UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

Meaning Change Detection

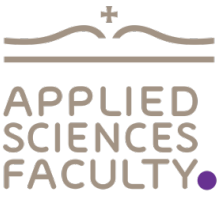
Author:
Iryna PASTUKHOVA

Supervisor:
Ph.D. Andrii LIUBONKO

A thesis submitted in fulfillment of the requirements
for the degree of Master of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences



Lviv 2024

Declaration of Authorship

I, Iryna PASTUKHOVA, declare that this thesis titled, "Meaning Change Detection" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"You can always recognize truth by its beauty and simplicity. "

Richard Feynman

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Master of Science

Meaning Change Detection

by Iryna PASTUKHOVA

Abstract

Detecting changes in the meaning of text after paraphrasing or editing is a challenging and non-trivial task in natural language processing (NLP). It is implicitly involved in other tasks such as translation, summarisation, and style transfer. Approaches to meaning change detection (or paraphrase identification) have evolved as the field of NLP has developed. Today, deep learning BERT-based models and Large Language Models (LLMs) provide state-of-the-art results. However, these methods need more interpretability and control and are computationally expensive. There are alternative methods based on linguistic and mathematical ideas that can overcome the shortcomings of LLMs and DL methods or complement them.

We aim to investigate the possibilities and limitations of one such alternative approach compared to state-of-the-art solutions for the paraphrase identification task.

Acknowledgements

I want to express my deepest gratitude to my master's thesis advisor, Andrii Liubonko, for his guidance, fruitful discussions, invaluable patience, and encouragement throughout the entire process of researching and writing this thesis.

I would also like to acknowledge the teachers of the Faculty of Applied Sciences for providing a nurturing, friendly, and intellectually stimulating environment.

Contents

List of Figures

List of Tables

List of Abbreviations

DL	**D**eep **L**earning
NLP	**N**atural **L**anguage **P**rocessing
PAWS	**P**araphrase **A**dversaries (from) **W**ord **S**crambling
PI	**P**araphrase **I**dentification
QNLP	**Q**uantum **N**atural **L**anguage **P**rocessing
QQP	**Q**uora **Q**uestion **P**airs

For everyone who cares

Chapter 1

Introduction

Detecting changes in the 'meaning' of the textual data presents a unique challenge in the field of Natural Language Processing (NLP). From a linguistic point of view, the understanding of 'text meaning' and its changes involves the analysis of syntax, semantics, pragmatics and the evolving nature of language itself. In NLP, the focus on detecting changes in text meaning primarily revolves around recognizing the shifts that occur following editing or paraphrasing activities. It is interconnected with other tasks such as translation, Grammar Error Correction (GEC), summarization, style transfer, paraphrasing, and others.

For example, in a translation task, recognizing changes in meaning is crucial to ensuring that the translated text truthfully conveys the original meaning. Similarly, in Grammar Error Correction (GEC), understanding the intended meaning is essential for making appropriate corrections without altering the original message. Summarization tasks depend heavily on identifying key ideas, requiring an acute sense of how meaning can be condensed without loss or distortion. Style transfer involves altering the tone or formality of a text while maintaining its original meaning. Lastly, paraphrasing involves generating semantically equivalent alternatives that maintain the original text's tone, style, and intent.

All these tasks require that the output preserves the original meaning of the input. This can be achieved implicitly or explicitly. When considered as a separate task, meaning change detection (also known as paraphrase identification) can be formulated as a classification task. It takes two versions of the text as input and outputs either a binary yes/no or a score. Some examples are given in Fig 1.1. When implemented, it can serve as an additional module in the corresponding NLP pipeline. This module ensures that the meaning is not changed.

Approaches to meaning change detection have evolved with advances in machine learning and natural language processing. In general, there are two main strategies for solving this problem Zhou, Qiu, and Acuna, 2022, which are illustrated in Fig 1.2:

- **Traditional approaches.** Traditional approaches are largely based on syntax, semantics, rule-based heuristics and statistical methods. This usually involves obtaining a structural representation of the data, which is then followed by some kind of matching algorithm.
- **Deep Learning-based approaches.** Deep learning-based approaches use neural network architectures and large datasets. They offer more context-aware solutions, but lack interpretability and require significant computational resources.
- **LLM (Prompt)-based approaches.** LLM-based approaches involve using the prompting of the model, that is, providing it with an initial input or "prompt" to guide its text generation or completion.

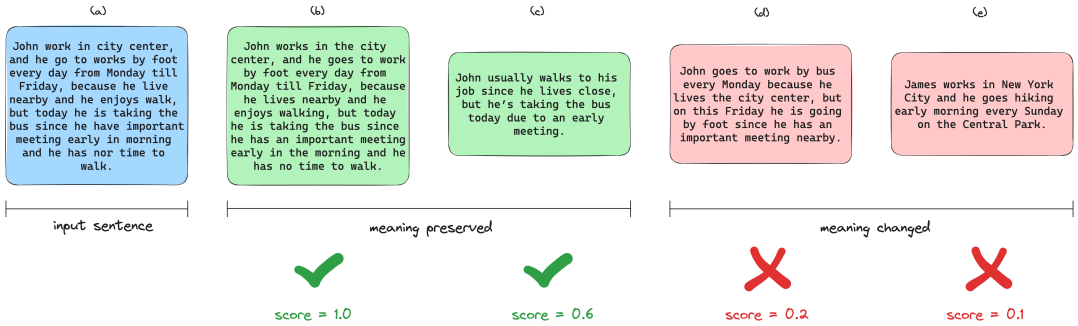


FIGURE 1.1: Four paraphrased versions of an original sentence are presented, each with a corresponding evaluation score. The examples range from high fidelity (a and b) to significant deviation (d and e) from the original input.

State-of-the-art (SOTA) results on relevant datasets for paraphrase identification are provided by BERT-like deep learning models (such as BERT, RoBERTa and ERNIE) Devlin et al., 2019, Liu et al., 2019, Sun et al., 2019. The SOTA results for paraphrase identification are likely to be updated soon with the recent emergence of GPT-based large language models (LLMs) such as LLAMA Touvron et al., 2023, MISTRAL[1], GPT4[2] and Anthropic[3]. While BERT-like and LLMs-based approaches are impressive, they have their shortcomings. Most notably, they lack interpretability and require significant computational resources.

Applied Category Theory (ACT) has recently emerged as a coherent approach to many problems. Within the NLP domain, ACT proposes to extend the categorical theory to model the structure and semantics of natural language by capturing its compositionality Coecke, Sadrzadeh, and Clark, 2010. Applying these ideas to NLP is still in its early stages and needs further development and validation.

DisCoCat Coecke, Sadrzadeh, and Clark, 2010 and its subsequent expansion through DisCoCirc Wang-Mascianica, Liu, and Coecke, 2023 offer a concrete realization of this idea. These frameworks combine category theory, linguistics, and quantum mechanics to provide a mathematical and diagrammatic foundation for understanding how meaning is composed in sentences and how different linguistic elements interact. Additionally, such structural representation of the meaning enables a more transparent view than large language models. Also, it lends itself more naturally to processing NLP problems on quantum hardware once it is widely accessible (more details are provided in the section 2.2 below). Consequently, it is sometimes dubbed "Quantum Natural Language Processing" (QNLP).

