Faculty of Human Sciences

Bhuvanesh Verma

810483

MSc. Cognitive Systems

An Exploratory Study of Research Design Classification in Scientific Literature

March 2023

Department of Linguistik

Abstract

Selecting an appropriate research design that aligns with the research question is crucial for any study. Identifying different designs used in scientific articles can aid researchers in recognizing the strengths and limitations of various approaches, assisting them in choosing the most suitable design for their research. However, with the overwhelming number of publications each year, it is challenging to keep track of them. Machine learning algorithms can be utilized to classify research designs in scientific articles, enabling researchers to locate existing techniques in their field and identify gaps in research designs. In this study, we investigated different input levels, including abstract, full text, section names, and selected section text, to classify the research design of scientific articles categorized as qualitative, quantitative, and mixed. Our findings revealed that the GNN-based method outperformed traditional machine learning classifiers, and carefully selected section text resulted in better performance.

Acknowledgements

I would like to take this opportunity to express my gratitude to all those who contributed to this project. I would like to thank my supervisors, Prof. Dr. Gerard de Melo and Babajide Owoyele, for their constant guidance, support, and encouragement throughout the project. I would like to extend my gratitude to the whole team at the Chair of Artificial Intelligence and Intelligent Systems Hasso Plattner Institute / University of Potsdam who provided me with access to resources and especially Victor Adelakun Omolaoye and Margarita Bugueño Pérez for their advice whenever needed.

Contents

List of Figures **iv**

List of Tables **v**

1 Introduction **2**

2 Related Work **4**

3 Data **6**

3.1 Data Collection 6

3.2 Data Extraction 6

3.3 Creating Balanced Dataset 7

3.4 Chunking 7

3.5 Research Designs Dataset 8

3.6 Citation Network Dataset (Dataset4) 9

4 Experiments **10**

4.1 Basic Machine Learning Classifiers 11

4.2 Transformer-based Models 11

4.2.1 Zero-Shot Learning 11

4.2.2 Few-Shot Learning 13

4.2.3 Research Design Classifier 14

4.3 Similarity based Models 14

4.3.1 Similar Vocabulary 15

4.3.2 Cosine Similarity 15

4.4 Graph Neural Network Based citation network 15

5 Results **17**

5.1 Basic ML Classifiers 17

5.2 Zero shot Learning 18

5.3 Few shot Learning 19

5.4 Similarity based Models 19

5.5 GNN 20

5.6 Research Design Classifiers 21

5.7 Discussion 21

6 Limitations and Future Scope **24**

7 Conclusion **26**

Bibliography **27**

A **30**

A.1 Scopus Search Query 30

A.2 Full Experiment Results 31

List of Figures

5.1 Comparing different techniques for the classification of research designs 22

5.2 Comparing different input levels for the classification of research designs 23

List of Tables

3.1 Distribution of Research Design Class. 7

4.1 Distribution of Node features. 16

5.1 Evaluation result for Basic Machine Learning Classifiers 17

5.2 Evaluation result for Zero-Shot experiments 18

5.3 Evaluation result for Few shot experiments using SetFit 19

5.4 Evaluation result for Cosine Similarity based experiments 19

5.5 Evaluation result for dictionary-based experiments 20

5.6 Evaluation result for GNN-based experiments with a combination of feature types: **struct+emb** means structural and sentence embedding as a feature, **emb** means only sentence embeddings and **all** means structural, categorical and sentence embedding as features 20

5.7 Evaluation result for fine-tuned GPT-2 models. **2 Labels model** is trained on Quantitative and Qualitative data only and **3 Labels model** includes data for Mixed as well. 21

A.1 Basic Machine Learning Classifier 32

A.2 Zero shot experiments 33

A.3 Few Shot Experiments Results 33

A.4 Cosine-similarity based experiment 34

A.5 Dictionary based experiments 34

A.6 Fine Tuned GPT-2 experiments 34

List of Algorithms

1 Chunk Text 8

2 Zero Shot Classification in auto mode 12

3 Zero Shot Classification in manual mode 13

4 RL for improving hypothesis prompts for NLI-based zero-shot classification . . . 18

Chapter 1

Introduction

The fundamental aim of the research is to either address the research question or test the research hypothesis. To accomplish this, a research design serves as a tailored plan or blueprint that outlines the specific procedures and strategies to answer the research question (Dulock, 1993). Selecting a research approach is an important decision that can be influenced by several factors, including the researcher's philosophical assumptions, research procedures, and specific research methods for data collection, analysis, and interpretation. Other factors that may influence the choice of research approach include the nature of the research problem, the researcher's personal experience, and the study's target population.

There are different types of research designs that researchers can use to answer their research questions or test their research hypotheses, depending on the nature of the research problem and the research objectives. In quantitative research, experimental designs, quasi-experimental designs, and survey designs are common research designs. In contrast, in qualitative research, case studies, ethnographic research, and grounded theory research are common research designs. In addition, researchers may use mixed-methods research designs that combine both quantitative and qualitative research methods to provide a more comprehensive understanding of the research problem. Regardless of which research design is used, it is important to ensure that the design is consistent with the research question and appropriate for the research context.

As important as it is to choose a research design that is consistent with the research question, it is also important to diversify research designs in a given academic field. For example, Zolfagharian et al. (2019) discussed the importance of research designs to achieve methodolog-

ical transparency and diversity in the field of transition studies. The authors emphasize the need to articulate the methods used and to balance qualitative and quantitative research. They also note that the path dependency of a field can lead to a preference for certain methodological options, which can limit the scope of knowledge and expertise.

Machine learning algorithms can be utilized to discern the research design employed in a given scientific article. This can help researchers to find the already existing strategies in their field or across other fields as well. Furthermore, it will allow researchers to find the gaps in the field with regards to the existing research designs. This process of classification not only assists in identifying the intended research methodologies, but also enhances the search capabilities of databases and search engines by introducing a new dimension. Incorporating research design as a criterion for search tasks can potentially optimize the retrieval of information from scientific articles.

Scientific articles have been subjected to classification tasks in the past by Kim and Gil (2019), Ech-Chouyyekh et al. (2019), Rivest et al. (2021), Kandimalla et al. (2021), and Romanov et al. (2016). In this work, we compared several approaches, from basic machine learning classifiers such as SVM and Naive Bayes to advanced transformer-based models for classifying research designs. Some studies also used the different levels of inputs for scientific article classificaiton tasks, for example, Rivest et al. (2021) and Cao et al. (2022). We also used different levels of inputs, from abstract to section titles to the full text. The main focus of this work was to achieve high performance for the classification task, which we consider to be the first of its kind. Previous work on the classification of research designs, such as. Burkett (1990) and Kramer and Boivin (1987), do not correspond to the research designs we discuss in this work. Aside from the classification task, we also wanted to understand whether the abstract provided enough information about the article under study so that it could be used for classification, or whether we needed to add more information by adding certain section or perhaps the full article text.

