Tracing the Flow of Knowledge From Science to Technology Using Deep Learning

Dietmar Harhoff Michael E. Rose Mainak Ghosh
Sebastian Erhardt Erik Buunk

Max Planck Institute for Innovation and Competition
Munich, Germany

Email: {dietmar.harhoff, michael.rose, mainak.ghosh,
sebastian.erhardt, erik.buunk }@ip.mpg.de

Final Report

This document is a comprehensive report on the EPO/ARP funded project "Tracing the Flow of Knowledge From Science to Technology Using Deep Learning", conducted from August 2022 to July 2024. It includes detailed information on the project's background, the data used, and the results achieved.

Executive Summary

This project aims to enhance the tracing of knowledge flows from scientific research to technology using advanced deep-learning techniques. By developing models such as Pat-SPECTER, and PaECTER, the project seeks to improve the accuracy of identifying connections between patents and scientific literature, surpassing the limitations of traditional citation-based analysis.

Key findings include the analysis of the performance of Pat-SPECTER in predicting scientific citations for patents and the performance of PaECTER in predicting citations among patents. The evaluation process was made difficult by the incompleteness of the open-access database OpenAlex, which lacks abstracts for a portion of the scientific literature. Real-world tests demonstrated Pat-SPECTER's effectiveness in identifying relevant prior art documents (patents and publications), improving the efficiency of prior art search.

The project highlights the potential of advanced machine learning models and advances their use in tracing knowledge flows. It provides tools that can enhance patent examination processes, innovation tracking, and research and development strategies. These efforts help foster innovation by revealing the intricate connections between science and technology.

All models and the training datasets
are publicly available on the Hugging Face platform
https://huggingface.co/mpi-inno-comp

Contents

1 Introduction **5**
- 1.1 Motivation . . . 5
- 1.2 Background . . . 6
- 1.3 Objectives . . . 9

2 Project Overview **10**

3 Data Sources **12**
- 3.1 PATSTAT . . . 12
- 3.2 OpenAlex . . . 12
- 3.3 Semantic Scholar . . . 13
- 3.4 SPECTER . . . 13
- 3.5 Reliance on Science . . . 13
- 3.6 SEARCHFORMER . . . 14
- 3.7 Logic Mill . . . 14

4 Training data **15**
- 4.1 Patent to Patent . . . 15
  - 4.1.1 Sampling . . . 15
  - 4.1.2 Positive Citations . . . 16
  - 4.1.3 Negative Citations . . . 17
- 4.2 Publication to Publication . . . 18

5 Training of the Document Similarity Models **19**
- 5.1 Training Pat-SPECTER . . . 19
- 5.2 Training PaECTER . . . 21
- 5.3 Training Pub-PaECTER . . . 22

6 Model Comparison **23**
- 6.1 Methodology . . . 23
- 6.2 Cross-Corpus Comparison . . . 23
- 6.3 Publication to Publication . . . 24
- 6.4 Patent to Patent . . . 25
- 6.5 Comparison with SEARCHFORMER . . . 27
- 6.6 Conclusion . . . 27

7 Applications **28**
- 7.1 Prediction of Patent Paper Pairs . . . 28
  - 7.1.1 Methodology . . . 28

7.1.2 Data . . . 30
7.1.3 Results . . . 32
7.1.4 Conclusion . . . 35
7.2 Prior Art Search . . . 35
7.2.1 Patents to Patents . . . 36
7.2.2 Patents to Publications . . . 37
7.2.3 Patent Paper Pairs . . . 39
7.3 Patent Landscaping Case Studies . . . 41
7.3.1 CRISPR/Cas9 . . . 41
7.3.2 mRNA Vaccine . . . 42
7.4 Conclusion . . . 43

8 Conclusion **45**

A **I**
A.1 Statistical Significance Test Pat-Specter . . . I
A.2 Additional Tables and Figures . . . II
A.3 PaECTER Paper . . . III
A.4 Pat-SPECTER Model . . . III
A.5 PaECTER Model . . . III
A.6 Dataset . . . III

Acknowledgements

We are grateful for the financial grant 2023/8401 from the European Patent Office - Academic Research Programme (EPO-ARP) and for the fruitful discussions and interactions we had with the participants of the ARP workshops and meetings. We also thank the team at the Max Planck Computing and Data Facility (MPCDF) for providing us with access to their high-performance computing cluster and the Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG) for letting us utilize their cloud solutions. Finally, we would like to thank Cheng Li, who joined our team during the project period, especially for his contributions to the patent paper pairs parts of this report.

Chapter 1

Introduction

1.1 Motivation

Tracing the flow of knowledge from science to technology is pivotal for understanding and fostering innovation processes. The dominant approach for tracing knowledge flows is the analysis of explicit citations in scientific publications and patents. Such citations (or references) acknowledge intellectual priority and delineate the foundations upon which new inventions are built. This method, however, has significant limitations, such as the sparsity of citation graphs and the strategic or opportunistic selection of citations. These limitations can obscure the true lineage of scientific and technological advancements.

In the patent domain, citations to scientific literature are used to link patented inventions to their scientific underpinnings. Studies have shown that highly cited scientific articles often serve as the foundation for particularly impactful patented inventions, highlighting the critical role of science in driving technological innovation. For example, a study by Poege et al. (2019) linked around 950,000 patent families to over 2.2 million scientific articles, demonstrating that foundational scientific research significantly contributes to breakthrough inventions. Related studies such as Marx and Fuegi (2020) also use large-scale citation graphs to derive conclusions for innovation policy.

Despite the valuable insights gained from citation analysis, this approach alone cannot capture the full spectrum of knowledge transfer. Many patents do not reference non-patent literature, and the citation graphs remain sparse. To address these gaps, researchers have explored the use of textual similarity to trace potential knowledge flows (Natterer, 2014). By comparing the text of documents, it is possible to gauge their similarity and infer possible connections that explicit citations may miss.

The advent of advanced machine learning techniques, particularly those involving document embeddings within the field of natural language processing, offers a promising alternative (Cohan et al., 2020). These techniques can transform the entire textual content of patents and scientific publications into numerical representations, allowing for scalable and accurate comparison of documents (Erhardt et al., 2022). High textual and semantic similarity between documents may indicate a flow of knowledge, even in the absence of explicit citations. It is important that neither semantic similarity nor citations are proof of such flows, but their detection can help to lay the foundation for subsequent in-depth analysis.

This project aims to leverage these methodological advancements to develop models capable of tracing potential knowledge flows from scientific publications to patents. By fine-tuning models like SPECTER for the patent domain (creating Pat-SPECTER) and developing new models such as PaECTER and Pub-PaECTER, the project seeks to improve the accuracy and reliability of identifying knowledge flow candidates. This approach enhances our understanding of innovation pathways and supports the development of tools for patent analysis, prior art search, and strategic planning of research and development activities. These efforts should help to support innovation efforts by revealing the intricate connections between science and technology.

1.2 Background

As many patents do not have direct references to the non-patent literature, Ahmadpoor and Jones (2017) propose the notion of "distance to the science frontier". Patents citing scientific publications directly are taken to be at the science frontier. Patents citing patents with references to science (but not citing scientific documents directly) are said to be one step away from the frontier. Generalizing this approach, the authors establish "distance from the frontier" as a measure of relatedness to science. The approach allows them to derive a somewhat denser representation of science-technology linkages than would be possible using direct references to scientific publications alone.

However, the use of explicit citations has serious disadvantages, even if indirect links are taken into account. First, the citation graph is sparse (even using the notion of "distance to the science frontier") – both for science-to-science as well as patent-to-science citations. Similarly, patent-to-patent referencing is infrequent relative to the overall set of possible choices. Moreover, citations may be selected strategically or opportunistically (Jaffe & Rassenfosse, 2017), such as to elevate own contributions (self-citation) or to please editors. Moreover, patent attorneys may be tempted to include many citations in order to obfuscate the true state of prior art. Therefore, searching for alternatives or complements to explicit citations has gained attention.

As an alternative to citations, scholars have investigated the possibility of using text as input data and comparing two documents to gauge their similarity. What inventors, patent assignees, and researchers wrote reveals important insights e.g., on the scope and nature of an invention similarly scientific publication. When vectors represent individual documents, vector operations (such as Cosine distance, Manhattan distance, Euclidean distance or others) yield similarity estimates. The similarity of documents can, under certain conditions, help in tracing knowledge flows in the absence of citations.

Thus, textual similarity has also been used in place of citations to study potential flows of knowledge. In his study, Natterer (2014) uses a term frequency–inverse document frequency (TF-IDF) model of textual similarity for technical texts. A more recent example in this growing literature is Kelly et al. (2021), who identify breakthrough patents as patents with low textual similarity to the existing patent text corpus. They then establish that sectors with many breakthrough patents experience higher growth. This relationship could not have been demonstrated convincingly using patent citations, as the authors point out.

Yet, turning massive amounts of text into data is challenging, and so far there exists no scalable solution that at the same time spans diverse text corpora. All applications intended to find similarities between patents or scientific publications suffer from one or several of the following shortcomings: they are specific to only one text corpus (i.e., only patents, such as patent maps); they do not account for semantic structure across different text corpora; they do not scale well.

A common yet simple approach to transform text into a vector is the bag-of-word approach. Typically researchers used a weighted incarnation of this approach, the so-called term frequency–inverse document frequency (TF-IDF). It yields a $k \times n$ matrix for k terms and n documents. The entries represent the weighted count of a given term in a given document. Any given document is represented by the corresponding column vector, which allows for vector operations. TF-IDF vectorization, simple as it is, has two important limitations. First, it ignores the relative positioning of terms (to each other and within the document) and scales badly. For example, including new documents in the corpus may require the re-computation of the entire matrix. Thus, it becomes computationally expensive with the growing number of documents. Secondly, the TF-IDF matrix is sparse and high-dimensional, which leads to higher memory consumption. The loss of information of the location of a term within a sentence, within a paragraph and within a document is presumably the most severe limitation. Kelly et al. (2021) is an example of the bag-of-words approach.

In the past decade, computer scientists have developed methods to represent words, paragraphs, and even whole documents as dense vectors. Word2vec significantly improved the generation of numerical representations of words, which in turn impacted the representation of documents. Introduced by Mikolov et al. (2013), word2vec models words as continuous vector spaces using neural networks. This method captures semantic relationships between words, enabling the transformation of words into numerical vectors that reflect their meanings and contexts. While word2vec itself focuses on word-level embeddings, these embeddings can be aggregated to represent documents. This paved the way for more advanced techniques in natural language processing.

The advent of Bidirectional Encoder Representations from Transformers (BERT), as introduced by Devlin et al. (2019), marked a significant milestone in the evolution of natural language processing (NLP) models. Since its release, BERT has demonstrated outstanding performance across various domains, including semantic search, text classification, and natural language understanding. Our models are descendants of this growing BERT family.

The BERT model leverages the attention layer (Vaswani et al., 2017), which efficiently estimates which tokens of a sentence to what extent are valuable in understanding the sentence. In this family of models, words in a sentence do not have equal weight, as is the case for TF-IDF. BERT is pre-trained on newspaper articles and Wikipedia and focuses on the 40k most important tokens[1]. The computational requirements to train a BERT from scratch are immense, however. Beltagy, Lo, and Cohan (2019) state on their SciBERT paper that 16 TPUv3 (Tensor Processing Units of the third generation) chips were used for 4 days to train BERT's largest model - 8 GPUs (Graphics Processor Unit) are expected to take 40-70 days for the same task. Training a base model can become very expensive and time-consuming.

[1] a token usually represents a word or a part of a word

So, in our approach, we fine-tune existing models, as this is more efficient and allows us to leverage the knowledge already incorporated in the more fundamental models.

Since BERT's token vocabulary is likely not representative of every domain, multiple domain-specific BERT derivatives have been trained. For example, SciBERT was developed by Beltagy, Lo, and Cohan (2019), whose 30k most important tokens overlap with the original general purpose BERT's vocabulary at the rate of 42%. In the patent domain, there are three models trained on patent text, namely PatentBERT by Lee and Hsiang (2020), Google's BERT for patents (Srebrovic & Yonamine, 2020) and the SEARCHFORMER by Vowinckel and Hähnke (2023). The latter model was developed during the course of our project and was therefore not available for systematic comparisons.