We aim to investigate the possibilities and limitations of DisCoCirc compared to state-of-the-art solutions for the paraphrase identification task.

The rest of the thesis is organized as follows: Chapter 2 is dedicated to the literature review; in particular, section 2.1 describes the main problem; section 2.2 gives an overview of the Distributional Compositional NLP frameworks; section 2.3 discusses the data used in the investigation. Chapter 3 is dedicated to applying the SOTA approaches to the solution of the PI task, and Chapter 4 is to the QNLP approaches. In particular, each sets its goals, describes the methods and tools used for the experiments, and reports the results. The last 5th Chapter summarizes the results and outlines the future work.

[1] https://mistral.ai/news/announcing-mistral-7b/
[2] https://openai.com/gpt-4
[3] https://www.anthropic.com/product

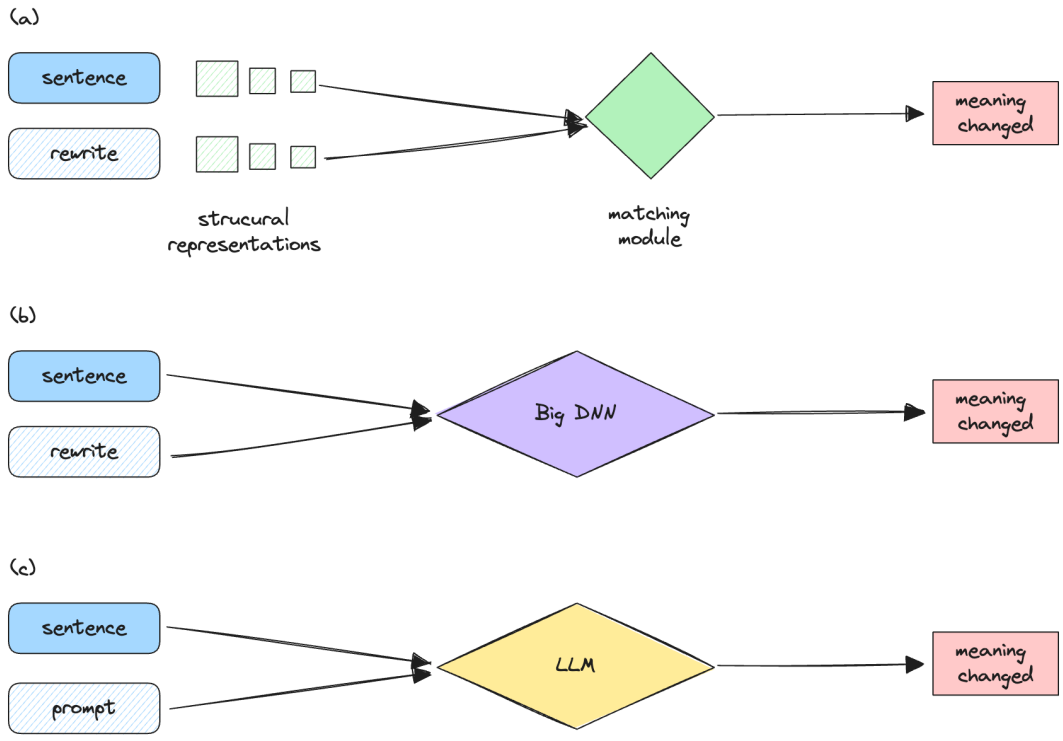


FIGURE 1.2: Three approaches to paraphrase identification. (a) Traditional approaches, (b) Deep-learning- and (c) LLM-based ones.

Chapter 2

Related Work

2.1 Paraphrase Identification

Paraphrase Identification (PI) is the task of determining whether two given sentences express the same or very similar meaning despite potentially using different words or structures. In some sense, it can be considered as a measuring tool of the feasibility of a meaning change detection method.

PI is a crucial NLP task having implications in many other tasks like question answering Dong et al., 2017, summarization Hardy and Vlachos, 2018, translation Thompson and Post, 2020, plagiarism detection Wahle, Gipp, and Ruas, 2023, etc. Zhou, Qiu, and Acuna, 2022. The last becomes especially important in light of the intensive development of generative models Becker et al., 2023.

In general, there are two main strategies for solving the PI problem: traditional and deep learning-based. The formers mostly use lexical structures and probabilistic methods for getting meaning and capturing similarities. They include so-called knowledge-based and corpus-based methods, focused on lexical and semantical text knowledge, respectively. The latter techniques provide more accurate solutions and are known to achieve state-of-the-art performance for detecting sophisticated paraphrases Zhou, Qiu, and Acuna, 2022.

Nowadays, the best performance in PI tasks is obtained by transformers Becker et al., 2023. Thus, a common approach is fine-tuning a pre-trained model on custom data. However, even well-performing state-of-the-art solutions can give unpredictable results on even such simple tasks as identifying pairs of two identical or randomly selected sentences Chen, Ji, and Evans, 2020; lack of efficiency is also observed for plagiarism detection Foltýnek et al., 2020 and limited abilities for questions paraphrasing Ribeiro et al., 2020.

Considering all the above problems and the black-boxiness of DL models, the alternative way to obtain a solution seems worth finding.

2.2 Distributional Compositional Models

There are two main approaches in NLP to understanding the meaning: compositional and distributional. The former assumes that the meaning of the sentence is determined by the meanings of its constituent parts and the way they are combined. The latter relies on the idea that words that occur in similar contexts tend to have similar meanings; that is, the distribution of words in a large corpus of text captures semantic relations Salton, Wong, and Yang, 1975.

FIGURE 2.1: The transformation of text into text circuits

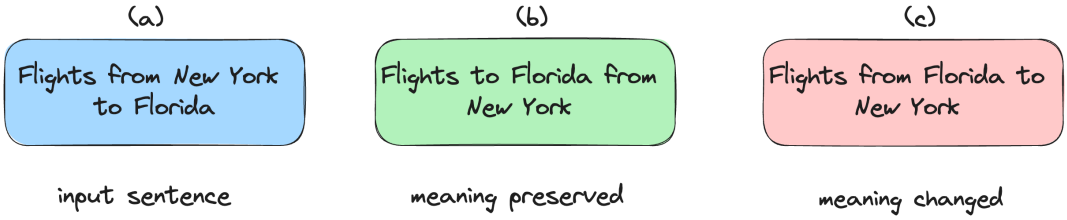


FIGURE 2.2: Examples of how slight word reordering changes the sentence meanings Zhang, Baldridge, and He, 2019

In Coecke, Sadrzadeh, and Clark, 2010, both approaches are unified and combined with the category theory into a DisCoCat (Categorical Compositional Distributional) framework, giving a mathematical foundation explaining how word interaction determines the sentence meaning. It relies on diagrammatic calculus and sentence string diagram representation.

