M.Sc. Data Analytics
Masters Dissertation

Applications of Machine Learning in Understanding and Predicting Academic Paper Retractions

Brian Collins - 2078876

Supervisor: Dr Ramazan Esmeli

November, 2023

School of Computing
University of Portsmouth

UNIVERSITY OF PORTSMOUTH

Contents

List of Figures

List of Tables

Abstract

Given the significant growth in academic publications in recent years, the scale of academic misconduct has grown significantly beyond that which can be identified and rooted out during the peer-review process. This work examines the impact that Machine Learning can have in supporting that process and the efforts of hard-working editors and reviewers globally, focusing on understanding historical patterns in retractions for individual subject areas and attempting to predict the likelihood of a new submission being retracted during its lifetime.

Article embeddings are generated using the Specter embeddings model, which then undergo dimensionality reduction using the OpenTSNE Python package. The resultant two-dimensional TSNEs are then clustered using DBSCAN, and commonalities are used to assess common causes of retraction on a per-cluster basis. Furthermore, the HuggingFace transformers pipeline generates multiple classification models, which are then compared for their efficacy to address the underlying problem.

The report highlights how effectively DBSCAN identifies sub-topics within larger subject categories and outlines how retraction patterns vary between those sub-disciplines. Consequently, the work demonstrates the effectiveness of few-shot learning in developing a binary classification algorithm that could be integrated with editorial offices to reduce editor workloads.

Keywords *Academic Publishing, Publication Ethics, Machine Learning, Retraction, Clustering, Classification, Prediction, Analysis, DBSCAN, Few-Shot Learning*

Acknowledgements My sincerest thanks to my dissertation supervisor, Dr Ramazan Esmeli, whose constant help and guidance were essential in the production of this work. I would also like to express my sincere thanks to the rest of the Wiley Intelligent Solutions team, whose code guidance and review were necessary to obtain the quality of the results displayed.

Chapter 1

Introduction

The discussion of publication ethics has become commonplace in the past decade due to the increase in reported cases of unethical behaviour by researchers, partly due to the transition of research publication from purely print media to one existing primarily in the electronic sphere. The shift in medium represents a double-edged sword for the research community due to the increased difficulty of maintaining ethical standards. On the one hand, instances of academic misconduct are far easier to detect thanks to the ability to use software to search an online database to discover instances of significant text duplication or to make direct comparisons between files to assess author bias (Morris et al., 2013). Conversely, the ability to replicate large portions of text at the touch of a button has made it remarkably easy to reproduce a large portion of work without attribution or to adapt figures to skew the data to lend credence to a different conclusion. Further exacerbating this is the rise of large language models and their impact on academic research (Mindzak & Eaton, 2021). Compounding all of this is the number of articles published worldwide remains in a period of explosive growth. The online database Dimensions, indexing articles from 104,000 journals, noted an increase in output from 4.2 million in 2014 to 6.7 million in 2021 (CAGR 6.67%).

Gilbert & Denison (2003) note that:

"*research misconduct jeopardises the reputations of research groups and institutions, reduces public confidence in the scientific community and can halt the progress of medical knowledge.*"

While the study above was explicitly focused on the field of Radiology, the theme can no doubt be transferred to any field of academia. In a profession where reputation is of the utmost importance, and when a single severe incidence of academic misconduct can end a career, the question of why these academics would risk their reputation and career is, of course, posed. This partly relates to the fact that in many instances, finding and retaining tenure, promotion and future employment is decided primarily on the number of articles published in a predefined set of journals. These journals are selected based on their relative impact, selectivity (rejection rate) and position in crucial department rankings (De Rond Miller, 2005). Barbour (2015) noted that researchers increasingly felt that the need to get their work published in these top journals left them:

"*feeling tempted or under pressure to compromise on research integrity and standards.*"

1.1 Project Aims and Design Objectives

This work examines machine learning's ability to support editorial offices' and journal management decisions around the suitability of submitted articles for publication. It prevents the reputational damage to journals, editors and authors associated with academic article retractions. Furthermore, by extension, this work supports the work of researchers and academics globally in the continued publication of ethically robust research.

Consequently, there are two distinct goals associated with this project:

1. Identify commonalities between existing article retractions with specific attention to how particular subject communities experience large-scale academic misconduct.

2. Develop a binary classifier, using article abstracts as input, to predict the likelihood that the associated article will be retracted at some point in the publication life cycle.

1.2 Project Cost, Limitations, Benefits and Ethics Concerns

To support this project, VSCode will be used as an Integrated Development Environment (IDE) to connect to GitHub Codespaces to provide appropriate computing resources to support the generation of Python code utilised to cluster and classify article retractions. The open-source Hugging Face NLP pipeline will further support the development of an appropriate classifier, and a combination of open-source Python libraries will assist article clustering efforts. A small budget will be required to provision a GPU to the Codespace to ensure that models can be trained within the project timeline, estimated to be around $0.70 per hour of runtime, approved by and paid for by the author's employer.

As with any commercially oriented project, the most significant limitation is time, as many of the engineering aspects of the work will need to be completed during operational hours, given the above-noted financial cost.

When completed, the resultant work will first generate appropriate intelligence for journal editors and managers regarding the common reasons for retraction across included subject communities. The increase in domain understanding will consequently allow editors and peer reviewers to be better informed as to what components of each submission might require closer investigation. Additionally, a binary classifier can also serve as a warning flag, facilitating the early warning to journal editors of articles the model deems potentially problematic. It should be noted that while the model will flag potentially inappropriate submissions, it will not have the power to action any decisions. Ultimately editorial integrity needs to be maintained. Final accept or reject decisions remain the sole domain of the appointed editorial teams and their associated subject matter expertise.

It should be noted that author names and institutional affiliations are present in the dataset, given the component articles' status as either being within the public domain when published open access or licensed by intellectual property holders. However, no other personally identifiable information is used, with only abstracts included in the tokenised training data. Consequently, no ethical issues are associated with his work, which has been ratified by the University of Portsmouth Ethics Screening Tool (ETHICS-10667). All data is utilised in adherence with license conditions or is otherwise part of the public record. All tools utilised in the project are fully open source and require no permission for usage.

1.3 Methodology

For the duration of the engineering portion of the project, the Kanban methodology will be utilised, with component portions of each project section added as tasks to a Kanban board hosted as a GitHub project (Figure 1.1.) The engineering portion of the project is broken down as follows:

- Review existing literature, reviewing language models and their application within domains overlapping or adjacent to the publishing industry.
- Investigate clustering methodologies and libraries to examine their applicability to grouping research papers, referencing approaches covered in the previously examined literature.
- Examine the Hugging Face transformers library and any new tools that might be useful in supporting the development of a binary classifier.
- The development portion of the project will be broken down into three component stages:
    1. Gathering and preparation of data from data warehouses and Application Programming Interfaces (API) before storage in both the GitHub Codespace and the Hugging Face hub.
    2. Cluster the component research articles using positive and negative classes to generate meaningful topic clusters.
    3. Train a binary classifier using the Hugging Face transformer pipeline or other associated methodology.
- With the previous stages complete, there will follow a period of iteration for both the clustering and classification portions of the study. This will ensure that produced analyses contain a story representative of the underlying data that is both digestible and contains a compelling story that can support the business. Refer to Figure 1.2 for a flowchart outlining the approach for the cluster engineering portion of the study.

1.4 Project Structure

What follows in this document will be broken up into five component sections:

- **Project Context and Literature Review** This section will frame the business context that necessitates the work conducted in this project. It will also examine the existing literature considering how similar projects have been conducted with particular regard to applications within the publishing industry or utilising academic research papers as a data source.
- **Data Gathering and Preparation** Cover the steps taken to gather and prepare the data used as inputs to the work's clustering and classification portions and will provide detail on the Python code written to pursue this goal.
- **Experiments and Results** Contains the processed outputs of the clustering and classification, detailing performance metrics and considering the non-metric-based measure of methodological efficacy where metrics cannot accurately represent model suitability.

- **Evaluation** Considers how the results outlined in the previous section address the project goals as previously outlined.
- **Conclusion** Presents concluding comments regarding the management of the project, timelines, coverage and future developmental opportunities for this project or those related to it.

FIGURE 1.1: Kanban management board, implemented as a GitHub project

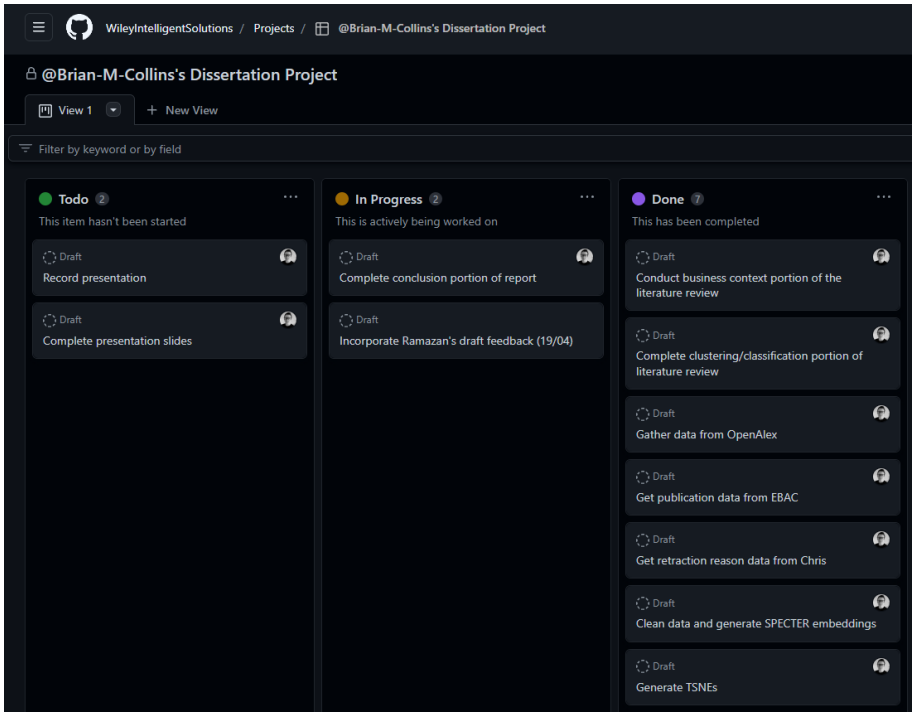
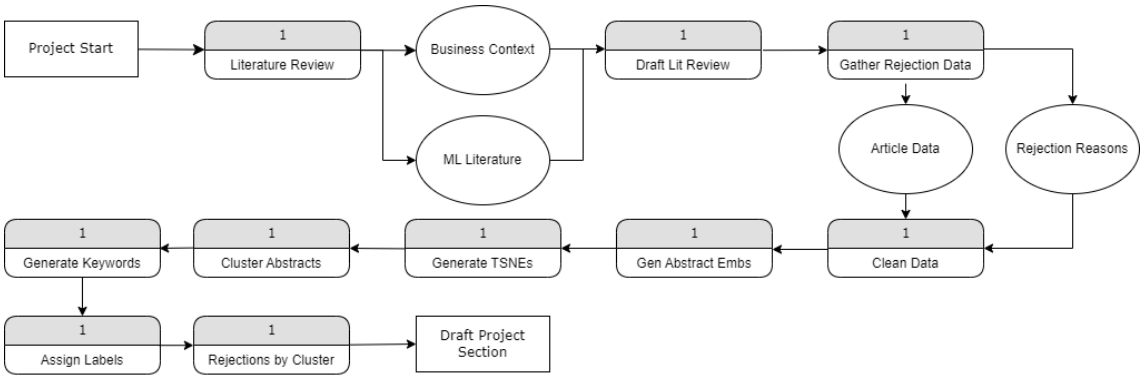


FIGURE 1.2: Clustering project flow chart



Chapter 2

Project Context and Literature Review

This section of the report aims to highlight and examine the factors that have led to the increased focus placed on research and academic integrity over the past decade, examining any contributing factors leading to increased academic misconduct. The research community's response to these ethical issues will also be examined from the viewpoint of multiple stakeholders – from those involved in the publishing industry directly, through journal editors and reviewers, to the contributors. Finally, there will be a discussion of previous works that have aimed to apply machine-learning techniques to domains overlapping or directly adjacent to those covered by this work.

2.1 The Changing Landscape of Publication Ethics

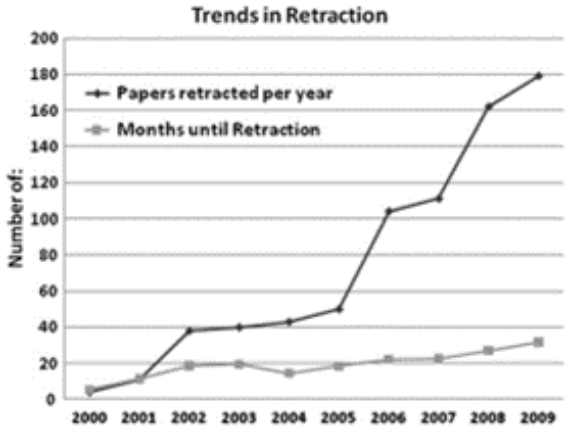
Several studies have been conducted to quantify and analyse the scale of research misconduct. These studies, however, have primarily concentrated on papers indexed in the Medline database, which concentrates on Bio-medicine and the Life Sciences. Steen (2011) analysed the 742 retractions over the period categorising them by eight different reasons (Table 2.1).

Reason for retraction	**Retracted papers, n (%)**
Fraud	**-**
Fabrication	111 (15.0)
Falsification	98 (13.2)
Error	**-**
Scientific mistake	234 (31.5)
Duplicate publication	117 (15.8)
Plagiarism	107 (14.4)
Ethical violations	76 (10.2)
Unstated reasons	61 (8.2)
Journal error	27 (3.6)

Table 2.1: Summary of reasons for retraction of 742 papers, 2000-2010.

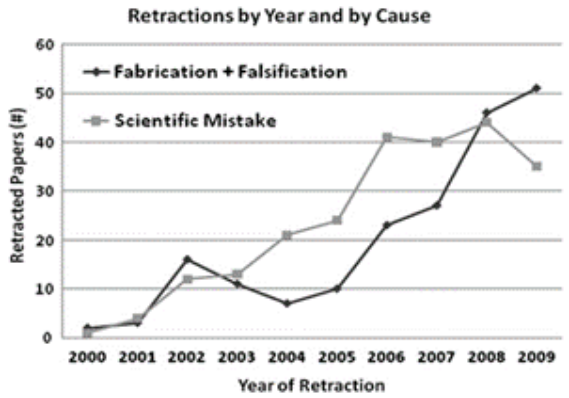
The study also analysed paper retractions over the study period, concluding that the frequency of retractions registered against papers indexed in the Pub Med database had increased significantly over the study period, notably so since 2005 (Figure 2.1). It was also noted that retractions due to fraud had increased seven-fold over the period (Figure 2.2), while the number of papers indexed in PubMed increased by only 35%.