PatentBERT and BERT for Patents differ in the underlying training data: PatentBERT was trained on 3 million granted USPTO patents available via Google Patents Public Datasets, while BERT for Patents was trained on more than 100 million patents from the same dataset. Most importantly, though, the BERT for Patents model utilized whole texts of the patents during training.

Because BERT models were trained to predict masked tokens and the next sentence, they do not perform effectively at identifying similar documents. This is a necessary prerequisite to identify potential knowledge flows between documents. The SPECTER model (Cohan et al., 2020) addresses this concern to a large extent. The model generates document-level representations of a scientific publication and uses SciBERT[2] (Beltagy, Lo, & Cohan, 2019) as a foundation. Furthermore, it uses document-relatedness information provided by the scientific publication citation graph. This citation-informed learning has proved to be very successful. SPECTER generates similar numerical representations for similar scientific publications. It exhibits higher accuracy in various tasks, including classifying scientific publications, predicting user activity, recommending scholarly documents, and predicting the co-citation of two articles. However, SPECTER was only trained on scientific publications.

The language specific to patents likely differs from that relevant to scientific publications, and that may impact the performance. To check the vocabulary presumption, we computed the overlap between BERT for Patents's vocabulary and SciBERT's vocabulary, finding it to be 50%. Our preliminary analysis in the patent domain with the pre-trained SPECTER model already showed promising results. SPECTER was not trained on the patent corpus; we can reasonably expect the prediction quality to increase when a model equivalent to SPECTER but specific to patent texts is utilized.

A SPECTER-like model dedicated to patents is the first step to identifying potential knowledge paths originating from scientific publications to patent documents. We argue that a model specific to one corpus needs to be fine-tuned on the other corpus, as their underlying language uses differ. Ex-ante, it is not possible to say whether a model specific to patents but fine-tuned on scientific publications outperforms a model specific to scientific publications but fine-tuned on patents. Statistical comparisons can be utilized to identify a superior approach. We consider it very likely that the winning model will also outperform existing TF-IDF-based solutions.

At the beginning of our project, there was no equivalent of the SPECTER model for patents. Vowinckel and Hähnke (2023) released the SEARCHFORMER model

[2]SciBERT is a BERT model pre-trained for scientific publications.

dedicated to support EPO examiners, yet the model is not publicly available. It also serves a different approach, as it works on paragraphs.

We see many fruitful avenues opening up once a BERT-based citation-informed full-document encoder model is available for the combined text corpus of patents and scientific publications. For example, such a model will enable us to obtain a much clearer picture of a given patent's distance to the frontier of science. The model would also allow researchers to conduct interesting case studies, such as an analysis of the scientific origins of mRNA or CRISPR-Cas9 technologies.

1.3 Objectives

The overall objective is to develop and apply state-of-the-art machine learning models capable of comparing the textual content of patents with other patents as well as scientific publications. Our approach is based on recent advances in the development of document encoding models. The output of our models are fixed-length dense vectors. This representation allows for fast comparison and makes the approach highly scalable. During the training of our models, we want to leverage the citation information present in the patent corpora to represent similar documents as similar numerical representations. These similarities indicate textual and, more importantly, semantic similarity. A high similarity may be interpreted as indicating a flow of knowledge under certain conditions.

Our strategy is to train models on one corpus using the available text citations within the corpus and then fine-tune them using the text and citations of the other. That is, the model is either specific to patents and fine-tuned on publications or vice versa. Given the immense computational costs of training language models from scratch, it is not feasible to train a common language model joint for both corpus, patent and publications. This is why we have chosen to fine-tune existing models rather than train from scratch. Since it is not clear beforehand which model will perform best in the inter-corpus citation predictions, we train both variants and determine the better model via a model comparison ("horse-race").

The winning model should contribute to the stream of data-driven research in innovation economics and industrial organization. Furthermore, it will also have implications for research related to machine learning and text mining because it illustrates a valuable application of such techniques.

The model developed in our project may also support researchers and patent practitioners in their analytical tasks. The generated similarity measure is all the more valuable in the absence of citations or in the presence of strategic citations. The inverse of similarity could be interpreted as a measure of the novelty of the underlying invention. The provided model is also relevant for practitioners wishing to find most similar documents, e.g., as a citation recommender for the search of prior art. Furthermore, the numerical representations of the patents may help to depict in which technological areas patentable innovations cluster.

Chapter 2

Project Overview

The aim of this project is to overcome current limitations by leveraging state-of-the-art machine learning models.

We develop multiple document encoders derived from the BERT model family. During training, we utilize the citation graph to create specialized training data, following a method similar to SPECTER (Cohan et al., 2020) and SPECTER2 (Singh et al., 2022). By learning the notion of similar documents during training, the model generates similar embeddings for similar documents. Similarities between these numerical representations thus indicate high textual and semantic similarity. As a result, our models can represent patent documents and scientific publications as comparable document embeddings (dense vectors of fixed length). These numerical representations can then be used for downstream tasks such as classification, similarity search, and tracing knowledge flows.

Beforehand, it is not clear whether models trained on publications and then fine-tuned on patents best capture similarities between patents and scientific publications or vice versa. Hence, we use different training strategies to find suitable models for different use cases. In the project, we fine-tune three variations of these encoders and compare their performance.

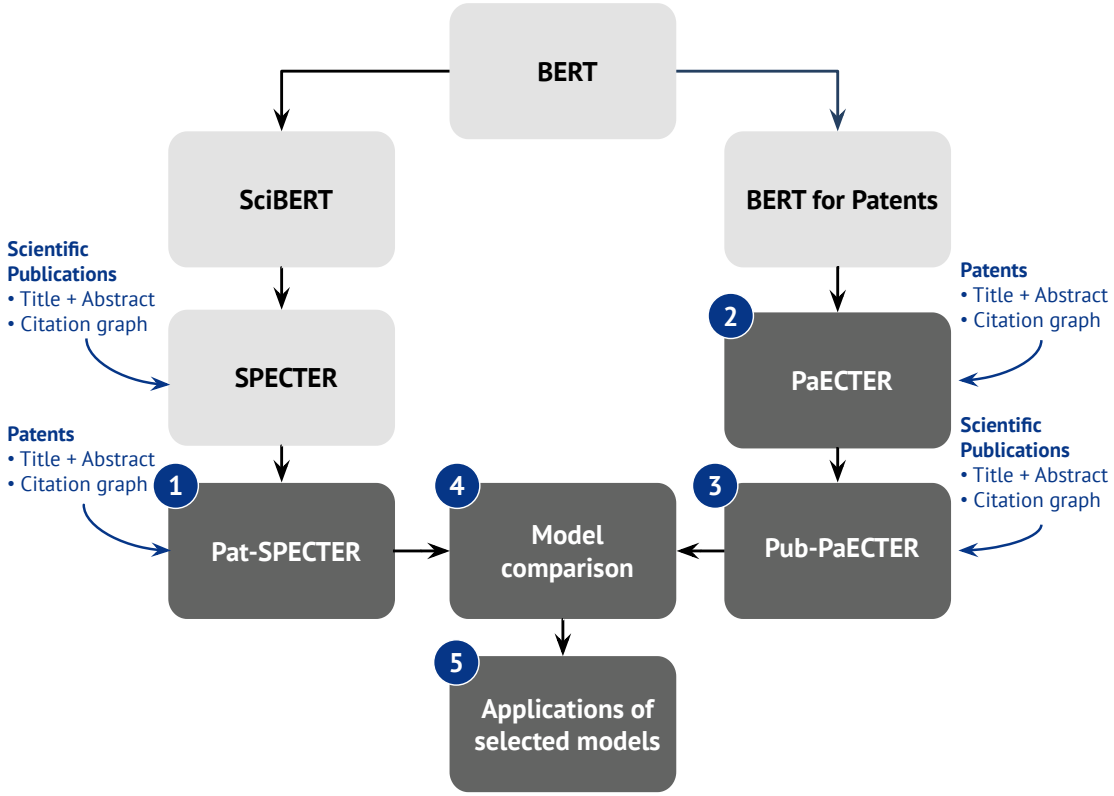
The first model (1) is the result of fine-tuning the existing SPECTER document similarity model for patents. This model is called Pat-SPECTER. We train the SPECTER model on our own curated dataset consisting of a selection of patents. We use citations to determine positive and negative training examples. In Figure 2.1, this is indicated as "citation graph".

The second model (2) is exclusively geared towards the patent domain and represents a stand-alone contribution of this project. Its name is PaECTER: "Patent-level Representation Learning using Citation-informed Transformers".[1] The basis for this model is BERT for Patents (Srebrovic & Yonamine, 2020) and we train it on the same patent dataset that we used for PAT-SPECTER. Although not foreseen at the onset of the project, we benchmark the PaECTER against the SEARCHFORMER model by Vowinckel and Hähnke (2023).

Our third model (3) is the result of fine-tuning the new PaECTER document similarity model on publications. We will call this model Pub-PaECTER. Here, we use the scientific publication dataset that was used to train SPECTER.

[1] We have released a pre-print paper on this model. Refer to Appendix A.3 for details. Note that parts of the content in Chapters 4 through 6 overlap with this publication.

Figure 2.1: Overview of the deliverables of the project



Notes: The dark grey parts are the deliverables we have completed. The light grey boxes are the models we build upon.

In a model comparison (4), we benchmark all of our models against each other and some existing models. The comparison uses different ranking metrics that are common in the machine learning domain. Since we are interested in tracing knowledge flow within and across different domains, we evaluate with three types of data pairs: patent to patent, publication to publication and patent to publications.

To make results more palpable, we apply the models to several real-world use cases (5). Finding similar documents in a small dataset is good for comparing the models. But we are especially interested in how a model performs when it has to search through thousands or even millions of documents.

We have chosen the following use cases:

1. Predicting patent paper pairs
2. Performance of Pat-SPECTER
3. The search for CRISPR/Cas9 related publications and patents, as well as those related to mRNA.

Chapters 3, 4, and 5 describe the data and the modeling of the three proposed models. Chapter 6 describes the horse race (model comparison). Chapter 7 describes the various applications projects.

Chapter 3

Data Sources

3.1 PATSTAT

PATSTAT (Patent Statistical Database) is a comprehensive database produced by the European Patent Office (EPO). It is designed to support research and analysis of worldwide patent data.

We use PATSTAT 2023 Spring version.[1]

3.2 OpenAlex

OpenAlex (Priem, Piwowar, & Orr, 2022) is a comprehensive, community-curated database of scholarly works, authors, institutions, and more. Developed by OurResearch, it is intended to be a modern replacement for the now-defunct Microsoft Academic Graph (MAG)[2].

OpenAlex provides information about scientific articles, in particular, titles and abstracts. Users access OpenAlex via a web API with multiple endpoints and a bulk download. It forms the foundation of several other datasets we use.

Likely due to its infancy, OpenAlex has several shortcomings. Notably, it misses a substantial amount of abstracts. In the May 2024 version, we found that as many as 45.5% of all works have no indexed abstract. These are usually papers not in English (or where the language of the article is not known). Another common cause is the lack of cooperation within the publishing house Springer, which refuses to share abstracts for all its journals (notably, the Nature family). If OpenAlex was not able to source to the abstract from other sources such as PubMed, it will be absent from the database. Missing abstracts prohibit proper representation of a document, which is why we cannot include them in our analyses. Thus we can only work with 54.5% of all works in OpenAlex.

Another shortcoming is the presence of documents that are not scientific articles, although published by journals. This includes front matters, back matters, table of contents, advertisements, etc. Since they are classified as works and cannot be detected, they tend to clutter the result sets of our analyses.

[1] See https://www.epo.org/searching-for-patents/business/patstat.html.

[2] See https://www.microsoft.com/en-us/research/project/microsoft-academic-graph

3.3 Semantic Scholar

Semantic Scholar[3] is a project of the Allen Institute for AI. It contains bibliographic information about scientific publications and it also has an API for retrieving the data. At the onset of the project, this was the only freely available publication dataset, thus forming the foundation of some early models dedicated to scientific publications.

Further, its AI-powered research tool helps researchers find relevant information quickly and efficiently by leveraging advanced machine learning and natural language processing techniques.