DisCoCirc was formulated as the further DisCoCat improvement in Coecke, 2021. It is also both compositional and distributive but additionally is able to track word-meaning evolution through the text and to represent the meaning of the text, not only single sentences. While DisCoCat represents the sentences in the form of string diagrams, DisCoCirc represents text in the form of text circuits. DisCoCirc allowed to build the correspondence between text and text circuits. In particular, the Text Circuit Theorem builds a surjection between the set of texts and text circuits; as a consequence – texts are equivalent if they have equal circuits (the corresponding text transformation chain is presented in Fig 2.1 Wang-Mascianica, Liu, and Coecke, 2023). The last statement potentially provides a solution to the PI task: sentences with similar circuits must be paraphrases. Moreover, such a structural approach gives more transparency in obtaining the solution compared to the DL one.

The formalism and ideas of DisCoCat and DisCoCirc are inspired by quantum theory and consequently should be highly effective on quantum computers after they become available Coecke, 2021.

2.3 Data Selection

It is difficult to overestimate the importance of data quality, quantity, and diversity in solving NLP tasks. While machine-generated and annotated paraphrases become more popular nowadays, according to Becker et al., 2023 human-authored ones still appear to be more difficult, diverse and suitable. In this context, paraphrase identification is especially sensitive and needs careful data selection.

Most paraphrase datasets lack sentence pairs with high lexical similarity but different meanings Zhang, Baldridge, and He, 2019, which significantly affects method efficiency. In particular, even state-of-the-art, well-performant models trained on such datasets are not able to differentiate between sentences even with slight changes in word order (see the relevant examples in Fig 2.2 Zhang, Baldridge, and He, 2019).

TABLE 2.1: PAWS and QQP datasets

	PAWS	**QQP**
Train set	49401	384335
Test set	8000	9999
Validation set	8000	9999

All three sentences in the figure have high bag-of-words (BOW) overlap. However, (2) is a paraphrase of (1), while (3) has a very different meaning from (1).

To resolve this issue, the authors of Zhang, Baldridge, and He, 2019 built a dataset of non-paraphrase sentence pairs having high lexical overlap called PAWS (Paraphrase Adversaries from Word Scrambling)[1]. Six models of different complexities (BOW, BiLSTM, ESIMChen et al., 2017, DecAttParikh et al., 2016, DIINGong, Luo, and Zhang, 2018, BERT) were tested for the ability to achieve a high sensitivity to the sentence structure during the training on PAWS. The initial accuracy on PAWS for all of them was less than 40%. The training step revealed their possibilities to learn and distinguish the structural features of the sentences: DIIN and BERT performance increased significantly to 84% and 85%, respectively; BiLSTM, ESIM, and DecAtt added from 23% to 34% to their accuracy; and BOW – only 1%, that is, it learns nothing.

The paraphrases in PAWS are built based on the text data from Quora and Wikipedia, and the dataset consists mostly of negative examples generated in 3 steps, including controlled word swapping, back translation into German, and, finally, thorough people judgments, thus making it a good choice. German was chosen because it offers more word reorder options and translation quality than other languages.

Our initial intention was to consider only the PAWS dataset for the investigation. However, the additional experiments with the Quora Question Pairs (QQP)[2] dataset revealed interesting, dependent on the data, differences in the models' behaviors (see Chapters 3, 4 for the details).

QQP consists of question pairs labeled 0s and 1s according to the duplication criteria. On the one hand, as was mentioned above, PAWS was built based on text data from Quora. On the other hand, PAWS is a more "refined" dataset, while QQP is less processed and, thus, in some sense, more "natural".

We used the QQP partition from the Wang, Hamza, and Florian, 2017 but reduced the QQP train set to a size comparable to the PAWS.

The key statistics about both datasets are presented in Table 2.1, and examples are provided in Tables 2.2 and 2.3, respectively.

[1] https://github.com/google-research-datasets/paws

[2] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

TABLE 2.2: PAWS examples

Sentence 1	**Sentence 2**	**Is Paraphrase**
In Paris , in October 1560 , he secretly met the English ambassador , Nicolas Throckmorton , asking him for a passport to return to England through Scotland	In October 1560 , he secretly met with the English ambassador , Nicolas Throckmorton , in Paris , and asked him for a passport to return to Scotland through England	0
When comparable rates of flow can be maintained , the results are high .	The results are high when comparable flow rates can be maintained .	1
Pluto was classified as the planet when the Grand Tour was proposed and was launched at the time " New Horizons " .	Note : Pluto was classified as a planet when the Grand Tour was launched and at the time " New Horizons " was proposed .	0

TABLE 2.3: QQP examples

Sentence 1	**Sentence 2**	**Is Paraphrase**
How do you start a bakery ?	How can one start a bakery business ?	1
What are the requirements to build my own server ?	What do I need to build my own server ?	0
Which programming Python or Java learn first ?	Should I learn python or Java first ?	1

Chapter 3

DL and LLM Experiments and Evaluation

3.1 Motivation and Approach

Based on the fact that the most state-of-the-art results in NLP are achieved by applying DL models and LLMs, it is natural to consider their application to the PI task.

This chapter aims to investigate and compare these approaches. In particular, we consider the RoBERTa to represent the DL approach and GPT-3.5-turbo and GPT-4.0 – the LLM ones.

The choice of the LLM model was motivated by its popularity and earlier release compared to other similar models, and the RoBERTa was due to the success of the BERT-like models in solving different NLP tasks.

Each approach will be applied and evaluated on two benchmark datasets: PAWS and QQP. The selection of these datasets was based on their diverse characteristics and wide usage in the PI research community, ensuring a comprehensive evaluation of our models.

3.2 Evaluation Metrics

As we consider the PI the usual binary classification task, the standard four metrics were chosen to evaluate and compare the performance of the used methods:

- **Accuracy**: The proportion of correctly identified paraphrase pairs out of the total number of pairs.
- **Precision**, **Recall**, and **F1 Score**: These metrics provide insights into the models' true positive rates, false positive rates, and the balance between precision and recall, respectively. Precision indicates the accuracy of positive predictions, recall shows the percentage of true positives captured by the model, and the F1 score provides a harmonic mean of precision and recall.

3.3 DL Experiments and Evaluation

3.3.1 Problem Statement and Goals

As mentioned above, we selected the RoBERTa Liu et al., 2019 base as a concrete representative of the DL model for our experiments. This decision was made due to the following facts:

- The BERT-like models are significantly influential in NLP providing the DL-based SOTA solutions for most of the NLP tasks.

- In Zhang, Baldridge, and He, 2019, the superiority of the considered PAWS dataset was reasoned by the estimation of the BERT model itself; in particular, the reported accuracy is 90.4%. So, it is natural to check the abilities of RoBERTa, the direct BERT improvement, on this dataset.

RoBERTA is built above the BERT architecture with several modifications which improve its performance and robustness. In particular, it was trained on the larger and more diverse data with bigger batches. It had a longer training duration, the static masking strategy was switched to the changing for each training epoch dynamic one, and the Next Sentence Prediction objective was removed from pre-training. With such modifications, RoBERTa is reported to beat the BERT in all the conducted experiments Liu et al., 2019.

