text generation, such as professional communication systems. In the context of outreach messaging, inconsistencies in generated messages can affect the professionalism and clarity of communication.

Several factors contribute to the variability of LLM outputs.

Factors Influencing Output Variability

- **Prompt Sensitivity**
  Small changes in prompt wording can lead to significantly different outputs.
- **Model Architecture Differences**
  Different language models may interpret prompts differently.
- **Randomness in Generation**
  Sampling methods used during generation can introduce variability.
- **Contextual Interpretation**
  Models may interpret contextual information in different ways.

1.3.3 Need for automated prompt optimization

Given the limitations of manual prompt engineering and the variability of language model outputs, it becomes necessary to develop systematic approaches for improving prompt design. Automated prompt optimization aims to address this challenge by using algorithmic methods to explore and refine prompts in order to achieve better performance.

Unlike manual prompt engineering, automated methods can explore a larger space of possible prompt configurations and identify those that produce higher-quality outputs. These methods can also adapt prompts dynamically based on feedback from evaluation metrics.

Several approaches have been proposed for automated prompt optimization, including gradient-based prompt tuning and reinforcement learning-based methods. Among these approaches, reinforcement learning provides a particularly promising framework because it allows prompts to be optimized through iterative interaction with the language model.

In reinforcement learning-based prompt optimization, prompts are treated as actions generated by an agent. The outputs produced by the language model are evaluated using predefined metrics, and the evaluation scores are used as reward signals. The agent then updates its prompt generation strategy in order to maximize the reward.

This approach enables the system to automatically discover prompts that improve the quality of generated messages without relying solely on human intuition. In the context of this thesis, reinforcement learning is applied to optimize prompts used for generating outreach messages, with the goal of improving their quality across multiple evaluation metrics.

1.4 Research Objectives

The primary goal of this thesis is to investigate methods for improving the quality and reliability of AI-generated outreach messages. As discussed in the previous sections, the performance of large language models is strongly influenced by the prompts used to guide the generation process. Manual prompt engineering often relies on trial-and-error experimentation and may not consistently produce optimal results.

To address these limitations, this thesis explores the use of **reinforcement learning-based prompt optimization techniques**. By automatically optimizing prompts, it becomes possible to improve the quality of generated messages across multiple evaluation metrics such as fluency, coherence, accuracy, and human alignment.

In addition to prompt optimization, the research also investigates **text style transfer**, which enables transformation between different communication styles (e.g., formal and casual). This allows the study to analyze how prompt optimization affects both the structure and tone of generated messages.

1.4.1 Main objective of the thesis

The main objective of this thesis is to **develop and evaluate an automated prompt optimization approach using reinforcement learning to improve the quality of AI-generated outreach messages**.

The research specifically focuses on applying the RLPrompt framework to optimize prompts used for message generation and analyzing how optimized prompts influence the performance of large language models in professional communication tasks.

1.4.2 Specific research questions

To achieve the main objective, the thesis addresses the following research questions:

- **RQ1:** How can reinforcement learning be applied to optimize prompts for large language models?
- **RQ2:** Does automated prompt optimization improve the quality of AI-generated outreach messages?
- **RQ3:** How does text style transfer affect the tone and effectiveness of generated messages?

Chapter 2 – Literature Review

2.1 Natural Language Generation with Large Language Models

Natural Language Generation (NLG) is a subfield of Natural Language Processing (NLP) that focuses on enabling machines to generate coherent and meaningful human language. Over the past decade, advances in deep learning have significantly improved the capabilities of language generation systems, leading to the development of powerful large language models (LLMs). These models are capable of producing high-quality text and performing a wide range of language-related tasks such as question answering. summarization, dialogue generation, and content creation.

Early approaches to natural language generation depended on rule-based systems and statistical models. While these approaches were useful for specific applications, they were constrained in their power to get the complexity and variability of natural language. The introduction of neural network-based models, particularly deep learning architectures, marked an important advancement in the area of NLP. Neural language models are capable of learning complex linguistic patterns directly from large-scale datasets without requiring extensive manual feature engineering.

A major breakthrough in NLP occurred with the introduction of the Transformer architecture, which helped the development of large-scale pretrained language models capable of learning contextual representations of text. These models are typically trained on massive corpora containing billions of words and can be adapted to various tasks using prompts or fine-tuning techniques. As a result, LLMs have become one of the most influential technologies in modern NLP research and applications.

Large language models have demonstrated impressive performance across multiple NLP benchmarks and have been widely adopted in both academic research and industry applications. Their ability to generate fluent, coherent and appropriate for the context text makes them particularly appropriate for tasks involving automated communication, including the generation of outreach messages and personalized content.

2.1.1 Transformer Architecture

The Transformer architecture, introduced by Vaswani et al. in 2017, represents a significant milestone in the development of modern NLP systems [1]. Unlike previous neural architectures such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, Transformers rely on a mechanism known as **self-attention** to process sequences of text.

Self-attention allows the model to analyze the relationships between different words in a sentence simultaneously rather than sequentially. This mechanism enables the model to capture long-range dependencies and contextual information more effectively than earlier architectures. As a result, Transformer-based models are able to process large text sequences efficiently and learn complex language patterns.

The Transformer architecture is composed of two primary components: an **encoder** and a **decoder**. The encoder takes the input text and generatest representations according to the context, while the decoder uses these representations to produce the output sequence. Each component is composed of multiple layers that include attention mechanisms and feed-forward neural networks.

The key advantages of the Transformer architecture include:

- Improved ability to capture long-range dependencies in text
- Parallel processing of input sequences
- Efficient training on large-scale datasets
- Scalability to models with billions of parameters

These advantages have made the Transformer architecture the foundation of most modern large language models.