3.4 SPECTER

We use the `cite_prediction` dataset from Cohan et al. (2020). This data is used to train the `SPECTER` model. It is available from Huggingface.[4]

Data consists of approx. 820k triplets and is pre-split into a validation set of 676k training triplets and 144k validation triplets. The training records have one focal publication with one positive and one negative publication. There is a separate test set that has 996 triplets. These triplets have one focal publication, five cited publications, and 25 negative (random) publications. These are taken from Semantic Scholar and have the Corpus ID included.

3.5 Reliance on Science

For our evaluations, we use the *Reliance on Science* dataset provided by Marx and Fuegi (2020) and Marx and Fuegi (2022). In particular, we look at Patent Paper Pairs and Patent-Paper Citations.

The curated nature of this dataset offers several distinct advantages. The efforts have resulted in a rigorous data collection. The corrections, by employing both automated and manual processes, make for a high degree of reliability and applicability. Furthermore, the dataset's direct references to OpenAlex streamline the often cumbersome process of locating titles and abstracts for patent-referenced papers, a significant challenge due to the textual nature of patent references and the frequent lack of explicit citations.

The patent to paper citations (`_pcs_oa.csv`) is a patent data set with all the cited publications. The set has around 47 million records, with 7 million unique patents. It includes many authorities such as USPTO (approx. 34 million), EPO(5.7 million), WIPO (4.1 million), and many others. In our case, we use USPTO or EPO, depending on the use case. The dataset contains the patent publication number and the Open Alex IDs.

A patent-paper pair (PPP) is a unique combination where a scientific publication (the paper) is the foundation for the corresponding patented invention which is described in the patent document (the patent). The patent-paper-pairs data set created by Marx and co-authors (`_patent_paper_pairs.csv`) has around 548k pairs with 310k unique USPTO patents and 336k unique publications.

[3] https://www.semanticscholar.org

[4] See https://huggingface.co/datasets/allenai/scirepeval.

In some cases, the PPC and PPP overlap. Yet in most of the cases, there are no direct citations between the patent and the paper or vice versa. In the most obvious case, this would not be possible. If a patent cites a publication on the same invention, it would destroy the novelty. However, there are many publications with an earlier publication date than the patent. The reason is the one-year grace period. In these cases, the publication already exists but is not cited by the patent.

The data are available from Zenodo, which we downloaded in April 2024.

We construct the following datasets from the Reliance on Science data:

1. Cross Corpus dataset for the "horse race" in this chapter (paragraph 6.2)
2. Data for the patent paper pair predictions (paragraph 7.1)
3. Data for the performance applications (paragraph 7.2)
4. Data for the mRNA and CRISPR applications (paragraphs 7.3.2 and 7.3.1)

3.6 SEARCHFORMER

Vowinckel and Hähnke (2023) have provided us with a part of the dataset used to train their SEARCHFORMER. This selection omits patents that have not been published yet. The remaining dataset covers around 90% of the original dataset: 1,795 focal patents and a total of 1,154,557 pairs.

The dataset includes the rank predictions of the SEARCHFORMER model, which is not publicly available. This allows us to compare the results with our models.

3.7 Logic Mill

Logic Mill[5] (Erhardt et al., 2022) is a scalable and openly accessible software system that identifies semantically similar documents within either one domain-specific corpus or multi-domain corpora. It uses advanced Natural Language Processing techniques to generate numerical representations of documents.

The system focuses on scientific publications and patent documents. It contains all works from OpenAlex that contain an English abstract ($\approx$ 116 Million documents) and all patent documents issued by either EPO, USPTO, or WIPO present in DocDB (almost 34 million patent publications). In some occasions, patent documents lack abstracts in DocDB. The data are continuously updated.

Logic Mill uses ElasticSearch which allows storing dense vectors that nearest-neighbor search algorithms can use. The system uses approximate Nearest Neighbor searches (ANN), which trades precision for lower computational and resource burden. This setup finds the most similar documents with a very high probability for any query document within milliseconds.

We rely on Logic Mill to evaluate our models, and it is through Logic Mill that the embeddings generated by the winning model are made available to the world.

[5] https://logic-mill.net

Chapter 4

Training data

The quality of every machine learning task depends crucially on the quality of the training data. The training of document similarity models is no exception.

To train our models, we rely on two training datasets. The first dataset embodies patent relationships and was used to train the Pat-SPECTER and the PaECTER models. We created this dataset specifically for this project. The other dataset derives from Cohan et al. (2020) and codifies publication relationships. All datasets use the title and the abstract of the patent or the publication. To be precise, the models use only the first 512 tokens of the available text.

4.1 Patent to Patent

The first dataset involves patents related to other patents. The dataset will be used for the PaECTER and Pat-SPECTER models. Our dataset follows the lead of Cohan et al. (2020) adapted to the patent domain. It is available at Hugging Face[1] and described in detail in Ghosh et al. (2024).

In essence, Cohan et al. (2020) apply contrastive learning, whereby the machine is tasked to learn the difference between a related document and a focal document, relative to the difference between an unrelated document and the focal document. Thus the training data consists of triplets in the form (focal patent, related patent unrelated patent). Triplet loss learning is an effective means for a model to learn which documents are similar and which are not. Cohan et al. (2020) argue that one training instance is not enough, hence for each focal patent they use five triplets. The related patent is called "positive citation", while the unrelated patent is called "negative citation". Negative citations are further separated into "easy negative" and "hard negative".

4.1.1 Sampling

In total, we use 300,000 patent families from the EPO using PATSTAT. They must satisfy the following criteria:

1. They are either EP-direct or Euro-PCT applications.

[1] See https://huggingface.co/datasets/mpi-inno-comp/paecter_dataset.

2. They were filed between 1985[2] and 2022.

3. They have been assigned a Cooperative Patent Classification (CPC).

4. They have a minimum of two backward citations of category $X/Y/I$, or at least one backward citation of category $X/Y/I$ and one of category A.

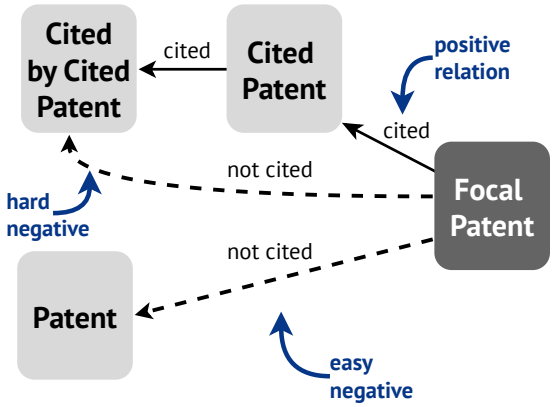
5. Their referenced patents collectively cite at least two other patents.

Given our focus on English language training, all patents, including focal, positive, and negative ones, must have at least an English abstract. In cases where this condition is not met, we strive to substitute it with the most fitting patent sibling from the same DOCDB family according to the following priority rule:

$$WO > US > GB > CA > AU > DE > CN > TW > KR > FR > JP$$

In total, 1,358,264 patent documents satisfy these criteria. From those, we randomly sample 300,000 documents. Finally, we split the training dataset into 85:15 ratio, such that all triplets for the same focal document remain together. The 15% fraction is reserved for validation purposes during the training process.

Figure 4.1: Positive, easy and hard negatives in patent training selection



4.1.2 Positive Citations

To generate a dataset of credible relatedness of patents, we rely on two unique features of the European Patent System. At the EPO, references in search reports are under the full control of examiners, mitigating any strategic bias typically associated with citations made by inventors or patent attorneys. Moreover, these references are classified with a unique emphasis on their contribution to the patent filed Webb et al. (2005), a feature not present in US patents. These categories are delineated by letters (see Table 4.1). Data on these categories are available through the search reports, which every patent application to the EPO receives.

Of these categories, X, Y, I and A are relevant for the training since the focal patents citing the patents of categories X, Y, and I have strong similarity with the

[2]We exclude patents filed prior to 1985 due to the ongoing transition to the EPO during that period.

Table 4.1: Citation Categories Provided by the EPO

Category	Meaning
X	Particularly relevant if taken alone - prejudicing novelty.
I	Particularly relevant if taken alone - prejudicing inventive step.
Y	Particularly relevant if combined with another document of the same category.
A	Documents defining the state of the art and not prejudicing novelty or inventive step.
O	Non-written disclosure.
P	Intermediate document.
T	Theory or principle underlying the invention.
E	Earlier patent application, but published after the filing date of the application searched (potentially conflicting patent documents).
D	Document cited in the application.
L	Document cited for other reasons.
&	Member of the same patent family, corresponding document

cited ones. The cited patent of kind X can even prejudice the novelty of the focal patents. The cited patent of kind A also reflects the state of the art for the focal patents.

Positive citations are thus selected exhaustively with replacement from the pool of patents whose parts are cited with X, Y, I, and A. We use 5 instances of these kinds of citations as the positive patents for a focal patent and construct 5 triplets for the focal patent in our sample. If A cited patents are available, the last two triplets sample exhaustively from this category.

In case of the unavailability of 5 such distinct positive patents, we sample with replacement out of all the available positive patents for the given focal patents. This strategy also reflects the necessity of choosing the focal patents having at least two backward citations of category X/Y or one backward citation of category X/Y and at least one backward citation of category A.

The sampling with replacement introduces duplicated positive citations at the level of the focal patent. Duplicates may also enter if they are cited by two included focal patents. The positive samples of the complete training set have 901,587 unique patent application IDs out of 1,500,000 rows. However, 60% of positive patents are used only once.

4.1.3 Negative Citations

Negative patents, representing dissimilarity with the focal patents, are selected with a nuanced approach. First, they must come from the universe of all patents ever

cited by any EPO patent, which was filed during the 1985-2022 period. Second, at least the abstract is available in English through the cited patent itself or its best substitute. Third, a CPC category must exist.

As the negative patents represent the patents that are not similar to the focal patents, the default choice could be the non-cited patents of the focal patents. However, following the approach of Cohan et al. (2020) with their SPECTER model, we further divide it into "easy negatives" and "hard negatives" for achieving better performance. Considering the patent-specific context, we take the following heuristic approach to define easy and hard negative:

- Easy Negatives: We consider all the uncited patents belonging to the same CPC class as the focal patent and also published within five years prior to the filing of the focal patent as the potential easy negative ones for the focal patent.
- Hard Negatives: We consider all the patents cited by the backward citations of the focal patent but not cited by the focal patent itself as the hard negatives for the focal patent. Although the focal patent and its hard negatives exhibit indirect citation relationship, but the lack of citation speaks in favor of subtle dissimilarity between them. We explicitly feed this dissimilarity signal to the model during training, so that the model treats them as dissimilar rather than as similar.

The selection process is illustrated in Figure 4.1. The number of all potential easy negatives is 6,484,216; there was no focal patent that did not meet the requirements of at least three easy negatives.

Taken 3 easy negatives and 2 hard negatives randomly out of the potential sets for a given focal patent together, we create 5 negative patent instances for the focal patent. Hence, combining these 5 negative instances with 5 positive instances for a given focal patent, we build 5 triplets for the said focal patent. Thus, replicating this strategy for all of the 150k focal patents, we construct a sample in triplet format. For the model's training and validation, we split this dataset using an 80:20 ratio as the training and validation set, respectively.

4.2 Publication to Publication

For training purposes, we use the SPECTER dataset (Paragraph 3.4), which includes titles and abstracts.

Some abstracts in the SPECTER dataset are missing, however. To maximize the dataset size, we attempt to retrieve these missing abstracts via OpenAlex. Via Semantic Scholar, we obtain external identifiers, such as MAG-ID and DOI. These identifiers allow us to search OpenAlex for the missing abstracts. Using this method, we can decrease the number of missing abstracts from almost 47 thousand to around 18 thousand. These are abstracts in any of the focal documents, positives or negatives of the training and validation data splits.

Chapter 5

Training of the Document Similarity Models

The aim of training a deep learning model for knowledge flow detection between documents is to teach the notion of similarity or dissimilarity between them using their textual information. As a consequence, the training helps the model to identify similar or dissimilar documents for an unseen document during inference on a large scale.

We train two models using the Patent to Patent dataset and another model using the Publication to Publication dataset. We build our models utilizing sentence transformers architecture (Reimers & Gurevych, 2019) and Hugging Face platform (Wolf et al., 2019).