Taking into account all the above, the following goals were stated:

- Fine-tune and evaluate the RoBERTa base model on the PAWS dataset and compare the obtained accuracy with the one reported in Zhang, Baldridge, and He, 2019.
- Fine-tune and evaluate the RoBERTa base model on the QQP dataset and compare its performance to the obtained on PAWS.
- Cross-evaluate the trained models on both datasets to get insights into the model's ability to generalize across datasets and explore its adaptability and robustness, and also check the claim made in Zhang, Baldridge, and He, 2019 about the universality of the training on the PAWS dataset.

3.3.2 Training and Evaluation

For our experiments, the HyggingFace's[1] RoBERTa base distribution was chosen. In particular, we used 'RobertaForSequenceClassification' – the customized with the "classification head" RoBERTa version[2]. This architecture modification allows the model to make binary decisions regarding the meaning similarity of sentence pairs.

Each sentence pair was concatenated into a single text and tokenized using the 'RobertaTokenizer' provided by HuggingFace. The sentences were separated using a special token, '<SEP>'. This format was necessary to transform the sentence pairs into a single input sequence that the RoBERTa model could process effectively.

In order to conduct the experiments corresponding to the above research plan, we started by training the RoBERTa model in the base configuration using the final labeled version of the PAWS dataset, containing 49.401 training samples alongside 8,000 validation samples and 8,000 testing samples. Unfortunately, from all the selected metrics, only accuracy was reported in the original PAWS paper for BERT (see Table 3.1). Comparing it with our results, one can conclude that RoBERTa performs better with the given setup.

Inspired by the above-obtained results, confirming the outperformance of RoBERTa over the BERT, we continued with fine-tuning RoBERTa base on QQP data distribution, with a random selection of 50000 training samples each epoch and 9999 testing and validation each.

The final metrics of all the experiments are reported in Table 3.2. Here, "PAWS on QQP" denotes the evaluation of the QQP test dataset by the model trained on PAWS; by analogy, the notation "QQP on PAWS" must be treated.

[1] https://huggingface.co/

[2] https://huggingface.co/docs/transformers/v4.40.2/en/model_doc/roberta

TABLE 3.1: BERT's accuracy vs. RoBERTa's accuracy on PAWS

	Accuracy
BERT	90.4 %
RoBERTa base	93.75 %

TABLE 3.2: RoBERTa experiments results on PAWS, QQP, and corresponding cross-evaluations

	Accuracy	**Precision**	**Recall**	**F1 score**
PAWS	93.75 %	95.36 %	90.94 %	93.10 %
QQP	87.00%	86.10 %	87.68 %	86.88 %
PAWS on QQP	67.75 %	85.40 %	63.11 %	72.58 %
QQP on PAWS	46.86 %	89.96 %	44.95 %	59.95 %

3.4 LLM Experiments and Evaluation

3.4.1 Prompt Engineering

It is known that the relevance and quality of the LLM answer strongly depend on the prompt used (White et al., 2023), and it is difficult to estimate the efficiency of the prompts.

Our experiments used zero-shot prompting, relying on the models' ability to generalize based on their pre-existing knowledge. In particular, the two GPT models were used: GPT-3.5-turbo[3] and GPT-4[4]. For the final testing phase, three distinct prompts were formulated.

Generating the final prompts included the following steps:

- **Defining the initial instruction.** The initial prompt had a very base form that clearly communicated the task to the models.
- **Iterative Debugging and Modification.** Through an iterative process, the initial prompt was refined and adjusted according to the GPT proposed strategies[5].
- **Reformulation Based on Model Feedback.** The final step entailed reformulating the prompt based on the model responses.

To debug the prompts, we used the paraphrases that were incorrectly predicted by the trained RoBERTa model. These mispredictions helped us to identify weak points in our prompts and make necessary adjustments to enhance their clarity and precision. Examples of the mispredicted paraphrases are presented in Table 3.3.

We started from the simplest version of the prompt and iteratively changed and formed it according to the several most relevant strategies recommended in the GPT prompting guide, such as:

- Applying formatting

[3] https://platform.openai.com/docs/models/gpt-3-5-turbo
[4] https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4
[5] https://platform.openai.com/docs/guides/prompt-engineering

TABLE 3.3: RoBERTa's incorrect predictions examples on PAWS

Sentence 1	Sentence 2	Actual	Predicted
The Chilean New Song movement was encouraged in the late 1950s and early 1960s by a renewed interest in traditional Chilean music and folklore.	The Chilean New Song movement was fired by the renewed interest in traditional Chilean music and folklore in the late 1950s and early 1960s.	0	1
The Culme family acquired the sub-manor of Great Champson in Molland and held Canonsleigh Abbey after the Dissolution of the Monasteries.	The Culme family acquired the Great Champson underground in Molland and , after the dissolution of the monasteries , held the Canonsleigh Abbey.	0	1
The first landing in Lae Airfield was possessed by Ernest Mustar on April 19 , 1927 in a De Havilland DH.37 by Guinea Gold Airways from Wau .	The first landing at Lae Airfield was made on 19 April 1927 by Ernest Mustar in a De Havilland DH.37 owned by Guinea Gold Airways from Wau .	1	0

- Including details in the query
- Adopting a persona

Our starting point was a straightforward instruction pattern:

```
Check if the following two sentences are paraphrases:
Sentence 1: ""
Sentence 2: ""
```

Testing this prompt on several samples quickly revealed incorrect predictions. Table 3.4 shows examples of mismatches between the model's output and the actual labels.

As the next improvement, we made the prompt to adopt the persona of a professional linguist. The resulting prompt had the following form:

```
You are a professional linguist
```

refined the instruction with:

```
with "True" if sentences have the same meaning and "False" otherwise.
```

and added an output validator:

```
Return the result for the following two sentences:
```

Testing this prompt version on the same samples showed that adding a persona made the model's answers "more human": they started coinciding with the actual ("human-annotated") labels (see the corresponding examples in Table 3.5).

TABLE 3.4: Comparison of the predictions obtained by the initial prompt and the actual values

Sentence 1	**Sentence 2**	**Actual**	**Prompt 1**
On July 21, 2014, after his two successful years at Las Palmas in Spain, Chrisantus signed a three-year contract with the Turkish Super Lig - Club Sivasspor.	On July 21, 2014, after his two successful years at Las Palmas in Spain, Chrisantus signed a three-year contract with Sivasspor at the Turkish Super Lig.	0	1
Alice Hopkins, daughter of Thomas Hopkins, married a merchant of London.	Lee married Alice Hopkins, daughter of Thomas Hopkins a merchant of London.	1	0

TABLE 3.5: Comparison of the predictions obtained by the two different prompts and the actual values

Sentence 1	**Sentence 2**	**Actual**	**Prompt 1**	**Prompt 2**
The Chilean New Song movement was encouraged in the late 1950s and early 1960s by a renewed interest in traditional Chilean music and folklore.	The Chilean New Song movement was fired by the renewed interest in traditional Chilean music and folklore in the late 1950s and early 1960s.	0	1	0
The Culme family acquired the sub-manor of Great Champson in Molland and held Canonsleigh Abbey after the Dissolution of the Monasteries.	The Culme family acquired the Great Champson underground in Molland and , after the dissolution of the monasteries , held the Canonsleigh Abbey.	0	1	0