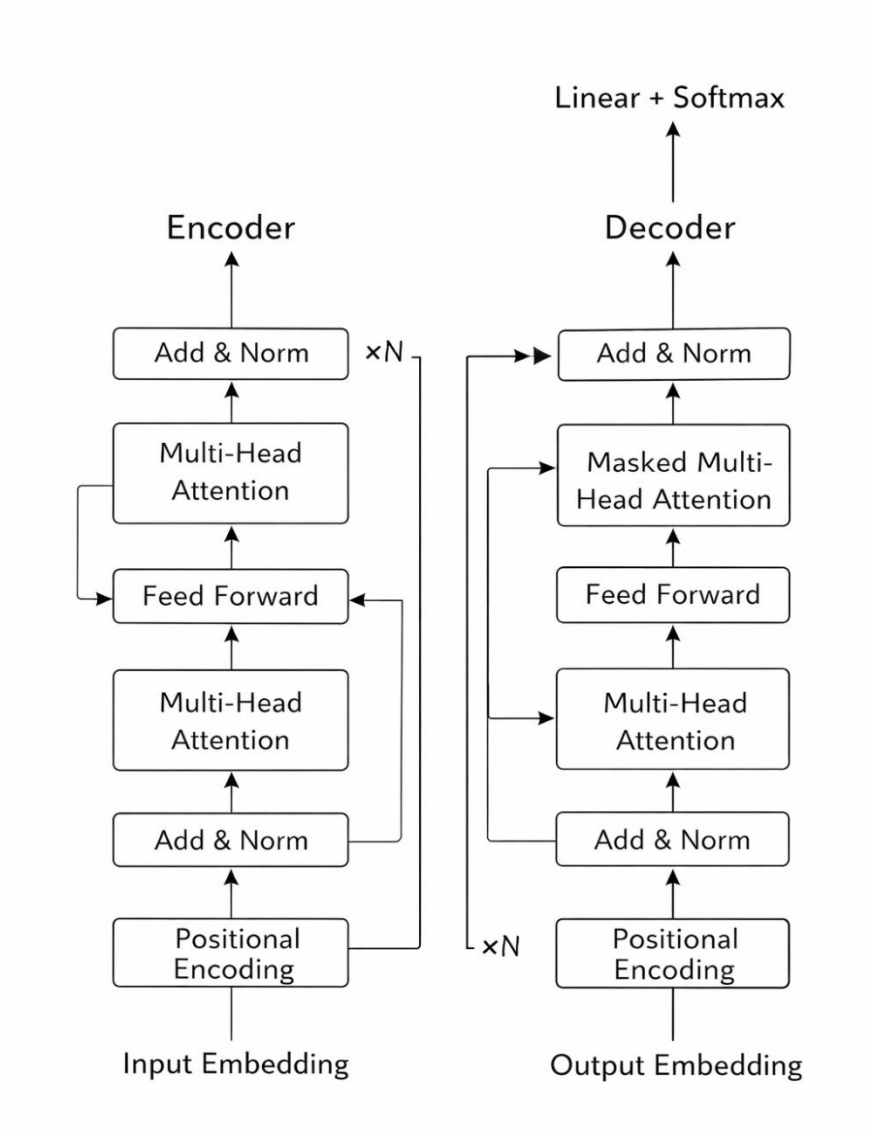

Figure 3: Architecture of transformer model introduced by Vaswani

2.1.2 Pretrained Language Models

The pretrained models are trained on large text corpora to learn general language representations before being applied to specific tasks. Pretraining allows the model to acquire knowledge about grammar, semantics, and contextual relationships within language.

The development of pretrained models has reduced the need for task-specific training data. Instead of training models from scratch for each task, pretrained models can be adapted through fine-tuning or prompting techniques. This approach has led to substantial improvements in performance across a wide range of NLP applications.

Two of the most influential pretrained language model architectures are **BERT** and the **GPT family** of models.

BERT

Bidirectional Encoder Representations from Transformers (BERT) was introduced by Devlin et al. in 2018 and represents one of the most important developments in pretrained language models [2]. BERT is based on the Transformer encoder architecture and is designed to learn bidirectional representations of text.

Unlike traditional language models that process text in a left-to-right or right-to-left manner, BERT analyzes both directions at the same time. This bidirectional context allows the model to understand the meaning of words in a better way within a sentence. As a result, BERT achieved state-of-the-art performance on several NLP benchmarks when it was introduced.

BERT is typically pretrained using two main training objectives:

- **Masked Language Modeling (MLM)** – Random words in the input sequence are masked, and the model learns to predict them based on the surrounding context.
- **Next Sentence Prediction (NSP)** – The model learns to determine whether two sentences follow each other in a text sequence.

These training objectives enable BERT to learn rich contextual representations that can be applied to tasks such as text classification, question answering, and sentiment analysis.

GPT family

The Generative Pre-trained Transformer (GPT) models represent another major class of pretrained language models. Developed by OpenAI, GPT models are based on the Transformer decoder architecture and are specifically designed for text generation tasks [3].

Unlike BERT, which focuses on understanding text, GPT models are designed primarily for **generating text sequences**. GPT models are trained using a language modeling objective that predicts the next word in a sequence given the preceding context.

The GPT family has evolved through multiple versions, each increasing in size and capability. Notable versions include:

- **GPT** – The original generative pretrained transformer model.
- **GPT-2** – Demonstrated strong performance in text generation tasks.
- **GPT-3** – Introduced large-scale language modeling with billions of parameters.
- **GPT-4** – Further improvements in reasoning, contextual understanding, and generation capabilities.

These models have demonstrated impressive performance across numerous tasks, including content generation, dialogue systems, and automated communication.

Table 5: Comparison between BERT and GPT pretrained language models.

Feature	BERT	GPT
Architecture	Transformer Encoder	Transformer Decoder
Training Direction	Bidirectional	Left-to-right
Main Objective	Text understanding	Text generation
Common Applications	Classification, QA	Text generation, dialogue

2.1.3 Applications of LLMs in Text Generation

Large language models have enabled a wide range of applications in automated text generation. Their ability to produce coherent and contextually relevant text has made them useful for both research and industrial applications.

Common applications of LLMs in text generation include:

- **Content Generation**
  Automatic generation of articles, summaries, and reports.
- **Conversational Agents**
  Dialogue systems used in customer support and virtual assistants.
- **Machine Translation**
  Translating text between different languages.
- **Question Answering Systems**
  Providing natural language answers to user queries.
- **Automated Communication**
  Generating emails, outreach messages, and marketing content.

Among these applications, automated communication has gained increasing attention due to its relevance for business environments. Organizations are exploring the use of LLMs to generate personalized messages for marketing campaigns, customer engagement, and professional networking.

However, despite their strong performance, LLMs can sometimes generate inconsistent or unreliable outputs. These limitations have motivated research into techniques that improve the controllability and reliability of language model outputs. One promising direction involves **prompt optimization**, which aims to design prompts that guide language models toward producing higher-quality text.

The research presented in this thesis builds upon these developments by investigating reinforcement learning-based prompt optimization techniques to improve the generation of outreach messages.

2.2 Prompting in Large Language Models

Prompting has emerged as a fundamental technique for interacting with large language models (LLMs). Instead of modifying model parameters through fine-tuning, prompting guides the model's behavior by providing instructions or contextual examples within the input text. This approach allows pretrained models to perform a wide variety of tasks without additional training.

Large language models are trained on vast amounts of textual data and acquire general linguistic knowledge during the pretraining phase. Prompting leverages this knowledge by framing tasks in natural language instructions that the model can interpret and respond to. As a result, tasks such as summarization, translation, question answering, and text generation can be performed simply by designing appropriate prompts.

Prompting has become particularly important with the development of large-scale models such as GPT-3 and GPT-4, which demonstrate strong capabilities in performing tasks using only task descriptions and examples [3]. However, the performance of LLMs is highly sensitive to prompt design. Even small changes in wording or structure can significantly influence the quality of generated outputs.

Consequently, researchers have explored various prompt design strategies to improve the effectiveness of language models. These strategies include prompt engineering, manual prompt design, and in-context learning.

2.2.1 Prompt Engineering

Prompt engineering refers to the process of designing and refining prompts to guide a language model toward producing the desired output. A prompt typically includes instructions, contextual information, and sometimes examples that help the model understand the task.

The concept of prompt engineering gained significant attention with the introduction of large generative language models capable of performing tasks without task-specific training. Researchers discovered that carefully designed prompts could significantly improve model performance across various tasks.

Prompts may contain several components:

- **Task instructions** that describe the objective of the task
- **Input context** providing relevant information for the model
- **Examples or demonstrations** showing how the task should be performed
- **Output format specifications** guiding the structure of the response

Prompt engineering allows developers to control the behavior of language models without modifying the underlying model parameters. However, designing effective prompts often

requires experimentation and domain expertise. This challenge has motivated research into automated prompt optimization methods that can systematically improve prompt design.

2.2.2 Manual Prompt Design

Manual prompt design is the most commonly used approach for interacting with large language models. In this approach, developers manually craft prompts and iteratively modify them to improve the quality of the model's responses.