Chapter 2

Related Work

In the past, scientific articles have been categorized through the use of various machine learning algorithms. Kim and Gil (2019) proposed a classification system that employs representative keywords extracted from article abstracts and identifies topics through Latent Dirichlet Allocation (LDA), subsequently using a K-Means clustering algorithm with TF-IDF values to classify articles into meaningful categories. Other studies have centered on the abstracts of scientific articles. For instance, Ech-Chouyyekh et al. (2019) developed a Convolutional Neural Network (CNN) approach to categorize scientific articles by their domains (7 distinct domains) using their abstracts. Kandimalla et al. (2021) proposed a Deep Attentional Neural Network (DANN) method to classify scholarly papers based on their abstracts, and they discussed the efficacy of the attention mechanism.

Besides classification, scientific articles have also been utilized for information extraction from both abstracts and full texts. Cao et al. (2022) compared topic modeling-based literature analysis using both the full text and abstracts of articles. The comparison between full text and abstracts was also conducted by Martin et al. (2004), Westergaard et al. (2018) and Nazemi et al. (2020). Galke et al. (2017) compared the performance of automatic semantic annotation using either full-text or title-only metadata of documents. Interestingly, the results revealed that using only titles can achieve over 90% quality compared to using full text in three out of four datasets.

In the study by Mishra and Jiang (2021), the author explored the classification of problem-solution patterns within scientific articles, utilizing machine learning classifiers and neural net-

works to identify problem and solution phrases. This research served as an inspiration for the current study to compare basic machine learning classifiers with language models. Similarly, Liu et al. (2013) focused on phrases or sentences within abstracts to extract information on the background, goal, method, and result of scientific articles using Support Vector Machines.

Kolchinsky et al. (2010) used a linear classifier to classify abstracts and introduced a novel Naive Bayes classifier that utilizes features extracted from the citation network of relevant literature for binary classification of full-text documents on protein-protein interactions. On the other hand, the work of Lachaud et al. (2022) focused on using Graph Neural Networks (GNNs) to represent data as graphs and trained GNN models on an academic citation network of computer science papers. They also explored the benefits of transforming the classification task into a multi-level problem.

Recent advancements in language models have greatly simplified the task of solving natural language processing problems. Techniques such as zero-shot and few-shot learning have made the process even more efficient. For example, Schopf et al. (2022) evaluated similarity-based and zero-shot approaches for text classification of unseen classes. Meanwhile, Tunstall et al. (2022) proposed a framework called SetFit (Sentence Transformer Fine-tuning), which is an efficient and prompt-free method for fine-tuning sentence transformers using as few as 8 labeled examples per class, yet still achieving high accuracy.

Chapter 3

Data

In our study, we utilized a dataset introduced in Zolfagharian et al. (2019), where authors presented a framework to evaluate 217 peer-reviewed papers in the area of transition studies. Through their empirical research, they identified three primary research methods commonly used by transition researchers: Qualitative, Quantitative, and Mixed. In the following sections, we will outline our data collection process, explain how we retrieved the necessary information from the collected data, and describe the preprocessing steps we applied. Furthermore, we will discuss the creation of additional datasets derived from the main dataset.

3.1 Data Collection

To collect our dataset, we used the same Scopus query (Appendix-A) employed by Zolfagharian et al. (2019) to collect 217 peer-reviewed papers. Around 75% of these articles were labeled as Qualitative, which could be attributed to the prevalence of qualitative research in transition studies.

3.2 Data Extraction

To extract relevant information from the dataset, we first converted the documents in pdf format into XML format using Grobid [1]. We then used the python library BeautifulSoup [2] to retrieve required information from the XML file, including the abstract, all sections with their

[1] `https://github.com/kermitt2/grobid_client_python`
[2] `https://www.crummy.com/software/BeautifulSoup/bs3/documentation.html`

	Qualitative	Quantitative	Mixed
Initial	179	19	19
Balanced	19	19	19

Table 3.1: Distribution of Research Design Class.

names, DOI, authors, year of publication, and location. After this, we verified the extracted data against the metadata CSV file obtained from Scopus, which contained abstracts and DOIs. Finally, we converted the data into JSON format.

3.3 Creating Balanced Dataset

Table 3.1 illustrates the significant data imbalance in the original dataset. To address this issue, we randomly selected 19 articles from the 'Qualitative' class and created a balanced dataset. While this decision reduced the size of our dataset to just 57 articles, it was necessary to avoid bias towards the 'Qualitative' class as we saw in our initial experiments. Although a smaller dataset presents a challenge for a relatively new task, we hoped that using advanced NLP techniques such as zero-shot and few-shot classification, and leveraging the full text of the articles, would help mitigate this issue. The balanced dataset is provided as JSON format, containing a list of text, raw abstract, title, section names, label, and DOI.

3.4 Chunking

One of the major challenges in utilizing the full potential of scientific articles is processing the full text of the article, which is difficult for deep learning models to contextualize. Chunking is a common technique used to overcome this issue, where large text is divided into smaller chunks of n tokens, which are then separately processed. We followed a similar approach but in two different ways.

1. **Dataset1**: We first preprocess the text data, by applying several common techniques used in NLP, such as removing stop words, lowercasing, lemmatizing, and removing in-text citations. Afterward, we combined all the sections of the document, including the Abstract, into one large string, which was separated by a separator (SBA). The separator (SBA) helps to identify different sections in the text. In case we want to use the full text for model processing, we can create chunks of a certain length from the string, which can

be processed separately by the model.

2. **Dataset2**: Algorithm 1 describes a procedure that divides the large text into chunks of variable length before preprocessing them. This technique is inspired by Schopf et al. (2022), who divided the text based on the paragraph. In our modified approach, we divides a large text into chunks of variable length based on the maximum number of tokens allowed per chunk. Each chunk consists of sentences, preserving the context of the text in each chunk and potentially capturing more information than with fixed-length chunks.