Several debugging iterations on the above-mentioned set of paraphrases allowed the forming of the smallest prompt able to predict labels coinciding with the human annotations. It had the following pattern:

```
You are a professional linguist who annotates pairs of sentences with
1 if sentences have the same meaning and 0 otherwise. Return
the result for the following two sentences:
Sentence 1: ""
Sentence 2: ""
```

As the final improvement, we considered reformulating the obtained prompt with words potentially more familiar to the model. We formed it by asking the model and using the answer that was generated accordingly. In particular, we requested a model in the following way: "Write a prompt for a language professional deciding if two sentences have the same or different meaning". Thus, the third version of the prompt used:

```
You are an expert in linguistic analysis tasked with evaluating whether
two given sentences convey the same or different meanings. Return 1
if the following two sentences convey essentially the same meaning and 0
if they convey slightly or substantially different meanings:
Sentence 1: ""
Sentence 2: ""
```

3.4.2 Tests

We ran the tests with different "prompt-model" combinations to compare the applicability of the above prompts for 500 samples from each dataset. The obtained results for the PAWS and QQP datasets are presented in Tables 3.6 and 3.7, respectively.

The first observation here is that, in general, both GPT-3.5-turbo and GPT-4.0 performed significantly better on QQP data. The most initial and intuitive conjecture is that GPT is able to understand less 'cultivated,' more straightforward in nature data better.

Another interesting fact is that GPT-3.5-turbo on PAWS seems to be more sensitive to the prompt's complications than GPT-4.0, which behaves almost constantly.

On the QQP dataset, an increase in prompt complexity negatively impacted the performance of both models, although the effect was more pronounced in GPT-3.5-turbo.

One more thing worth mentioning is that, overall, on both datasets, GPT-3.5-turbo showed stronger sensitivity to the prompt's changes.

Basically, GPT model complexity is in some sense inversely proportional to prompt complexity: GPT-3.5-turbo performs better on the more detailed prompts, while GPT-4 shows the best results on the simpler prompts and loses the performance with added instructions. This suggests that the more advanced GPT-4.0 can infer complex tasks from minimal instructions, whereas GPT-3.5-turbo requires more explicit guidance.

TABLE 3.6: Evaluation results of prompts on the PAWS dataset

	Accuracy	**Precision**	**Recall**	**F1 score**
Prompt 1 gpt-3.5-turbo	66.80 %	92.76 %	57.75 %	71.18 %
Prompt 1 gpt-4	73.60 %	96.83 %	63.13 %	76.43 %
Prompt 2 gpt-3.5-turbo	63.40 %	85.97 %	55.56 %	67.50 %
Prompt 2 gpt-4	**74.60 %**	**94.57 %**	**64.51 %**	**76.70 %**
Prompt 3 gpt-3.5-turbo	**76.20 %**	**64.25 %**	**78.02 %**	**70.47 %**
Prompt 3 gpt-4	73.40 %	94.12 %	63.41 %	75.77 %

TABLE 3.7: Evaluation results of prompts on the QQP dataset

	Accuracy	**Precision**	**Recall**	**F1 score**
Prompt 1 gpt-3.5-turbo	**81.80 %**	**76.80 %**	**85.33 %**	**80.84 %**
Prompt 1 gpt-4	80.20 %	68.00 %	89.95 %	77.45 %
Prompt 2 gpt-3.5-turbo	77.40 %	76.80 %	77.73 %	77.26 %
Prompt 2 gpt-4	78.40 %	61.60 %	92.77 %	74.04 %
Prompt 3 gpt-3.5-turbo	61.40 %	24.80 %	92.54 %	39.12 %
Prompt 3 gpt-4	78.00 %	60.40 %	93.21 %	73.30 %

3.4.3 Final Evaluation

As the final evaluation step for GPT as an LLM method of gaining the PI task solution, the experiments were conducted on test partitions of both PAWS and QQP datasets.

Relying primarily on accuracy and F1 score, according to our tests, the optimal "prompt-model" combinations are "Prompt 2 gpt-4" and "Prompt 3 gpt-3.5-turbo" on PAWS, and "Prompt 1 gpt-3.5-turbo" for QQP. Thus, for consistent comparison using the same model version, "Prompt 3 gpt-3.5-turbo" on PAWS and "Prompt 1 gpt-3.5-turbo" on QQP were used as benchmarks. Table 3.8 presents the final results, highlighting the comparative performance insights across datasets.

TABLE 3.8: Performance of GPT-3.5-turbo on PAWS and QQP test subsets

	Accuracy	**Precision**	**Recall**	**F1 score**
Prompt 3 gpt-3.5-turbo on PAWS	77.53 %	66.54 %	79.28 %	72.36 %
Prompt 1 gpt-3.5-turbo on QQP	81.03 %	77.26 %	83.56 %	80.28 %

3.5 Conclusions

In the chapter, we systematically evaluated two SOTA solutions for the PI task: DL-based and LLM-based ones. The results of the conducted experiments imply several insights and observations:

- They one more time reaffirmed the superiority of RoBERTa over the BERT.
- Despite the inherent difficulty of PAWS, RoBERTa outperformed its performance on QQP. This re-confirms the model's ability to capture complex patterns and represents the refined nature of the PAWS dataset.
- Cross-evaluation of the pre-trained models, in general, approved the beneficial impact of the PAWS dataset in enhancing the model's generalization capabilities stated in Zhang, Baldridge, and He, 2019: indeed, the pre-training

on PAWS provided much higher accuracy on QQP than the converse case. At the same time, these cross-evaluation results are much lower than the ones obtained by the model trained on QQP; that is, PAWS are still not perfect in generalizing, as could be expected.

- Confirmed an intuitive assumption about LLMs abilities: despite their capacity and general-purpose capabilities, the LLM determines the paraphrases worse than the fine-tuned DL model.
- Revealed an interesting distinction in the performance of both approaches: RoBERTa learned better from the more complex and more refined PAWS dataset, while GPT gave more relevant responses for simpler and more straightforward QQP data.
- Detected some interesting behavior peculiarities and regularities of the considered GPT model versions, like differences in sensitivity to prompt-data combinations or in reacting to the detalization of the prompt instructions.

Chapter 4

QNLP Experiments and Evaluation

4.1 Problem Statement and Goals

Considering the complexity and black-box nature of the interpretation of gaining the results of the above-mentioned SOTA approaches, the QNLP approach is additionally considered an alternative and not-so-widespread potential solution.

This chapter aims to investigate this approach in order to compare it with the previous ones. In particular, we consider the lambeq toolkit Kartsaklis et al., 2021 as a concrete implementation of the QNLP processing.

4.2 Approach and Tools

There is a tendency to use quite a theoretical mathematical method of the Category theory in different spheres of sciences and engineering. This approach is known as an Applied Category theory.

Category theory is a branch of mathematics that studies the structures and relationships between them in the most abstract way. In other words, it enables the understanding of mathematical structures and their properties in a very general and unified way by representing them in the form of objects (the essential elements of the category) and morphisms or arrows (relationships or mappings) between them. Each category possesses a binary associative operation of composition of morphisms, and for every object, there exists an identity morphism, which is the unit according to the composition operation.