The process typically involves a trial-and-error methodology in which prompts are repeatedly tested and refined. Developers experiment with different wording, instructions, and examples until the model produces satisfactory outputs.

While manual prompt design is relatively simple to implement, it presents several limitations. The effectiveness of prompts often depends heavily on the intuition and experience of the designer. Furthermore, manual experimentation can be time-consuming, especially when exploring multiple prompt variations.

Key challenges of manual prompt design include:

- dependence on human intuition and expertise
- lack of systematic optimization methods
- limited exploration of possible prompt configurations
- inconsistent results across different models

These limitations highlight the need for automated approaches capable of optimizing prompts more efficiently.

Table 6Comparison between manual prompt design and automated prompt optimization approaches.:

Aspect	**Manual Prompt Design**	**Automated Prompt Optimization**
Design process	Human-driven	Algorithm-driven
Efficiency	Low	Higher
Exploration of prompt space	Limited	Extensive
Scalability	Difficult	More scalable

2.2.3 In-Context Learning

In-context learning is an ability of large language models that enables them to perform tasks by learning from examples provided within the prompt itself. Instead of updating model parameters, the model infers the task pattern from the examples included in the prompt and applies it to new inputs.

This concept was extensively studied with the introduction of GPT-3, which demonstrated strong few-shot learning capabilities [3]. By providing a small number of input-output examples within the prompt, the model can generalize the pattern and generate appropriate responses.

In-context learning can be implemented in several forms:

- **Zero-shot prompting** – the model receives only task instructions without examples
- **One-shot prompting** – the prompt contains a single example demonstrating the task
- **Few-shot prompting** – the prompt includes multiple examples that illustrate the task

These approaches allow language models to adapt to new tasks without retraining, making them highly flexible tools for many NLP applications.

Despite its advantages, in-context learning still depends heavily on prompt quality and example selection. Poorly chosen examples may lead to suboptimal outputs, which further emphasizes the importance of effective prompt design and optimization techniques.

The research presented in this thesis builds upon these ideas by investigating reinforcement learning-based methods for optimizing prompts in order to improve the generation of outreach messages.

2.3 Prompt Optimization Techniques

Several prompt optimization strategies have been proposed in the literature. These approaches can generally be categorized into **soft prompt tuning**, **discrete prompt optimization**, and **gradient-based prompt optimization**. Each method differs in how prompts are represented and optimized.

2.3.1 Soft Prompt Tuning

Soft prompt tuning is a technique in which prompts are represented as continuous embedding vectors rather than discrete text tokens. Instead of manually designing prompts using natural language, a set of trainable embedding vectors—known as **soft prompts**—is appended to the input of a frozen language model.

During training, these embedding vectors are optimized using gradient-based methods while the parameters of the language model remain fixed. The optimized embeddings guide the model toward producing better outputs for the target task.

Soft prompt tuning offers several advantages:

- reduces the need for manual prompt design
- requires updating only a small number of parameters
- allows efficient adaptation of large pretrained models

However, soft prompts are not interpretable in natural language because they exist only in embedding space.

2.3.2 Discrete Prompt Optimization

Discrete prompt optimization focuses on optimizing prompts that are represented as actual text tokens rather than continuous embeddings. In this approach, prompts remain human-readable and interpretable.

The optimization process typically involves searching through possible combinations of tokens to find prompts that maximize a performance metric. Various search strategies can be used for this purpose, including heuristic search, reinforcement learning, and evolutionary algorithms.

Discrete prompt optimization is particularly useful when interpretability is important because the resulting prompts remain understandable to humans. However, searching through the large space of possible text prompts can be computationally challenging.

Key characteristics of discrete prompt optimization include:

- prompts remain interpretable natural language
- optimization occurs over discrete token space
- often requires search-based or reinforcement learning methods

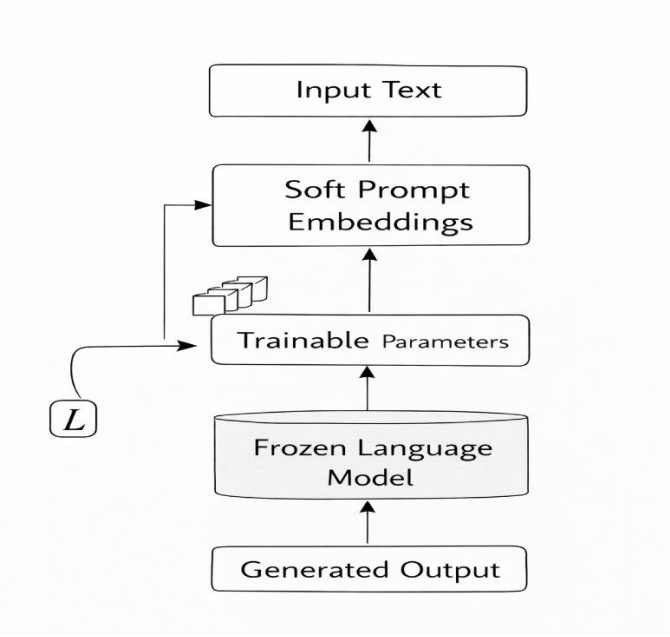

Figure 4: Soft prompting architecture

2.3.3 Gradient-Based Prompt Optimization

Gradient-based prompt optimization is a method that uses gradient information from the language model to refine prompts. In this approach, prompts are represented as parameters that can be updated using gradient descent to improve model performance.

The optimization process typically involves calculating gradients of a loss function with respect to the prompt parameters. These gradients indicate how the prompt should be modified in order to produce better outputs.

This approach enables systematic prompt improvement and can efficiently explore the prompt space. However, gradient-based optimization often requires access to model internals, which may not always be available when working with closed-source language models.

Advantages of gradient-based prompt optimization include:

- efficient exploration of prompt space
- systematic optimization process
- strong performance in several NLP tasks

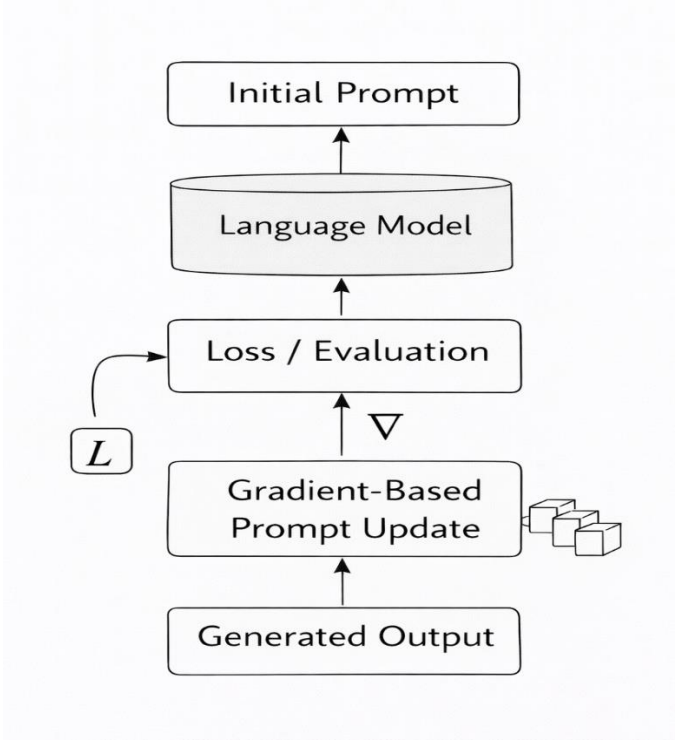

Figure 5: Gradient based optimization architecture

2.4 Reinforcement Learning in NLP

Reinforcement Learning (RL) is a type of machine learning where an agent learns how to make decisions by interacting with an environment and getting rewards as feedback. In supervised learning, models learn from data that has been labeled. In reinforcement learning, on the other hand, models learn the best actions by trying things out and seeing what happens.