FIGURE 2.1: Trends in paper retractions of PubMed papers 2000-2009. Source: Steen (2011)



An important final point to note regarding this study is that over the timescale, the time from publication to retraction also increased significantly (Figure 2.3). This suggests that while journal editors and publishers remain concerned with resolving ethics cases, the wider academic community remains primarily responsible for raising cases of ethical misconduct.

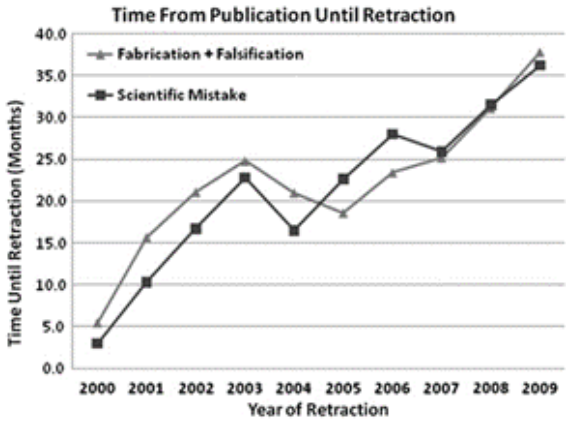
FIGURE 2.2: Trends in paper retractions for instances of fraud and mistake in PubMed papers 2000-2009. Source: Steen (2011)



2.2 Impact of Developing Economies

When considering the explosive growth of article publication, it is essential to consider the role of developing economies in contributing to the number of articles released globally. Figure 2.4 displays a bubble plot comparing publication output and economic status by country, with the size of the bubble denoting the total number of articles published and the colour coding to geographic region.

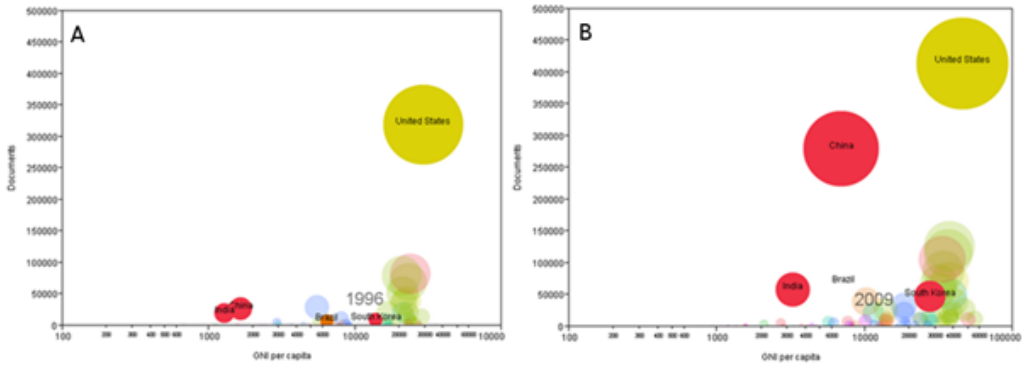
FIGURE 2.3: The average time from publication to retraction in PubMed retracted papers 2000-2009. Source: Steen (2011)



When referring to the 1996 plot (Figure 2.4a), it is clear that the Western world and Japan dominated Science, Technological and Medical (STM) publishing, countries that typically have invested much into Higher Education and Research & Development. However, compared to the 2009 plot (Figure 2.4b), it is apparent that China is rapidly undergoing academic and economic growth. Academic growth is bolstered by the fact that as of 2014, China was spending 2.09% of its burgeoning GDP on R&D (UNESCO, 2015) and is expected to surpass the USA as the world's leading R&D spender by 2019 (OECD, 2014). The same can be said for both India and Brazil, although to a lesser extent in both cases.

In China, however, the academic standing of a particular university or institution is measured by the number of papers published by the institution in journals indexed in the Science Citation Index, Engineering Index and Index to Scientific & Technical Proceedings. The pressure to publish in these prestigious journals leads many institutions to use monetary rewards (Table 2.2) to drive the submission and publication of articles (Jufang & Huiyun, 2011).

FIGURE 2.4: Bubble plot of 1996 (left) and 2009 (right) economic status vs publishing output worldwide (Davis, 2011).



With the growth of emerging economies, it is vital to consider the cultural differences in those societies and how they differ from traditional values. Al-Adawi et al. (2016) pointed out that traditionally, eastern cultures function hierarchically. As such, the head of the hierarchy is shown respect and loyalty. Functionally this leads to heads of departments being included as honorary authors on papers despite having no hand in the generation of

the work; this lies contrary to the generally accepted criteria for the definition of authorship, as seen below.

Journal classification	**Monetary award (2011 GBP Equivalent)**
Nature/Science	£19,252
IF $<$ 1	£192
1 $>=$ IF $<$ 3	£289
3 $>=$ IF $<$ 5	£385
5 $>=$ IF $<$ 10	£481
IF $>$ 10	£1,348
EI journals	£173
ISTP	£58

TABLE 2.2: Monetary reward system in Zhejiang University (Shao & Shen, 2011)

Al-Adawi et al. (2016) proposed that because many journals in developing countries do not have any provision for copyright, which can subsequently lead to authors submitting the article again to other flagship journals, which is considered a form of duplication.

2.3 Review of Clustering Literature

An extensive body of literature already examines the application of standard document clustering techniques, given its ability to provide context to text mining and information retrieval of documents. These techniques have historically provided a method of improving document retrieval systems, suggesting to users other documents that might be of interest based on their viewing history (Zamir et al., 1997). Cutting et al. (1992) highlight that an important consideration with the application of document clustering for document retrieval systems lies in mutually similar documents being relevant to similar search terms, facilitating the suggestion of alternative documents to users based on their search terms. Their study, however, points out that clustering time is typically quadratic to the number of documents. As such, larger corpora have substantial processing time, limiting their contemporary effectiveness given the significant increase in document output, as described in the previous section. However, they illustrate that rather than simply supplementing document retrieval systems, clustering algorithms remain an essential tool for information-gathering processes in their own right. Supporting this viewpoint, Aggarwal et al. (1999) noted that clustering can be utilised to identify natural clusters in document space, using these clusters to produce an accurate document classifier for new documents - an important factor when considered in the context of this study.

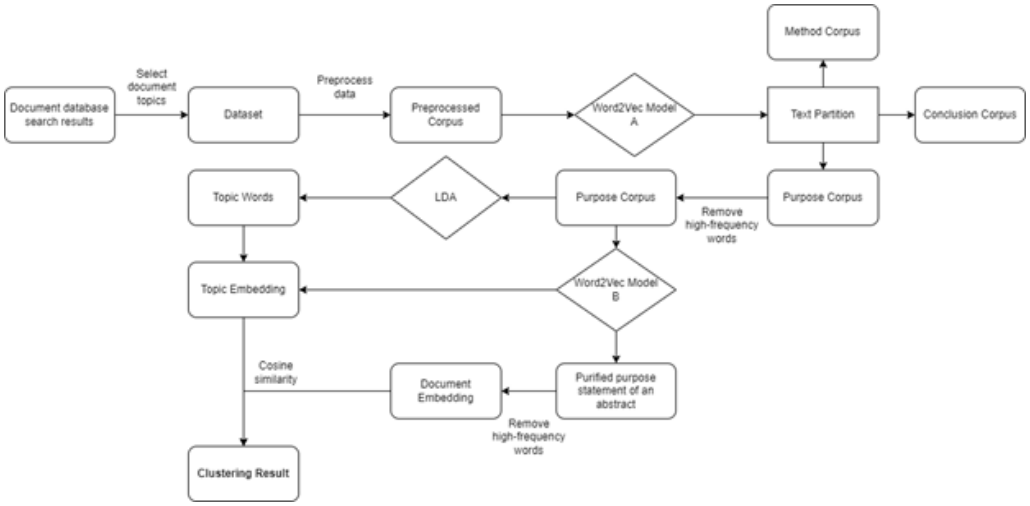
Steinbach et al. (2000) note that Agglomerative Hierarchical Clustering (AHC) and K-means are two approaches historically used for document clustering processes. At the same time, AHC is typically seen as a more effective tool than K-means producing more accurate results, albeit slower in processing time. This study utilised the vector space model, in which similar words are clustered together, meaning that 'Computer' would be proximal to 'Calculator' and distant from 'Country'. It is important to note that this study also weighted terms based on their inverse document frequency, which discounts frequent items, such as conjunctive words, which bring little semantic meaning. A significant limitation of this approach is applying the vector space model, which is useful when mapping single

unrelated words. However, in this application – reviewing academic articles, we know that academic articles hold semantic meaning between words and, importantly, two different sentences can ultimately hold the same meaning.

As referenced previously, commonly accepted clustering techniques begin to struggle when applied to narrow-domain short-form texts such as academic abstracts. While these algorithms could be applied against full-text documents, their typical length of 7,000-15,000 words would either significantly limit the number of documents included in the work or significantly increase the computational requirement and, thus, cost. Article keywords are commonly published alongside papers, often supplied by either article authors or simple language models. Initial consideration suggests that these might present an opportunity to represent papers and act as a seed for the clustering process. However, it is essential to note that given the relatively small number of words representing each document (typically three to five keywords), keyword sparsity would significantly hamper the accuracy of the produced clustering results.

Millar et al. (2009) note the compelling power of Latent Dirichlet allocation (LDA) in document clustering tasks. LDA again sufferers from the requirement of being based on the complete document instead of the isolated abstract, requiring users to again either accept poorer clustering results or longer processing times. However, Li et al. (2018) propose a combination of both LDA and Word2Vec, a framework focused on generating word embeddings to attempt to represent the words within a document within dimensional space. This approach recognises that academic abstracts are regularly split into three distinct portions, a purpose, method and conclusion, and focuses specifically on text extracted programmatically from the purpose section, as this most adequately represents the overall focus of the study and is as such relevant when attempting to cluster academic articles.

FIGURE 2.5: A flowchart representing the complete workflow suggested by Li et al. (2018), adapted from the article.



An initial application of Word2Vec splits the individual abstracts within the corpus into the three described partitions. The purpose corpus then being preprocessed to remove high-frequency words, akin to applying the inverse document frequency method referred to previously (TF-IDF). The processed corpora are then passed to a second Word2Vec model focused on extracting the purified purpose sentence of a document, which is then averaged to produce a document embedding. At the same time, an LDA model extracts topic words from the corpus in order to determine a topic embedding. These topic and

document embeddings have conjoined cosine similarity calculated to produce a clustering result (Figure 2.5).

Sivakumar et al. (2020) note, however, that a limitation of the application of classic embedding techniques like Word2Vec lies in the fact that words will always receive the same embedding regardless of the context of their usage, which presents a problem for document clustering accuracy. For instance, taking the word branch as an example:

- "*I must visit the bank's local branch to deposit a few cheques*"
- "*The birds nest on the path must have fallen from a branch of one of those trees*"

Even with this in mind, pre-trained word embeddings remain the most effective method of extracting information from unlabelled data in Natural Language Processing (NLP) tasks. In response to these limitations, Radford et al. (2018) propose an alternative approach whereby a language model is pre-trained on a large corpus of documents, with the distinct goal of identifying the long-term dependencies in text, i.e. facilitating the understanding of the context of a word's usage, not just identifying individual tokens. This pre-training is primarily completed in an unsupervised fashion. The goal is to find an initialisation point that makes the resultant model effective when generalising to multiple downstream tasks. The second stage of this process is then to conduct supervised fine-tuning using labelled data for specific objectives.

An example of this approach is the Bidirectional Encoder Representation from Transformers (BERT) model developed by the Google AI team (Devlin et al., 2019). BERT was pre-trained, focusing on two unsupervised tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). NSP was established as necessary due to the likelihood of downstream fine-tuning to include tasks such as Question Answering and Natural Language Inference, where understanding the relationship between two discrete sentences is critical. The task was achieved by passing two sentences to the model trainer. In 50% of training cases, sentences that appeared together in the training data were passed to the model trainer, with sentences chosen randomly from across the corpora for the remaining 50%.

Traditionally language models are trained either left-to-right or right-to-left, given that training bi-directionally will allow a word to effectively see itself, meaning that subsequent word prediction would be challenging to generalise outside the training set. MLM circumvents this by masking a random selection of 15% of all input tokens to model training. It should be noted that due to the mask token not being present during fine-tuning, the ith token is replaced with the mask 80% of the time, with a random alternative token 10% of the time, and remains unchanged the remaining 10%.

Beltagy et al. (2019) have taken this work further by fine-tuning the BERT-base model on a random sample of 1.14 million scientific papers from the Semantic Scholar Index, producing SciBERT. Each item within the corpus has an average length of 154 sentences (2,769 tokens), producing approximately 3.17 billion tokens (similar to the corpus that served as the basis for the BERT base). Training, as with BERT, focused on several tasks representative of the scientific community. The fine-tuned model outperformed BERT-Base significantly when applied to NLP tasks within the scientific domain:

1. **Named Entity Recognition**: Identifying critical information in the text and assigning it to one of any predefined categories.
2. **PICO Extraction**: Similar to NER but focusing on describing the sections of a typical clinical trial report (Participants, Interventions, Comparisons, Outcomes).

3. **Text Classification**: Similar to NER, focusing instead on the document as a whole, placing it into predefined categories.

4. **Relation Classification**: Predicts the type of relationship that exists between entities.

5. **Dependency Parsing**: Assessing words in a sentence to analyse the grammatical structure.

However, an essential consideration with BERT-centric language models is that as previously mentioned, they consider the context of words within a document, disregarding any inter-document communication. Since this project aims to discover the links between retracted documents, an approach that considers these links is vital. Cohan et al. (2020) introduced SPECTER, which is based on the SciBERT LM but, importantly, was trained with the specific task of recognising document inter-relatedness using article citations as a primary method of connection. The model was trained to consider those documents that cited one another as closely represented, while a lack of citations indicated a distant representation.

The output of the embedding functions of the SPECTER LM described above is represented as an array of vectors in 768 dimensions, practical as inputs to other fine-tuning tasks to pre-existing models but not particularly helpful when considering visualising data or applying clustering techniques. Therefore, before passing data to clustering methodologies, applying dimensionality reduction to the array to produce coordinates representable in two- or three-dimensional space is necessary. Maaten & Hinton (2008) introduce a variation of Stochastic Neighbor Embedding, which facilitates data dimensionality reduction, t-Distributed Stochastic Neighbor Embedding (t-SNE). t-SNE aims to produce an output that reflects the underlying relationships in the input data so that if two points are proximal in high-dimensional space, they will remain close in low-dimensional space and vice versa. It should be noted that in the event that t-SNE produces poor overall results, an alternative would be to utalise Uniform Manifold Approximation and Projection (UMAP) to achieve the similar dimensionality reduction results.