Algorithm 1 Chunk Text

procedure CHUNK_TEXT($text, max\_tokens = 256$)
  Define regular expression exp for splitting text into sentences.
  Tokenize $text$ into sentences using exp.
  $text\_chunks \leftarrow list$, $current\_chunk \leftarrow list$.
  for $i \leftarrow 0, \text{len}(sentences) - 1$ **do**
    Tokenize $current\_sentence$ into individual words using a BERT-based tokenizer.
    if $\text{len}(current\_chunk) + \text{len}(current\_sentence) \leq max\_tokens$ **then**
      Append $current\_sentence$ to $current\_chunk$
    else
      if $\text{len}(current\_chunk) > 0$ **then**
        Append $current\_chunk$ to $text\_chunks$.
      end if
      Set $current\_chunk$ to $current\_sentence$.
    end if
  end for
  return $text\_chunks$
end procedure

3.5 Research Designs Dataset

We also developed a supplementary dataset (**Dataset3**) which aimed at assisting large language models in understanding the context of research design classes. This dataset contains text related to Qualitative, Quantitative, and Mixed research designs, obtained from various sources such as Adebiyi and Abayomi (2016), Creswell and Creswell (2017), Bhat (2023), and Fleetwood (2023). To prepare this dataset, we again utilized Algorithm 1 to partition the text into smaller, more manageable chunks.

3.6 Citation Network Dataset (Dataset4)

In order to construct a citation network, we required a list of references for each scientific article in our dataset. We were able to obtain metadata and reference lists for each article using the CrossRef API[3], utilizing the DOI of each article in our dataset. We also obtained metadata for any articles referenced in the original articles, provided their DOI information was available. By creating a mapping from DOI to respective metadata and a citation network consisting of a mapping from DOI to the list of DOIs of cited articles, we were able to establish the necessary structure for our graph-based experiments. More details on the graph structure are available in the Experiments.

[3] `https://api.crossref.org/`

Chapter 4

Experiments

In our study, we conducted a range of experiments to classify research designs using different levels of inputs and feature representations. These experiments encompassed various NLP techniques, including basic machine learning approaches and zero/few-shot learning based classification. Our experiments were carried out on the available dataset described in last chapter, and we discuss these experiments in detail in this section.

One of the objectives of this exploratory study was to investigate how different levels of input affect the classification results. To accomplish this, we conducted experiments using four levels of input:

1. Abstract: This level uses only the abstract of a scientific article as input.
2. Full Text: This level uses the full text of a scientific article, including the abstract, as input.
3. Section Name: This level uses only the section names of a scientific article, such as Abstract, Introduction, Methodology, etc.
4. Section Text: This level uses text from specific sections of a scientific article, such as Introduction, Methodology, Conclusion, Discussion, Results, Concluding remarks, Method, and Data.

Our hypothesis for selecting specific sections was that not all sections in a scientific article may be relevant for research design classification. Additionally, some sections may introduce noise that can be avoided if we hand-pick relevant sections. To identify such sections, we collected

all the sections in the corpus and selected them based on their frequency and relevancy.

4.1 Basic Machine Learning Classifiers

We conducted experiments using various machine learning classifiers to establish a baseline for the classification task. Specifically, we used Support Vector Machine (SVM), k-Nearest Neighbours (kNN), and Naive Bayes (NB) algorithms. To avoid overfitting, we performed nested cross-validation by splitting the data into 8:2 train-test proportions and tuning the classifier parameters using 5-fold cross-validation on the training set. We repeated this process 5 times to obtain different train-test splits. For text representation in vector format, we used a simple bag of words (bow) and term frequency-inverse document frequency (tf-idf).

It should be noted that for these classifiers, we used Dataset1 without chunking as they do not have any restrictions on the number of words that can be processed at a given time.

4.2 Transformer-based Models

Due to the small size of our dataset, training deep learning models from scratch was not feasible. Instead, we utilized pre-trained large language models (LLMs) to perform the classification tasks.

4.2.1 Zero-Shot Learning

Our initial approach involved using a Natural Language Inference (NLI)-based zero-shot classification pipeline from HuggingFace[1], which leverages NLI to classify unseen data. NLI is an NLP task that determines the relationship between a given hypothesis and a sentence (or premise), indicating whether the hypothesis entails, contradicts, or is neutral to the premise. For zero-shot classification, the text instance is treated as the premise, and the label is treated as the hypothesis. As an example, suppose we have the following text and hypotheses:

Text: "The new iPhone is released today."

hypothesis 1: This is related to **Technology**.

hypothesis 2: This is related to **Politics**.

[1] `https://huggingface.co/tasks/zero-shot-classification`

hypothesis 3: This is related to **Sports**.

In this case, we want to classify the text into one of the three classes: *Technology*, *Politics*, or *Sports*. Using the NLI-based zero-shot classification pipeline from HuggingFace, we can treat the text as the premise and the hypotheses as possible labels. The LLM will then provide an entailment score for each hypothesis, indicating the degree to which the hypothesis is entailed by the text. The hypothesis with the highest entailment score would be chosen as the label for the text.

We used two different methods to apply zero-shot classification on our dataset: automatic and manual. In the automatic method, we kept the default prompt for the hypothesis, which is "This is an example of {label}". The steps for this method are outlined in Algorithm 2.

Algorithm 2 Zero Shot Classification in auto mode

```
procedure ZS_AUTO(zero_shot_nli_model, articles, labels, max_tokens = 256)
    predicted_labels ← list
    for each article in articles do
        Create chunks based on max_tokens
        for each chunk in chunks do
            zero_shot_nli_model(chunk)
            Collect entailment score for each label
        end for
        Aggregate scores for each chunk
        Choose the label with the highest score as the final predicted label for the article
        Append predicted label to predicted_labels
    end for
    Calculate Accuracy or F1 score using labels and predicted_labels
end procedure
```

In manual mode of zero-shot classification, we use more specific and contextualized hypotheses for each label. For instance,

Qualitative Hypothesis: Given text is primarily focused on understanding subjective experiences and social phenomena using interviews, observations, or case studies.

Quantitative Hypothesis: Given text is primarily numerical using statistical methods and relies on measurements and calculations.

Unlike in auto mode, we don't have a mixed label option, and instead, we have designed the algorithm to determine if the text is a mixture of both qualitative and quantitative designs. The algorithm used in manual mode, as presented in Algorithm 3, is similar to the one used in auto

mode with two major changes. Firstly, we decide if a given text is a mixture of both qualitative and quantitative research designs by checking if any of the scores are between 0.33 and 0.67 or if no scores are available. Secondly, we ignore the chunks that result in neutral inferences during the NLI process. This method aims to extract the relevant chunks of information from the whole document for the classification process, as not all chunks of information may be suitable for the classification of research designs.