In recent years, reinforcement learning has been increasingly applied to Natural Language Processing (NLP) tasks, particularly in scenarios where direct supervision is difficult or where the quality of generated outputs must be evaluated using complex metrics. RL provides a framework for optimizing sequence generation models by directly optimizing task-specific objectives.

In the context of language models, reinforcement learning enables the optimization of text generation processes by treating the generation of words or prompts as a sequential decision-making problem. The model receives feedback based on the quality of generated outputs and updates its strategy accordingly.

2.4.1 Fundamentals of Reinforcement Learning

Reinforcement learning is based on the interaction between an **agent**, an **environment**, and a **reward signal**. The agent selects actions based on its current policy, and the environment provides feedback in the form of rewards. The goal of the agent is to learn a policy that maximizes the cumulative reward over time.

The key components of reinforcement learning include:

- **Agent** – the decision-making entity that selects actions
- **Environment** – the system with which the agent interacts
- **State** – the current situation observed by the agent
- **Action** – the decision made by the agent
- **Reward** – feedback indicating the quality of the action
- **Policy** – the strategy used by the agent to select actions

The learning process typically involves updating the policy based on observed rewards in order to maximize long-term performance.

Reinforcement learning algorithms such as **policy gradient methods** and **Q-learning** have been widely used in various domains, including robotics, game playing, and natural language processing.

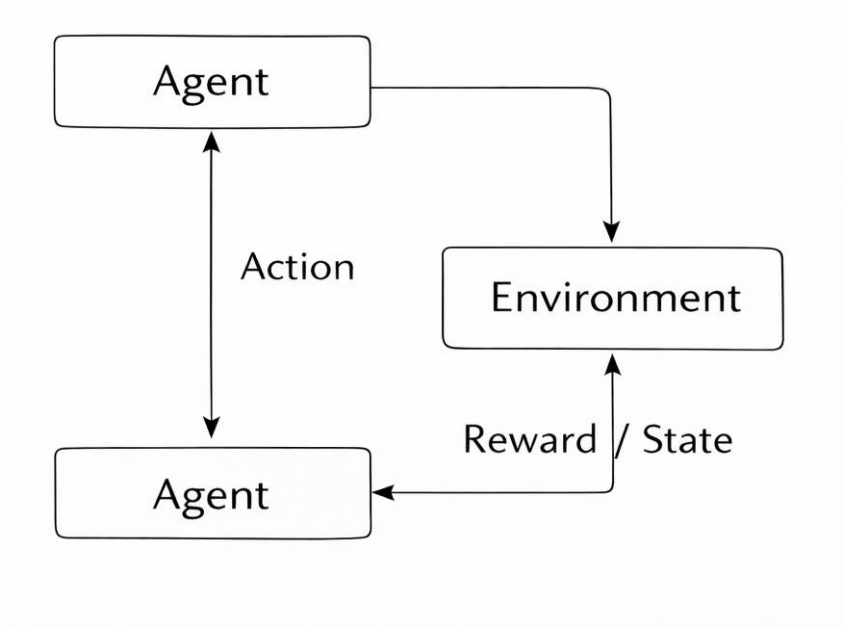

Figure 6: Interaction cycle between the reinforcement learning agent and the environment.

2.4.2 RL for Text Generation

Reinforcement learning has been applied to text generation tasks to improve the quality of generated sequences. Traditional language models are typically trained using maximum likelihood estimation (MLE), which optimizes the probability of predicting the next word in a sequence. However, this objective does not always align with the desired evaluation metrics for generated text.

Reinforcement learning provides an alternative approach by optimizing text generation according to task-specific reward functions. In this setting, the language model acts as the agent, and the generated text sequence represents the actions taken by the agent.

During training, the model generates candidate text sequences, which are then evaluated using predefined metrics such as fluency, relevance, or task-specific performance measures. These evaluation scores are used as reward signals to update the model's policy.

Applications of reinforcement learning in text generation include:

- dialogue generation
- machine translation
- summarization
- content generation

By optimizing models using reward-based objectives, reinforcement learning can improve the quality and coherence of generated text.

2.4.3 RL for Prompt Optimization

Reinforcement learning has also been explored as a method for optimizing prompts used in large language models. In prompt optimization tasks, prompts are treated as actions generated by a reinforcement learning agent. The goal of the agent is to discover prompts that maximize the performance of the language model according to predefined evaluation metrics.

In this framework, the agent generates candidate prompts, which are then used as inputs to the language model. The model produces outputs based on these prompts, and the outputs are evaluated using quality metrics. The evaluation results are converted into reward signals that guide the optimization process.

One notable approach in this area is **RLPrompt**, which applies reinforcement learning techniques to search for effective discrete prompts for language models [11]. Unlike gradient-based methods that rely on differentiable prompt representations, RL-based methods can operate directly in the discrete token space, making them suitable for optimizing interpretable prompts.

The reinforcement learning framework allows the system to explore a large space of possible prompt configurations and gradually learn prompts that produce higher-quality outputs.

This approach is particularly relevant to the research presented in this thesis, which investigates reinforcement learning-based prompt optimization to improve the generation of outreach messages.

2.5 Text Style Transfer

Text style transfer is an important task in natural language processing that focuses on modifying the stylistic properties of a text while preserving its original semantic meaning. In recent years, the development of large language models has significantly improved the ability of automated systems to perform style transformations such as converting informal text into formal language or adapting writing styles for specific contexts.

Traditional approaches to text generation primarily focused on producing coherent sentences without explicitly controlling stylistic properties. However, many real-world applications require the ability to modify writing style while retaining the underlying content. For example, style transfer can be used to convert casual language into professional communication, adjust sentiment in text, or adapt content for different audiences.

The increasing capabilities of large language models have made style transfer more accessible, allowing systems to modify linguistic characteristics such as tone, formality, and politeness. In this thesis, text style transfer is applied to transform outreach messages between **formal and**

casual communication styles, enabling more flexible and context-appropriate automated message generation.

2.5.1 Definition of Text Style Transfer

Text style transfer refers to the process of transforming a piece of text from one stylistic form to another while preserving its original meaning. The objective is to separate **content** from **style**, allowing modifications to stylistic attributes without altering the semantic information contained in the text.

In the context of natural language processing, style may refer to several linguistic properties, including:

- formality level
- sentiment
- politeness
- writing tone
- domain-specific language

A successful style transfer system must satisfy two main objectives:

1. **Style accuracy** – the generated text should reflect the target style.
2. **Content preservation** – the semantic meaning of the original text should remain intact.

Achieving both objectives simultaneously is challenging because changes in style often affect the structure and wording of sentences. Consequently, style transfer has become an active area of research within the NLP community.

2.5.2 Approaches to Style Transfer

Several approaches have been proposed for performing text style transfer. These methods differ in how they separate stylistic features from semantic content and how the transformation process is implemented.

One common approach involves **rule-based transformations**, where predefined linguistic rules are applied to modify stylistic elements of text. While this approach can work for simple tasks, it is limited in flexibility and scalability.