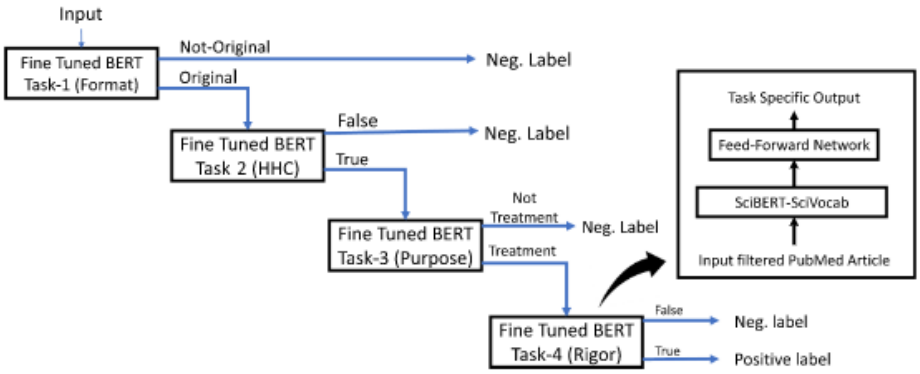
2.4 Review of Classification Literature

Classification algorithms and their application to text documents have consistently displayed their effectiveness in classifying online documents such as news articles (Masand et al., 1992) and web pages (Rongbo Du et al., 2003). However, studies examining these technologies' applications in classifying academic papers are far more sparse.

Caragea et al. (2011) published an early paper examining the process of article classification using abstracts alone as a data source. The work examined the effect of reducing feature complexity given its tendency to cause poor generalisation performance by the overfitting that tends to arise due to the high dimensionality commonly seen in 'bag of words' applications. This study presented a process that established the hierarchy of words within a given abstract, scoring those words based on multiple information criteria to identify a good point in the hierarchy to cut words, reducing the number of tokens passed to the clustering process and lowering complexity. This approach indicated improved performance compared to the bag of words, top-ranked features (selected by mutual information with the class identifier), and topic words established by the application of Latent Dirichlet allocation.

Considering the application of clustering to the scientific domain, Zhou et al. (2015) examined the application of classification algorithms on academic papers sourced from the online preprint archive: arXiv. Their work focused on programmatically identifying computer science papers from within the study set, assessing the viability of different models (Multinomial Naïve Bayes – MNB, and Logistic Regression - LR) and how model efficacy was affected by differing preprocessing techniques: stop word removal combined with stemming, stop word only, stemming only and original text. Findings reported that stop word removal improved F1 scores, with stemming resulting in no significant change when scores across ten cross-validated passes were averaged. Notably, MNB outperformed LR with an F1 score of 0.95.

FIGURE 2.6: Cascade learner model architecture (Ambalavanan & Devarakonda, 2020)



Revisiting the previously discussed BERT LM (Ambalavanan & Devarakonda, 2020) reviewed the applicability of using the technology in a multi-criteria classification problem by screening biomedical papers against four distinct criteria (Table 2.3). The study trained an Individual Task Learner (ITL), Cascade Learner (Figure 2.6), ensemble-Boolean and Feed-forward network ensemble. In the case o f all approaches bar the ITL, each consisted of four BERT models fine-tuned on the specific tasks outlined, with the ITL instead comprising a single integrated model of all criteria. Results indicated that the ITL displayed consistently high recall. In contrast, the Cascade Learner consistently displayed high precision and f-measure when averaging results from each 10-fold cross-validation methodology. This differentiation in model performance indicated a requirement to consider the application of the model when selecting the appropriate approach.

Task	**Description**
Format	assessment of papers' originality and type (article, review, case report, misc)
HHC	Whether the article relates to Human Health Care
Purpose	Whether the article focuses on aetiology, prognosis, diagnosis, treatment (or prevention), costs, economics, disease-related prediction, qualitative study, or something else
Rigour	assessment of experiment design criteria (i.e. whether a clinical trial utilised treatment vs control methodology)

TABLE 2.3: Criteria for acceptance, adapted from Ambalavanan & Devarakonda (2020)

2.5 Clustering Algorithms

This section introduces and evaluates the clustering methodologies utilised in this study.

DBSCAN: Whilst approaches like K-Means attempt to find compact and well-separated clusters within a dataset (Steinley, 2006), DBSCAN looks for areas of high point density in other less dense space. The algorithm initialises with two parameters: 1. an epsilon value - if two points have a distance less than the epsilon value, they are considered neighbours and 2. a minimum cluster size, the mininum number of points in a group before it can be considered a cluster. These values classify each point as either core, border or noise. Core points are defined by having more points within its epsilon distance than the minimum cluster size. A minimum number of points per cluster must scale with the overall size of the underlying dataset. Border points have fewer points within their epsilon distance but are within range of a core point and noise, an observation that does not fall into the previous categories.

An important consideration with this approach is that a given point may be considered part of two or more clusters depending on the density of data points as presented. However, from a user's point of view, allocating each point to a single cluster is enforced by traditionally allocating these bridging points to the first cluster that 'discovers' the observation (Schubert et al., 2017). Secondly, it is essential to consider the impact of the unclustered results and how that applies to the domain of study. In the case of the application of assessment of scientific papers, these unclustered points may represent multi-disciplinary papers with no clear distinction, or they may also represent emerging fields of research without the critical mass of papers to be identified, given the algorithm's initialisation parameters.

Typically clustering performance can be evaluated using metrics such as the Silhouette index, which assesses the average similarity of objects within an identified cluster, their cohesion and how well individual clusters are separated from one another (Dudek, 2020). However, a significant limitation of this approach is that the Silhouette score does not consider the noise (unassigned points) typically seen in density-based clustering approaches. With this in mind, a combination of cluster counts, the proportion of points clustered and the silhouette score will need to be utilised to select appropriate hyperparameters.

2.6 Binary Classification

As discussed previously, the applicability of language models to this study is essential. However, for reasons of time and cost, training a new language model from scratch would fall significantly outside the bounds of feasibility for this study. As such, a transformer architecture can fine-tune a pre-trained language model, retaining the model body (weights) and training a new head on new data. For this purpose, the Hugging Face Transformers API (Wolf et al., 2019) was utilised. The API can handle all processes of fine-tuning a new model, from loading a pre-existing model checkpoint to handling and tokenising new datasets and training the model. Notably, after model validation, the new model can be uploaded to the Hugging Face hub using the Transformers API for easy use in downstream tasks.

Hugging Face's AutoModelForSequenceClassification enables the selection of a base model checkpoint, loading its weights and associated tokeniser. The latter is used to transform the input strings into an array expected by the trainer. Parameters of the training loop are then defined, such as the number of epochs and the selected optimiser, with model training taking place using the trainer class, facilitating evaluation as each

training epoch passes.

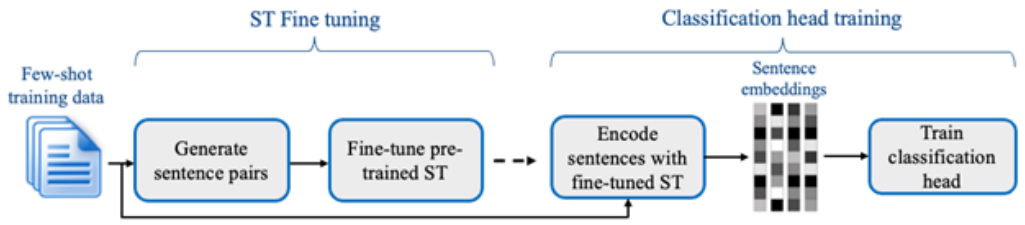
2.7 Limitations of Traditional Approaches

One of the most significant limitations of traditional classifier approaches is that even model fine-tuning requires a considerable amount of labelled data with required observations ranging into the hundreds of thousands to produce a model that can be generalised to new data (Parnami & Lee, 2022). Failing to meet these criteria may not provide the diversity and amount of training data required to facilitate gradient-based model optimisation. As such, determining how to train a model based on a smaller number of training examples becomes vital.

2.8 Few-Shot Learning (FSL)

Few-shot learning has started to gain momentum as a response to the above limitations (Parnami & Lee, 2022), with Hugging Face releasing the SetFit framework based on the few-shot fine-tuning of sentence transformers (Eun Seo Jo et al., 2022). This approach initially capitalises on the available positive samples by generating vector embeddings of positive and negative sample document pairs before training a classification head on the generated dense embeddings for those pairs, utilising the known class labels (Figure 2.7).

FIGURE 2.7: SetFit training architecture



2.9 Evaluating Binary Classifier Model Efficacy

While accuracy can serve as a defacto model evaluation method, problems associated with underlying data make it a problematic measure when viewed in isolation. Therefore, to determine the effectiveness of produced models, literature (Ling et al., 2003; Parker, 2013; Provost & Fawcett, 2001) point instead towards using F1-measure, precision, recall and ROC Area Under the Curve as appropriate evaluation metrics. However, it is also essential to consider the downstream application of any produced model, with journal managers and editors being the most likely end-users. Suppose a model is tuned with recall in mind, ensuring appropriate coverage of True Positive predictions at the cost of an increase in False Positives. In that case, end users may likely develop a degree of scepticism of the effectiveness of model predictions and turn to ignore any resultant warnings due to the increase in editorial workload resulting from moving False Positives through the editorial process. Finally, consideration should be paid to the impact of increasing machine learning scepticism regarding expanding but arguably still unproven technologies (Shah et al., 2019). As such, training with precision in mind, considering how many predicted retractions were from retracted papers, would likely result in an end product with flags taken seriously by downstream editorial teams.

2.10 Conclusion

This section has outlined the literature relevant to the study and the technologies and methodologies utilised in the report. Specifically, DBSCAN is the most suitable clustering technique, given its ability to produce arbitrarily shaped clusters without requiring user-defined cluster counts. Secondly, the Hugging Face API will serve as the basis for the classification portion of the study, given its ability to facilitate the rapid prototyping of new models, fine-tuned on the pre-existing large language models.

Chapter 3

Data Gathering and Preparation

This section of the report aims to explore the various data sources that served as a basis for the study and outlines the numerous stages of preprocessing required to facilitate these data's use as inputs to both clustering and classification model development.

3.1 Overview of Data Sources and Acquisition

This study relies upon several data sources to support the analytical work with separate sources required to identify retracted papers, gather publication details of those retracted papers, and outline reasons for retraction where that information existed. The primary sources for each dataset were found in the OpenAlex index (Priem et al., 2022) for retractions, the Web of Science index for article publication data and the Retraction Watch Database (The Center for Scientific Integrity, 2018) for article retraction data.

OpenAlex, as the name suggests, is a fully open-access/open-source scientific knowledge graph which a group of researchers established to replace the discontinued Microsoft Academic Graph (Sinha et al., 2015) after its discontinuation in 2022. It provides data on 209 million works, with c. 50,000 added daily. These works can be queried using the databases concepts keys which contain an array of article meta-data, including the key 'is_retracted', which is critical for the application of this study. This database can be queried using the OpenAlex Application Programming Interface (API), or the entire dataset can be downloaded as a snapshot. For this study, a pre-existing snapshot of the database existed within a GraphQL database which could be sent paginated queries to return a list of Digital Object Identifiers (DOI). Note the first query, including the complete syntax, which was then shortened in all subsequent queries to include just the 'scrollId' provided. The code extract used to perform this search can be seen in Appendix B.1.

The Web of Science (WoS) presents a paywalled citation index of 85.9 million academic papers sourced from 21,000 peer-reviewed journals with index coverage ranging from 1900 to the current day. Importantly, however, compared to OpenAlex, journals have to surpass strict editorial criteria to be admitted to the index, with data provided to Clarivate Analytics, the owner of the WoS, directly by journal publishers. Consequently, data sourced from the WoS is complete, accurate and consistent; for this reason, it was selected as the primary data source for this study. However, it should also be noted that because of the editorial criteria for inclusion in the index, many DOIs sourced previously would not have associated records in the WoS. While data can be accessed directly from the WoS online portal, with an appropriate publisher login, the online search would be impractical for this study. However, Clarivate does provide weekly data exports to its publishing partner, which in the author's organisation is stored in a cloud-based data warehouse, Snowflake, which can

be queried using SQL. DOIs were chunked to avoid query length caps into batches of n = 1000; queries were sent to Snowflake using the Python package SQLAlchemy, with batched results concatenated into a single Pandas DataFrame. Importantly, it should be noted that Web of Science subject categories could serve as an alternative to clustering when considering tying specific articles to responsible journal teams. This smaller resolution may not enable the distinction of separate clusters within the same subject category, such as Internal Medicine vs Medical Sciences – General.

Article abstracts were saved in a separate table with multiple rows per abstract, representing paragraphs in the original document. As such, all relevant rows needed to be extracted from the table, with the multiple paragraphs needing to be concatenated together to result in a single string per document (Appendix B.2). Notably, not every retracted DOI has a related WoS entry and not every article with a WoS entry had abstract text available. Of the original 22,449 retracted DOIs extracted from GraphQL, 6,113 had associated entries within WoS. In order to gather the 'negative' class, article counts were generated for the positive class, grouped by their respective WoS subject categories. Subsequently, a random selection of articles, based on the generated proportions per subject category, were then downloaded using the same syntax as in Appendix B.3.

Retraction Watch is a blog that covers the retractions of scientific papers, developed to try and provide insight into the nature of academic misconduct and fraud within scholarly publications. For this study, data relevant to the 6,113 identified retracted papers were provided as an extract from the database for which the author's organisation holds a license. It is provided as an online search tool with complete snapshots of the database made available to licensees in CSV form, which can be ingested into existing databases.

3.2 Text Preprocessing

For the classification portion of the study, preprocessing input data reduces the impact of unnecessary data on model training performance and evaluation metrics (Kadhim, 2018). Consequently, four preprocessing steps were applied to the training data to reduce its complexity: punctuation and stopword removal, lowering and Lemmatisation. Python's native String library was used for punctuation removal by applying a list comprehension over the individual input strings. Applying this process in a lambda function (a simple anonymous Python function) facilitated input text lowered to remove letter capitalisation (Appendix B.4).

Stop Word removal, and Lemmatisation was achieved using the Natural Language Toolkit (NLTK) package. Stop words were imported as a list of words from a specified input language with input strings passed through a comprehension with all words not included in the stop word list allowed to pass to the output. Similarly, Lemmatisation was achieved by importing a word net, serving as a dictionary with all input words passed through the NLTK net to match keys to appropriate lemmatised values. In both cases, the preprocessing functions were applied individually to abstracts in the DataFrame through a lambda function (Appendix B.5).