Algorithm 3 Zero Shot Classification in manual mode

```
procedure ZS_MANUAL(zero_shot_nli_model, articles, labels, max_tokens = 256)
    predicted_labels ← list
    for each article in articles do
        Create chunks based on max_tokens
        for each chunk in chunks do
            for each hypothesis in hypotheses do
                zero_shot_nli_model(chunk, hypothesis)
                if hypothesis is not neutral then
                    Collect entailment score for the label
                end if
            end for
            Collect entailment score for each label
        end for
        Aggregate scores for each chunk and then normalize over both labels
        if any score is between 0.33 and 0.67 or no score is available then ▷ scores are unavailable when all the chunks results as neutral against every hypothesis
            Assign Mixed as predicted label for the article
        else
            Choose the label with the highest score as the predicted label for the article
        end if
        Append predicted label to predicted_labels
    end for
    Calculate Accuracy or F1 score using labels and predicted_labels
end procedure
```

It should be noted that in this experiment, we used Dataset1 and applied chunking to the articles.

4.2.2 Few-Shot Learning

For our experiment, we utilized SetFit, a framework for few-shot fine-tuning of Sentence Transformers that does not require prompts and is highly efficient (Tunstall et al., 2022). SetFit employs a two-step process: first, a Sentence Transformer model is fine-tuned on a small set of labeled examples (usually 8-16 per class), and then a classification head is trained on the embeddings generated from the fine-tuned Sentence Transformer.

We performed 5-fold cross-validation on our training data using SetFit, which involved training the model five times on five different splits of the training and test data with the same split ratio. As the sentence transformer can handle up to 512 tokens as input, we did not split the abstract and section names, but chunked the section text and full-text input.

4.2.3 Research Design Classifier

As discussed in the previous chapter, we created a small dataset (Dataset3) that can help a large language model to learn more context about research design classes. For this purpose we fined-tuned GPT-2 on a classification task for the research design using small chunks of text for each label. It was observed that using smaller chunks resulted in better data availability and therefore improved the accuracy of the classifier.

To fine-tune GPT-2, the data instances were converted into prompts of a specific format. The prompt format consisted of the following parts:

1. The beginning of the text marker "STARTOFTEXT"
2. The input text chunk that pertains to a specific research design.
3. The label for the research design.
4. The end of the text marker "ENDOFTEXT"

Therefore, the final prompt format looked like the following:

STARTOFTEXT SENTENCE: {INPUT CHUNK TEXT} RESEARCH DESIGN: {LABEL} ENDOFTEXT

This allowed the GPT-2 model to fine-tune on the research design classification task with the given input text chunk and corresponding label.

As a result, we conducted two experiments, first, we used fine-tuning GPT model to evaluate a subset of the research design dataset with different settings and second we used the fine-tuning model to evaluate the main dataset.

4.3 Similarity based Models

The technique of similarity-based models is an unsupervised way of classifying scientific articles. Several simple similarity-based techniques such as cosine similarity and similar vocabulary have been tried.

4.3.1 Similar Vocabulary

For the research design dataset (Dataset3), a vocabulary for research design classes was created using the most frequent unigrams, bigrams, and trigrams. Similarly, a vocabulary was created for each scientific article, and the proportion of labels and vocabulary present in the article was determined.

4.3.2 Cosine Similarity

The label text from the Dataset3 was converted into embeddings using SentenceTransformer. Similarly, each scientific article was converted into embeddings. However, since SentenceTransformer can only use up to 512 tokens, chunked embeddings were created for each article using Dataset2. There are two ways to find similarities between the article and the label. First, similarities can be found between each chunk and the label. Based on the weighted aggregate of label score, the article can be classified. Second, the mean of all chunked embeddings can be taken, and meaned embedding can be used to find the most similar label.

4.4 Graph Neural Network Based citation network

The motivation behind using a citation network is that a research article will cite a lot of articles having similar research designs. Therefore, to model the citation network, we utilized the power of a Graph Neural Network (GNN) which can extract the information required to classify an article based on its citation. We constructed a directed graph where each node represented an article, and each directed edge represented the citation relationship between two articles. To extract useful features for modeling, we computed various structural properties of the graph such as in-degree, out-degree, average shortest path length, clustering coefficient, and page rank. Additionally, we also incorporated some categorical features such as publisher name, type of article, and subject (e.g., "Ecology", "Strategy and Management", "Renewable Energy, Sustainability and the Environment") obtained from the article metadata. Furthermore, we used some properties like journal title, article titles, and different input levels as embedding features. These embedding features were used to capture the semantic information of the article, which can be useful in predicting the research design.

Table 4.1 displays the contribution of each type of feature for the final node feature representation. We use SentenceTransformer to obtain embeddings for each property and then take an

Features	Count
Structural	5
Categorical	500
Word Embedding	768
Total	1273

Table 4.1: Distribution of Node features.

average as an embedding feature. In the end, we get a matrix of size $N \times M$, where N represents the number of nodes or articles in the citation network, and $M = 1273$.

After converting the citation network into a directed graph with 1453 nodes and 1788 edges, we prepared a dataset for training. It is important to note that, splitting data in the graph is not as straightforward as in other machine-learning experiments. GNN follows two approaches for splitting data, transductive and inductive. In the transductive approach, the model utilizes information from the whole network but can only see labels for the nodes set as training nodes whereas, in the inductive approach, the graph is split into train and test. The transductive approach is useful when the dataset is small since it uses whole network knowledge to learn from, therefore we use the transductive approach in our experiments. We investigated different types of features and language models which we present in the next chapter.

Chapter 5

Results

5.1 Basic ML Classifiers

In Table 5.1, we present the top results obtained from the three basic machine learning classifiers that we employed. The complete evaluation results with all data types are provided in the Appendix-A. The Naive Bayes classifier with tf-idf features achieved the highest performance among all the basic classifiers. Furthermore, we observed that tf-idf features outperformed bag-of-words features.

Model Type	Feature Type	Data Type	Best F1	Average F1	Average Accuracy
SVM	`tf-idf`	sec-text	0.83	**0.68**	0.72
	`bow`	sec-text	0.81	0.66	0.68
KNN	`tf-idf`	sec-text	0.83	**0.61**	0.65
	`bow`	abstract	0.7	0.51	0.57
NB	`tf-idf`	sec-text	0.84	**0.74**	0.75
	`tf-idf`	sec-name	0.84	0.68	0.68

Table 5.1: Evaluation result for Basic Machine Learning Classifiers

Our approach of selecting relevant section text for classification is effective as it yields the best results across all basic classifiers. Specifically, NB with tf-idf features on section text achieves an average f1 score of 0.74 and the highest f1 score of 0.84. However, there is still room for improvement, as tuning the hyperparameters of NB or SVM models may yield better results than the current best setting.

5.2 Zero shot Learning

The preliminary results of the zero-shot classification experiment indicate that section text yielded the highest f1 score among all the data types, with a score of 0.61. However, this result is lower than the f1 score achieved with the NB model, and no parameter tuning was conducted for this experiment. We experimented with several other NLI models, but $facebook/bart-large-mnli$ provided the most effective results.