More advanced techniques rely on **machine learning models**, particularly deep learning architectures, which can learn style transformations from data. These models are typically trained using parallel or non-parallel datasets that contain examples of texts written in different styles.

Recent research has increasingly focused on leveraging large language models for style transfer. By providing appropriate prompts or instructions, LLMs can generate text that follows specific stylistic constraints while preserving the underlying meaning.

Common approaches to style transfer include:

- **Rule-based methods** – predefined linguistic transformations
- **Supervised learning approaches** – trained using parallel style datasets
- **Unsupervised methods** – learn style transformations without paired data
- **Prompt-based generation using LLMs**

Among these approaches, prompt-based methods have gained popularity due to their flexibility and ability to leverage pretrained language models without requiring extensive labeled data.

Table 7: Overview of different approaches used for text style transfer.

Approach	Description	Advantages	Limitations
Rule-based	Uses predefined linguistic rules	Simple to implement	Limited flexibility
Supervised learning	Uses parallel datasets	High accuracy	Requires labeled data
Unsupervised learning	Learns from unpaired text	More scalable	Harder to train
Prompt-based methods	Uses LLM prompts to control style	Flexible and adaptable	Sensitive to prompt design

2.5.3 Evaluation Metrics for Style Transfer

Evaluating text style transfer systems is challenging because the quality of generated text depends on multiple factors. A successful style transfer system should produce text that reflects the target style, preserves the original meaning, and maintains grammatical fluency.

Researchers typically evaluate style transfer models using a combination of automatic metrics and human evaluation methods. These metrics measure different aspects of the generated text.

Common evaluation criteria include:

- **Style Accuracy** – measures whether the generated text matches the target style.
- **Content Preservation** – evaluates how well the original meaning is retained.
- **Fluency** – assesses grammatical correctness and readability.

In practical applications, additional evaluation metrics may be introduced depending on the specific task. In the context of automated message generation, evaluation may include metrics related to clarity, coherence, safety, and human alignment.

Chapter 3 – Methodology

3.1 Research Framework

This chapter presents the methodology used to investigate reinforcement learning-based prompt optimization for improving AI-generated outreach messages. The proposed research framework integrates large language models, prompt optimization techniques, and text style transfer experiments to analyze how optimized prompts influence the quality of generated communication.

The overall framework of this research consists of several sequential stages, beginning with the preparation of the dataset and continuing through prompt optimization, message generation, and evaluation. The goal of this framework is to systematically examine how reinforcement learning can be applied to discover effective prompts that guide large language models toward producing higher-quality outputs.

The methodology is designed to combine both **prompt optimization** and **text style transfer** in order to analyze the behavior of language models under different stylistic constraints. Reinforcement learning is used to automatically generate and refine prompts, while the generated messages are evaluated using multiple quality metrics provided in the Gemini evaluation dataset.

The main components of the proposed research framework include:

- **Dataset Preparation**
  The Gemini evaluation dataset is used as the primary data source. It contains company information, lead names, generated messages, and evaluation scores across multiple quality metrics.
- **Prompt Generation and Optimization**
  Prompts are optimized using the RLPrompt framework, which applies reinforcement learning to search for effective prompt structures.
- **Message Generation using Language Models**
  Optimized prompts are provided as input to large language models to generate outreach messages.

- **Text Style Transfer Experiments**
  Generated messages are transformed between different communication styles, specifically formal and casual styles.
- **Evaluation of Generated Messages**
  The generated outputs are evaluated using predefined metrics such as accuracy, fluency, coherence, and human alignment.

This research framework enables a systematic investigation of how prompt optimization techniques influence message generation quality and style adaptation. By integrating reinforcement learning and style transfer within a unified pipeline, the methodology allows for a comprehensive analysis of automated communication generation using large language models.

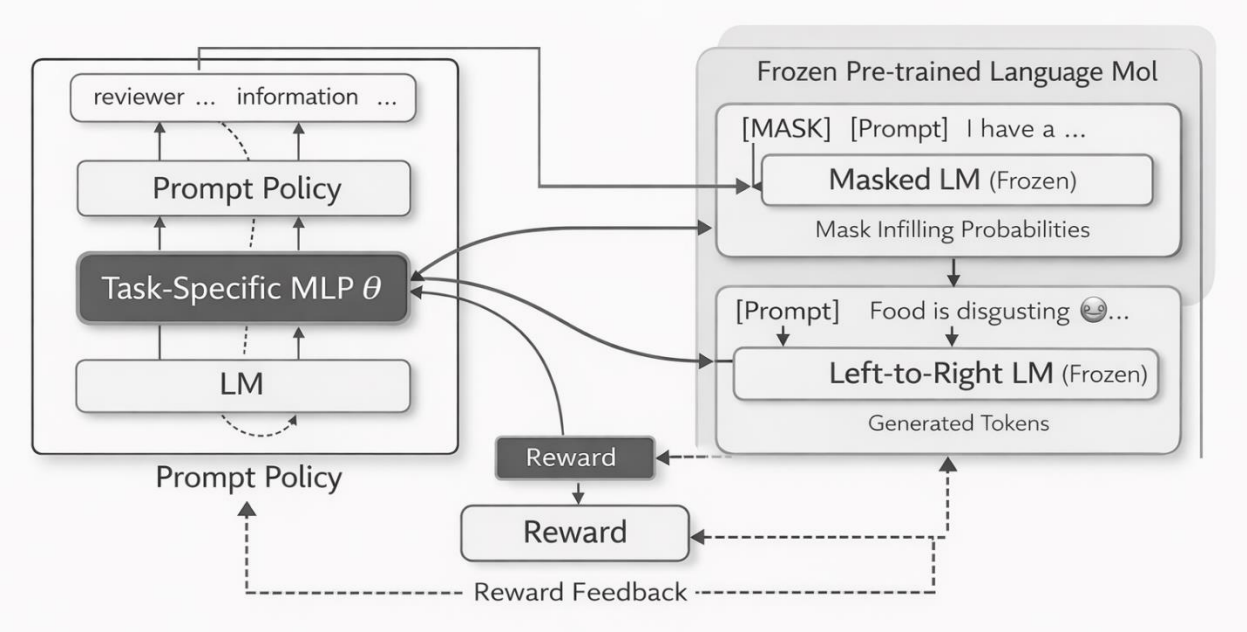

Figure 7: Overall framework used for prompt optimization and style transfer experiments

3.2 Dataset Preparation

The experiments in this thesis use the **Gemini evaluation dataset**, which contains AI-generated outreach messages along with evaluation scores across multiple quality metrics. The dataset was developed during an industrial internship project focused on automated business communication using large language models.

The dataset entry includes information about the **company**, the **lead**, the **language model used**, and the **generated message**, along with evaluation scores assessing different aspects of message quality. This dataset provides a structured basis for analyzing the performance of language models and evaluating the effectiveness of prompt optimization techniques.

3.2.1 Gemini Evaluation Dataset

The Gemini evaluation dataset is organized in a tabular format where each row represents a single generated message and its associated metadata. The dataset contains information about the company, the lead to whom the message is directed, the model used for generation, and the evaluation scores assigned to the message.

3.2.2 Data Structure

The main attributes included in the dataset are summarized in the following table.