3.3 Vectorising Paper Content

For the clustering portion of the study, Scientific Paper Embeddings using Citationinformed TransformERs (SPECTER) embeddings needed to be generated from the concatenated article titles and abstracts for both the positive and negative classes. Firstly, the data was prepared by applying a Python function to ensure all data was present with columns

named correctly and had unnecessary white space stripped from the title and abstract columns (Appendix B.6).

The resultant DataFrame produced by this function was then used as the input to a second function which loads a tokeniser and then passes chunks of the data frame set to whatever batch size is used as an argument for the function load (Appendix B.7). Importantly, this operation could be completed on a Central Processing Unit (CPU). However, given the scale of operations completed, using a Graphical Processing Unit (GPU) significantly decreased processing times from over 3 hours on a CPU to less than 1 hour on a GPU. However, completion on a GPU also necessitated selecting an appropriate batch size parameter, with four used in this case. Smaller chunk sizes lead to long computation times, while large chunks have higher memory requirements, potentially greater than that available by the provided GPU.

While the output of this work could serve as input to the binary classification portion of the study, Hugging Face has its tokeniser pipeline, utilising the vocabulary of the base model. This tokeniser is instantiated within a function and then mapped to each abstract string of the dataset (Appendix B.8). Note the requirement for Hugging Face data ingests to be in the Datasets format, which could be converted from a Pandas DataFrame. Furthermore, it should be noted that a 70%/30% train test split was selected, given the underlying dataset size, and was stratified on the label class to ensure that the same proportion of positive and negative classes was present in both the training and test dataset.

3.4 Dimensionality Reduction

t-SNE dimensionality reduction (Maaten & Hinton, 2008) was achieved through the OpenTSNE package. Inputs to the process were article DOI and the pre-generated SPECTER embeddings, with the latter reshaped by having individual arrays stacked vertically using Numpy's vstack function. Affinities between data points were calculated with an initialisation object and established to generate initial coordinates for embeddings. Embeddings were then optimised through an early exaggeration phase before optimisation, allowing neighbouring articles to find their neighbours before converging into a stable state in the second regular optimisation phase. Default values were utilised for all tuning parameters except for the perplexity value, which was increased to 500 based on the documentation recommendations, given the size of the training dataset. A full excerpt of the Python code producing the t-SNE coordinates can be seen in Appendix B.1.

3.5 Conclusion

This section has presented the data sources, and preparation required to gather data suitable for training the machine learning models utilised in this report and many downstream tasks.

Chapter 4

Experiments and Results

This section will cover the experimental approaches used in both portions of the project, detailing how results were produced and business intelligence extracted from the underlying dataset.

4.1 Clustering

Given the t-SNE coordinates generated, as outlined in the previous section, it is first necessary to plot these coordinates into two dimensions to ensure the integrity of the previous output (Figure 4.1). This plot displays all publications in the study regardless of retraction status and serves as the basis of the following clustering work. The absence of x and y labels on these plots relates to the fact that the coordinates for each point are ultimately arbitrary. Only the relationship of the points to one another carries any valuable intelligence.

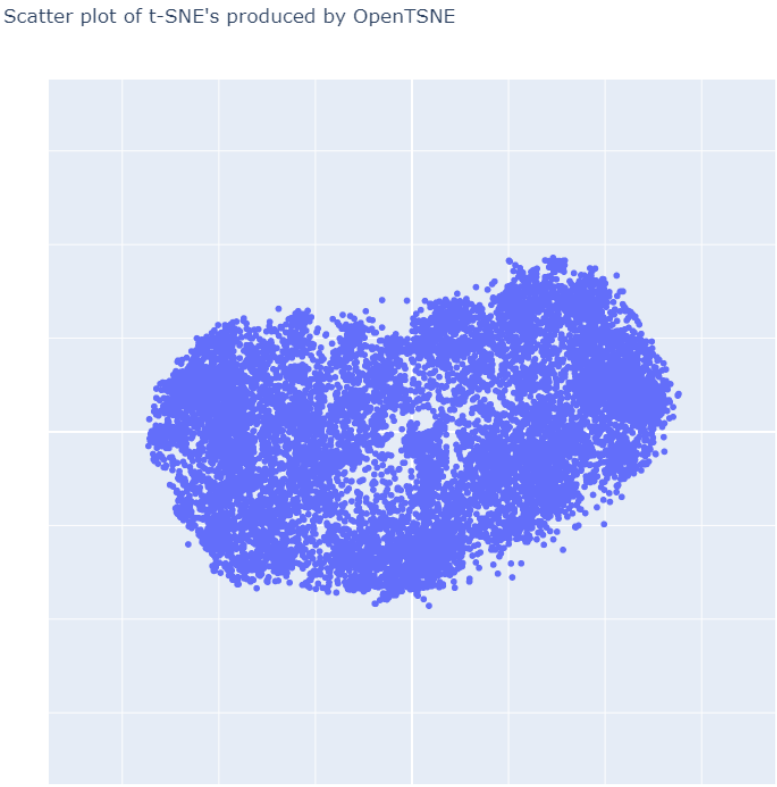
With the coherence of the data points established, optimal hyperparameters for DBSCAN clustering were established via a grid search. As such, clustering was performed on the dataset using a range of minimum distances (0.1, 0.2, 0.3, 1.5) and minimum cluster sizes ranging from 50 to 1500 in increments of 50 articles. Based on the results displayed in Appendix A.5, a minimum cluster size of 300 articles and 0.3 minimum distance value was selected. Details of the clustering performance of the selected hyperparameters can be seen in Table 4.1.

Evaluation Metric	**Score**
Minimum Cluster Size	300
Minimum Distance (eps.)	0.3
Number of Identified Clusters	12
Pct of Article Pool Clustered	71.76%
Silhouette Score	0.0914257

Table 4.1: Clustering performance of selected hyperparameters

With clustering data applied to each observation within the underlying dataset, a second plot was produced in line with the approach seen in Figure 4.1, adding colouring to data points to represent identified clusters (Figure 4.2). Note that the transparency added

FIGURE 4.1: Scatter plot displaying unclustered t-SNEs



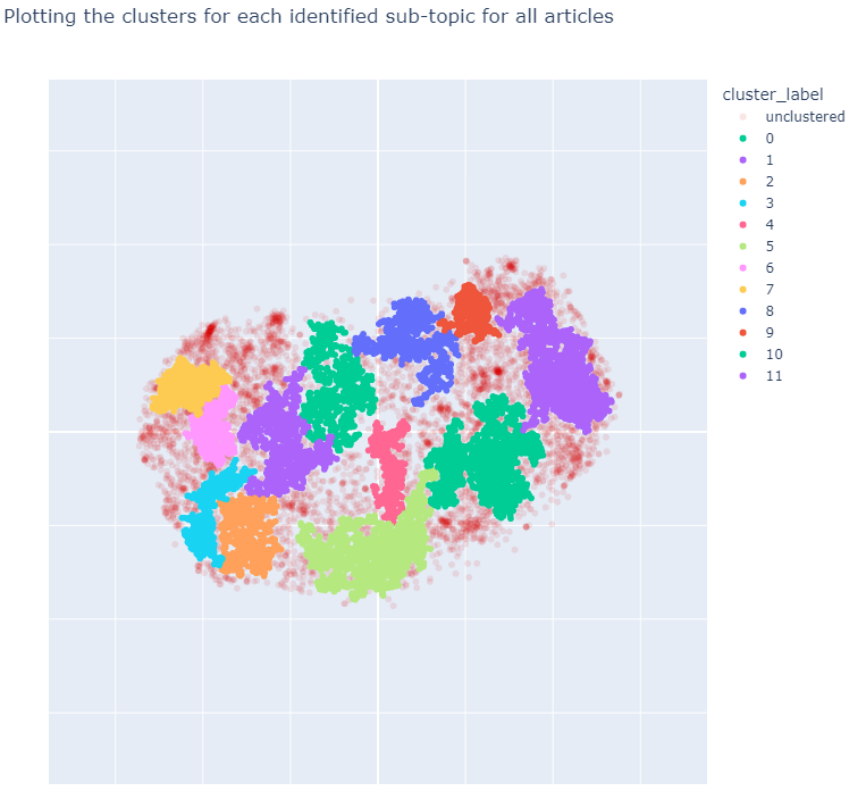
to unclustered points clarifies the borders of the identified subclusters and can serve as confirmation of identified clusters displaying the expected degree of coherence.

In order to clarify the content of each cluster article, keywords were then produced using KeyBERT (Sharma & Li, 2019), focusing on extracting n-grams with a length of between one and three from each clustered abstract (Appendix B.9). These keywords could then be assessed for frequency per cluster to determine the most used and, by extension, use this as a basis for assigning topic labels to the identified clusters. Table 4.2 displays the top 5 most frequent keywords for clusters 0-4. A complete list of the most frequent keywords by cluster can be seen in Appendix Table A.1.

Further to the keywords, exemplar articles were selected from the clustering results. These exemplars are returned as a list by the HDBSCAN clustering object and represent data points at the heart of each cluster and, by extension, can be considered the most representative of that cluster. Notably, a list of exemplar papers is returned due to the asymmetrical nature of DBSCAN clusters, and as such, a single exemplar could not be considered representative. A link to the complete list of exemplar articles can be seen in Appendix A.1.

KeyBERT keywords were used to determine preliminary topic labels for each cluster based on the frequency of keywords within those clusters. However, due to the large number of papers in each cluster and the potential for variation in underlying articles, some care had to be taken to ensure that large multi-disciplinary clusters were appropriately represented. Article exemplars were used to verify the preliminary labels, making adjustments where

FIGURE 4.2: Scatter plot of the identified subclusters



required to produce the final list seen in Table 4.3. It should be noted that subject matter experts have not verified the labels and thus should not be considered truly representative.

As an alternative labelling approach, KeyBERT keywords produced for each article were concatenated into a single comma-separated string, grouped by cluster label. These were then used as input to a second KeyBERT model looking for Ngrams with a max length of one. The top three resultant keywords by the score were then selected and concatenated into a single underscore-separated string, seen in Table 4.4.

4.2 Bibliometric Analysis of Retracted Articles

With the labels established, it was possible to utilise article counts and citations to perform a bibliometric analysis of each cluster, highlighting whether the number of retractions had grown or decreased over the study period and identifying citation patterns per cluster (Figure 4.3). Computer science displayed a significant amount of retraction count growth over the study period and is joined by Public Health and Materials Science (small scale), with both seeing a growth in the number of retractions; all other categories saw retraction count decline. Notably, a random selection of the same count of published papers attracted an average of 5 citations per paper, with retracted papers tripling that figure with 16 citations per paper.

Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
essential oils	microbial fuel	fuzzy lyapunov method	bifurcations occur delay	osteoarthritis oa
essential oil	microbial fuel cells	fuzzy lyapunov	hopf bifurcations stability	knee osteoarthritis
silver nanoparticles agnps	microbial fuel cell	adaptive neuro fuzzy	pipe heat exchanger	hydroxyapatite
green fluorescent protein	aerobic microbial degradation	particle swarm optimization	periodic solutions	evaluate bone implant
drosophila genome	microwave assisted extraction	neuro fuzzy inference	fractional differential equations	ankle arthroplasty

Table 4.2: Five most frequent KeyBERT keywords for clusters 0 - 4

Cluster	Preliminary Label	Cluster	Preliminary Label
0	Biological Sciences	**6**	Materials Science - Large Scale
1	Environmental Sciences	**7**	Materials Science - Small Scale
2	Computer Science	**8**	Microbiology
3	Applied Mathematics	**9**	Internal Medicine
4	Medical Sciences - General	**10**	Biochemistry
5	Public Health	**11**	Oncology

Table 4.3: Preliminary cluster labels

Cluster	KeyBERT Label
0	triazolyltetrazoles_oxazolidinones_triazoles
1	taxonomic_biorefinery_ecology
2	svm_classifiers_features
3	magnetohydrodynamic_plasmas_gases
4	dentomaxillofacial_fibromyalgia_graft
5	perceptions_psychometric_psychometrically
6	concretes_concrete_nanocomposites
7	nanocatalysts_nanosheets_nanomaterials
8	resveratrol_galectin_galectins
9	pharmacologic_hypoperfusion_indications
10	apoptosis_antiapoptosis_microglia
11	microrna185_micrornas_microrna

TABLE 4.4: AI-generated topic labels using KeyBERT

FIGURE 4.3: Scatter plot of article output and citational activity of the identified clusters

4.3 Reasons for Article Retractions

Reasons for article retraction were provided as a parquet file with a single string containing multiple reasons separated by a comma. Therefore, it would be possible to determine the most frequent reason for retraction by individual cluster by expanding this list into distinct values and then counting frequency as with keywords. Refer to Table 4.5 for the complete output, note that column percentages do not add up to 100% because they reflect only the top 10 most common reasons per identified cluster.

4.4 Building a Sequence Classifier

As outlined in the previous chapter, model checkpoints and tokenisers could be loaded from Hugging Face's hub, facilitating quick preparation of the underlying text strings for model training; these models were then transferred to a GPU to speed up training times. A trainer object was created with arguments defining the evaluation strategy and the desire to train all models for ten epochs. The training runs with an average run time of 5 minutes and 30 seconds per epoch across model bases. Finally, a metric computation function was set to extract each epoch's accuracy, precision, recall and f1-score. Model

evaluation metrics can be seen in Table 4.6.