During the training of each model, we follow the leads of Cohan et al. (2020), in that we apply the triplet loss learning strategy[1]. We minimize the following triplet margin loss function during training:

$$max\{(\|V_F - V_P\|_2 - \|V_F - V_N\|_2 + m), 0\} \tag{5.1}$$

where $\|.\|_2$ is L_2 norm distance and V_F, V_P, V_N are numerical representations for focal patent (P_F), positive patent (P_P), and negative patent (P_N) respectively. The margin variable m reflects that the corresponding positive document is at least m distance units closer to its focal document than the corresponding negative document. While training, we feed each triplet to the model, which optimizes the loss function, given in eq. 5.1, by comparing the numerical representations of all of three documents in that triplet.

What changes throughout the models is simply the underlying model as well as the training dataset.

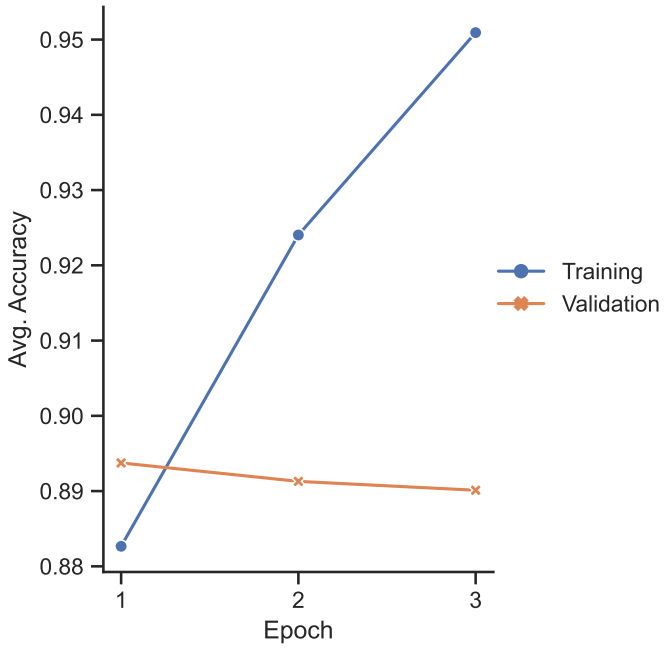
5.1 Training Pat-SPECTER

The first model is a fine-tuned version of the SPECTER2 model, with the aforementioned Patent to Patent citation dataset.

The underlying model is connected to an input module at the preceding layer and also connected to a document level embedding module at the following layer. Given a patent's title and abstract to the input module, it concatenates them to a single

[1]Vowinckel and Hähnke (2023) also used the same strategy

Figure 5.1: Average training and validation accuracy for the Pat-SPECTER model by epoch



string. Then it passes on concatenated text to the SPECTER and subsequently it goes to the document level embedding module, which convert the text to a 768 dimensional $[CLS]$ token.

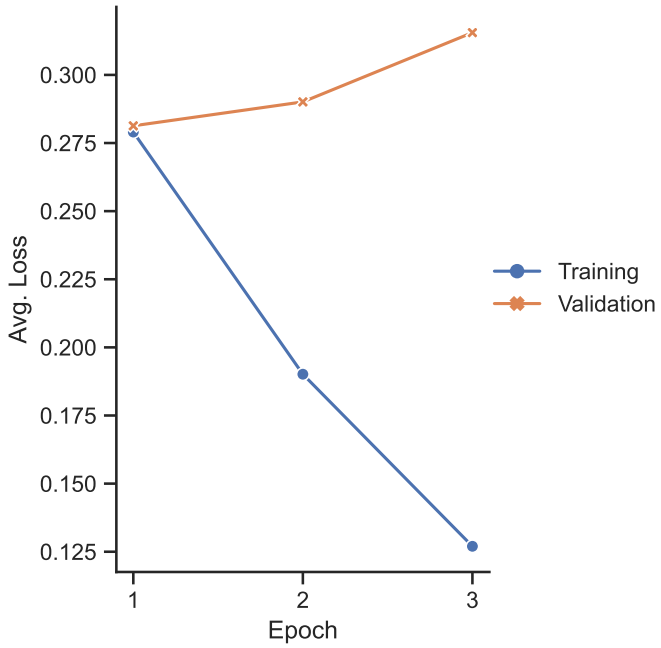
To train the model with multiple GPUs, we adapt the code base of sentence transformers package[2] and also utilize HuggingFace's Accelerator[3] module.

Following SPECTER and SEARCHFORMER models, we optimize a minimal set of hyperparameters as needed to improve validation accuracy. We train our model with a batch size of 8 on 4 NVIDIA A100-SXM4-40GB for training and achieving an effective batch size of 32. We use *Decoupled Weight Decay Regularization* (Loshchilov & Hutter, 2019) as an optimizer, a default option provided by sentence transformers (Reimers & Gurevych, 2019), along with learning rate $(LR) = 1e-5$ and triplet loss margin $(m) = 1$. We train the model for a single epoch, as training for additional epochs results in decreased accuracy and increased loss on the validation dataset (see Figure 5.1 and 5.2).

The computations have been performed on the high-performance computing (HPC) system Raven at the Max Planck Computing and Data Facility (MPCDF), where training this model for 3 epochs with evaluation steps 2000 took approximately 20 hours.

[2] https://www.sbert.net/index.html
[3] https://huggingface.co/docs/accelerate/package_reference/accelerator

Figure 5.2: Average training and validation loss for the Pat-SPECTER model by epoch



5.2 Training PaECTER

We train the PaECTER model using the same training data and training strategy. The primary differences are in the choice of the base model and hyperparameter tuning.

Specifically, PaECTER utilizes Google's BERT for Patents as its base model, a language model pre-trained on over 100 million patent documents. This use of a patent-specific language model, combined with further refinement over patent citations, enriches PaECTER with more knowledge in the patent domain.

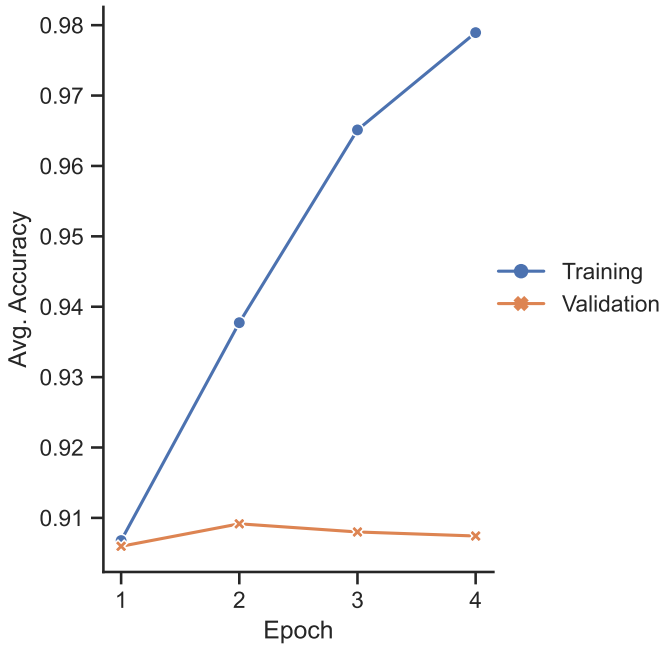
Post-tuning, we use a learning rate of $1e$-5, 10% of the training data for warmup steps, and triplet loss margin (m) of 1 for training in addition to the optimizer, *Decoupled Weight Decay Regularization* (Loshchilov & Hutter, 2019).

We train our model for 4 epochs, but the training does not improve after the second epoch (see Figure 5.3). Each training epoch spans approximately 20 hours, with validation occurring every 4000 steps.

We use a batch size of 4 (maximum GPU capacity) and gradient accumulation of 8 on four NVIDIA A100-SXM4-40GB GPUs for training and achieving an effective batch size of 128. To train the model with multiple GPUs, we have adapted the code base of the sentence transformers package slightly[4], and also utilized Hugging Face's Accelerate library.

[4]See the forked GitHub repository: https://github.com/max-planck-innovation-competition/sentence-transformers

Figure 5.3: Average training and validation accuracy for the PaECTER model by epoch



5.3 Training Pub-PaECTER

PaECTER, by design, may lack insights into scientific knowledge and understanding, as it is entirely focused on patent texts and citations among patents. However, it is important to incorporate scientific knowledge into the PaECTER model to enable tracing knowledge flows between scientific articles and patents. Therefore, we fine-tune the model in the scientific domain, resulting in the third model, Pub-PaECTER.

During fine-tuning, the model learns to differentiate between similar and dissimilar scientific articles. Since the SPECTER model was trained with the same objective, we use its corresponding dataset for fine-tuning PaECTER on scientific articles.

The fine-tuning process closely follows the training pipeline documented for Pat-SPECTER. Each step uses a triplet of scientific articles: a focal article and its respective positive and negative citations. We concatenate their titles and abstracts and feed them into the PaECTER model. Using mean pooling, the model generates 1024-dimensional document embeddings for these articles. These embeddings are then used to optimize the triplet loss function, similar to Pat-SPECTER (see Equation 5.1), resulting in the Pub-PaECTER model.

We fine-tune the model for 2 epochs with a batch size of 4 and gradient accumulation of 2 on 4 GPUs, leading to an effective batch size of 32. The learning rate (lr) and triplet margin (m) are set to $5e-6$ and 1, respectively, with validation occurring every 2000 steps. We use *Decoupled Weight Decay Regularization* (Loshchilov & Hutter, 2019) as the optimizer with 10,000 warm-up steps.

Chapter 6

Model Comparison

6.1 Methodology

Equipped with these three models (Pat-SPECTER, PaECTER, and Pub-PaECTER), our goal is to assess which one traces knowledge flows from science to technology best. For comparison, we include prior models with a similar focus, namely BERT, SciBERT, BERT for Patents, SPECTER, and SPECTER2. We assess the models' citation prediction capabilities in the following contexts:

1. from patents to scientific publications
2. within scientific publications
3. within patents

Additionally, we benchmark Pat-SPECTER and PaECTER against SEARCH-FORMER, developed by the European Patent Office (EPO), for potential prior art search.

First, we examine the *Rank First Relevant (RFR)*, which tells us the position of the first correct item in the model's list. A lower RFR is preferable, as it means the model is placing relevant information higher up in the rankings.

Next, we use *Mean Average Precision (MAP)*, a comprehensive metric that takes into account both the precision (the fraction of relevant items retrieved) and the recall (the fraction of relevant items retrieved out of all relevant items) at every position where a relevant item appears. A higher MAP score means the model is consistently retrieving and ranking relevant items well.

Finally, we look at the *Mean Reciprocal Rank at k=10 (MRR@10)*, which focuses on the top 10 results. It calculates the reciprocal of the rank of the first relevant item within those top 10 results. A higher MRR@10 indicates the model excels at placing relevant items near the top of the list, with a particular emphasis on the very first relevant result.

6.2 Cross-Corpus Comparison

The cross-corpus dataset contains pairs of patent and cited publications and is taken from the Reliance on Science PPC data (Marx & Fuegi, 2022). Our dataset consists of 1,000 randomly selected triplets, which is constructed the same way as the

SPECTER validation set: Every record contains a focal EPO patent[1], five cited publications (positives), and 25 publications (negatives). We sampled the negatives randomly from OpenAlex, provided their abstract is in English. PATSTAT provides patent titles and abstracts, while OpenAlex provides publication titles and abstracts.

The PPC dataset provides confidence scores based on how certain their algorithm and human judgment are that the reference and the scientific publication belong together. To be on the conservative side, we only selected observations from the PPC involving the highest confidence scores.

Table 6.1: Rank-Aware Evaluation of Different Models on a Cross-Corpus Dataset: From Patents to Publications

	Avg. RFR		MAP		MRR@10	
Model	CLS	Mean	CLS	Mean	CLS	Mean
BERT	2.52	1.29	46.76	79.05	69.86	91.52
SciBERT	2.48	1.37	49.23	71.14	71.75	90.55
BERT for Patents	1.20	1.10	78.51	85.99	93.14	96.97
SPECTER	1.08	1.13	**91.64**	86.57	97.72	96.23
SPECTER2	1.11	1.35	88.29	76.51	95.94	91.62
Our Models						
Pat-SPECTER	**1.05**	1.12	91.38	87.24	**98.04**	96.06
PaECTER	1.13	1.07	86.45	89.72	96.12	97.55
Pub-PaECTER	1.32	1.25	76.49	79.57	92.51	94.15

The results in Table 6.1 reveal that Pat-SPECTER performs well in terms of *avg. RFR* and *MRR@10* compared to other models. However, it comes out only second to SPECTER in terms of *MAP* metric, though the difference is not statistically significant (see Table S1 in Appendix).