Column	**Description**
Company	Name of the organization to which the message is addressed
Lead	Name of the target individual within the company
Model	Language model used to generate the message
Message	Generated outreach message
Accuracy	Evaluation score measuring factual correctness
Hallucination	Score indicating the presence of fabricated information
Efficacy	Score measuring effectiveness of the communication
Fluency	Score assessing grammatical quality and readability
Coherence	Score evaluating logical structure of the message
Transparency	Score indicating clarity and interpretability
Safety	Score measuring the absence of harmful or inappropriate content
Human-alignment	Score reflecting alignment with human communication standards

Each evaluation metric is assigned a score ranging from **1 to 3**, where higher values indicate better performance.

3.4 Reinforcement Learning Prompt Optimization

This research applies reinforcement learning to automatically optimize prompts used for generating outreach messages. Instead of manually designing prompts, the **RLPrompt framework** is used to treat prompt generation as a reinforcement learning problem.

In this framework, a **policy network** generates candidate prompts that are provided to a pretrained language model. The generated outputs are evaluated using predefined criteria, and the resulting scores are used as reward signals to guide the optimization process.

The reinforcement learning agent then updates the prompt generation policy to maximize the reward, gradually discovering prompts that improve the quality of generated outputs.

3.4.1 RLPrompt Framework

RLPrompt is a reinforcement learning-based method designed to optimize **discrete text prompts** for pretrained language models. Unlike soft prompt tuning methods that operate in continuous embedding space, RLPrompt generates prompts composed of actual text tokens.

In the RLPrompt framework, the pretrained language model remains **frozen**, and only the prompt generation policy is updated during training. The policy network generates candidate prompts, which are then evaluated based on the quality of the language model outputs.

The optimization process follows three main steps:

1. **Prompt Generation** – The policy network produces candidate prompts.
2. **Language Model Evaluation** – The prompts are provided to a pretrained language model to generate outputs.
3. **Reward Feedback** – The generated outputs are evaluated and used to compute reward signals for updating the policy network.

Through repeated interactions, the RLPrompt framework learns prompts that improve the performance of the language model on the target task.

```
for epoch in range(num_epochs):

    logits = policy()
    dist = torch.distributions.Categorical(logits=logits)

    prompt_token_ids = dist.sample()
    log_probs = dist.log_prob(prompt_token_ids).sum()

    reward, generated_text, input_data, prompt_text = env.step(
        prompt_token_ids.tolist()
    )

    loss = -log_probs * reward

    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

Figure 8: Training loop implementing the REINFORCE algorithm for prompt optimization.

3.4.2 Prompt Policy Network

The prompt policy network is responsible for generating candidate prompts during the reinforcement learning process. The policy network learns a probability distribution over possible prompt tokens and selects sequences of tokens that form a prompt.

At each iteration, the policy network produces a prompt that is combined with the task input and passed to the language model. The generated output is evaluated, and the resulting reward signal is used to update the parameters of the policy network.

The policy network is typically implemented as a lightweight neural network that generates prompts token by token. During training, reinforcement learning algorithms such as policy gradient methods are used to adjust the network parameters so that prompts leading to higher rewards become more likely to be generated in future iterations.

```
class PromptPolicy(nn.Module):
    def __init__(self, vocab_size, prompt_len=5, hidden_size=128):
        super(PromptPolicy, self).__init__()

        self.embeddings = nn.Parameter(torch.randn(prompt_len, hidden_size))

        self.mlp = nn.Sequential(
            nn.Linear(hidden_size, hidden_size),
            nn.ReLU(),
            nn.Linear(hidden_size, vocab_size)
        )

    def forward(self):
        logits = self.mlp(self.embeddings)
        return logits
```

Figure 9: Prompt policy network used to generate candidate prompts.

3.4.3 Reward Function Design

The reward function plays a crucial role in the reinforcement learning process because it determines how the quality of generated outputs is evaluated. The reward signal guides the prompt optimization process by encouraging prompts that lead to better model outputs.

The reward function combines several evaluation metrics from the dataset to guide the reinforcement learning process. The implementation of the reward function used in the experiments is shown in code snippet below:

```
def reward_fn(generated_text: str, sample: Dict) -> float:
    reference_scores = sample["rewards"]
    hallucination_score = 3.0 - reference_scores["hallucination"]

    reward = (
        0.4 * reference_scores["accuracy"] +
        0.4 * hallucination_score +
        0.2 * reference_scores["coherence"]
    )
    return reward
```

Figure 10: Reward function used for prompt optimization.

where the reward value reflects how well the generated message satisfies the desired evaluation criteria. Higher reward values indicate better performance and encourage the reinforcement learning agent to generate similar prompts.

By optimizing prompts according to this reward signal, the RLPrompt framework is able to discover prompt structures that improve the quality of language model outputs.

3.5 Text Style Transfer Experiment

In addition to prompt optimization, this research investigates the use of text style transfer to transform outreach messages between different communication styles. Style transfer enables the modification of stylistic characteristics of text while preserving the original semantic meaning.

The experiments conducted in this study focus on transforming outreach messages between formal and casual communication styles. This allows the analysis of how style variations affect the tone and effectiveness of generated messages.

The style transfer experiments are performed in two directions: converting formal messages into casual messages and converting casual messages into formal messages.

3.5.1 Formal to Casual Transformation

In the first experiment, outreach messages written in a formal style are transformed into a casual style. The goal of this transformation is to generate messages that maintain the original meaning while adopting a more conversational tone.

Formal messages typically use structured language and professional vocabulary. In contrast, casual messages tend to use simpler language, shorter sentences, and a more informal tone.

During the transformation process, the language model modifies stylistic elements of the message while preserving the core content. This experiment allows the analysis of how style changes affect readability and communication effectiveness.

3.5.2 Casual to Formal Transformation

The second experiment focuses on transforming casual messages into formal communication. This process involves converting conversational language into a more professional tone suitable for business communication.

Formal messages generally exhibit characteristics such as:

- more structured sentence formation
- professional vocabulary
- polite and respectful tone

The transformation process ensures that the semantic meaning of the original message is preserved while improving its suitability for professional contexts.

By evaluating both transformation directions, the experiments provide insights into how style transfer techniques can be applied to automated communication systems.

3.6 Implementation Setup

3.6.1 Model Architecture

The implementation of the proposed approach is based on the RLPrompt framework, which applies reinforcement learning to optimize discrete prompts for pretrained language models. In this architecture, a prompt policy network is responsible for generating candidate prompts, while the pretrained language model remains frozen during the optimization process.

The generated prompts are combined with task inputs and passed to the language model to produce text outputs. These outputs are evaluated using predefined evaluation criteria, and the resulting reward signals are used to update the policy network.

This architecture allows the system to optimize prompts without modifying the parameters of the underlying language model.

3.6.2 Training Procedure

The training process follows the reinforcement learning workflow defined in the RLPrompt framework. At each iteration, the policy network generates candidate prompts that are used as input to the language model.

The language model produces text outputs based on the generated prompts, and these outputs are evaluated using the defined reward function. The reward signal is then used to update the policy network using a policy optimization algorithm.

Through repeated iterations, the policy network learns to generate prompts that improve the quality of the generated outputs.

3.6.3 Hyperparameters

The RLPrompt framework includes several hyperparameters that control the training process, such as the learning rate, prompt length, and the number of training iterations. These parameters influence how the policy network explores and updates candidate prompts during optimization.

In this study, the hyperparameters were set according to the default configuration provided in the RLPrompt implementation to ensure stable training and reproducibility of the experiments.

Chapter 4 – Experimental Setup

4.1 Prompt Optimization Setup

The experiments conducted in this study focus on applying reinforcement learning-based prompt optimization to generate outreach messages and analyze their transformation across different communication styles. The RLPrompt framework is used to automatically generate prompts that guide the language model during message generation.

In this setup, the prompt generation process is treated as a reinforcement learning problem. A prompt policy network generates candidate prompts, which are combined with task inputs and provided to a pretrained language model. The generated outputs are then evaluated, and the resulting reward signals are used to update the prompt policy.

The objective of the reinforcement learning process is to learn prompt structures that improve the quality and effectiveness of generated messages.

The optimization objective can be expressed as:

$$\max_{\theta} \mathbb{E}_{p_\theta(z)}[R(z)]$$

where:

- $z$represents the generated prompt
- $p_\theta(z)$represents the prompt policy parameterized by θ
- $R(z)$represents the reward associated with the generated output

Through iterative training, the policy network learns prompts that maximize the expected reward.

4.2 Training Configuration

4.2.1 Hardware and Software Environment

The implementation of the experiments was carried out using the RLPrompt framework in a Python-based machine learning environment. The experiments were executed using commonly used deep learning libraries and tools for reinforcement learning-based prompt optimization.

The experimental environment includes:

- Python programming language
- PyTorch deep learning framework
- RLPrompt implementation for prompt optimization
- pretrained language models for text generation

The pretrained language model remains frozen during training, while the prompt policy network is updated using reinforcement learning.

4.2.2 Training Procedure

The training process follows the reinforcement learning workflow defined in the RLPrompt framework. During each training iteration, the prompt policy network generates candidate prompts that are combined with the input data and passed to the language model.

The language model produces text outputs based on the generated prompts. These outputs are evaluated using a reward function that reflects the quality of the generated messages.

The policy network is then updated using a policy gradient optimization method. The policy gradient update can be expressed as:

$$\nabla_\theta J(\theta) = \mathbb{E}_{p_\theta(z)}[R(z)\nabla_\theta \log p_\theta(z)]$$

where:

- $J(\theta)$represents the expected reward
- $p_\theta(z)$represents the prompt generation policy
- $R(z)$represents the reward obtained from the generated output

Through repeated training iterations, the policy network gradually learns to generate prompts that improve the quality of language model outputs.

4.3 Evaluation Methodology

The generated outputs are evaluated using standard metrics commonly applied in text style transfer research. The evaluation focuses on three key aspects: style accuracy, content preservation, and fluency. Additional metrics such as joint score, geometric mean, BLEU, BERTScore, and perplexity are also used to provide a comprehensive assessment of the generated messages.

4.3.1 Style Accuracy

Style accuracy measures whether the generated message successfully adopts the intended target style. A pretrained style classifier is used to predict the style of each generated message.

$$Style = \frac{1}{N}\sum_{i=1}^{N} I(C(y_i) = t_i) \times 100$$

where $N$is the number of samples, $C(y_i)$is the predicted style label of the generated message y_i, and $t_i$is the target style label.