Model Name	Experiment Type	Data Type	F1 Score	Accuracy
`xlm-roberta-large-xnli`	auto	full-text	**0.45**	0.46
	manual	abstract	0.25	0.41
`mDeBERTa-v3-base-mnli-xnli`	auto	abstract	0.30	0.40
	manual	sec-text	**0.47**	0.51
`bart-large-mnli`	auto	sec-name	0.31	0.37
	manual	sec-text	**0.61**	0.63

Table 5.2: Evaluation result for Zero-Shot experiments

The results also show a considerable improvement when we manually modify the hypothesis prompt. This finding prompted us to explore the potential of using Reinforcement Learning (RL) in an experiment where we generate prompts with a pre-trained generative model and measure accuracy for each prompt. Accuracy would serve as a reward, and the generative model would aim to generate prompts that improve the model based on this reward. We describe this approach in Algorithm-4.

Algorithm 4 RL for improving hypothesis prompts for NLI-based zero-shot classification

```
procedure RL_GENERATIVE(zero_manual, generative_model)
    accuracy ← 0
    while accuracy < 70 do
        Generate hypothesis for each label using generative_model
        new_accuracy = zero_manual on dataset using generated hypotheses
        if new_accuracy > accuracy then
            reward = 1
        else
            reward = −1
        end if
        Update generative model based on the reward
    end while
end procedure
```

5.3 Few shot Learning

It appears that the results obtained from few-shot training have been an improvement over the zero-shot experiments. It is interesting to note that some of the data splits performed exceptionally well, achieving an f1 score of 0.92. Additionally, section text and section names emerged as the dominant data types that yielded better results when compared to other data types.

Model Type	Data Type	Best F1	Average F1	Average Accuracy
`paraphrase-mpnet-base-v2`	sec-name	0.75	**0.66**	0.67
	sec-text	0.58	0.53	0.55
`all-mpnet-base-v2`	sec-name	0.75	0.61	0.63
	sec-text	0.92	**0.61**	0.63

Table 5.3: Evaluation result for Few shot experiments using SetFit

Other than the variation shown in Table 5.3 we also tune the learning rate ($1e-5$) for the contrastive model, learning rate ($1e-3$) for the classification head, batch size (16), and a number of iterations (15).

5.4 Similarity based Models

Table 5.4 displays the results of the similarity-based experiments. It can be observed that similarity-based on section names as the data type seems to perform same with both embedding types. Therefore, we only present it once in the table and show the second best-performing data type for the averaged embedding type.

Model Name	Embedding Type	Data Type	F1	Accuracy
`paraphrase-mpnet-base-v2`	chunked	sec-name	**0.48**	0.49
	mean	abstract	0.37	0.39
`all-mpnet-base-v2`	chunked	sec-name	**0.44**	0.49
	mean	sec-text	0.34	0.44

Table 5.4: Evaluation result for Cosine Similarity based experiments

The dictionary-based approach yielded similar results to the cosine similarity approach with the highest f1 score of 0.48 obtained using abstract with unigram features. Interestingly, unlike the previous approaches, the impact of section name or section text was not significant.

Model Type	Data Type	F1	Accuracy
unigram	abstract	**0.48**	0.51
	full-text	0.43	0.46
unigram + bigram	abstract	**0.46**	0.49
	sec-name	0.39	0.39

Table 5.5: Evaluation result for dictionary-based experiments

5.5 GNN

In our experiment with different levels of input and different feature types including structural (struc), categorical (cat), and embedding (emb), we present results for the top-performing data types only (Table-5.6). Although both models performed similarly, we found **all-mpnet-base-v2** to be better overall compared to **paraphrase-mpnet-base-v2**, hence we only show results for **all-mpnet-base-v2**. Full results can be found in our project repository mentioned in Appendix-A. The use of embedding as a feature representation of the nodes in the graph played a significant role in the top-performing models.

Data Type	Feature Type	Avg F1	Best F1	Avg Accuracy
abstract	all	**0.80**	0.92	0.80
	emb	0.75	0.83	0.75
full_text	all	**0.78**	0.92	0.78
	struct+emb	0.77	0.83	0.75

Table 5.6: Evaluation result for GNN-based experiments with a combination of feature types: **struct+emb** means structural and sentence embedding as a feature, **emb** means only sentence embeddings and **all** means structural, categorical and sentence embedding as features

We conducted hyperparameter tuning to find the optimal combination of parameters. We used the Weights & Bias platform to track two experiments. The first experiment[1] involved a combination of categorical and sentence embedding features, while the second experiment[2] contained all features combined. With a learning rate of 0.0086, a dropout rate of 0.425, 8142 epochs, and a patience of 1177, we obtained an average f1 score of 0.80 when the features were represented using structural, categorical, and sentence embeddings. Therefore, this model outperformed all the other models presented in this study.

[1] https://wandb.ai/hpi-dc/cat_emb_features_transductive/sweeps/tfg1wmfv
[2] https://wandb.ai/hpi-dc/all_features_transductive/sweeps/xtd8upta

5.6 Research Design Classifiers

We initially trained GPT-2 with a small dataset related to research design, not for the purpose of classification but to provide context for research design in GPT-2. However, we later utilized the trained model to evaluate the Dataset1.

During the fine-tuning process of GPT-2 with the research design dataset, we partitioned the dataset into chunks of 32, resulting in a total of 580 labeled chunks for Quantitative and Qualitative classes. This model achieved an accuracy of 81% and an f1 score of 0.80. Subsequently, we added a Mixed class to the dataset, resulting in a total of 780 labeled chunks with three labels. Fine-tuning on this dataset resulted in 69% accuracy and an f1 score of 0.66 on the test set. We experimented with various chunk sizes, but a chunk size of 32 produced the best results.

Model Type	Data Type	Accuracy	F1
`2 Labels model`	sec-text	0.56	**0.57**
	abstract	0.54	0.55
`3 Labels model`	abstract	**0.51**	0.47
	sec-text	0.38	0.28

Table 5.7: Evaluation result for fine-tuned GPT-2 models. **2 Labels model** is trained on Quantitative and Qualitative data only and **3 Labels model** includes data for Mixed as well.

After selecting the best models from the previous experiments, we evaluated the Dataset1 by dividing it into chunks of 32 tokens and using the same approach described in Algorithm 3, but with a fine-tuned model instead of the NLI model. As shown in Table 5.7, the section text data type was once again found to be useful for research design classification.

5.7 Discussion

Research design classification is a challenging task based on the fact that we have a small dataset to perform experiments on. Nonetheless, we experimented with various techniques to make the best use of the small labeled dataset that we have.