TABLE 4.5: Reasons for retraction per identified cluster

Biological Sciences	Environmental Sciences	Computer Science	Applied Mathematics	Medical Sciences - General	Public Health	Materials Science - Large Scale	Materials Science - Small Scale	Microbiology	Internal Medicine	Biochemistry	Oncology
Unreliable Results (6.64%)	Duplication of Article (7.83%)	**Fake Peer Review (14.49%)**	**Plagiarism of Article (12.41%)**	Duplication of Article (6.62%)	Error in Analyses (5.91%)	**Duplication of Article (16.11%)**	Unreliable Results (7.72%)	Concerns/Issues About Data (8.58%)	**Investigation by Company/Institution (11.09%)**	Duplication of Image (9.35%)	**Duplication of Image (10.53%)**
Fake Peer Review (5.09%)	Plagiarism of Article (6.81%)	Investigation by Journal/Publisher (11.05%)	**Duplication of Article (11.26%)**	Investigation by Journal/Publisher (6.36%)	Investigation by Company/Institution (5.82%)	Investigation by Journal/Publisher (7.58%)	Duplication of Image (7.39%)	Investigation by Company/Institution (7.34%)	Misconduct by author (8.91%)	Unreliable Results (7.38%)	Concerns/Issues About Data (9.17%)
Investigation by Journal/Publisher (4.74%)	Investigation by Journal/Publisher (4.49%)	Concerns/Issues about Referencing/Attributions (8.32%)	Fake Peer Review (6.9%)	Investigation by Company/Institution (5.34%)	Falsification/Fabrication of Data (4.74%)	Duplication of Image (6.16%)	Duplication of Article (6.73%)	Duplication of Image (4.74%)	Misconduct - Official Investigation/Finding (8.48%)	Concerns/Issues About Data (6.96%)	Investigation by Third Party (8.75%)
Duplication of Image (4.5%)	Error in Analyses (4.49%)	Duplication of Article (8.32%)	Error in Results and/or Conclusions (4.83%)	Duplication of Image (5.09%)	Plagiarism of Article (4.66%)	Plagiarism of Article (5.53%)	Concerns/Issues About Data (6.24%)	Misconduct by author (4.63%)	Lack of IRB/IACUC Approval (8.04%)	Investigation by Third Party (5.93%)	Investigation by Journal/Publisher (8.57%)
Error in Data (4.27%)	Error in Results and/or Conclusions (4.06%)	Plagiarism of Article (8.18%)	Concerns/Issues About Authorship (4.83%)	Plagiarism of Article (5.09%)	Duplication of Article (4.48%)	Date of Retraction/Other Unknown (4.58%)	Investigation by Journal/Publisher (3.61%)	Unreliable Results (4.29%)	Investigation by Third Party (7.83%)	Investigation by Company/Institution (5.56%)	Unreliable Results (7.9%)
Concerns/Issues About Data (4.27%)	Unreliable Results (3.91%)	Investigation by Company/Institution (5.45%)	False/Forged Authorship (4.37%)	Lack of IRB/IACUC Approval (4.58%)	Fake Peer Review (4.3%)	Fake Peer Review (4.27%)	Plagiarism of Article (3.61%)	Misconduct - Official Investigation/Finding (3.84%)	Investigation by Journal/Publisher (5.22%)	Investigation by Journal/Publisher (4.57%)	Concerns/Issues About Image (3.84%)
Error in Results and/or Conclusions (3.67%)	Fake Peer Review (3.91%)	Euphemisms for Plagiarism (3.87%)	Error in Analyses (4.14%)	Concerns/Issues About Data (4.33%)	Error in Data (4.12%)	False/Forged Authorship (3.95%)	Error in Results and/or Conclusions (3.28%)	Error in Data (3.5%)	Duplication of Article (4.13%)	Manipulation of Images (4.33%)	Fake Peer Review (3.39%)
Results Not Reproducible (3.55%)	Error in Data (3.48%)	Misconduct by author (3.44%)	Euphemisms for Plagiarism (3.45%)	Unreliable Data (4.33%)	Misconduct - Official Investigation/Finding (4.12%)	Unreliable Results (3.48%)	Unreliable Data (2.96%)	Results Not Reproducible (3.39%)	Concerns/Issues About Data (3.91%)	Concerns/Issues About Image (3.63%)	Investigation by Company/Institution (3.15%)
Error in Analyses (3.44%)	Euphemisms for Plagiarism (3.48%)	Concerns/Issues About Authorship (2.87%)	Concerns/Issues about Referencing/Attributions (3.22%)	Misconduct by author (4.07%)	Investigation by Journal/Publisher (4.03%)	Complaints about author (2.84%)	Fake Peer Review (2.96%)	Manipulation of Images (2.82%)	Ethical Violations by Author (3.7%)	Misconduct by author (3.34%)	Original Data not Provided (3.12%)
Manipulation of Images (3.32%)	Concerns/Issues About Data (3.33%)	Misconduct by Third Party (2.58%)	Concerns/Issues About Results (2.76%)	Plagiarism of Text (3.82%)	Error in Results and/or Conclusions (4.03%)	Concerns/Issues About Data (2.84%)	Error in Analyses (2.79%)	Concerns/Issues About Results (2.71%)	Error in Data (3.7%)	Original Data not Provided (3.13%)	Author Unresponsive (3.12%)

Base Model	Accuracy	Precision	Recall	f1-Score	TP	TN	FP	FN	AUC
specter	0.6858	0.7098	0.6106	0.6565	1,120	1,438	458	714	0.68
specter2	0.6863	0.7024	0.6281	0.6632	1,152	1,408	488	682	0.69
multicite-multilabel-scibert	0.6775	0.6986	0.6052	0.6485	1,110	1,417	479	724	0.68
scibert_scivocab_uncased	0.6649	0.6657	0.6396	0.6524	1,173	1,307	589	661	0.66

TABLE 4.6: Evaluation Metrics for the Binary Classifiers

To better understand the component predictions and how they are compared to observation labels, a confidence matrix was produced using Seaborn, a Python visualisation library acting as an API for Matplotlib (Figure 4.4). Across the board, the fine-tuned classifiers displayed high specificity, with an average of 503.5 (26.56%) predictions being False Positive and a further 1392.5 (73.44%) True Negative. However, the models consistently displayed low sensitivity with 1138.75 (62.09%) True Positive and 695.25 (37.91%) False Negatives.

FIGURE 4.4: Confidence matrix for predictions made by fine-tuned scibert_scivocab_uncased model

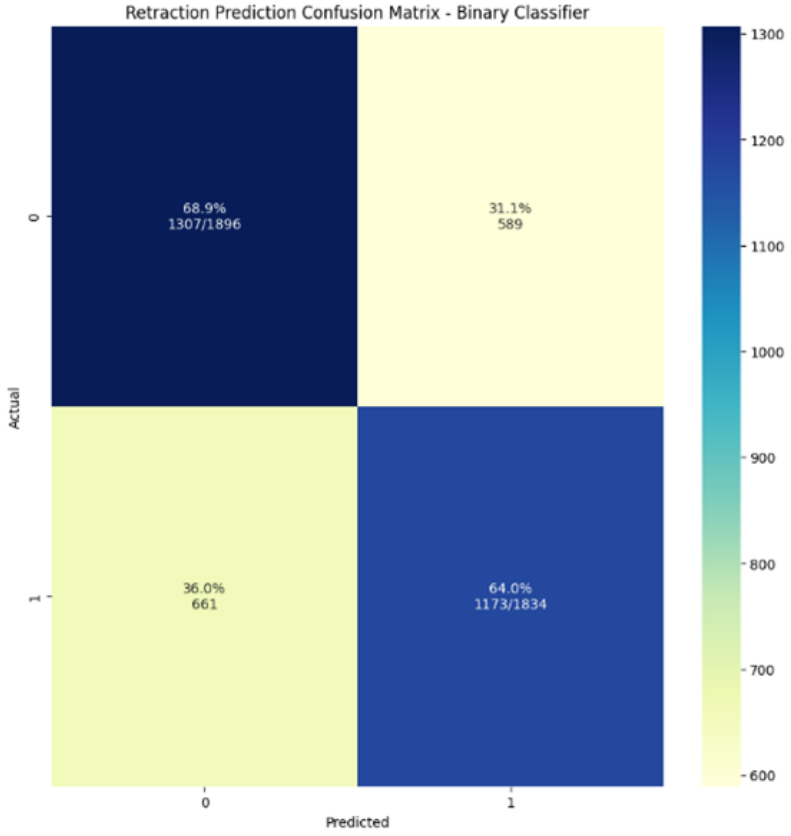
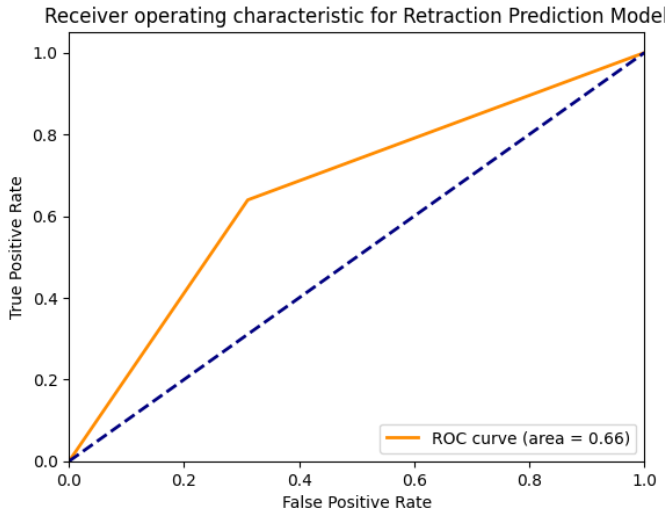


Figure 4.5 displays the Receiving Operator Characteristic for the scibert_scivocab_uncased fine-tuned model, illustrating the typically low predictive power with an Area Under the Curve of 0.66.

4.5 Trialling Few-Shot Learning with HF SetFit

Given the ability to load previously generated and cleaned datasets onto the Hugging Face hub, it was possible to load the data prepared for this report using the HF load_data

FIGURE 4.5: ROC curve for scibert_scivocab_uncased fine-tuned binary classifier



function, tokenising based on the new model's vocabulary. Given its ability to effectively fine-tune for many downstream tasks, the paraphrasempnetbasev2 base and paraphrase-albertsmallv2 base models were utilised. As was the case for the previous models, a trainer was established, utilising Cosine Similarity loss to measure model performance over multiple iterations, with 20 text pairs generated for each example of the positive class. Given the substantial number of generative pairs (348,040 training examples), the models were trained for two epochs, taking an average of 4 hours and 11 minutes to train for the paraphrase-mpnetbasev model and 1 hour 18 minutes for the paraphrasealbertsmallv2 base. Table 4.7 displays the evaluation metrics for the resultant Few Shot models.

Base Model	Accuracy	Precision	Recall	f1-Score	TP	TN	FP	FN	AUC
paraphrase-mpnet-base-v	0.8903	0.8663	0.9188	0.8918	1685	1636	260	149	0.89
paraphrase-albert-small-v2	0.6263	0.6131	0.6505	0.6312	1193	1143	753	641	0.63

TABLE 4.7: Evaluation metrics for SetFit classifiers

Figure 4.6 presents the confidence matrix for the SetFit model predictions. In the case of the paraphrase-albert-small-v2 base model, performance was comparable to the previous binary classifiers. However, the paraphrase-mpnet-base-v performed substantially better, again displaying high specificity, with only 260 (13.7%) predictions being False Positive and 1,636 (86.3%) True Negative. This time, however, the model's sensitivity rose significantly, with 1,685 (91.9%) True Positive and 149 (8.1%) False Negatives.

Figure 4.7 displays the Receiving Operator Characteristic for the fine-tuned paraphrase-mpnet-base-v model, significantly improving the sequence classification model. Given the model's performance, it is possible to maintain a significant True Positive Rate without significantly increasing the frequency of false positives. The model also achieved a significant Area Under the Curve of 0.89, significantly improving over the sequence classification model.

4.6 Conclusion

This section has presented the primary evaluation metrics and results associated with the engineering portion of the project and is necessary for the evaluation of model efficacy.

FIGURE 4.6: Confidence matrix for Setfit model

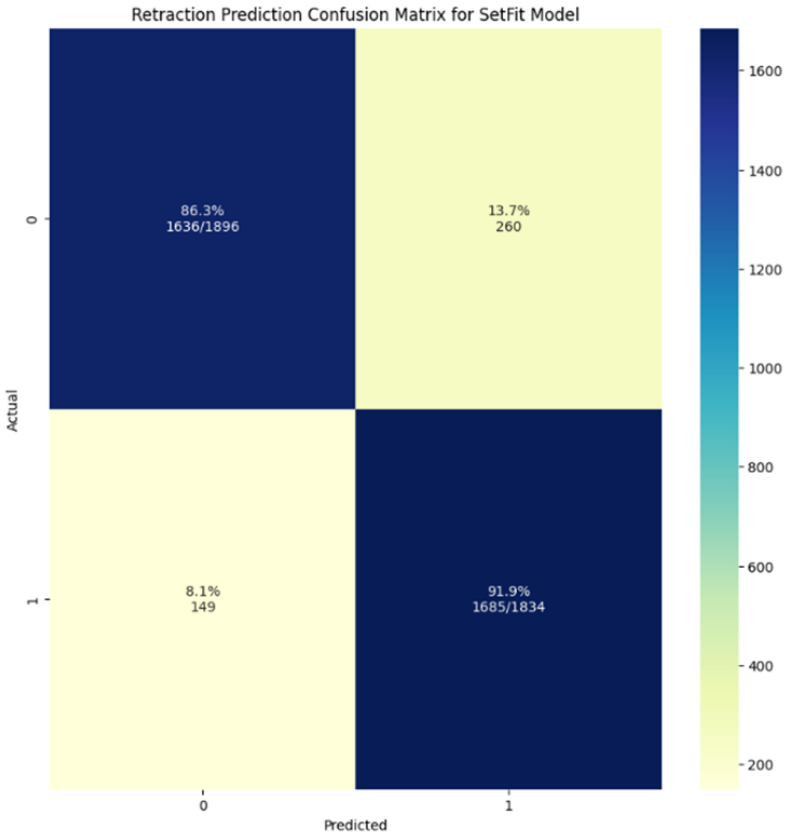
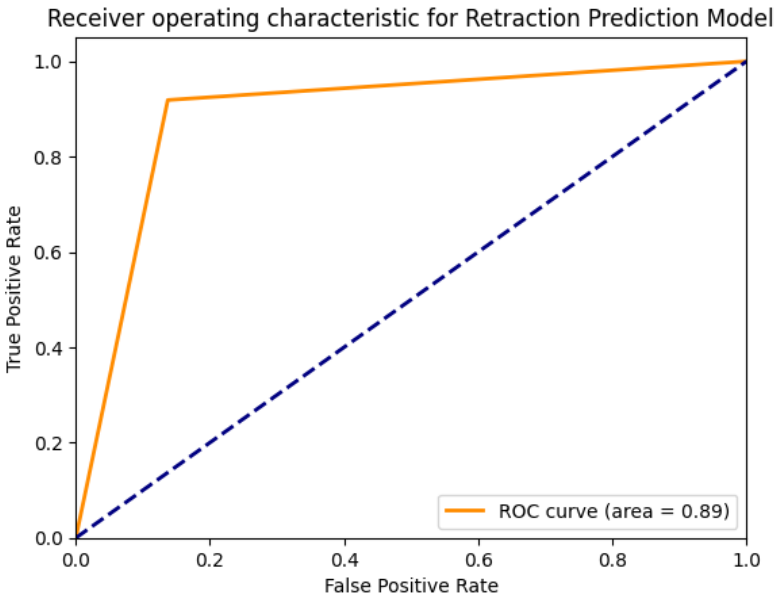


FIGURE 4.7: ROC curve for Setfit classifier



Chapter 5

Evaluation

This section of the report aims to discuss the implications of the analyses presented in the previous section, with particular concern for the impact of model performance on the underlying business objective of the study.