```
style_corrects = []
for i, c in enumerate(self.style_classifier(output_dataset)):
    style_corrects.append(int(c['label'] == target_labels[i]))

style = round(100 * np.array(style_corrects).mean(), 1)
```

Figure 11: Style accuracy computation using a pretrained classifier.

4.3.2 Fluency

Fluency measures the grammatical correctness and readability of the generated text. A grammaticality classifier is used to determine whether each generated sentence is linguistically acceptable.

$$Fluency = \frac{1}{N}\sum_{i=1}^{N} I(F(y_i) = 1) \times 100$$

where $F(y_i)$denotes the output of the grammaticality classifier.

4.3.3 Joint Score

A joint score is computed to evaluate style accuracy, content preservation, and fluency simultaneously.

$$Joint = \frac{1}{N}\sum_{i=1}^{N} Conten\,t_i \cdot Style_i \cdot Fluency_i$$

4.3.5 Geometric Mean

To obtain a balanced evaluation across the three primary metrics, the geometric mean is calculated as:

$$GM = (Content \cdot Style \cdot Fluency)^{1/3}$$

4.3.6 Additional Metrics

Additional metrics are used to further evaluate the generated outputs.

BLEU Score

$$BLEU = \frac{1}{N}\sum_{i=1}^{N} BLEU(y_i, r_i)$$

where $r_i$denotes the reference text.

Perplexity

Perplexity measures the likelihood of the generated text under a language model:

$$PPL = \exp\left(\frac{\sum NLL}{\sum T}\right)$$

where NLLrepresents the negative log-likelihood and Tdenotes the token length of the sentence.

Chapter 5 – Results and Analysis

5.1 Generated Message Examples

This section presents examples of generated outreach messages before and after the prompt optimization process. These examples illustrate how optimized prompts influence the structure, clarity, and tone of generated messages.

Example 1

Input:

Company: Namirial
Lead: Alessandro Mandolini

Generated Message (Initial Prompt):

```
Hello Alessandro, I noticed your role at Namirial and would like to connect with you to learn more about your work.
```

Generated Message (Optimized Prompt):

```
Hi Alessandro, I came across your work at Namirial and found it really interesting. I'd love to connect and exchange insights about your experience in the industry.
```

The optimized prompt produces a message that appears more natural and conversational while maintaining the core intent of the outreach.

Example 2

Input:

Company: TechSolutions
Lead: Maria Rossi

Generated Message (Initial Prompt):

```
Dear Maria, I am interested in learning more about your role at TechSolutions and would appreciate the opportunity to connect.
```

Generated Message (Optimized Prompt):

```
Hi Maria, I saw your profile while exploring TechSolutions and would enjoy connecting to learn more about your work and experiences.
```

5.2 Quantitative Results

The evaluation metrics include content preservation, style accuracy, and fluency, along with additional measures such as BLEU, BERTScore, and perplexity.

Metric	**Score**
Content Preservation	72.5
Style Accuracy	90.1
Fluency	86.1
Joint Score	58.2
Geometric Mean	83.5
BLEU	23.5
BERTScore	56.2
Perplexity	30.4

Higher values for content preservation, style accuracy, fluency, BLEU, and BERTScore indicate better performance, while lower perplexity indicates more fluent generated text.

The learning behavior of the RLPrompt framework during training is illustrated in Figure 5.1. The reward values gradually increase across training epochs, indicating that the prompt policy network successfully learns prompts that improve the quality of the generated outputs.

Figure 12: Reward progression during reinforcement learning training. As training progresses, the reward values gradually increase, indicating that the prompt policy network learns prompts that generate higher-quality outputs.

5.3 Style Transfer Results

In addition to prompt optimization, the experiments examine the performance of text style transfer applied to generated outreach messages. The style transfer experiments focus on transforming messages between formal and casual communication styles.

5.3.1 Formal → Casual Results

In the formal-to-casual transformation task, messages written in a professional tone are converted into a more conversational style. The results indicate that the generated messages maintain their original meaning while adopting simpler and more informal language.

Example transformation:

Formal message:

```
Dear Alessandro, I would appreciate the opportunity to connect and learn more about your professional experience.
```

Casual message:

```
Hi Alessandro, I'd love to connect and hear more about your work and experiences.
```

This transformation demonstrates that stylistic adjustments can be applied without significantly altering the core message content.

5.3.2 Casual → Formal Results

In the casual-to-formal transformation task, conversational messages are converted into more structured and professional communication.

Example transformation:

Casual message:

```
Hi Maria, I saw your profile and thought it would be great to connect.
```

Formal message:

```
Dear Maria, I came across your professional profile and would appreciate the opportunity to connect with you.
```

These transformations illustrate how the system can adapt messages to different communication contexts.

5.4 Analysis of Optimized Prompts

5.4.1 Learned prompts

During training, the prompt policy network gradually learns prompts that maximize the reward signal. These prompts often consist of token sequences that guide the language model toward generating more coherent and stylistically appropriate messages.

Examples of learned prompts include token combinations that emphasize professional communication, networking, and engagement with the target individual.

5.4.2 Prompt characteristics

The optimized prompts exhibit several notable characteristics:

- inclusion of context-related keywords such as *connect*, *experience*, and *opportunity*
- prompts that encourage conversational language
- prompts that guide the language model toward outreach-oriented text generation

These characteristics suggest that reinforcement learning can discover prompts that effectively steer the language model toward desired outputs.

5.4.3 Observed patterns

Several patterns were observed during the prompt optimization process:

- reward values gradually increased during training iterations
- optimized prompts produced more consistent message structures
- generated messages became more coherent and contextually relevant

These observations indicate that reinforcement learning-based prompt optimization can effectively guide language model behavior.

5.5 Discussion

5.5.1 Interpretation of results

The results demonstrate that reinforcement learning-based prompt optimization can improve the quality and stylistic adaptability of generated outreach messages. The optimized prompts guide the language model toward generating more natural and contextually appropriate messages while preserving the core information of the original input.

The style transfer experiments further show that generated messages can be adapted to different communication styles without significantly altering their semantic meaning.

5.5.2 Limitations of the approach

Despite the promising results, several limitations should be acknowledged.

First, the reward function relies on proxy evaluation metrics derived from the dataset, which may not fully capture the quality of newly generated messages. Second, the evaluation process depends on pretrained classifiers and semantic similarity measures, which may introduce biases in the evaluation results.

Additionally, the experiments were conducted using a limited dataset and a single language model architecture, which may restrict the generalizability of the findings.

Future work could explore larger datasets, alternative reward formulations, and more advanced language models to further improve prompt optimization and style transfer performance.

Chapter 6 – Conclusion and Future Work

6.1 Summary of Findings

This thesis investigated the use of reinforcement learning for prompt optimization in the context of automated outreach message generation. The RLPrompt framework was applied to optimize prompts that guide a pretrained language model in generating high-quality messages. In addition, text style transfer techniques were explored to transform messages between formal and casual communication styles.

The experimental results demonstrate that reinforcement learning can effectively optimize prompts, leading to improved message quality and stylistic adaptation. The evaluation results indicate that the generated messages maintain semantic consistency while successfully adopting the intended style.

6.2 Contributions of the Research

The main contributions of this research are summarized as follows:

- Application of reinforcement learning–based prompt optimization for automated outreach message generation.
- Implementation of the RLPrompt framework to learn prompts that improve the quality of generated messages.
- Integration of text style transfer to adapt generated messages between formal and casual communication styles.

- Evaluation of generated outputs using style transfer metrics including style accuracy, content preservation, and fluency.

These contributions demonstrate the potential of reinforcement learning techniques for improving prompt-based language model applications.

6.3 Limitations

Despite the promising results, several limitations should be acknowledged. The experiments were conducted on a relatively limited dataset of outreach messages, which may restrict the generalizability of the findings. Additionally, the reward function relies on proxy evaluation signals, which may not fully capture all aspects of message quality.

Furthermore, the experiments were performed using a single language model architecture, which may limit the exploration of alternative model capabilities.

6.4 Future Research Directions

Future work could extend this research in several directions. First, larger and more diverse datasets could be used to improve the robustness of the prompt optimization process. Second, more advanced language models could be explored to further enhance the quality of generated messages. Additionally, improved reward functions and evaluation strategies could be developed to better capture stylistic and semantic aspects of generated text.

Finally, the proposed approach could be applied to other natural language generation tasks, such as automated email generation, conversational agents, and personalized communication systems.

References

[1] Vaswani et al., *Attention Is All You Need*, 2017
[2] Devlin et al., *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, 2018
[3] Brown et al., *Language Models are Few-Shot Learners*, 2020.
[4] Liu et al., *Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in NLP*, 2021.
[5] Reynolds and McDonell, *Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm*, 2021.

[6] Lester et al., *The Power of Scale for Parameter-Efficient Prompt Tuning*, 2021.

[7] Shin et al., *AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts*, 2020.

[8] Li and Liang, *Prefix-Tuning: Optimizing Continuous Prompts for Generation*, 2021.

[9] Sutton and Barto, *Reinforcement Learning: An Introduction*, 2018.
[10] Ranzato et al., *Sequence Level Training with Recurrent Neural Networks*, 2016.
[11] Deng et al., *RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning*, 2022.

[12] Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, 2020.

[13] Radford et al., "Learning Transferable Visual Models From Natural Language Supervision," *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[14] Ouyang et al., "Training Language Models to Follow Instructions with Human Feedback," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[15] Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[16] Kojima et al., "Large Language Models are Zero-Shot Reasoners," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[17] OpenAI, "GPT-4 Technical Report," *arXiv preprint arXiv:2303.08774*, 2023.

[18] Dong et al., "A Survey on In-Context Learning," *arXiv preprint arXiv:2301.00234*, 2023.

[19] Zhou et al., "Large Language Models Are Human-Level Prompt Engineers," *International Conference on Learning Representations (ICLR)*, 2023.

[20] Liu et al., "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in NLP," *ACM Computing Surveys*, 2023.

[21] Min et al., “Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?” *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

[22] Jin et al., “Deep Reinforcement Learning for Text Generation: A Survey,” *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[23] Lample et al., “Multiple-Attribute Text Rewriting,” *International Conference on Learning Representations (ICLR)*, 2019.

[24] Hu et al., “LoRA: Low-Rank Adaptation of Large Language Models,” *International Conference on Learning Representations (ICLR)*, 2022.

[25] Touvron et al., “LLaMA: Open and Efficient Foundation Language Models,” *arXiv preprint arXiv:2302.13971*, 2023.

[26] Touvron et al., “LLaMA 2: Open Foundation and Fine-Tuned Chat Models,” *arXiv preprint arXiv:2307.09288*, 2023.

Appendices

Possible appendices:

Appendix A – Generated Message Samples

“Salve Vito,

È un piacere scoprire la tua esperienza inerunne la crescita di Namirial. Con la nostra esperienza in digital tailored solutions, CX-UX-UI design, app mobile e strategie digitali, crediamo di poter collaborare con successo al fine di raggiungere i vostri obiettivi di digitalizzazione.

Vorremmo conoscere meglio la vostra organizzazione e le sue esigenze per poterle presentare soluzioni mirate.

Vorremmo essere lieti di fissare un incontro per approfondire la questione.

Saluti saluti!

Davide Morra”

"Salve Andrea,

Mi chiamo Davide Morra e sono cofondatore della Volcanic Minds, una company di tecnologia digitale dedicata a soluzioni personalizzate di alta qualità.

Mi ha fatto piacere scoprire la vostra azienda, Namirial, e la vostra esperienza nel settore della gestione digitale delle transazioni.

Ci piacerebbe explores la possibilità di collaborare con voi per affrontare i vostri obiettivi aziendali in materia di tecnologia.

Vorremmo fissare un incontro per approfondire la questione e sapere come possiamo essere d'aiuto.

Grazie mille per il vostro tempo e la vostra attenzione.

Saluti saluti,

Davide Morra"