Since *avg. RFR* and *MRR@10* evaluate the rank of the most similar publication for a given patent, Pat-SPECTER's well performance in these metrics suggests that it excels at early detection of the most relevant publications. In general, its slightly lower performance in *MAP*, which considers the ranks of all relevant publications, indicates that it may need to retrieve more publications to identify all relevant ones.

6.3 Publication to Publication

The second test case involves citations among publications. Our evaluation set is the test set of SPECTER with 996 triplets (see section 4.2). Every record contains a focal publication, five cited publications (positives), and 25 random publications (negatives).

[1]There are further restrictions: An abstract in English must exist, at least five different scientific articles are cited, and the kind code is one of A1, A2, or B1.

Table 6.2: Rank-Aware Evaluation of Different Models on SPECTER Test Dataset

	Avg. RFR		MAP		MRR@10	
Model	CLS	Mean	CLS	Mean	CLS	Mean
BERT	2.74	1.46	43.87	67.41	69.09	88.12
SciBERT	2.11	1.32	54.76	71.64	76.32	91.71
BERT for Patents	1.39	1.26	72.33	72.96	90.79	93.02
SPECTER	1.04	1.06	91.75	86.45	98.58	97.86
SPECTER2	**1.03**	1.08	**91.87**	84.14	**98.97**	97.30
Our Models						
Pat-SPECTER	1.07	1.09	86.73	82.92	97.27	96.77
PaECTER	1.18	1.10	79.53	83.66	94.86	96.43
PubPaECTER	1.33	1.30	67.56	68.74	89.15	90.52

Table 6.2 shows that Pub-PaECTER performs poorly in predicting citations within scientific articles, while SPECTER2 consistently exhibits the best performance. This suggests that Pat-SPECTER fails to capture the notion of similarity and dissimilarity between publications, though its performance is only slightly worse than SPECTER2.

6.4 Patent to Patent

The third test case involves Patent-to-Patent citations detailed in section 4.1. For the test dataset we have constructed 1000 triplets consisting of 1 focal patent (provided they cite a sufficient number of patents) with 5 positive samples and 25 negative samples. The negative samples are selected similarly to the training test data for the PaECTER model described in section 4.1.2. This means that these are not completely random but fall under the category of easy negatives.

The test dataset has no overlap with the training data. None of the focal, negative, or positive patent selections are present in the training dataset.

Since the dataset is smaller, only focal patents were selected where there were enough positive and negative examples. We do not sample with replacement.

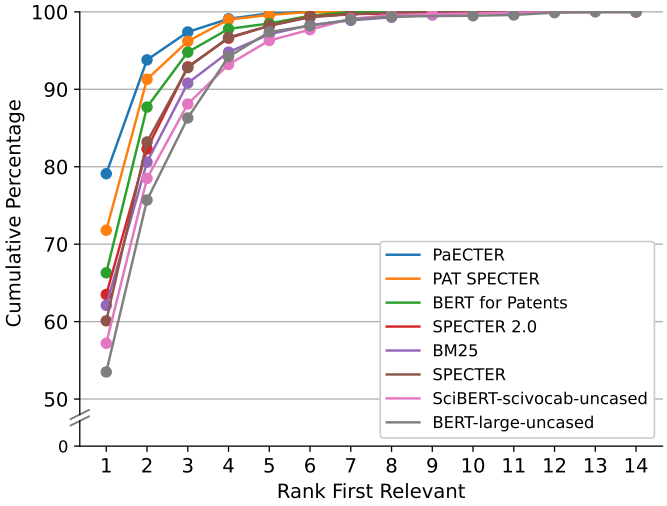
Table 6.3 and Figure 6.1 illustrate the results. We have included BM25 as a baseline method for prior art search. PaECTER turns out to be superior to other models in detecting potentially citable patents, predicting the first relevant patent at a rank of 1.32 on average. Pat-SPECTER and BERT for Patents are the second-best and third-best performers, respectively.

These findings provide three insights: First, domain-specific vocabulary enhances performance, as seen with BERT for Patents. Second, fine-tuning on patent citations improves prediction accuracy, as evident from Pat-SPECTER's performance over SPECTER2. Lastly, the combination of patent-specific vocabulary and fine-tuning over patent citations potentially make the PaECTER model superior to other models considered in this evaluation.

Table 6.3: Rank-aware Evaluation of Different Models on Patent Dataset

	Avg. RFR		MAP		MRR@10	
Model	CLS	Mean	CLS	Mean	CLS	Mean
BM25	1.79		51.84		76.61	
BERT	2.29	1.98	38.17	51.69	58.95	71.06
SciBERT	3.07	1.91	37.76	51.31	58.80	73.45
SPECTER	1.74	1.70	55.86	56.10	77.04	76.40
SPECTER2	1.79	1.68	56.12	56.96	75.47	77.96
BERT for Patents	2.44	1.55	44.44	60.32	67.14	80.49
Our Models						
Pat-SPECTER	1.42	1.55	63.42	60.63	84.07	80.75
PaECTER	1.32	**1.31**	66.91	**68.17**	88.03	**88.25**
Pub-PaECTER	1.96	1.84	50.94	52.55	72.23	74.21

Figure 6.1: Emprical Cumulative Density Function for the distribution of RFR scores across different models



Note: The distribution of *Rank First Relevant* (RFR) scores for PaECTER and other models used in comparison. All models use their best pooling strategy, as determined from Table 6.3.

6.5 Comparison with SEARCHFORMER

Lastly, we evaluate the models that we have trained, namely Pat-SPECTER and PaECTER, against the EPO's SEARCHFORMER model. We use the SEARCHFORMER data for comparison (Paragraph 3.6).

There are two key differences between our dataset and this dataset:

- This dataset uses patent claim texts as opposed to title and abstract that we have used for training and testing.
- The dataset is not as balanced as our datasets: the number of positive examples (cited) ranges from 1 to 14 and the number of negative examples (non-cited) ranges from 197 to 955.

The goal of this exercise is thus to suggest prior art given the first claim of a hypothetical patent application.

Table 6.4: Rank-aware evaluation of our models against SEARCHFORMER using the SEARCHFORMER dataset

Model	Avg. RFR	MAP	MRR@10
SEARCHFORMER	58.73	9.38	15.33
Our Models			
Pat-SPECTER	69.01	7.90	12.68
PaECTER	**49.66**	**11.13**	**17.93**

Table 6.4 portrays the results of this exercise. While Pat-SPECTER performed worse, PaECTER outperforms SEARCHFORMER in all metrics. On average, an examiner would have to inspect 50 documents suggested by PaECTER to find relevant prior art, which is 9 documents fewer than with SEARCHFORMER.

6.6 Conclusion

The evaluation in the publications-patents dataset suggests that Pat-SPECTER is the most suitable model. The runner-up model is SPECTER, but Pat-SPECTER statistically dominates SPECTER. The other evaluations suggest it to be a suitable model, though not the best.

On the other hand, PaECTER outperforms other models considered in this project for predicting citations among patents only. Pub-PaECTER, the fine-tuned version of PaECTER in the scientific domain, performs poorly in predicting citations among scientific articles.

The conducted tests have a notable limitation due to their lack of real-world applicability, as only a small number of documents were evaluated. In practical scenarios, the model must identify and rank the most relevant documents from an extensive universe of possibilities rather than being confined to a set of just 30 documents.

Chapter 7

Applications

In this chapter, we describe several applications we performed to evaluate the models in more realistic scenarios and use cases, helping readers understand the potential of the Pat-SPECTER model. We leverage our internal tool Logic Mill (Erhardt et al., 2022) for this, as it enables efficient nearest neighbor search using embeddings.

7.1 Prediction of Patent Paper Pairs

The goal of the first application is to predict Patent-Paper-Pairs (PPPs). These are instances where a paper itself is a patent.

Compared to metrics in the previous chapter, we demonstrate here that we are able to identify specific PPPs in large text corpora. Working with such large datasets is a more realistic scenario than working with small document samples, as in the previous chapter.

We use our trained transformer models to create embeddings and then identify textually similar candidates for a PPP. We aim to determine a similarity cut-off value that we can use as a threshold to separate pairings with high similarity from actual PPPs. Alternatively, we employ a logistic regression model to determine PPPs.

Since we need to differentiate between two uses of the term "models", we reserve the term "model" for the trained and existing transformer models (such as PaECTER and SPECTER), and we will use the term "separation method" for all approaches meant to separate PPPs from other pairings.

7.1.1 Methodology

In our dataset, we have three different classifications. The PPPs, the Non-patent literature (NPL), and patents paired with random publications. We try to identify the PPPs in the total set.

We employ the cosine similarity metric in combination with various separation methods to predict PPPs. This approach is essential as it reduces the decision variable into a single scalar metric. As proposed here, our method does not utilize supplementary information such as dates, confidence scores, author names, CPC classes, or concepts, which could potentially enhance the predictive power. The results should, therefore, be taken as a very conservative approach. This approach differs from previous methods, such as those used by Marx and Scharfmann (2024),

which incorporate additional metadata like author names, similarities, and institutional distance into their prediction model. Our decision is based on efficiency and scalability. We aim to achieve high-quality outcomes with minimal input data.

In practical applications, this additional metadata may serve as pre- or post-filtering criteria. For instance, author names could be used to pre-filter documents, refining the search space for identifying PPPs. Within this refined subset, a nearest-neighbor search would be conducted. Documents exceeding the similarity threshold would then be treated as PPPs. We choose not to apply such filtering in order to demonstrate the unconstrained performance of our algorithms.

We use the following separation methods:

- **Cut-off by F1**. The F1 score measures the accuracy of a machine learning model by combining precision and specificity of the model (Eq. 7.4). We aim to determine the cosine similarity value where the F1 score is maximized, indicating the best balance between precision and specificity for identifying PPPs.

- **Logistic regression with 2 categories** (PPP vs. the rest). A simple logistic regression model is used to predict if a pair is a PPP or not, based on the cosine similarity values.

- **Logistic regression with 3 categories** (PPP, NPL and random). This is a multiclass classification problem where we train a logistic regression model to predict three categories: PPP, non-patent literature (NPL), and random, based on the cosine similarity values. In essence, the logistic regression model learns to differentiate PPPs from other two categories based on the cosine similarity value.

- **Ordered logistic regression** with three categories. This approach takes a two-stage approach where the first stage considers random pairings vs. the joint group of PPPs and simple (non-PPP) NPL references, and then the joint group is split into NPL references and PPPs.

In the evaluation, we use the F1 score to determine the performance of the prediction of the patent-paper pairs (PPPs). The F1 score is chosen because it combines both precision and recall, providing a more comprehensive measure of the prediction accuracy.

Table 7.1 outlines the definitions of True Positive, False Positive, True Negative, and False Negative, which are used by these metrics.

In many cases, the metric *Accuracy* is used to determine how good the predictions are. With accuracy, we only measure the share of the correct predictions, i.e., the share of True Positives and True Negatives out of all predictions (Eq. 7.1).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7.1}$$

Since PPPs are only a small fraction of the total dataset, we have an imbalanced dataset. Imbalanced datasets are those where the classes or categories are not equally distributed, thus a model might achieve high accuracy by simply predicting the majority class. In such cases, relying solely on accuracy can be misleading.

Term	Definition
TP (True Positive)	Positive examples correctly identified by the model.
FP (False Positive)	Negative examples incorrectly identified as positive by the model.
TN (True Negative)	Negative examples correctly identified by the model.
FN (False Negative)	Positive examples incorrectly identified as negative by the model.

Table 7.1: Definition of True Positives, False Positives, True Negatives, and False Negatives

Given that only 10% of the dataset consists of PPPs, a model that always predicts non-PPP would yield an accuracy of as high as 90%. However, the model would fail to identify any of the PPPs within the dataset, which fails to serve our objective.

Unlike accuracy, the F1 metric provides a more comprehensive view of prediction accuracy, as it accounts for both precision and recall for imbalanced datasets.