A mapping between categories preserving identity morphisms and composition of morphisms is called a functor.

Recent advances in this area possibly allow extending the list of solutions for the PI task with the ones that are better interpretable and more intuitive. In other words, the Applied Category Theory approach is able to model the structure and semantics of natural language.

The general idea of this approach is to map the text into the text diagrams or, by analogy, with the quantum theory, text circuits. Such an analogy with quantum circuits is supposed to make this approach applicable and highly efficient while running on quantum computers. According to the most recent results in this direction (see Wang-Mascianica, Liu, and Coecke, 2023), the text circuit represents text meaning, and the sentences mapped into similar circuits should have similar meanings. This fact makes it possible to apply this theory to the PI task.

There are several frameworks and toolkits that realize these ideas, providing the structural representation of the text data and allowing quantum natural language processing. In particular, we considered lambeq[1], DisCoPy [2], and DisCoCirc[3].

As mentioned in the official documentation[4], lambeq introduces the string diagrams as an abstraction, allowing the NLP design on quantum hardware. In particular, string diagrams operate in a monoidal category (a category equipped with the associative monoidal or tensor product and an identity object), which perfectly models the computations and processing on a quantum computer.

Despite being a perfect modeling abstraction of the quantum circuits, string diagrams are applicable to any hardware decisions[5].

The lambeq toolkit provides several approaches to NLP processing: classical, hybrid, and purely quantum. Each of them is accomplished with the corresponding models, text converters, and simulations. Since the last one requires quantum computations, only the first two were chosen for the investigation.

The initial idea of applying the QNLP approach to the PI task was motivated by the results obtained inWang-Mascianica, Liu, and Coecke, 2023, stating that texts having similar diagrammatic representations should convey similar meanings. The authors additionally announced the implementation of the latest framework, DisCoCirc, extending the previously released DisCoPy by enabling the creation of the circuit representations not only for separate sentences but also for texts.

Relying on the above facts, the general solution plan had the following steps:

- Use DisCoCirc to convert paraphrase datasets into circuits
- Train and evaluate the hybrid model using lambeq's PennyLaneModel
- Train and evaluate the classical model using lambeq's PytorchModel
- Compare the results with the DL and LLM ones

4.3 Implementation Details

The implementation of the above-mentioned plan appeared not as smooth as expected and was adjusted correspondingly:

- Implementation of the first step showed that the DisCoCirc framework seems to need to be more mature for the proposed approach. It is indeed able to generate the circuits for text, but we were not able to make them compatible with the models. As a result, we switched to using the sentence converters of the previously released DisCoPy framework, which must also be suitable as most of the paraphrase samples from the datasets contain precisely two sentences.
- The lambeq's hybrid approach consists of direct use or customization of the PennyLaneModel class on text circuits. Unfortunately, we did not manage to instantiate the model for running the pipeline due to toolkit errors, which were present in its different release versions.

[1] https://github.com/CQCL/lambeq/tree/main
[2] https://github.com/discopy/discopy
[3] https://github.com/CQCL/text_to_discocirc/tree/main
[4] https://cqcl.github.io/lambeq/string-diagrams.html
[5] https://cqcl.github.io/lambeq/glossary.htm

- For the classical pipeline, we customized and trained the model implementation provided in the lambeq's documentation[6].

The final implementation is provided in the GitHub[7].

4.4 Model Customization

The initial model[8] takes two diagrams as an input and outputs the predicted binary label. Under the hood, it additionally has a simple neural network with two linear layers:

```
nn.Sequential(
        nn.Linear(4, 10),
        nn.ReLU(),
        nn.Linear(10, 1),
        nn.Sigmoid()
        )
```

Due to the inability to instantiate the hybrid model, we changed the base class from PennyLaneModel to PythorchModel.

The initial training of this model on the PAWS and QQP subsets provided an accuracy of 52% and 55%, respectively. These results are significantly lower than those obtained by the model we based on: 88%. As a result, we doubled the number of parameters of the linear layers in the underlying neural network:

```
nn.Sequential(
        nn.Linear(4, 20),
        nn.ReLU(),
        nn.Linear(20, 1),
        nn.Sigmoid()
        )
```

Although our model did not manage to achieve the same performance, additional parameters showed increasing in accuracy.

We assume that lower performance is due to the complexity of the PAWS and QQP datasets samples and possibly because the hybrid model approach is more effective than the classical.

4.5 Sentence Diagrams Generation

Each sentence goes through the following four steps to become an amenable model input (see Fig 4.1):

- **Tokenization** with the SpacyTokenizer, which is, according to the lambeq documentation, based on the NLP package SpaCy[9].

[6] https://cqcl.github.io/lambeq/examples/pennylane.html
[7] https://github.com/irynapast/thoth
[8] https://cqcl.github.io/lambeq/examples/pennylane.html
[9] https://spacy.io

- **Convertion to the string diagram** of the tokenized sentence via the provided parser BobcatParser. On the one hand, string diagrams model the abstraction of quantum computations and data processing. On the other - they represent compositional relations between words in the sentence[10]. Some examples of the parsed sentences are provided in Figs 4.2, 4.3.
- **Rewriting the diagram** to a simplified and proper form using the RemoveSwapsRewriter. In general, the rewriting is a diagram transformation to a simplified and suitable form aimed at reducing resource usage and reducing training time.
- **Mapping the string diagram via ansatz** to the more low-level, but concrete representation, suitable for further model training and experiments[11]. In general, ansatz converts the diagram to the tensor or quantum representation with a specified number of qubits or dimensionality of the wires, respectively. According to our model choice, TensorAnsatz should be used; the resulting diagram representations are presented in Figs 4.4, 4.5.

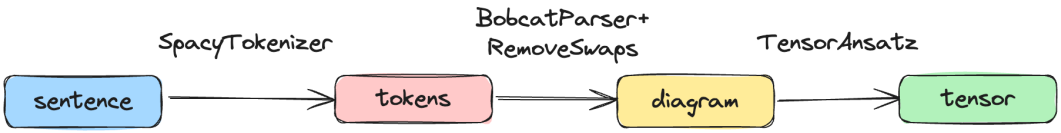


FIGURE 4.1: Sentence-to-tensor transformation

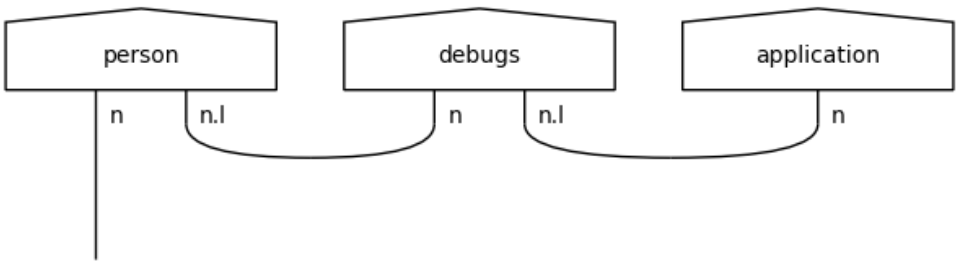


FIGURE 4.2: Sentence diagram with connection types

To form the datasets for experiments, we applied this flow to the subsets of the PAWS and QQP datasets. It is worth mentioning that a part of the samples failed to be converted to diagrams.