Figure-5.1 compares different techniques for classifying research designs, irrespective of the model type, feature type, or input level type. Both GNN and Set-Fit-based few-shot classification achieved the best F1 score of 0.92 for a particular split. However, GNN had the highest average F1 score of 0.80 overall splits. Zero-shot and similarity-based classification had the

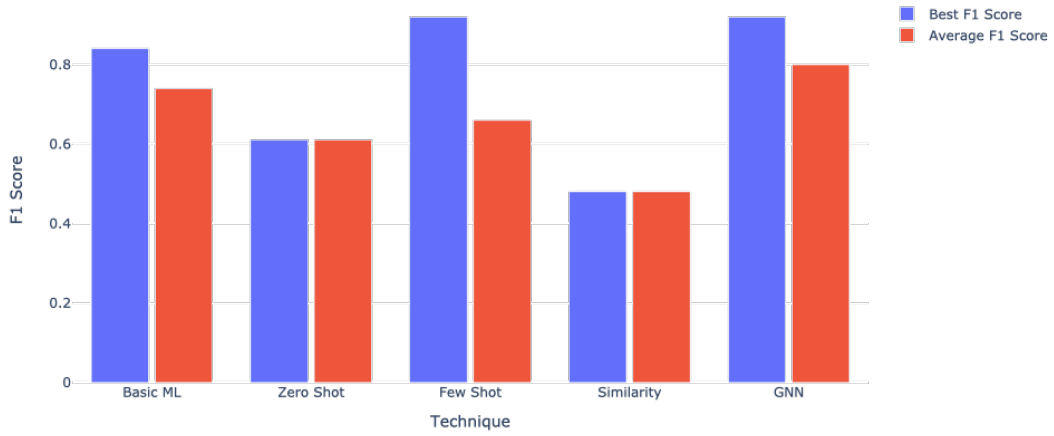

Figure 5.1: Comparing different techniques for the classification of research designs

same value for the best and average F1 scores since these are unsupervised techniques, and we did not perform split-based experiments on them.

We also compared different levels of inputs for classifying research designs, irrespective of the model and feature type. It is evident from Figure-5.2 that GNN outperformed all other techniques on each input level. However, we observed that section text and section name performed surprisingly better in most experiments. Although the best performance was based on the abstract, we have sufficient evidence to demonstrate that other input levels performed better than the abstract.

Experiments conducted using unsupervised techniques such as zero-shot experiments relied more on the large text as the results indicated lower scores for the abstract and section name as compared to section text and full text. Conversely, similarity-based experiments were more inclined towards smaller text inputs.

Although there is no significant variation in the overall scores of the different input levels, the outcomes of the supervised experiments indicate interesting trends. Specifically, we observed that when section-based input outperforms, the scores for abstract and full text are slightly lower, and conversely.

The supervised experiments were able to strike a balance between the amount of text and the relevant information, unlike the unsupervised experiments that were performed at the extremes of input levels. Section Name, although containing less text, provides highly specific information

Figure 5.2: Comparing different input levels for the classification of research designs

that can capture the properties of different research designs, which may be difficult to achieve with other alternatives of less text input such as the Abstract. On the other hand, the Section Text consists of carefully selected sections that we deemed useful for the classification of the research design. While the volume of text is large, the content is highly specific to the research design as compared to the Full Text.

The performance of GNN can be attributed to the fact that the classification of a research article is not only dependent on the input text but also on other features such as those from neighboring articles in the citation network. Even in models that only utilize embeddings as features, the performance is still influenced by the embedding features of neighboring articles. Thus, the combination of these different factors may have contributed to the improved performance of abstract and full-text input levels.

Chapter 6

Limitations and Future Scope

We conducted experiments to determine the most suitable input level for the scientific article classification tasks, but we believe that a different dataset would provide a better basis for comparison. For future work, researchers could follow our approach and use different datasets to compare results. Since we used a small dataset, it would be interesting to see how the experiment results compare with a well-defined, larger dataset.

In our exploration of different input levels for scientific article classification, we did not investigate the potential of combining inputs to achieve improved results. Additionally, while we selected specific sections of articles for research design classification, future research could investigate whether utilizing a subset of these sections can enhance performance. Furthermore, we found that the section text we selected was more useful for classification than the full text in most experiments, sometimes resulting in better overall performance. Therefore, automating the process of selecting relevant sections of an article based on section names could be a fruitful area of research.

Algorithm-4 was proposed as a means to generate more effective prompts and potentially enhance zero-shot results, as we observed in our manual zero-shot experiments. While we did not test the algorithm in our current study, it would be worthwhile to investigate its impact on the results. Future research could also focus on more discrete research designs such as modeling, simulation, conceptual, and experimental, among others as studied by Antons and Breidbach (2018). With models that can identify articles based on research design, we can study specific research designs in various academic fields. For example, how are experimental designs presented

in Physics compared to Computer Science? What new mathematical concepts can be applied in other fields? These questions would be easier to answer if we had access to experimental and conceptual studies from different fields to compare.

Chapter 7

Conclusion

Our study aimed to address the challenging task of classifying research designs in scientific articles using various methods. We began by exploring the effectiveness of existing and newly created datasets, including a citation network, sentence-based dataset, and research design label dataset, in improving classification performance. Initially, we found that a basic machine learning classifier like Naive Bayes with tf-idf features outperformed transformer-based zero-shot and few-shot learning methods. However, after converting the citation network into a Graph Neural Network, we were able to achieve an f1 score of 0.80, surpassing the performance of Naive Bayes. We also experimented with different input levels for scientific article classification and discovered that carefully selected section text can balance the amount of text input from the abstract and full text, leading to improved performance.

Furthermore, we proposed several improvements for future research, including exploring the potential of combining inputs, selecting subsets of article sections, and using our proposed Algorithm-4 to generate better prompts to improve zero-shot results. Additionally, we suggested investigating the classification of discrete research designs, such as modeling, simulation, conceptual, and experimental, to compare them across different fields of study.

In conclusion, our study provides a starting point for further research in identifying research methods used in various fields and how their classification can facilitate information exchange between different fields. Our findings can serve as a basis for future studies on improving research design classification performance and exploring its applications in interdisciplinary studies.

Bibliography

J Adebiyi and T Abayomi. 2016. Research design: A review of features and emerging developments. *European Journal of Business and Management*, 8(11):113–118.

David Antons and Christoph F Breidbach. 2018. Big data, big insights? advancing service innovation and design with machine learning. *Journal of Service Research*, 21(1):17–39.

Adi Bhat. 2023. Qualitative research: Definition, types, methods and examples.

Gary L Burkett. 1990. Classifying basic research designs. *Family medicine*, 22(2):143–148.