5.1 Clustering Performance Evaluation

Naturally, due to the complications surrounding evaluation metrics and their applicability to density-based clustering methods, a metric-centric evaluation of the efficacy of the methodology becomes unachievable. As seen in Appendix A.5, selecting an epsilon value of 0.3 and a minimum cluster size of 1,400 would produce the most significant silhouette score. However, due to the produced cluster count being three, topic coverage would likely be far too interdisciplinary. As such, any associated analysis would fail to produce practical insight when considered within the context of the business question.

However, effective clustering is achieved by extracting helpful insight from the underlying data. Thankfully, examining the efficacy of the methodology is still possible due to the nature of the clustered observations, that of academic articles. As such, selecting appropriate hyperparameters that produce a suitable number of clusters for the size and type of input articles while ensuring that an appropriate proportion of articles are included in clusters becomes the primary objective. Once clustered, reviewing the lists of articles and their component titles and abstracts to determine the approach's effectiveness is possible. While this means confirmation of this approach remains centred around the expertise of subject matter experts (SMEs), initial evaluation can be completed by anyone able to fully comprehend the titles and abstracts of component articles; obviously, this may be far more complicated when considering the heavily technical subject matter. The manually produced labels and those generated by the concatenation of the KerBERT keywords for individual clusters display a significant degree of crossover. As such, the latter could be used if no SME validation is available for the work or if this methodology needs to be reproduced programmatically in the future. While labour-intensive, the accuracy of the resultant labels enables easy identification of the relevant team within the organisation for whom this intelligence is valuable. Finally, the ability to perform bibliometric analysis per cluster basis is compelling because it enables direct comparison of retracted article clusters to one another and articles still in circulation. While the negative class articles represent a random subset and may not truly represent their respective subject community alone, by virtue of sample size, completing the same work while reviewing a specific subject community as a whole would provide truly representative results if this work were to be repeated with a different subset.

5.2 Retraction Reason by Identified Cluster

In most cases, there was a relatively even distribution of retraction reason frequencies. However, in the case of Computer Science, there was a significant proportion of retractions due to Fake Peer reviews, typically representative of paper authors reviewing their paper under the guise of a third-party independent peer reviewer. Applied Mathematics saw a skew towards both plagiarism cases and those where articles had been duplicated, potentially indicative of salami publishing, where a single topic is published in smaller, purportedly unrelated articles to increase one's publication count. Alternatively, duplication can reflect an article being published, in its entirety, in more than one academic journal. Materials Science (large scale) and Oncology also saw a disproportionate frequency of retractions for article duplication. At the same time, Internal Medicine primarily had papers retracted due to Investigation by Company/Institution, with reasons unspecified. Further to the previous points on the case for the application of machine learning over WoS subject categories, DBSCAN clustering is highlighted effectively highlighted the vastly different causes of retraction, as reflected by the two Materials Science clusters.

5.3 Clustering Limitations

As alluded to in previous sections of this report, density-based clustering has two distinct disadvantages when considering the domain of academic texts. Firstly, the established clusters represent the areas of the highest density of the produced TSNEs after dimensionality reduction based on the tuning parameters of the clustering object. Nevertheless, clusters of material outside the parameters may still hold observations attractive to the business. However, suppose parameters are tuned to highlight these clusters by reducing minimum cluster size; larger coherent clusters may be splintered into two or more component clusters while providing no additional actionable intelligence.

Secondly, albeit with the assistance of cluster keywords in evaluating and verifying the coherence of produced clusters, refining the cluster parameters and verifying using article titles reflects the core tenants of supervised learning. Verification and iteration are required before clusters can be confirmed. Consequently, deploying this process as a self-service application to enable non-technical colleagues to complete this work independently, given the somewhat abstract nature of the clustering parameters, would likely be too complex to provide appropriately digestible training. As such, its use outside of stand-alone projects may well be limited.

5.4 Evaluation of the Binary Classification Model

Perhaps unsurprisingly, given the limited sample size, traditional binary classification approaches produced a model only marginally better than random chance with an average ROC Area Under the Curve of 0.67. While it would be possible to undertake a concerted feature engineering effort to improve the model's predictive power, examining the potential of the Few Shot model was considered the best investment in processing time.

The two Few Shot models diverged significantly concerning their evaluation metrics. The model trained from the paraphrase-mpnet-base-v weights displayed a comparably excellent predictive power with far higher sensitivity. Furthermore, consideration should also be paid to the significant reduction in the False Positive rate in the resultant model. As described previously, given the downstream application of these classifiers, increasing the True Positive rate while working to restrict the incidence of False Positives remains

the most effective method of gaining the kind of editorial buy-in required to make this a practical application of the technology.

Finally, it should be noted that given the lack of model checkpoints available that provide a suitable basis for Few Shot learning. Consequently, there remains significant potential for these metrics to be improved upon, with the release of new checkpoint models trained specifically on academic publications with Few Shot applications in mind.

5.5 Model Limitations

When fine-tuning a binary classifier, passing predictions through a softmax function would typically be possible, producing the probabilities for both the positive and negative classes. With probabilities available, it would be possible to plot these concerning their specific impact on both True and False Positive prediction rates to tune the model to suit the underlying needs of the business, requiring the selection of a Positive/Negative threshold value to suit. However, the Few Shot processes, specifically its implementation within the Hugging Face SetFit pipeline, means that it is impossible to extract these probabilities. As such, threshold tuning cannot be conducted.

A critical consideration also lies in the limited number of articles that could be used as the positive class when training these models. Academic articles vary significantly across different subject domains, with a marked variation in the structure and appearance of article abstracts, with some subject communities following structured vs unstructured formats or the addition of lay summaries, particularly within the Social Sciences, which were notably absent from the training data. With this in mind, it would be reasonable to suggest that any trained model would generalise poorly to domains not covered by the training data. As such, considering an alternative source for article abstracts (considering more limited coverage of the Web of Science) may lead to a more well-rounded training set that might generalise to new domains more effectively.

5.6 Implications For Project Objectives

When considering how effectively the analysis described in the previous chapters has helped to address the underlying project criteria, it is perhaps necessary to revisit the project's objectives. The first aim was to use machine learning to understand the commonalities between retracted articles in subject communities. Ultimately, the DBSCAN-centric approach still requires much human-led validation of the extracted clusters, with prospective clusters required to meet the criteria established by the model's parameters (heavy emphasis is placed on the minimum cluster size). Consequently, clusters of articles representing emerging or niche subject disciplines may be overlooked, necessitating an alternative approach to extract their still vital business intelligence.

However, even considering the above limitations, the ability to cluster subject communities at a level illustrative of the underlying content is evident. Drawing specifically on the ability to differentiate between different scales of materials science as an example, which both displayed vastly different trends in retraction reason, indicates valuable information may be hidden if one looks at retraction reasons using Web of Science subject categories as a method of article differentiation alone. It should also be noted that if a single large subject community is identified, repeating the methodology outlined on the identified subset, generating a fresh set of TSNEs and then reclustering using smaller parameters would also enable the ability to view a given subject discipline's component sub-topics.

The project's second goal focused on developing a binary classifier to predict possible retractions effectively. Notably, the release of the Hugging Faces SetFit pipeline and the paraphrase-mpnet-base-v base model during the project's runtime enabled the production of an effective avenue of prediction. Importantly predictions made by the model displayed a significantly lower False Positive rate. Given the requirement for deploying this model within the academic review pipeline, reducing the unnecessary workload caused by false flags would ensure successful integration into existing workflows.

However, as noted previously, while the SetFit model displayed its power when classifying data with a small number of both classes, the ability of this model to generalise to data outside the sample set may prove a roadblock. This drawback is further highlighted due to noted variations in the structure and language used in academic abstracts of varying subject communities. Therefore, either building a large model with more than the c. 12k papers used as the training set for this study or building multiple smaller models to capture the idiosyncrasies of varying disciplines effectively would be necessary.

Chapter 6

Conclusion

6.1 Project Management

Largely the time management plan proposed from the outside stayed true to the project's development lifecycle, with a small amount of deviation, as displayed in Appendix Table A.2, with orange displaying extra time added and red displaying unused time. However, the project's start and finish times remain unchanged. The single most significant variation was observed during model building, primarily due to the need to pivot late in the development lifecycle to use few-shot learning.

6.2 Problems Encountered

The project faced two roadblocks during the development lifecycle. Firstly, gathering the DOIs of retracted articles proved difficult, ultimately requiring a more prolonged time investment to gain familiarity with the syntax of GraphQL queries and the subsequent requirement to paginate results.

The more significant problem was identified during the training time of the SetFit models, related to its significant training time per epoch and the size of the input data. As a result of the training data being academic abstracts of varying lengths (between 300 and 600 words on average), there was a significant variety of tokens passed to the trainer, with some shorter abstracts using several padding tokens to reach the max length. In contrast, others would need to be truncated during tokenisation. Resultant CUDA memory errors, seen when memory requirements exceed the 16GB of VRAM available, would result in a trainer crash, regularly losing all progress at the mid-point of training. The CUDA error was resolved by downward experimentation of the training batch size, sacrificing training speed for trainer stability.

6.3 Future Opportunities

Perhaps the area with the most significant potential benefit would be the experimentation of the viability of a single SetFit model for all new submissions vs multiple models tailored for individual subject areas. Given the variability in the frequency of retractions outlined above, these individual models could be trained on a risk-posed priority, and potentially produced results could be more accurate. However, the limitation of this approach is the availability of training data.

Linked to the previous point is whether alternative data sources could be identified using

web-scraping or some multi-publisher coordination to explore whether greater availability of training data could reinforce the resultant training model(s), given the c. 6000 positive examples. Furthermore, expanding the negative classes observed in the training data to a 2-to-1 ratio of the negative to positive class might result in a more robust model.

6.4 Contributions

This project underlines the potential for Natural Language Processing to contribute to evaluating incoming manuscripts to editorial offices. It is essential to bear in mind the impact of the explosive growth in article output which has only exacerbated the editorial workload of editors and peer-reviews in recent years. Any tool that results in a net loss to the workload of academics involved in the publishing process will be warmly received as long as its benefit is apparent and its impact transparent. Furthermore, while machine learning has received, albeit slow, uptake in the academic publishing sector, the contribution of this literature, or excerpts from it, will be hugely helpful in allowing journal managers to make compelling cases to editorial offices for testing and potential implementation in the immediate future.

Chapter 7

Bibliography

Aggarwal, C. C., Gates, S. C., Yu, P. S. (1999). On the merits of building categorization systems by supervised clustering. Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '99, 352–356. https://doi.org/10.1145/312129.312279

Al-Adawi, S., Ali, B. H., Al-Zakwani, I. (2016). Research Misconduct: The Peril of Publish or Perish. Oman Medical Journal, 31(1), 5–11. https://doi.org/10.5001/omj.2016.02

Ambalavanan, A. K., Devarakonda, M. V. (2020). Using the contextual language model BERT for multi-criteria classification of scientific articles. Journal of Biomedical Informatics, 112, 103578. https://doi.org/10.1016/j.jbi.2020.103578

Barbour, V. (2015, September 23). Publish or perish culture encourages scientists to cut corners. Australian National University Newsroom.

Beltagy, I., Lo, K., Cohan, A. (2019). SciBERT: A Pretrained Language Model for Scientific Text. Arxiv. http://arxiv.org/abs/1903.10676

Caragea, C., Silvescu, A., Kataria, S., Caragea, D., Mitra, P. (2011). Classifying Scientific Publications Using Abstract Features. SARA 2011 - Proceedings of the 9th Symposium on Abstraction, Reformulation, and Approximation, 26–33.

Cohan, A., Feldman, S., Beltagy, I., Downey, D., Weld, D. S. (2020). SPECTER: Document-level Representation Learning using Citation-informed Transformers. In arxiv. http://arxiv.org/abs/2004.07180

Cutting, D. R., Karger, D. R., Pedersen, J. O., Tukey, J. W. (1992). Scatter/Gather: a cluster-based approach to browsing large document collections. Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '92, 318–329. https://doi.org/10.1145/133160.133214

De Rond, M., Miller, A. N. (2005). Publish or Perish. Journal of Management Inquiry, 14(4), 321–329. https://doi.org/10.1177/1056492605276850

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT,

4171–4186. https://github.com/tensorflow/tensor2tensor

Dudek, A. (2020). Classification and Data Analysis (K. Jajuga, J. Batóg, M. Walesiak, Eds.). Springer International Publishing. https://doi.org/10.1007/978-3-030-52348-0

Eun Seo Jo, U., Tunstall, L., Bates, L., Korat, D., Pereg, O. (2022, September 26). SetFit: Efficient Few-Shot Learning Without Prompts. Hugging Face Blog.

Gilbert, F. J., Denison, A. R. (2003). Research Misconduct. Clinical Radiology, 58(7), 499–504. https://doi.org/10.1016/S0009-9260(03)00176-4

Jufang, S., Huiyun, S. (2011). The outflow of academic papers from China: why is it happening and can it be stemmed? Learned Publishing, 24(2), 95–97. https://doi.org/10.1087/20110203

Kadhim, A. I. (2018). An Evaluation of Preprocessing Techniques for Text Classification . International Journal of Computer Science and Information Security, 16(6), 22–32.

Li, C., Lu, Y., Wu, J., Zhang, Y., Xia, Z., Wang, T., Yu, D., Chen, X., Liu, P., Guo, J. (2018). LDA Meets Word2Vec: A Novel Model for Academic Abstract Clustering. The Web Conference 2018 - Companion of the World Wide Web Conference, WWW 2018, 1699–1706. https://doi.org/10.1145/3184558.3191629

Ling, C. X., Huang, J., Zhang, harry. (2003). AUC: a Statistically Consistent and more Discriminating Measure than Accuracy. IJCAI.

Maaten, L. van der, Hinton, G. (2008). Visualizing Data using t-SNE. Journal of Machine Learning Research, 9, 2579–2605.

Masand, B., Linoff, G., Waltz, D. (1992). Classifying news stories using memory based reasoning. Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '92, 59–65. https://doi.org/10.1145/133160.133177

Millar, J. R., Peterson, G. L., Mendenhall, M. J. (2009). Document Clustering and Visualization with Latent Dirichlet Allocation and Self-Organizing Maps. FLAIRS Conference

Mindzak, M., Eaton, S. E. (2021). Artificial intelligence is getting better at writing, and universities should worry about plagiarism. The Conversation. https://theconversation.com

Morris, S., Barnas, E., LaFrenier, D., Reich, M. (2013). The Handbook of Journal Publishing (Vol. 1). Cambridge University Press.

OECD. (2014). OECD Science, Technology and Industry Outlook 2014.

Parker, C. (2013). On measuring the performance of binary classifiers. Knowledge and Information Systems, 35(1), 131–152. https://doi.org/10.1007/s10115-012-0558-x

Parnami, A., Lee, M. (2022). Learning from Few Examples: A Summary of Approaches to Few-Shot Learning.

Priem, J., Piwowar, H., Orr, R. (2022). OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts.