Moreover, there is a distinction between failure rates among the datasets. In particular, from the total parsed 8167 and 9367 samples of PAWS and QQP, respectively, nearly 2% of sentence pairs did not succeed in diagram conversion for PAWS, and almost 15% for QQP.

Despite all the diagrams being built with the same conversion pipeline, some of the diagrams obtained incorrect structures and were not suitable for model training,

[10] https://cqcl.github.io/lambeq/string-diagrams
[11] https://cqcl.github.io/lambeq/tutorials/parameterise

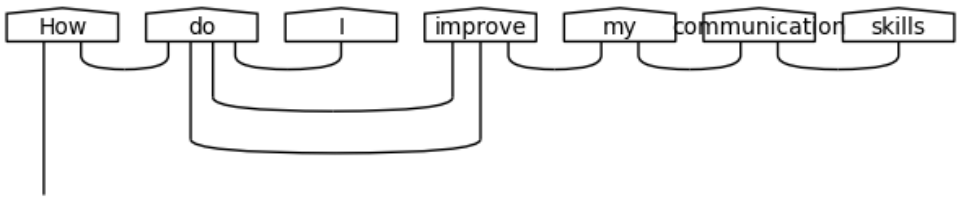


FIGURE 4.3: Sentence diagram

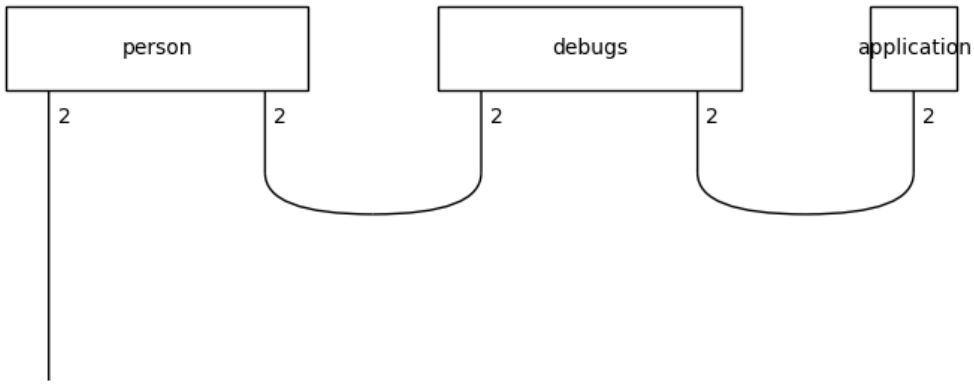


FIGURE 4.4: Tensor diagram with dim=2

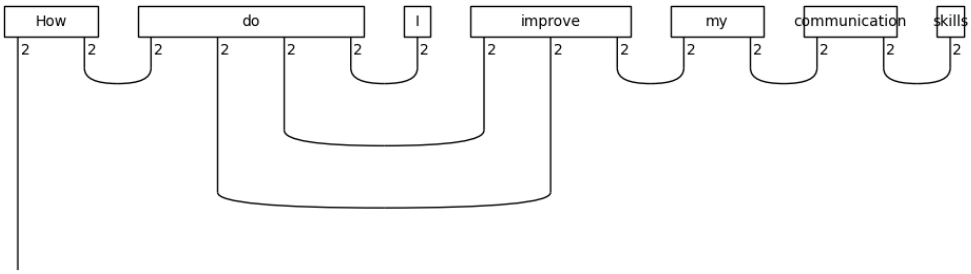


FIGURE 4.5: Tensor diagram with dim=2

which added one more filtration step to the data selection: nearly 1% of data was removed from both datasets.

We assume that diagram failures were caused due to the structural diversity and complexity of the considered datasets.

The statistics for the final datasets are presented in Table 4.1; the datasets can be downloaded by the link[12].

[12] https://drive.google.com/drive/folders/1USbV37zwPJvs3KzeGQmfdNC4gt3qOVUg?usp=sharing

TABLE 4.1: Data for QNLP experiments in numbers

	Train	Validation	Test
PAWS	4955	1494	1487
QQP	4939	1477	1479

4.6 Training and Evaluation

In our experiments, we stuck to quite a similar training strategy with only a few minor parameter changes as for the base model: with Adam optimizer, Binary cross entropy loss function, and early stopping if the validation accuracy did not improve after 20 epochs.

Similar to other experiments, the standard evaluation metrics (accuracy, precision, recall, f_1) were used.

We conducted the experiments on both PAWS and QQP subsets. Unfortunately, the model failed to learn enough from the data to be able to perform well on our data. The main observation is that it is not able to determine the paraphrases correctly on PAWS at all; on QQP, the performance is significantly higher, which is quite an intuitive behavior due to the considerable complexity of PAWS sentences, but it is still too low compared to the previous methods. The corresponding results are presented in Table 4.2.

TABLE 4.2: QNLP experiments results

	Accuracy	Precision	Recall	F1 score
PAWS	54.90 %	1.21 %	30.77 %	2.33 %
QQP	66.3 %	60.0 %	56.0 %	57.9 %

4.7 Conclusions

This chapter was meant to investigate the possibility of solving the PI task with the QNLP approaches. Despite not fully achieving our initial goals and the quality of the obtained solution being lower compared to DL and LLM, the experiments yielded numerous insights and valuable experiences in the categorical NLP.

We assume that the obtained results can be significantly improved by picking up more suitable model architectures. The model we based on provided 88% accuracy, but on a really simple dataset consisting of sentences related to only two topics and having quite a simple and similar structure, while the QQP and PAWS sentences are complex in structure and relate to very different topics. Another possible reason for decreasing the obtained accuracy could be the change in the model type: we managed to use only the classical approach, while the model taken as the base example was a hybrid one.

Besides, applying more advanced data preprocessing and training on bigger datasets can potentially improve the solution. And finally, the purely quantum approach should definitely be tried as a future work.

Chapter 5

Conclusions and Future Work

The thesis investigates the problem of detecting changes in the meaning of the textual data by the precise consideration of the Paraphrase Identification task.

We aimed to approach and compare the solutions with different NLP methods: the widespread ones, such as DL-based techniques and prompting LLMs for the solution, and an alternative one, QNLP, that only enters the active phase of development and is becoming common.

An additional experiment diversity was gained by running them on two different structure and complexity datasets: one containing only sentences that are questions and another 'cultivated' specifically for the PI task.

Investigation of the DL-based approach approved the superiority of RoBERTa over BERT. The experiment results on both datasets reaffirmed its ability to effectively learn and capture complex syntactic structures; cross-evaluation analysis demonstrated the beneficial impact of the PAWS dataset compared to QQP in enhancing the model's generalization capabilities.

Overall, the DL-based approach demonstrated remarkable performance in solving the PI task. It does not seem to need any further investigation unless exploring larger versions such as RoBERTa large or other BERT-like models.

Although both explored GPT models did not manage to approach the RoBERTa's performance level, experimenting with them revealed several non-obvious insights and behavior patterns, like less sensitivity to the prompt complications of the later version of the model or a surprising inverse proportionality between the complexities of the prompt and the model.