Qiang Cao, Xian Cheng, and Shaoyi Liao. 2022. A comparison study of topic modeling based literature analysis by using full texts and abstracts of scientific articles: a case of covid-19 research. *Library Hi Tech*, (ahead-of-print).

John W Creswell and J David Creswell. 2017. *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.

Helen L Dulock. 1993. Research design: Descriptive research. *Journal of Pediatric Oncology Nursing*, 10(4):154–157.

Monir Ech-Chouyyekh, Hicham Omara, and Mohamed Lazaar. 2019. Scientific paper classification using convolutional neural networks. In *Proceedings of the 4th international conference on big data and internet of things*, pages 1–6.

Dan Fleetwood. 2023. Quantitative research: What it is, tips and examples.

Lukas Galke, Florian Mai, Alan Schelten, Dennis Brunsch, and Ansgar Scherp. 2017. Using titles vs. full-text as source for automated semantic document annotation. In *Proceedings of the Knowledge Capture Conference*, pages 1–4.

Bharath Kandimalla, Shaurya Rohatgi, Jian Wu, and C Lee Giles. 2021. Large scale subject

category classification of scholarly papers with deep attentive neural networks. *Frontiers in research metrics and analytics*, 5:600382.

Sang-Woon Kim and Joon-Min Gil. 2019. Research paper classification systems based on tf-idf and lda schemes. *Human-centric Computing and Information Sciences*, 9:1–21.

Artemy Kolchinsky, Alaa Abi-Haidar, Jasleen Kaur, Ahmed Abdeen Hamed, and Luis M Rocha. 2010. Classification of protein-protein interaction full-text documents using text and citation network features. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(3):400–411.

Michael S Kramer and Jean-François Boivin. 1987. Toward an "unconfounded" classification of epidemiologic research design. *Journal of Chronic Diseases*, 40(7):683–688.

Guillaume Lachaud, Patricia Conde-Cespedes, and Maria Trocan. 2022. Graph neural networks-based multilabel classification of citation network. In *Intelligent Information and Database Systems: 14th Asian Conference, ACIIDS 2022, Ho Chi Minh City, Vietnam, November 28–30, 2022, Proceedings, Part II*, pages 128–140. Springer.

Yuanchao Liu, Feng Wu, Ming Liu, and Bingquan Liu. 2013. Abstract sentence classification for scientific papers based on transductive svm. *Computer and Information Science*, 6(4):125.

Eric PG Martin, Eric G Bremer, Marie-Claude Guerin, Catherine DeSesa, and Olivier Jouve. 2004. Analysis of protein/protein interactions through biomedical literature: Text mining of abstracts vs. text mining of full text articles. In *KELSI*, pages 96–108. Springer.

Rohit Bhuvaneshwar Mishra and Hongbing Jiang. 2021. Classification of problem and solution strings in scientific texts: Evaluation of the effectiveness of machine learning classifiers and deep neural networks. *Applied Sciences*, 11(21):9997.

Kawa Nazemi, Maike J Klepsch, Dirk Burkhardt, and Lukas Kaupp. 2020. Comparison of full-text articles and abstracts for visual trend analytics through natural language processing. In *2020 24th International Conference Information Visualisation (IV)*, pages 360–367. IEEE.

Maxime Rivest, Etienne Vignola-Gagné, and Éric Archambault. 2021. level classification of scientific publications: A comparison of deep learning, direct citation and bibliographic coupling. *PloS one*, 16(5):e0251493.

A Yu Romanov, KE Lomotin, ES Kozlova, and AL Kolesnichenko. 2016. Research of neural

networks application efficiency in automatic scientific articles classification according to udc. In *2016 International Siberian Conference on Control and Communications (SIBCON)*, pages 1–5. IEEE.

Tim Schopf, Daniel Braun, and Florian Matthes. 2022. Evaluating unsupervised text classification: zero-shot and similarity-based approaches. *arXiv preprint arXiv:2211.16285*.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts. *arXiv preprint arXiv:2209.11055*.

David Westergaard, Hans-Henrik Stærfeldt, Christian Tønsberg, Lars Juhl Jensen, and Søren Brunak. 2018. A comprehensive and quantitative comparison of text-mining in 15 million full-text articles versus their corresponding abstracts. *PLoS computational biology*, 14(2):e1005962.

Mohammadreza Zolfagharian, Bob Walrave, Rob Raven, and A. Georges L. Romme. 2019. Studying transitions: Past, present, and future. *Research Policy*, 48(9):103788.

Appendix A

A.1 Scopus Search Query

The following query was used to search for the required 217 articles.

((REFTITLE (Regime shifts to sustainability through processes of niche formation: The approach of strategic niche management) AND REFPUBYEAR = 1998) OR (REFTITLE (Technological transitions as evolutionary reconfiguration processes: A multi-level perspective and a case-study) AND REFPUBYEAR = 2002) OR (REFTITLE (On the nature, function and composition of technological systems) AND REFPUBYEAR = 1991) OR (REFTITLE (Understanding carbon lock-in) AND REFPUBYEAR = 2000) OR (REFTITLE (More evolution than revolution: Transition management in public policy) AND REFPUBYEAR = 2001) OR (REFTITLE (From sectoral systems of innovation to socio-technical systems: Insights about dynamics and change from sociology and institutional theory) AND REFPUBYEAR = 2004) OR (REFTITLE (The governance of sustainable socio-technical transitions) AND REFPUBYEAR = 2005) OR (REFTITLE (Typology of sociotechnical transition pathways) AND REFPUBYEAR = 2007) OR (REFTITLE (The diffusion of renewable energy technology: An analytical framework and key issues for research) AND REFPUBYEAR = 2000) OR (REFTITLE (Bricolage versus breakthrough: Distributed and embedded agency in technology entrepreneurship) AND REFPUBYEAR = 2000) OR (REFTITLE (The past and future of constructive technology assessment) AND REFPUBYEAR = 1997) OR (REFTITLE (Functions of innovation systems: A new approach for analysing technological change) AND REFPUBYEAR = 2007) OR (REFTITLE (Transforming the energy sector: the evolution of technological systems in renewable energy technology) AND REFPUBYEAR = 2004) OR (REFTITLE (Strategies for Shifting Technological Systems: The Case of the Automobile System) AND REFPUBYEAR