The F1 score (Eq. 7.4) is the harmonic mean of precision (Eq. 7.2) and recall (Eq. 7.3), where precision is defined as the share of True Positives (TP) among all predicted positives and recall is the ratio of positive instances that are correctly detected by the classifier.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7.2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7.3}$$

$$\text{F1 Score} = \frac{2}{\text{Recall}^{-1} + \text{Precision}^{-1}} = \frac{2TP}{2TP + FP + FN} \tag{7.4}$$

Regardless of the separation method, with 2 or 3 catogories, we use the PPPs as positives and the rest as negatives. That means that we combine NPL and random for the performance metrics in all cases.

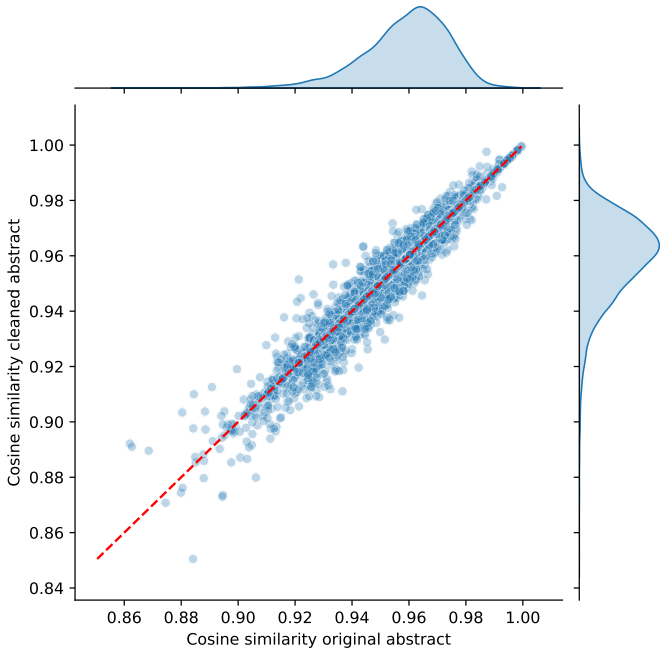
7.1.2 Data

We sampled the datasets patent-paper-pairs (PPP) and non-patent literature citations (NPL), and generated random patent and publication pairs (Random). The data is constructed from the Reliance on Science datasets (see paragraph 3.5) For the random data set, we add random publications to the patents of PPP and NPL. We use the random API endpoint of Open Alex API. Publications must have a title and an abstract in English.

All files are pre-processed in the following way. First, we retrieve the patent title and abstract from PATSTAT. Via Open Alex, we retrieve the title and abstract for the publications.

The next step is the cleaning of the Open Alex abstracts. Some abstracts contain structure elements (such as headings) or contain copyright statements. These elements are typical for certain journals. Since the abstracts appear to be more similar, it may introduce an unwanted bias. We remove these elements with our abstract cleaning tool.

Figure 7.1: Differences in cosine similarities before and after abstract cleaning. The dotted red line indicates equal values.



Upfront it was unclear how big consequences the abstract cleaning would be. To investigate the actual effect, we took a sample of 50,000 patent paper pairs. Our statistics show that in around 34% of the cases, abstracts are changed in some way. However, the overall effect on the cosine similarities between the patent and the paper is very small. We found a Cohen's D value of $d = 0.04$ and a Pearson correlation of $r = 0.994$. (Figure 7.1)

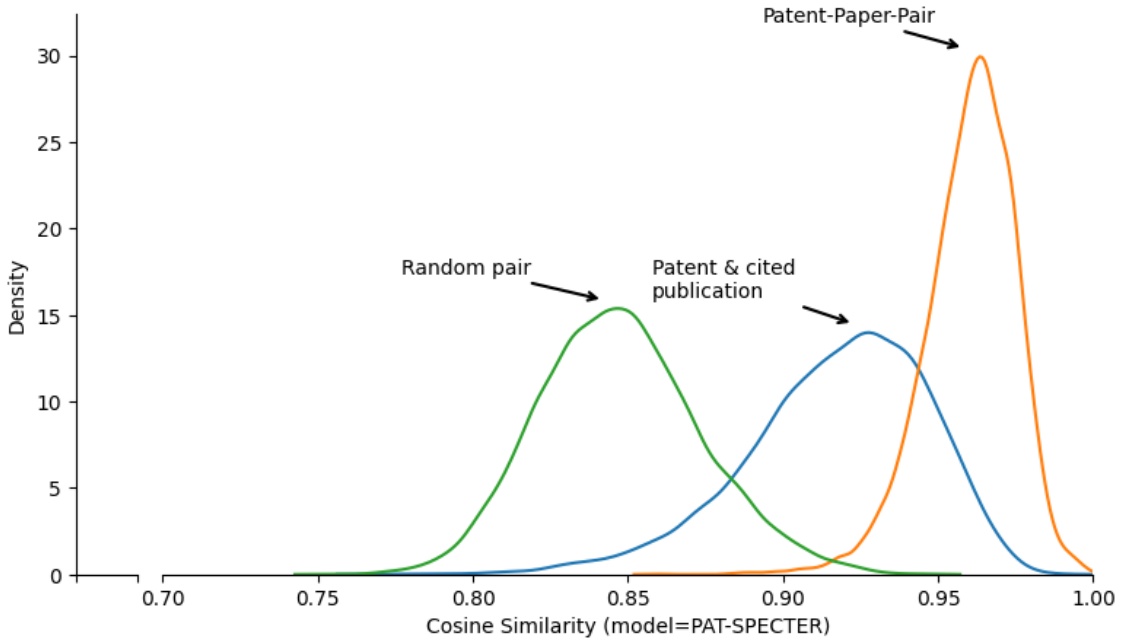
For our analysis, we utilize a 10,000-document sample. An unrestricted sample is taken for Patent-Paper Pairs (PPP). For Non-Patent Literature (NPL), we specifically select patents from the United States Patent and Trademark Office (USPTO). This approach ensures a consistent citation strategy, as citation rules may vary across different jurisdictions. Consequently, the resulting datasets contain fewer than 10,000 records due to the absence of some abstracts in the OpenAlex database. These records are subsequently excluded from the analysis.

For the random dataset, we use the PPP and NPL patent selection and add the random publications. This leads to a sample that is almost twice as large.

The next step is the creation of dense numerical representations (embeddings). We use our models PAT-SPECTER and PaECTER. SPECTER and SPECTER2-base are used as baselines. The source for all models is the Huggingface website [1]. To create the embedding, we use the title and abstract and concatenate these with the `sep_token` from Autotokenizer. This is a special token separating two different sentences in the same input (used by BERT and its descendants) and defaults to `[SEP]`. PaECTER creates embedding of size 1024. The other models generate embeddings with 768 dimensions.

[1] https://huggingface.co/

Figure 7.2: KDE of cosine distribution PAT-SPECTER



After this, we calculate the cosine similarity between the embeddings of the pairs. For this (and some other tasks) we use Scikit-learn (Pedregosa et al., 2011).

We use an 80/20 split for training and testing.

Additionally, we need to adjust the fraction of PPPs compared to the other pair types. Patent-paper-pairs are not so common in the real world; hence, we adjust the fraction of PPPs. If the fraction is too high, the results will be unrealistic compared to the real world. If the fraction is too low, the modeling results will be inaccurate since we have too few positives. In the latter case, whether a single item is found will make a big difference in the results. In our setup, we have chosen for 10% PPPs and 90% random and NPL. An iterative approach is used to make sure the resulting data set sizes are still around the 80/20 split. We ensure that the training and test sets do not have any overlapping papers or patents. We end up with a split of 13,474 pairs for training and 3,189 for testing.

As a final step, we add normalized values of the cosine similarities. The normalized values have a mean of 0 and a standard deviation of 1. First, we normalize the training set. After that, the same scaling is applied to the test dataset. We used this in the beginning, but later, we found out that it was not necessary.

7.1.3 Results

The results show that there is a separation between Random, NPL, and PPP pairs (KDE plots Figure 7.3), but the ability to separate them is limited. The reason is that the distributions overlap, and we only use one number as a threshold to separate the datasets. (Figure 7.3. The results seem to be consistent over the different models and changes in selection don't have a big difference within a specific model.

The models (such as SPECTER or PaECTER) and the separation methods (such as logistic regression or cutoff by F1) perform differently depending on the

Figure 7.3: KDE of cosine distributions TEST and TRAIN non-normalized data

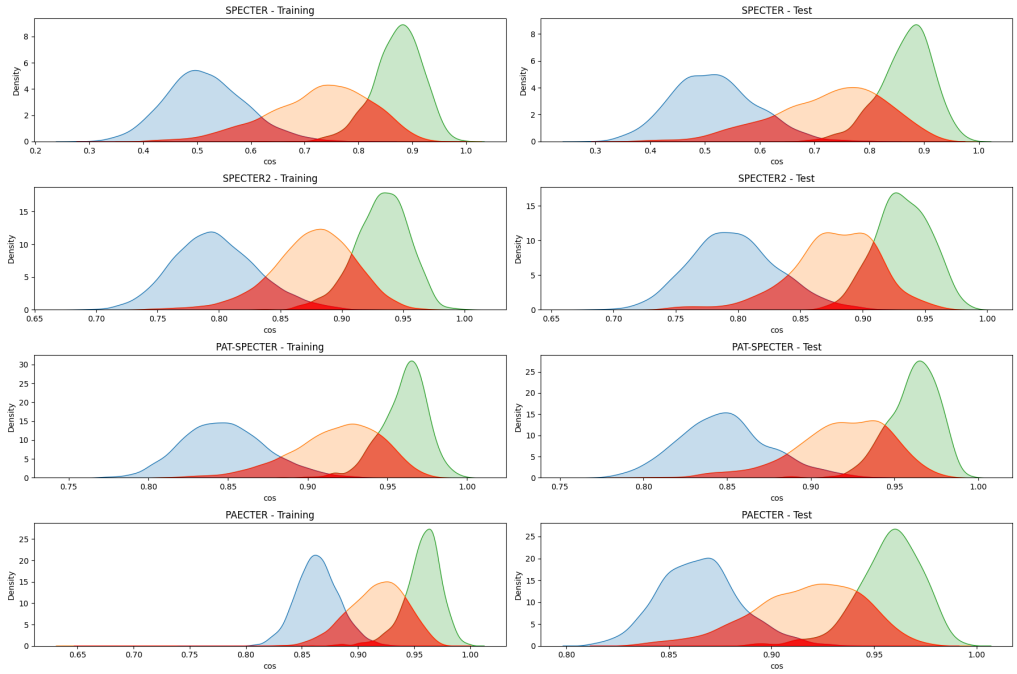


Table 7.2: Optimal F1 cut-off values for prediction of PPPs

Model	Optimal Cut-off value
PaECTER	0.948
Pat-SPECTER	0.950
SPECTER	0.835
SPECTER2	0.914

implementation details. (Separation token, outliers in the set, fraction of PPP in the datasets), but is fairly stable within a specific experiment. The order of magnitude in all cases seems to remain the same, with F1 scores roughly between 0.70 and 0.74) When optimizing the model parameters (by doing a grid search over the penalty applied (None, *L1*, *L2*) and the type of solver), there is hardly any difference between the separation methods. Normalizing the data or the use of the unscaled data does not matter anymore. The results are summarized in the heatmap (Figure 7.4). This means that we can use the simplest method possible to get the optimal results. For Pat-SPECTER, we can use a cut-off value of 0.95 as the optimal threshold value for the F1 score (Figure 7.5). We do not need to run our results through an additional regression model for the same result. The F1 cutoff values for all models can be found in Table 7.2.

PaECTER seems to be more sensitive to outliers in the dataset. The bump around $cos = 0.74$ in Figure 7.3 in the PaECTER model is caused by a few abstracts that are not in English or have some other irregularities. In the other models, such as Pat-SPECTER, these values disappear in the regular bell curves (Figure 7.2).