It is clear that the LLM's performance is highly dependent on the quality of the prompt, which suggests potential directions for future improvement. In particular, using a few-shot learning prompts, which is known for improving models' performance, is a great candidate for future work. Fine-tuning the GPT models and exploring other LLMs could also be the options.

Experimenting with the QNLP approaches appeared to be the most unexpected and unsuccessful in terms of achieving a satisfactory accuracy level on the one hand but the most challenging and exciting on the other.

The initial investigation plan was adjusted several times due to the inability to compile and run the needed tools. However, we still managed to experiment with the categorical approach in NLP. Although these experiments provided the lowest results compared to the two previous methods, they have the most significant improvement potential. In particular, the first obvious steps to take are picking up more suitable model architectures and applying more advanced data preprocessing and training on the more extensive dataset. Using hybrid and purely quantum models is definitely the very next candidate for future work. Finally, as QNLP evolves, continuous innovations will likely generate new methodologies to apply to the PI task.

In conclusion, the thesis results are essential for understanding the methods of solving the PI task as well as their limitations. In particular, they imply the following:

- With the rapid development of LLMs, the more traditional LM, such as RoBERTa, is still the better choice for this particular task and datasets.
- The QNLP approach (at least in the settings we considered) is not yet ready for practical use and falls short in terms of accuracy compared to LLM and Roberta-based models.

Bibliography

Becker, Jonas et al. (2023). *Paraphrase Detection: Human vs. Machine Content*. arXiv: 2303.13989 [cs.CL].

Chen, Hannah, Yangfeng Ji, and David Evans (July 2020). "Pointwise Paraphrase Appraisal is Potentially Problematic". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. Ed. by Shruti Rijhwani et al. Online: Association for Computational Linguistics, pp. 150–155. DOI: 10.18653/v1/2020.acl-srw.20. URL: https://aclanthology.org/2020.acl-srw.20.

Chen, Qian et al. (July 2017). "Enhanced LSTM for Natural Language Inference". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Regina Barzilay and Min-Yen Kan. Vancouver, Canada: Association for Computational Linguistics, pp. 1657–1668. DOI: 10.18653/v1/P17-1152. URL: https://aclanthology.org/P17-1152.

Coecke, Bob (2021). "The Mathematics of Text Structure". In: *Joachim Lambek: The Interplay of Mathematics, Logic, and Linguistics*. Ed. by Claudia Casadio and Philip J. Scott. Cham: Springer International Publishing, pp. 181–217. ISBN: 978-3-030-66545-6. DOI: 10.1007/978-3-030-66545-6_6. URL: https://doi.org/10.1007/978-3-030-66545-6_6.

Coecke, Bob, Mehrnoosh Sadrzadeh, and Stephen Clark (Mar. 2010). "Mathematical Foundations for a Compositional Distributional Model of Meaning". In: *Lambek Festschrift Linguistic Analysis* 36, pp. 345–384.

Devlin, Jacob et al. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: 1810.04805 [cs.CL].

Dong, Li et al. (Sept. 2017). "Learning to Paraphrase for Question Answering". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Copenhagen, Denmark: Association for Computational Linguistics, pp. 875–886. DOI: 10.18653/v1/D17-1091. URL: https://aclanthology.org/D17-1091.

Foltýnek, Tomáš et al. (July 2020). "Testing of support tools for plagiarism detection". In: *International Journal of Educational Technology in Higher Education* 17.1. ISSN: 2365-9440. DOI: 10.1186/s41239-020-00192-4. URL: http://dx.doi.org/10.1186/s41239-020-00192-4.

Gong, Yichen, Heng Luo, and Jian Zhang (2018). *Natural Language Inference over Interaction Space*. arXiv: 1709.04348 [cs.CL].

Hardy, Hardy and Andreas Vlachos (2018). "Guided Neural Language Generation for Abstractive Summarization using Abstract Meaning Representation". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ed. by Ellen Riloff et al. Brussels, Belgium: Association for Computational Linguistics, pp. 768–773. DOI: 10.18653/v1/D18-1086. URL: https://aclanthology.org/D18-1086.

Kartsaklis, Dimitri et al. (2021). *lambeq: An Efficient High-Level Python Library for Quantum NLP*. arXiv: 2110.04236 [cs.CL].

Liu, Yinhan et al. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. arXiv: 1907.11692 [cs.CL].

Parikh, Ankur et al. (Nov. 2016). "A Decomposable Attention Model for Natural Language Inference". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Ed. by Jian Su, Kevin Duh, and Xavier Carreras. Austin, Texas: Association for Computational Linguistics, pp. 2249–2255. DOI: 10.18653/v1/D16-1244. URL: https://aclanthology.org/D16-1244.

Ribeiro, Marco Tulio et al. (July 2020). "Beyond Accuracy: Behavioral Testing of NLP Models with CheckList". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, pp. 4902–4912. DOI: 10.18653/v1/2020.acl-main.442. URL: https://aclanthology.org/2020.acl-main.442.

Salton, G., A. Wong, and C. S. Yang (1975). "A vector space model for automatic indexing". In: *Commun. ACM* 18.11, 613–620. ISSN: 0001-0782. DOI: 10.1145/361219.361220. URL: https://doi.org/10.1145/361219.361220.

Sun, Yu et al. (2019). *ERNIE: Enhanced Representation through Knowledge Integration*. arXiv: 1904.09223 [cs.CL].

Thompson, Brian and Matt Post (Nov. 2020). "Automatic Machine Translation Evaluation in Many Languages via Zero-Shot Paraphrasing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber et al. Online: Association for Computational Linguistics, pp. 90–121. DOI: 10.18653/v1/2020.emnlp-main.8. URL: https://aclanthology.org/2020.emnlp-main.8.

Touvron, Hugo et al. (2023). *LLaMA: Open and Efficient Foundation Language Models*. arXiv: 2302.13971 [cs.CL].

Wahle, Jan, Bela Gipp, and Terry Ruas (2023). "Paraphrase Types for Generation and Detection". en. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, pp. 12148–12164. DOI: 10.18653/v1/2023.emnlp-main.746. URL: https://aclanthology.org/2023.emnlp-main.746 (visited on 12/18/2023).

Wang, Zhiguo, Wael Hamza, and Radu Florian (2017). *Bilateral Multi-Perspective Matching for Natural Language Sentences*. arXiv: 1702.03814 [cs.AI].

Wang-Mascianica, Vincent, Jonathon Liu, and Bob Coecke (2023). *Distilling Text into Circuits*. arXiv: 2301.10595 [cs.CL].

White, Jules et al. (2023). *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT*. arXiv: 2302.11382 [cs.SE].

Zhang, Yuan, Jason Baldridge, and Luheng He (June 2019). "PAWS: Paraphrase Adversaries from Word Scrambling". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1298–1308. DOI: 10.18653/v1/N19-1131. URL: https://aclanthology.org/N19-1131.

Zhou, Chao, Cheng Qiu, and Daniel E. Acuna (2022). *Paraphrase Identification with Deep Learning: A Review of Datasets and Methods*. arXiv: 2212.06933 [cs.CL].