= 1994) OR (REFTITLE (The politics and policy of energy system transformation - Explaining the German diffusion of renewable energy technology) AND REFPUBYEAR = 2006) OR (REFTITLE (Analyzing the functional dynamics of technological innovation systems: A scheme of analysis) AND REFPUBYEAR = 2008) OR (REFTITLE (Technological innovation systems and the multi-level perspective: towards an integrated framework) AND REFPUBYEAR = 2008) OR (REFTITLE (CAUTION! Transitions ahead: politics, practice and sustainable transition management) AND REFPUBYEAR = 2007) OR (REFTITLE (Processes and patterns in transitions and system innovations: Refining the co-evolutionary multi-level perspective) AND REFPUBYEAR = 2005)) AND (TITLE-ABS-KEY ((sustainab* OR environmental* OR bio* OR renewable OR socio-technical) AND (transition OR transform* OR "system innovation" OR "radical innovation" OR shift OR change))) AND (LIMIT-TO (EXACTSRCTITLE, "Energy Policy") OR LIMIT-TO (EXACTSRCTITLE, "Technological Forecasting And Social Change") OR LIMIT-TO (EXACTSRCTITLE, "Environmental Innovation And Societal Transitions") OR LIMIT-TO (EXACTSRCTITLE, "Research Policy") OR LIMIT-TO (EXACTSRCTITLE, "Technology Analysis And Strategic Management")) AND (LIMIT-TO (PUBYEAR, 2019) OR LIMIT-TO (PUBYEAR, 2018) OR LIMIT-TO (PUBYEAR, 2017) OR LIMIT-TO (PUBYEAR, 2016))

A.2 Full Experiment Results

This section presents tabulated results of all experiments conducted with various configurations except for GNN which contains 56 combinations. Results can be found in the experiments folder of the project repository on Github at `https://github.com/Bhuvanesh-Verma/sciplore`. Dataset and code are also provided for the reproducibility of the project.

model_type	feature_type	data_type	best_f1	avg_f1	best_accuracy	avg_accuracy
svm	tfidf	abstract	0.63	0.48	0.67	0.53
svm	tfidf	full_text	0.65	0.56	0.67	0.62
svm	tfidf	sec-name	0.74	0.57	0.75	0.6
svm	tfidf	sec-text	0.83	0.68	0.83	0.72
svm	bow	abstract	0.6	0.52	0.58	0.55
svm	bow	full_text	0.65	0.63	0.67	0.65
svm	bow	sec-name	0.67	0.48	0.67	0.48
svm	bow	sec-text	0.81	0.66	0.83	0.68
knn	tfidf	abstract	0.56	0.46	0.58	0.48
knn	tfidf	full_text	0.63	0.52	0.67	0.55
knn	tfidf	sec-name	0.56	0.48	0.58	0.52
knn	tfidf	sec-text	0.83	0.61	0.83	0.65
knn	bow	abstract	0.7	0.51	0.75	0.57
knn	bow	full_text	0.67	0.49	0.67	0.53
knn	bow	sec-name	0.4	0.36	0.42	0.38
knn	bow	sec-text	0.62	0.47	0.67	0.52
bayes	tfidf	abstract	0.64	0.46	0.67	0.5
bayes	tfidf	full_text	0.67	0.6	0.67	0.62
bayes	tfidf	sec-name	0.84	0.68	0.83	0.68
bayes	tfidf	sec-text	0.84	0.74	0.83	0.75
bayes	bow	abstract	0.64	0.56	0.67	0.57
bayes	bow	full_text	0.74	0.67	0.75	0.68
bayes	bow	sec-name	0.76	0.64	0.75	0.65
bayes	bow	sec-text	0.81	0.67	0.83	0.7

Table A.1: Basic Machine Learning Classifier

model_name	experiment_type	data_type	f1_score	accuracy
xlm-roberta-large-xnli	auto	abstract	0.27	0.28
	auto	full_text	0.45	0.46
	auto	sec-name	0.23	0.3
	auto	sec-text	0.28	0.33
	manual	abstract	0.25	0.41
	manual	full_text	0.24	0.32
	manual	sec-name	0.13	0.25
	manual	sec-text	0.2	0.29
mDeBERTa-v3-base-mnli-xnli	auto	abstract	0.3	0.4
	auto	full_text	0.17	0.33
	auto	sec-name	0.28	0.33
	auto	sec-text	0.2	0.35
	manual	abstract	0.31	0.38
	manual	full_text	0.21	0.32
	manual	sec-name	0.2	0.27
	manual	sec-text	0.47	0.51
bart-large-mnli	auto	abstract	0.29	0.37
	auto	full_text	0.17	0.33
	auto	sec-name	0.31	0.37
	auto	sec-text	0.2	0.35
	manual	abstract	0.24	0.36
	manual	full_text	0.6	0.62
	manual	sec-name	0.2	0.34
	manual	sec-text	0.61	0.63

Table A.2: Zero shot experiments

model_type	data_type	best_accuracy	avg_accuracy	best_f1	avg_f1
paraphrase-mpnet-base-v2	abstract	0.58	0.52	0.58	0.49
	full_text	0.67	0.53	0.67	0.48
	sec-text	0.58	0.55	0.58	0.53
	sec-name	0.75	0.67	0.75	0.66
all-mpnet-base-v2	abstract	0.83	0.53	0.83	0.52
	full_text	0.67	0.58	0.67	0.53
	sec-text	0.92	0.63	0.92	0.61
	sec-name	0.75	0.63	0.75	0.61

Table A.3: Few Shot Experiments Results

model_name	embed_type	data_type	f1_score	accuracy
paraphrase-mpnet-base-v2	chunked	abstract	0.37	0.39
	chunked	full_text	0.35	0.42
	chunked	sec-name	0.48	0.49
	chunked	sec-text	0.31	0.37
	avg	abstract	0.37	0.39
	avg	full_text	0.26	0.37
	avg	sec-name	0.48	0.49
	avg	sec-text	0.29	0.39
all-mpnet-base-v2	chunked	abstract	0.33	0.42
	chunked	full_text	0.36	0.46
	chunked	sec-name	0.44	0.49
	chunked	sec-text	0.33	0.42
	avg	abstract	0.33	0.42
	avg	full_text	0.32	0.42
	avg	sec-name	0.44	0.49
	avg	sec-text	0.34	0.44

Table A.4: Cosine-similarity based experiment

model_type	data_type	f1_score	accuracy
uni	abstract	0.48	0.51
	full_text	0.43	0.46
	sec-text	0.32	0.35
	sec-name	0.39	0.39
uni-bi	abstract	0.46	0.49
	full_text	0.39	0.42
	sec-text	0.34	0.39
	sec-name	0.39	0.39

Table A.5: Dictionary based experiments

model_type	data_type	best_accuracy	best_f1
2_labels_32	abstract	0.54	0.547
	full_text	0.526	0.528
	sec-text	0.56	0.566
	sec-name	0.508	0.51
3_labels_32	abstract	0.508	0.47
	full_text	0.386	0.278
	sec-text	0.386	0.28
	sec-name	0.368	0.345

Table A.6: Fine Tuned GPT-2 experiments