Figure 7.4: Heatmap of the F1 Scores (with original cosine values)

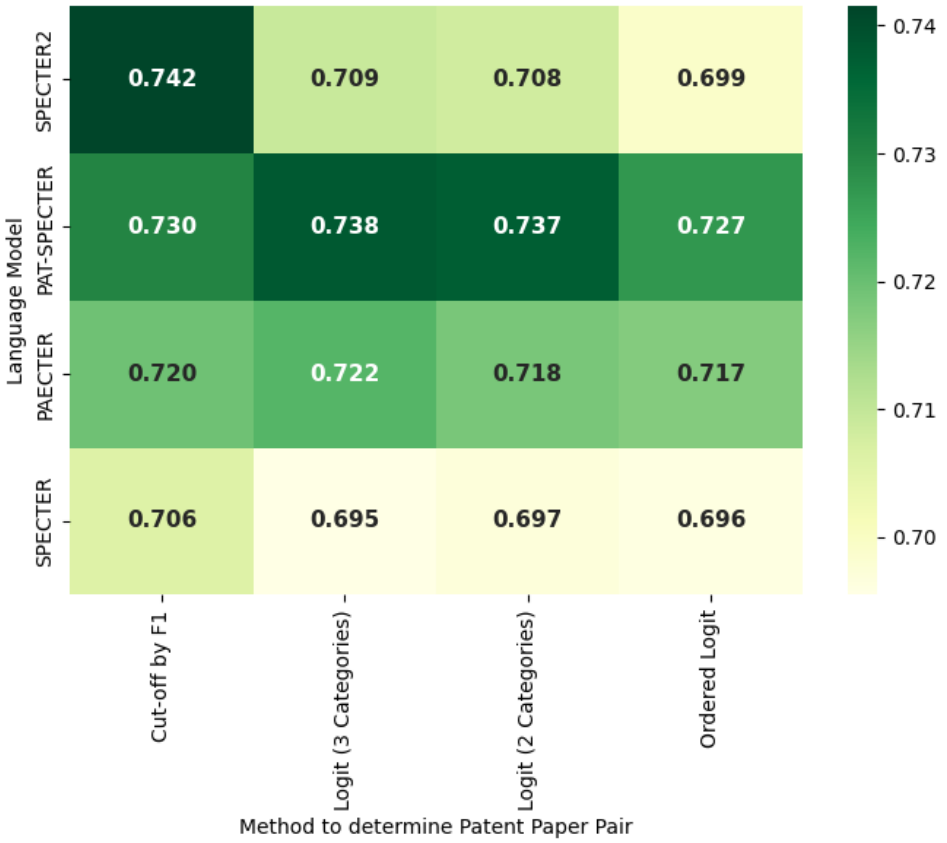
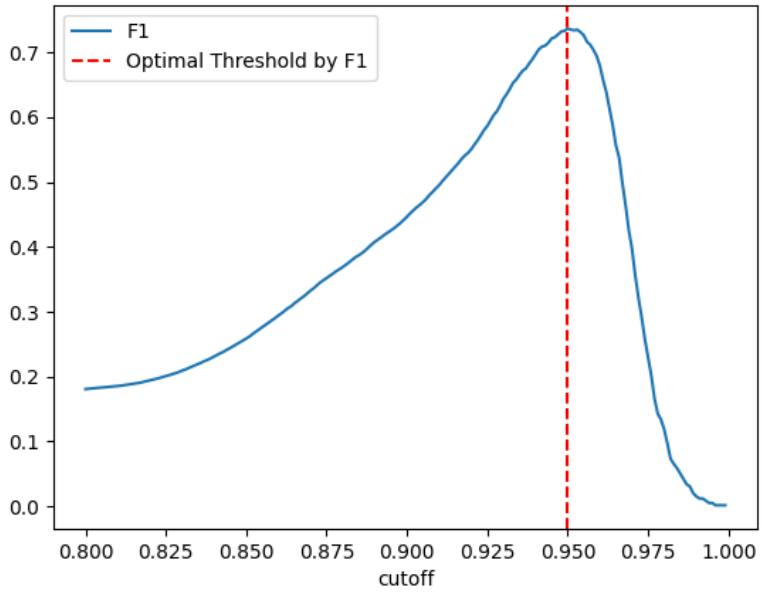


Figure 7.5: Optimal PPP classifier by F1 score for Pat-SPECTER



7.1.4 Conclusion

The models are capable of picking up differences between the text corpora. The results show that the different models lead to similar results, and the separation methods do not outperform each other.

However, two remarks are important. First, this is only one particular application where we use the model, and the PPPs are not always similar in text. If the text is not similar, the model will not find it. Second, other applications might benefit from a specialized patent or a cross-corpus model. The benchmarks in the horse race show that different models pick up different patterns, hence they might be advantageous in differentiated settings.

The F1 cutoff value can be directly used as a guide as to which documents are to be selected. In this case, the (Approximate) Nearest Neighbor search operates on one similarity metric, thus allowing for considerable speed, even while searching through millions of documents. After the initial selection, one could build a more informed machine learning model that uses more metadata of the documents (such as authors, dates, CPC-classes, and concepts). This model can be more precise and only needs to be applied to a very limited set of documents.

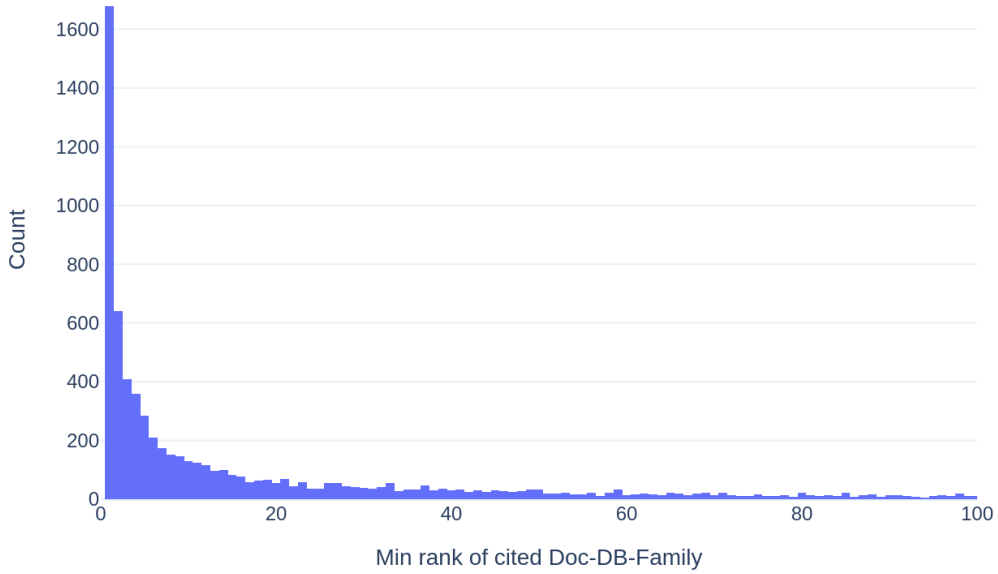
For the following applications, we have chosen to proceed with the Pat-SPECTER model, despite similar results across models. We do so for several reasons. First, our project aims to develop a cross-corpus model, and Pat-SPECTER ranked highest in the previous chapter. SPECTER, on the other hand, underperformed in this exercise and also does not show a significant difference in performance when compared to Pat-SPECTER (see Table S1). SPECTER will, therefore, not be considered further. Second, in this cross-corpus exercise, Pat-SPECTER outperformed other models in 3 out of 4 cases. Third, the PaECTER model, while larger and having larger embeddings, requires more resources to operate. Pat-SPECTER, being smaller, allows for more efficient searching through large datasets. Lastly, although SPECTER2 performed well in one case, it was outperformed by both PaECTER and Pat-SPECTER in the other three cases.

7.2 Prior Art Search

This evaluation is analogous to the prior art search by patent offices. Prior art search is a crucial task in the patent examination process. It is designed to uncover existing patents, publications, and other relevant articles that might challenge the novelty of the patent application under prosecution.

It is important to acknowledge that the experimental results provide conservative estimates of performance. This conservatism is primarily due to limitations in the data. As described earlier, the most significant issue encountered was the absence of abstracts in both the OpenAlex dataset and the DocDB. Since the model relies on both titles and abstracts to generate meaningful embeddings, many documents are excluded from our database. Consequently, the model's performance may appear lower than its actual potential.

Figure 7.6: Distribution of the first ranked patent cited by the reference patent (



Notes: Figure uses the most similar patent per DocDB family member cited by the reference patent (N=6,912 of 10,000).

7.2.1 Patents to Patents

The first task is to identify patent-to-patent citations for 10,000 randomly selected patents granted by the EPO whose family consists of at least 10 documents. The goal is to predict the patent citations to prior patents on a patent family level. We then search our DocDB index for 1,000 approximated nearest neighbors present at the time of the earliest filing date and retrieve the closest patent documents from the EPO, the USPTO, and the WIPO. The approximate nearest neighbor results are then aggregated on a patent family basis and re-ranked.

Of 10,000 test cases, we were able to identify at least one correct family citation within the first 1,000 nearest neighbors in 8,641 cases ($\approx$ 86%). In 6,912 cases, at least one family citation is among the 100 nearest neighbors ($\approx$ 69%). Indeed, most of the first family citations are within the first 20 results (cf. Figure 7.6).

Table 7.3 shows the Mean Reciprocal Rank (MRR) and the Mean Average Precision (MAP) for various cutoff values $k \in \{5, 10, 20, 50, 100\}$. For $k = 5$, the MRR is equal to 0.264 and the MAP equals 0.06.

The results presented in Table 7.3 demonstrate the Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP) for patent-to-patent citations at various cut-off levels (k). As k increases from 5 to 100, both MRR and MAP values show an upward trend. Specifically, MRR starts at 26.3874 for k=5 and rises to 28.8638 for k=100, while MAP begins at 6.2466 for k=5 and reaches 8.2916 for k=100. This indicates that the model's retrieval performance improves with larger k values, suggesting that higher recall levels yield better precision and ranking quality.

Interpreting MRR and MAP scores is challenging without another system for comparison, making it difficult to determine what constitutes good or bad values.

Table 7.3: MRR and MAP for patent-to-patent citations

k	MRR	MAP
5	26.3874	6.2466
10	27.6368	6.5272
20	28.3207	7.0677
50	28.7384	7.8185
100	28.8638	8.2916

Given the results of Table 6.1 where Pat-SPECTER has an $MRR@10 > 95$ in a controlled sample of 30 documents, the performance in this real-world scenario where millions of documents have to be considered shows promising results with an $MRR@10 > 27$. This indicates that, on average, users find in $6,912/10,000 \approx 69\%$ of the cases a relevant item at position $1/0.27 \approx 4$ given a universe of 30+ million patent documents.

7.2.2 Patents to Publications

Our second task focuses on identifying relevant publications that a patent should cite. To establish a ground truth for these citations, we leverage the Reliance on Science dataset by Marx and Fuegi (2020, 2022), which links USPTO patent IDs to their corresponding citations within the OpenAlex database. Each patent-paper citation in this dataset is assigned a confidence score ranging from 1 to 10. For our analysis, we exclusively consider citations with the highest confidence score of 10. The search parameters employed for this task remain consistent with those used in the previous analysis.

As stated before, there are instances where we are not able to obtain an embedding. A common reason is the absence of English abstracts. Another reason is that OpenAlex has changed IDs of some papers, while the Reliance on Science dataset contains their old IDs.

Of initially 10,000 randomly selected samples, we found at least one cited publication among the top 100 results in 3,740 cases ($\approx 37\%$). As seen in Figure 7.7, the distribution is not as steep as in the patent-to-patent case. Table 7.4 presents the MRR and MAP for various $k \in \{5, 10, 20, 50, 100\}$.

The results presented in Table 7.4 indicate that, in $3,740/10,000 \approx 37\%$ of the cases, on average, users find a relevant item at position $1/0.16 \approx 6$ given a universe of 100+ million publication documents.