Provost, F., Fawcett, T. (2001). Robust Classification for Imprecise Environments. Machine Learning, 203–231.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. (2018). Improving Language Understanding by Generative Pre-Training. Preprint.
https://gluebenchmark.com/leaderboard

Rongbo Du, Safavi-Naini, R., Susilo, W. (2003). Web filtering using text classification. The 11th IEEE International Conference on Networks. ICON2003., 325–330.
https://doi.org/10.1109/ICON.2003.1266211

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., Xu, X. (2017). DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. ACM Transactions on Database Systems, 42(3), 1–21. https://doi.org/10.1145/3068335

Shah, P., Kendall, F., Khozin, S., Goosen, R., Hu, J., Laramie, J., Ringel, M., Schork, N. (2019). Artificial intelligence and machine learning in clinical development: a translational perspective. Npj Digital Medicine, 2(1), 69. https://doi.org/10.1038/s41746-019-0148-3

Sharma, P., Li, Y. (2019). Self-Supervised Contextual Keyword and Keyphrase Retrieval with Self-Labelling. https://doi.org/10.20944/preprints201908.0073.v1

Sinha, A., Shen, Z., Song, Y., Ma, H., Eide, D., Hsu, B.-J. (Paul), Wang, K. (2015). An Overview of Microsoft Academic Service (MAS) and Applications. Proceedings of the 24th International Conference on World Wide Web, 243–246.
https://doi.org/10.1145/2740908.2742839

Sivakumar, S., Videla, L. S., Rajesh Kumar, T., Nagaraj, J., Itnal, S., Haritha, D. (2020). Review on Word2Vec Word Embedding Neural Net. 2020 International Conference on Smart Electronics and Communication (ICOSEC), 282–290.
https://doi.org/10.1109/ICOSEC49089.2020.9215319

Steen, R. G. (2011). Retractions in the scientific literature: is the incidence of research fraud increasing? Journal of Medical Ethics, 37(4), 249–253.
https://doi.org/10.1136/jme.2010.040923

Steinbach, M., Karypis, G., Kumar, V. (2000). A Comparison of Document Clustering Techniques.

Steinley, Douglas. (2006). K-means clustering: A half-century synthesis. British Journal of Mathematical and Statistical Psychology, 59(1), 1–34.
https://doi.org/10.1348/000711005X48266

The Center for Scientific Integrity. (2018). The Retraction Watch Database. http://retractiondatabase.org/

UNESCO. (2015). UNESCO science report: towards 2030.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. Le, Gugger, S., ... Rush, A. M. (2019). HuggingFace's Transformers: State-of-the-art Natural Language Processing.

Zamir, O., Etzioni, O., Madani, O., Karp, R. M. (1997). Fast and Intuitive Clustering of Web Documents. KDD-97, 287–290.

Zhou, T., Zhang, Y., Lu, J. (2015). Classifying Computer Science Papers. Proceedings of the International Joint Conference on Artificial Intelligence.

Appendix A

Appendix

A.1 Cluster exemplars

https://1drv.ms/u/s!AiZ8E$_G$$IEpwAgaRP1wHwObRY1_XVmg?e = 61$$Hgkf$

A.2 Project GitHub Repo

https://github.com/Brian-M-Collins/academic-paper-retractions

Can either be built as a Codespace, or cloned locally (import paths will need to be updated)

A.3 Signed Project Specification

UNIVERSITY OF PORTSMOUTH

School of Computing Postgraduate Programme

MSc in Data Analytics *(online)*

Project Specification

Brian Collins

Project Specification

1. Basic details

Student name:	Brian Collins
Draft project title:	Using machine learning techniques to identify commonalities amongst academic article retractions
Course and year:	MSc Data Analytics – 2022
Client organisation:	John Wiley & Sons Ltd
Client contact name:	David Flanagan
Project supervisor:	Ramazan Esmeli

2. Outline of the project environment

John Wiley & Sons Ltd is a commercial publishing company based in the United States which focuses on academic publishing. Given the annual increase in article submissions, the academic peer review process is under considerable stress related to submitting ethically questionable articles. These submissions, once retracted, lead to significant damage to the profile of not just Wiley as the publisher but the independent editorial teams that manage the journals published by Wiley.

3. The problem to be solved

The work aims to provide intelligence on the commonalities between retractions at the publisher, subject community, geographic, or journal level would facilitate the development of other techniques to head off problematic papers during the submission process.

The work can also extend to looking at only Wiley's data, such as peer review times of retracted papers and reviewer reports. This portion of the work will highlight whether review times differ significantly from the mean for subject communities and whether reviewer reports show an indication of manipulation of the peer review process through highlighting text duplication across reviews.

4. Breakdown of tasks

A significant portion of this work's early part will focus on gaining access to the Retraction Watch database, which contains a robust index of retracted papers with an accompanying justification for that retraction.

Once access to the appropriate database of retracted papers has been achieved, the next step will be to validate data and proceed with feature engineering to associate each retraction not just with its publishing journal but also with the publishing group, Web of Science subject category (where possible). A likely next step will be to develop vectorised embeddings of the associated retraction notices, with subsequent t-SNE dimensionality reduction. Once documents have been plotted in two-dimensional space, DBSCAN clustering can be applied, given the high likelihood of arbitrarily shaped clusters resulting from the feature mentioned above engineering steps. Once these clusters have been

identified, they can be compared to the Retraction Watch justification notices and then EDA can be completed to identify the commonalities as described in section 2.

5. Project deliverables

The project aims to develop a report detailing the common themes among retracted articles; this will detail the geographies of authors, retraction justifications per subject community, and journals with the highest proportion of retracted papers per n publication.

6. Requirements

The client requires a greater understanding of the context around the significant increase in academic article retractions over the last ten years.

7. Legal, ethical, professional, social issues

While the bulk of the data is freely available on the retraction watch database, they explicitly require that the database is not published in its entirety. As such, only high-level data analysis will be publishable, and the entire underlying dataset cannot be supplied alongside the published article.

Similarly, as described during earlier portions of this specification, the client's data will be used as a supplementary addition to the data provided by the Retraction Watch dataset. This data, given its proprietary and confidential nature, cannot be provided in its entirety with any published portion of the study.

While reviewer comments will be analysed, any personally identifiable information (PII) can be obfuscated through the application of hashing through Python's now native Hashlib package. Consequently, there should be no ethical considerations with using any of the described datasets.

8. Facilities and resources

All work will be completed using Python and its associated library of packages. Given the computing requirements of some of these steps, data will be processed using a Github codespace with the support of the client. There are no funding implications as these are all typical resources available in conjunction with my employment by the client.

9. Project plan

See attached project Gantt chart for information on workflow and schedule.

10. Supervision meetings

Supervision meetings have been described with Dr Ramazan Esmeli on a fortnightly basis throughout this project. No concerns have been raised regarding the potential for absences of Dr Esmeli. However, short-duration absences should present no barrier to effective project completion, provided the bulk of the schedule is adhered to.

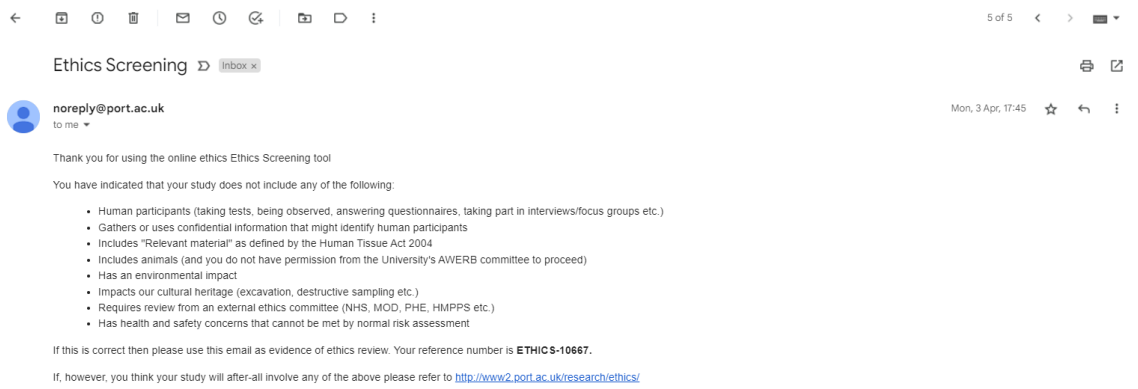
11. Project mode

	Please delete as appropriate	
Registration mode		Part Time
Project mode		Part Time
Planned submission deadline	19th May 2023	

12. Signatures

	Signature:	Date:
Student	Brian Collins	25.04.2023
Client	David WE	
Project supervisor		

A.4 Ethics Screening Tool Email



5 of 5

Ethics Screening Inbox ×

noreply@port.ac.uk
to me

Mon, 3 Apr, 17:45

Thank you for using the online ethics Ethics Screening tool

You have indicated that your study does not include any of the following:

- Human participants (taking tests, being observed, answering questionnaires, taking part in interviews/focus groups etc.)
- Gathers or uses confidential information that might identify human participants
- Includes "Relevant material" as defined by the Human Tissue Act 2004
- Includes animals (and you do not have permission from the University's AWERB committee to proceed)
- Has an environmental impact
- Impacts our cultural heritage (excavation, destructive sampling etc.)
- Requires review from an external ethics committee (NHS, MOD, PHE, HMPPS etc.)
- Has health and safety concerns that cannot be met by normal risk assessment

If this is correct then please use this email as evidence of ethics review. Your reference number is **ETHICS-10667.**

If, however, you think your study will after-all involve any of the above please refer to http://www2.port.ac.uk/research/ethics/

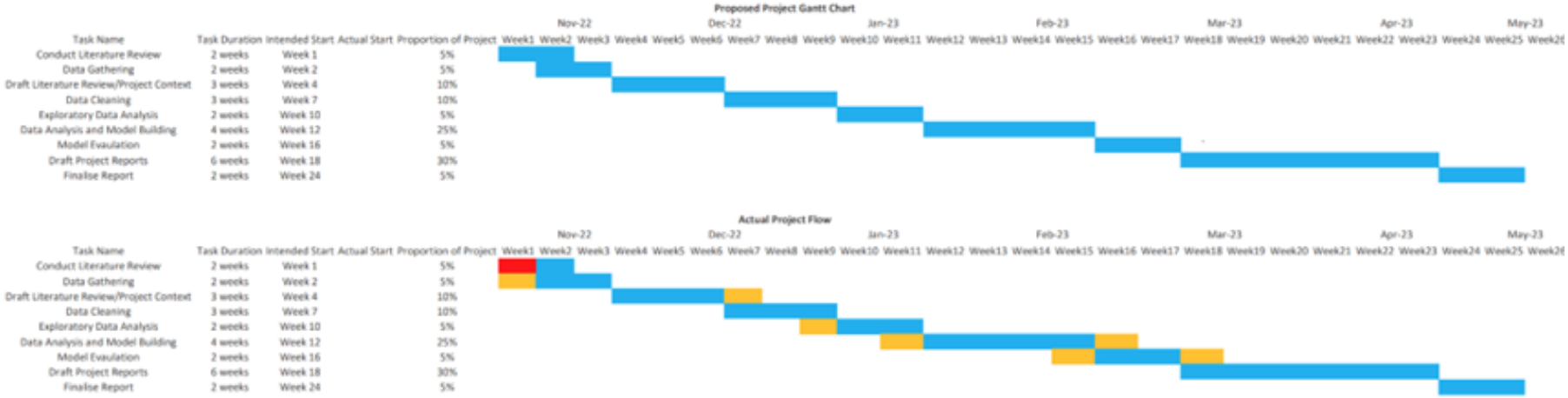
A.5 Clustering grid search results

Size	Distance	Num Clusters	Pct Clusters	Score	Size	Distance	Num Clusters	Pct Clusters	Score
1400	0.3	3	0.873934	0.2709907	**900**	1.5	2	0.920354	0.2542186
1450	0.3	3	0.873934	0.2709907	**1450**	1.5	2	0.920354	0.2542186
1350	0.3	3	0.873934	0.2709907	**600**	1.5	2	0.920354	0.2542186
1450	0.1	3	0.873934	0.2709907	**650**	1.5	2	0.920354	0.2542186
1400	0.1	3	0.873934	0.2709907	**1400**	1.5	2	0.920354	0.2542186
1350	0.1	3	0.873934	0.2709907	**1350**	1.5	2	0.920354	0.2542186
1300	0.1	3	0.873934	0.2709907	**1300**	1.5	2	0.920354	0.2542186
1300	0.3	3	0.873934	0.2709907	**1250**	1.5	2	0.920354	0.2542186
1450	0.2	3	0.873934	0.2709907	**1200**	1.5	2	0.920354	0.2542186
1400	0.2	3	0.873934	0.2709907	**1150**	1.5	2	0.920354	0.2542186
1350	0.2	3	0.873934	0.2709907	**1100**	1.5	2	0.920354	0.2542186
1300	0.2	3	0.873934	0.2709907	**1050**	1.5	2	0.920354	0.2542186
1150	0.2	4	0.7353178	0.2542881	**1000**	1.5	2	0.920354	0.2542186
1100	0.2	4	0.7353178	0.2542881	**500**	0.2	8	0.7352373	0.1268618
1150	0.3	4	0.7353178	0.2542881	**500**	0.3	8	0.7352373	0.1268618
1250	0.2	4	0.7353178	0.2542881	**500**	0.1	8	0.7352373	0.1268618
1100	0.3	4	0.7353178	0.2542881	**550**	0.1	7	0.7365245	0.1220835
1050	0.3	4	0.7353178	0.2542881	**600**	0.2	7	0.7365245	0.1220835
1000	0.3	4	0.7353178	0.2542881	**550**	0.2	7	0.7365245	0.1220835
950	0.3	4	0.7353178	0.2542881	**600**	0.1	7	0.7365245	0.1220835
900	0.3	4	0.7353178	0.2542881	**600**	0.3	7	0.7365245	0.1220835
850	0.3	4	0.7353178	0.2542881	**550**	0.3	7	0.7365245	0.1220835
800	0.3	4	0.7353178	0.2542881	**650**	0.2	6	0.6851971	0.101111
750	0.3	4	0.7353178	0.2542881	**650**	0.3	6	0.6851971	0.101111
1200	0.3	4	0.7353178	0.2542881	**650**	0.1	6	0.6851971	0.101111
800	0.2	4	0.7353178	0.2542881	**450**	0.2	10	0.6945294	0.0999735
850	0.2	4	0.7353178	0.2542881	**450**	0.1	10	0.6945294	0.0999735
1200	0.2	4	0.7353178	0.2542881	**450**	0.3	10	0.6945294	0.0999735
900	0.2	4	0.7353178	0.2542881	**400**	0.1	11	0.6923572	0.0936971
950	0.2	4	0.7353178	0.2542881	**350**	0.1	11	0.6923572	0.0936971
1000	0.2	4	0.7353178	0.2542881	**350**	0.2	11	0.6923572	0.0936971
750	0.2	4	0.7353178	0.2542881	**400**	0.2	11	0.6923572	0.0936971
1050	0.2	4	0.7353178	0.2542881	**400**	0.3	11	0.6923572	0.0936971
1250	0.3	4	0.7353178	0.2542881	**350**	0.3	11	0.6923572	0.0936971
1000	0.1	4	0.7353178	0.2542881	**700**	0.1	5	0.6294449	0.0929715
750	0.1	4	0.7353178	0.2542881	**700**	0.2	5	0.6294449	0.0929715
800	0.1	4	0.7353178	0.2542881	**700**	0.3	5	0.6294449	0.0929715
850	0.1	4	0.7353178	0.2542881	**300**	0.2	12	0.7176187	0.0914257
950	0.1	4	0.7353178	0.2542881	**300**	0.1	12	0.7176187	0.0914257
900	0.1	4	0.7353178	0.2542881	**300**	0.3	12	0.7176187	0.0914257
1050	0.1	4	0.7353178	0.2542881	**100**	0.3	32	0.7097345	0.0312588
1100	0.1	4	0.7353178	0.2542881	**50**	0.3	57	0.7156074	0.0143191
1150	0.1	4	0.7353178	0.2542881	**100**	0.2	35	0.6374095	-0.0197571
1200	0.1	4	0.7353178	0.2542881	**250**	0.3	14	0.6234111	-0.0203072
1250	0.1	4	0.7353178	0.2542881	**250**	0.2	14	0.6234111	-0.0203072
300	1.5	2	0.920354	0.2542186	**250**	0.1	14	0.6234111	-0.0203072
550	1.5	2	0.920354	0.2542186	**150**	0.3	25	0.6549477	-0.0209123
500	1.5	2	0.920354	0.2542186	**50**	0.2	71	0.6419147	-0.0396831
450	1.5	2	0.920354	0.2542186	**50**	0.1	72	0.6251006	-0.0627978
400	1.5	2	0.920354	0.2542186	**100**	0.1	36	0.6040225	-0.065064
350	1.5	2	0.920354	0.2542186	**150**	0.2	27	0.586967	-0.0683716
800	1.5	2	0.920354	0.2542186	**150**	0.1	27	0.586967	-0.0683716
250	1.5	2	0.920354	0.2542186	**200**	0.3	16	0.6054706	-0.0695618
200	1.5	2	0.920354	0.2542186	**200**	0.2	17	0.5938053	-0.0894368
850	1.5	2	0.920354	0.2542186	**200**	0.1	17	0.5938053	-0.0894368
700	1.5	2	0.920354	0.2542186	**150**	1.5	2	0.9729686	-0.146338
750	1.5	2	0.920354	0.2542186	**100**	1.5	2	0.9729686	-0.146338
950	1.5	2	0.920354	0.2542186	**50**	1.5	2	0.9729686	-0.146338

TABLE A.1: KeyBERT keywords by article cluster

Cluster_0	Cluster_1	Cluster_2	Cluster_3	Cluster_4	Cluster_5
essential oils	microbial fuel	fuzzy lyapunov method	bifurcations occur delay	osteoarthritis oa	breast cancer survivors
essential oil	microbial fuel cells	fuzzy lyapunov	hopf bifurcations stability	knee osteoarthritis	working memory
silver nanoparticles agnps	microbial fuel cell	adaptive neuro fuzzy	pipe heat exchanger	hydroxyapatite	physical activity pa
green fluorescent protein	aerobic microbial degradation	particle swarm optimization	periodic solutions	evaluate bone implant	traumatic brain injury
drosophila genome	microwave assisted extraction	neuro fuzzy inference	fractional differential equations	ankle arthroplasty	multiple sclerosis ms
fluorescent protein	remote sensing	support vector machine	convection heat transfer	pain questionnaire	knowledge management km
inhibitory activity complement	bacterial communities	fuzzy controller design	nanofluid flow	mesenchymal stem cells	depression scale
sensitive bacteria produce	bio electrochemical	control recurrent laguerre	harmonic functions	knee osteoarthritis oa	neonatal mortality
bacteria produce rare	salts chemoorganotrophic bacteria	background estimation method	asymptotic behavior solutions	femoral neck fracture	adolescent depression scale
plant salt tolerance	mitochondrial genome	adaptive control	delay bifurcation	bone graft	cancer survivors
Cluster_6	**Cluster_7**	**Cluster_8**	**Cluster_9**	**Cluster_10**	**Cluster_11**
functionally graded steels	graphene oxide	dendritic cells	hypertrophic cardiomyopathy hcm	mesenchymal stem cells	cell lung cancer
graded steels	graphene	dendritic cells dcs	acute ischemic stroke	cerebral ischemia	lung cancer nsclc
neural networks model	perovskite solar	newcastle disease virus	pulmonary artery thermodilution	alzheimer disease	hepatocellular carcinoma hcc
functionally graded steel	carbon nanotubes	respiratory syndrome coronavirus	pain injection propofol	apoptosis inflammatory	lung cancer
titanium alloy	perovskite solar cell	coronavirus sars cov	aortic dissection	diabetic nephropathy	micrornas
neural networks models	photocatalytic degradation	rheumatoid arthritis	ischemic stroke	endothelial progenitor cells	hepatocellular carcinoma
graded steels produced	magnetic anisotropy	clostridium difficile infection	atrial fibrillation af	inflammatory cytokines	renal cell carcinoma
nanoparticles concrete	graphene oxide rgo	staphylococcus aureus	cardiac arrest	alzheimer disease ad	circular rnas circrnas
artificial neural networks	perovskite solar cells	rheumatoid arthritis ra	hypertrophic cardiomyopathy	blood brain barrier	ovarian cancer cells
graded ferritic austenitic	reduced graphene oxide	allergic asthma	atherosclerotic plaques	cell apoptosis inflammatory	colorectal cancer

TABLE A.2: Proposed vs actual project Gantt chart



Appendix B

Appendix: Python Extracts

B.1 Python code used to query GraphQL to return a complete list of retracted articles using the native Requests library

```
query = """
    query{
        IPublicationCount(retracted: true)
    }
"""

r = requests.post(ENDPOINT, json={"query": query}, headers=headers)
retracted_counts = r.json()["data"]["IPublicationCount"]
doi_out_list = []

first_query = """
    query{publist: IPublications(retracted: true, scrollId: "
                                            cf7b9d17a3a911f24664", size: 500)
                                             {
        doi
    }
    }
"""

r = requests.post(ENDPOINT, json={"query": first_query}, headers=headers)

for x in r.json()["data"]["publist"]:
    doi_out_list.append(x["doi"])

for num in range(0, math.ceil((retracted_counts/CHUNKSIZE)-1)):
    subsequent_query = """
        IPublications(scrollId: "cf7b9d17a3a911f24664") {
            doi
        }
    """
    r = requests.post(ENDPOINT, json={"query": subsequent_query}, headers=
                                            headers)
    for x in r.json()["data"]["publist"]:
        doi_out_list.append(x["doi"])
```

B.2 Gathering abstract data from EBAC, including concatenating multiple paragraphs into a single Python string

```
    abstract_data = pd.DataFrame()
abstract_dois = [article_data[i : i + n] for i in range(0, article_data.
                                        shape[0], n)]

for chunk in abstract_dois:
    STMT = f"""
        SELECT      article.doi,
                    abstract.abstract_text
        FROM "PROD_EDW". "EBAC". "DW_ARTICLE_EXTN" article
        JOIN "PROD_EDW". "EBAC". "DW_ABSTRACT" abstract on article.
                                                article_id = abstract.
                                                article_id
        WHERE article.doi in {tuple(chunk["doi"].to_list())}
    """

    conn = snowflake_utils.connect_to_snowflake()
    df = pd.read_sql(STMT, conn)
    abstract_data = pd.concat([abstract_data, df], axis=0).dropna(
        subset=["abstract_text"]
    )

papers_grouped = (
    abstract_data.drop_duplicates(["doi", "abstract_text"], keep="first")
    .groupby(["doi"])["abstract_text"]
    .apply(",".join)
    .reset_index()
)

papers_grouped.columns = ["doi", "concat_abstract"]
```

B.3 Querying SQL using SQLAlchemy from Python

```
    retracted_dois = pd.read_csv(workspaces/analysing_retractions/
                                        clustering/data/retracted_dois.
                                        csv"
).dropna()

n = 1000
chunked_dois = [retracted_dois[i : i + n] for i in range(0, retracted_dois.
                                    shape[0], n)]

article_data = pd.DataFrame()

for chunk in chunked_dois:
    STMT = f"""
        SELECT      article.doi,
                    article.article_id,
                    article.article_title,
                    article.full_source_title,
                    subject.subject_cat_desc,
                    article.publisher_group,
                    article.year_published,
                    article.ARTICLE_OPEN_ACCESS_STATUS,
                    article.FWCI,
                    metrics.citations
        FROM "PROD_EDW". "EBAC". "DW_ARTICLE_EXTN" article
        JOIN "PROD_EDW". "EBAC". "DW_ABSTRACT" abstract on article.
                                        article_id = abstract.
                                        article_id
        JOIN "PROD_EDW". "EBAC". "ARTICLE_METRICS_AGG" metrics on article.
                                        article_id = metrics.
                                        article_id
        JOIN "PROD_EDW". "EBAC". "DW_SUBJECT_CATEGORY_EXTN" subject on
                                        article.article_id = subject.
                                        article_id
        WHERE article.doi in {tuple(chunk["doi"].to_list())}
    """

    conn = snowflake_utils.connect_to_snowflake()
    df = pd.read_sql(STMT, conn)
    article_data = pd.concat([article_data, df], axis=0).drop_duplicates("
                                        doi")
```

B.4 Punctuation removal and text lowering function

```
    import string
def remove_punctuation(text):
    return "".join([i for i in text if i not in string.punctuation])

merged["concat_abstract"] = merged["concat_abstract"].apply(
    lambda x: remove_punctuation(x.lower()))
```

B.5 Stop word removal and Lemmatisation using NLTK

```
    stop_words = set(stopwords.words("english"))

def remove_stopwords(text):
    return " ".join([word for word in text.split() if word not in
                                        stop_words])

merged["concat_abstract"] = merged["concat_abstract"].apply(lambda x:
                                    remove_stopwords(x))


lemmatizer = WordNetLemmatizer()

def lemm_abstract(text):
    return " ".join([lemmatizer.lemmatize(w) for w in nltk.word_tokenize(
                                        text)])

merged["concat_abstract"] = merged["concat_abstract"].apply(lambda x:
                                    lemm_abstract(x))
```

B.6 Data preparation function

```
def prepare_data(dataframe:str):
"""
A function to check for the presence of correct columns and strip
                                        excess columns from a source
                                        dataframe.
Parameters
---------
dataframe (str): the source dataframe
Returns
---------
output(dataframe): a dataframe with columns doi, title, abstract
"""
df = dataframe
df = df.rename(columns={"article_title": "title", "concat_abstract":"
                                        abstract"})
# check the data has the necessary columns to read and stop if it doesn
                                        't
assert (
    "doi" in df.columns
), "doi column not found, please check your dataframe and column name
                                        case."
assert (
    "title" in df.columns
), "title column not found, please check your dataframe and column name
                                         case."
assert (
    "abstract" in df.columns
), "abstract column not found, please check your dataframe and column
                                        name case."

#cut down to the three required columns.
output = pd.DataFrame(df[["doi", "title", "abstract"]])
output["title"] = output["title"].str.strip()
output["abstract"] = output["abstract"].str.strip()

return output
```

B.7 t-SNE training Python code excerpt

```
    n_jobs = -1  # use all available cores
    METHOD = "pynndescent"  # NN calculation method
    METRIC = "cosine" # cosine performs well for high dimensional data
    RANDOM_STATE = 42
    PERPLEXITY = 500  # based on openTSNE documentation, 500 for large
                                        datasets

    x_train = np.vstack(combined["embeddings"])
    y_train = combined["doi"]

    affinities_train = PerplexityBasedNN(
        x_train,
        perplexity=PERPLEXITY,
        method=METHOD,
        metric=METRIC,
        n_jobs=n_jobs,
        random_state=RANDOM_STATE,
    )

    init_train = initialization.pca(x_train, random_state=RANDOM_STATE)

    embedding_train = TSNEEmbedding(
        init_train,
        affinities_train,
        negative_gradient_method= "fft",
        n_jobs=n_jobs,
    )

    # Optimise embedding: 1 Early exaggeration phase
    embedding_train_1 = embedding_train.optimize(
        n_iter=250,
        exaggeration=12,
        momentum=0.5,
    )

    embedding_train_2 = embedding_train_1.optimize(n_iter=750, momentum=0.8
                                        )

    tsne_df_train = pd.DataFrame(embedding_train_2, columns=["tsne_1", "
                                        tsne_2"]).assign(doi=y_train)

    retracted_articles = retracted_articles.merge(tsne_df_train, on="doi",
                                        how="left")
    negative_class = negative_class.merge(tsne_df_train, on="doi", how="
                                        left")
```

B.8 Embedding function

```
def embed_df_abstracts(abstracts, batch_size=4):
tokenizer = AutoTokenizer.from_pretrained("allenai/specter")
embedded_df = pd.DataFrame()

for chunk in tqdm(
    np.split(abstracts, np.arange(batch_size, len(abstracts),
                                          batch_size))
):

    title_abs = [
        (d.get("title") or "") + tokenizer.sep_token + (d.get("abstract
                                                ") or "")
        for d in chunk.to_dict("records")
    ]

    inputs = tokenizer(
        title_abs,
        padding=True,
        truncation=True,
        return_tensors= "pt",
        max_length=512,
    ).to(device)

    results = model(**inputs)

    embedding_list = results.last_hidden_state[:, 0, :].detach().cpu().
                                            numpy().tolist()
    chunk["embeddings"] = [np.asarray(i) for i in embedding_list]

    embedded_df = pd.concat([embedded_df, chunk], axis=0)

torch.cuda.empty_cache()

df2 = pd.DataFrame(embedded_df).reset_index(drop=True)
output_dataframe = pd.DataFrame(df2[["doi", "title", "embeddings"]])

return output_dataframe
```

B.9 Application of KeyBERT to article abstracts

```
kb_model = KeyBERT()

combined["keybert_keywords"] = combined["concat_abstract"].
                                    progress_apply(
    lambda x: kb_model.extract_keywords(x, keyphrase_ngram_range=(1, 3)
                                        ))

combined["keywords"] = combined["keybert_keywords"].apply(
    lambda x: ", ".join([a_tuple[0] for a_tuple in x]))
```