To understand the omission of a significant share of citations, we manually examined anecdotal cases where matches were not found within our approximate nearest neighbor (ANN) results. Our analysis reveals that many of these citations were applicant-added rather than examiner-cited. These applicant-added citations often appear in the patent description, a context not considered during model training nor encoded in our system. Additionally, a lack of semantic similarity between the cited publication and the focal patent's abstract makes associating the two challenging even for humans. Another critical aspect is the incompleteness of the underlying OpenAlex database. For instance, a significant number of articles, such as those

Figure 7.7: Distribution of the first ranked publication cited by the reference patent (N=3,740 of 10,000)

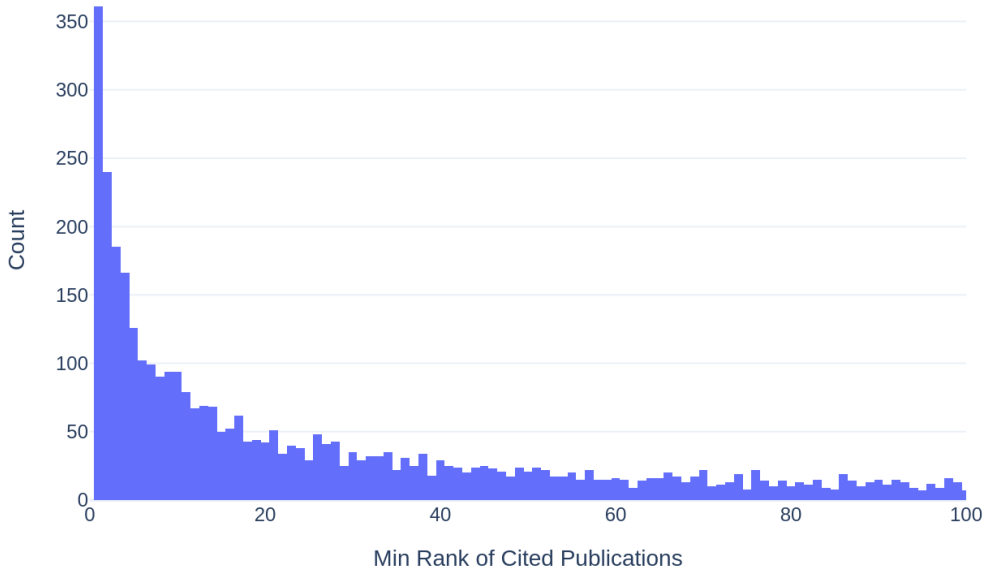


Table 7.4: MRR and MAP for patent-to-publication citations

k	MRR	MAP
5	16.2932	5.2162
10	17.9573	6.6367
20	19.0259	8.0288
50	19.7886	9.3623
100	20.0598	10.0609

Notes: Table shows the Mean Reciprocal Rank (MRR) and Mean Average Rank (MAP) at different rank cutoffs k.

published in the Nature journal, lack available abstracts. This absence of abstracts presents a considerable challenge, as the abstract is essential for understanding the core content and relevance of the scientific work. Without this crucial information, the system's ability to accurately match patents with relevant scientific literature is compromised. This leads to potential gaps in citation prediction and knowledge flow analysis.

7.2.3 Patent Paper Pairs

Our third task is to identify papers that are very similar or identical to the patent, using the patent-paper-pair (PPP) dataset from the Reliance on Science project as a ground truth (Marx & Scharfmann, 2024). The goal is to evaluate the performance of our model to predict PPPs.

The PPP dataset includes 548,315 PPPs, constructed using a trained predictive model to link US patents to the scientific articles in OpenAlex. Based on the model's predictions, each pair is assigned to one of four categories reflecting the confidence score: very high for category 4 (prediction greater than 0.99), high for category 3 (prediction between 0.90 and 0.99), medium for category 2 (prediction between 0.80 and 0.90), and low for category 1 (prediction between 0.70 and 0.80). For each unique US patent in the PPP dataset, we retrieve its embedding from our DocDB index and then search for the corresponding 1,000 approximated nearest neighbors in our OpenAlex index. Furthermore, we calculate the rank of the paired papers for each focal patent in the PPP dataset when they appear within the top 1000 results.

In the PPP dataset, we were able to match 342,252 pairs ($\approx$ 62%) within the first 1,000 nearest neighbors, where the missing instances are mainly due to the imperfect datasets as discussed in section 7.2. For the matched PPPs, about 65% have a rank between 1 and 100, and nearly 90% have a rank between 1 and 500. The cumulative distribution of the rank is skewed towards lower values and gradually levels off as the rank increases (Figure 7.8).

Further, we investigated the distribution of ranks from the matched PPPs across different categories, highlighting similarities and differences in the distribution patterns within each category by violin and box plots in Figure 7.9. Across all categories, the PPP ranks are predominantly low, as shown by the high density of data points at lower ranks. Categories 1 and 2 show more variability in ranks, with Category 1 having the most dispersed distribution. In contrast, Categories 3 and 4 have a higher concentration of lower-ranked PPPs and less variability, with Category 4 being the most concentrated. Hence, our predictions are more confident in categories 3 and 4 than in categories 1 and 2. This is consistent with the definition of the confidence level in Marx and Scharfmann (2024).

Figure 7.8: Empirical Cumulative Distribution of the rank for the matched PPPs

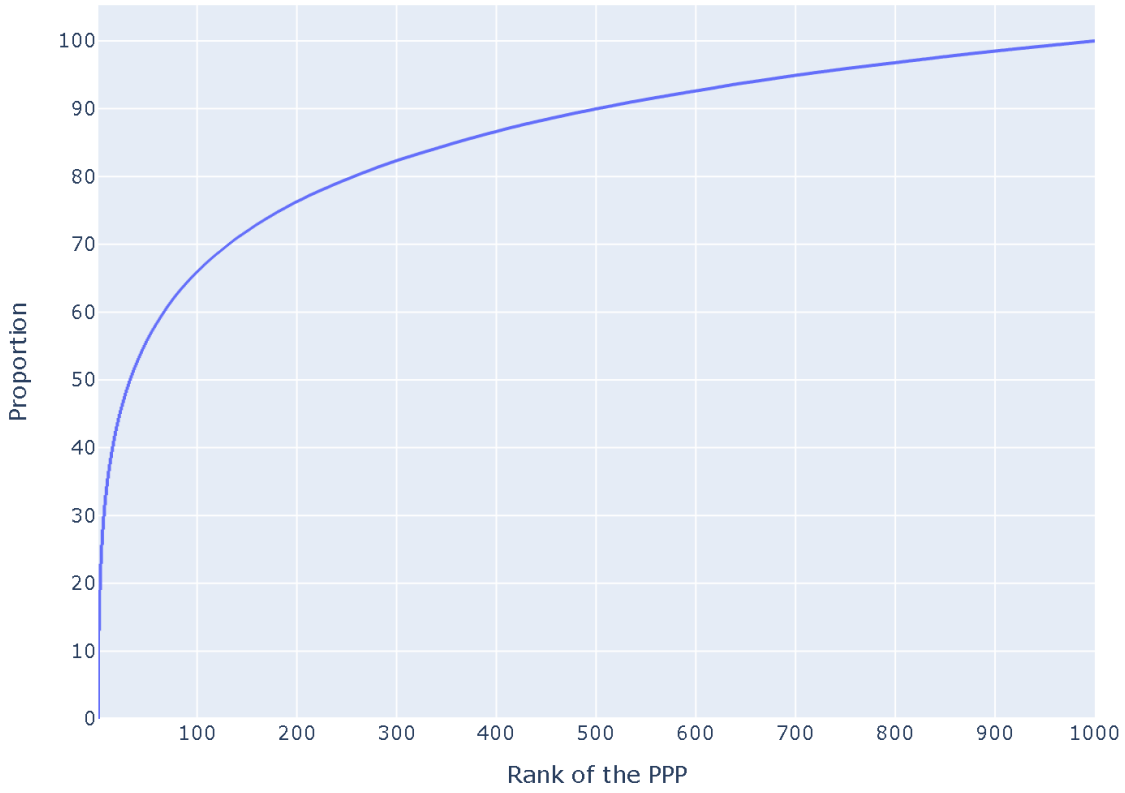
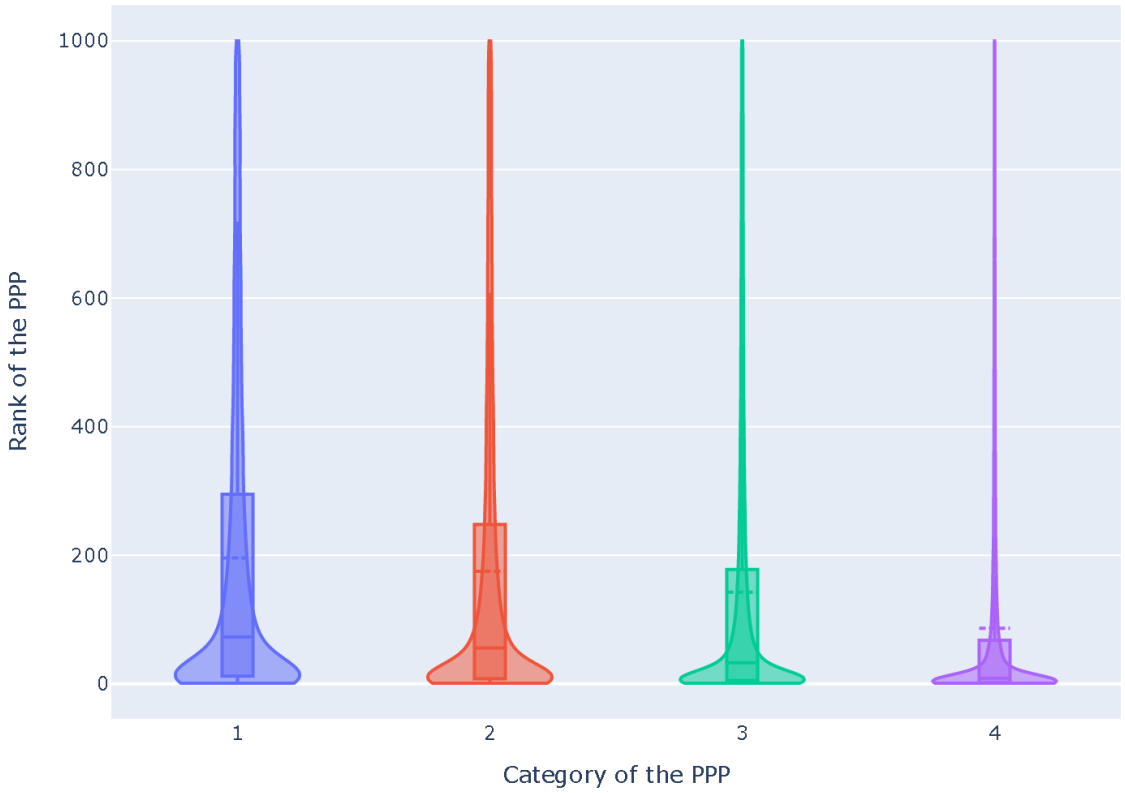


Figure 7.9: Distribution of the rank for the matched PPPs in different categories



Notes: Categories relate to algorithmic confidence of the Reliance of Science dataset, with category 4 being the highest confidence.

7.3 Patent Landscaping Case Studies

To make the analysis more palpable, we conduct two patent landscaping exercises. We have chosen the CRISPR/Cas9 technology and the mRNA technology, as both were recently awarded Nobel prizes and sparked a lot of technological development.

In both cases, we source seed documents from the Nobel decisions and target patents from published patent landscaping studies. It should be noted that none of these sets represent 'gold standards' in the sense that these and only these publications or patents are related to the technology.

We then ask, how often do we find target patents among the documents most similar to the scientific publications, and which others do we find? The goal is to demonstrate how far Pat-SPECTER (in conjunction with Logic Mill) can speed up patent landscaping studies.

7.3.1 CRISPR/Cas9

In 2020, Emmanuelle Charpentier and Jennifer A. Doudna were jointly awarded the Nobel Prize in Chemistry for the invention of CRISPR/Cas9. It has been dubbed a 'gene scissor' as it allows targeted manipulation of genome editing.

We retrieve a list of publications related to CRISPR/Cas9 from the Nobel committees' Advanced information, which provides extensive scientific background and relevant publications (Gustafsson, 2020). From this, we determine four key publications by the Nobel laureates (see Table S2).

The set of related patents stems from Martin-Laffon, Kuntz, and Ricroch (2019), who landscape CRISPR patents whose priority date lies before 31 December 2017. They perform iterative queries on the three databases, Orbit Intelligence, PatentPulse, and Patent Lens, with several manual steps in between. In their Supplementary Table 1, the authors identify 2,019 relevant patent families. However, many do not include documents published by the EPO, the USPTO, or the WIPO, reducing the number of target patent families to 1,188.

First, we search for the 5,000 closest patent documents for each of the four seed papers in Logic Mill. We restrict the search to patent families with priority date before 31 December 2017. This yields 8,211 distinct patent documents from 7,163 DocDB families.

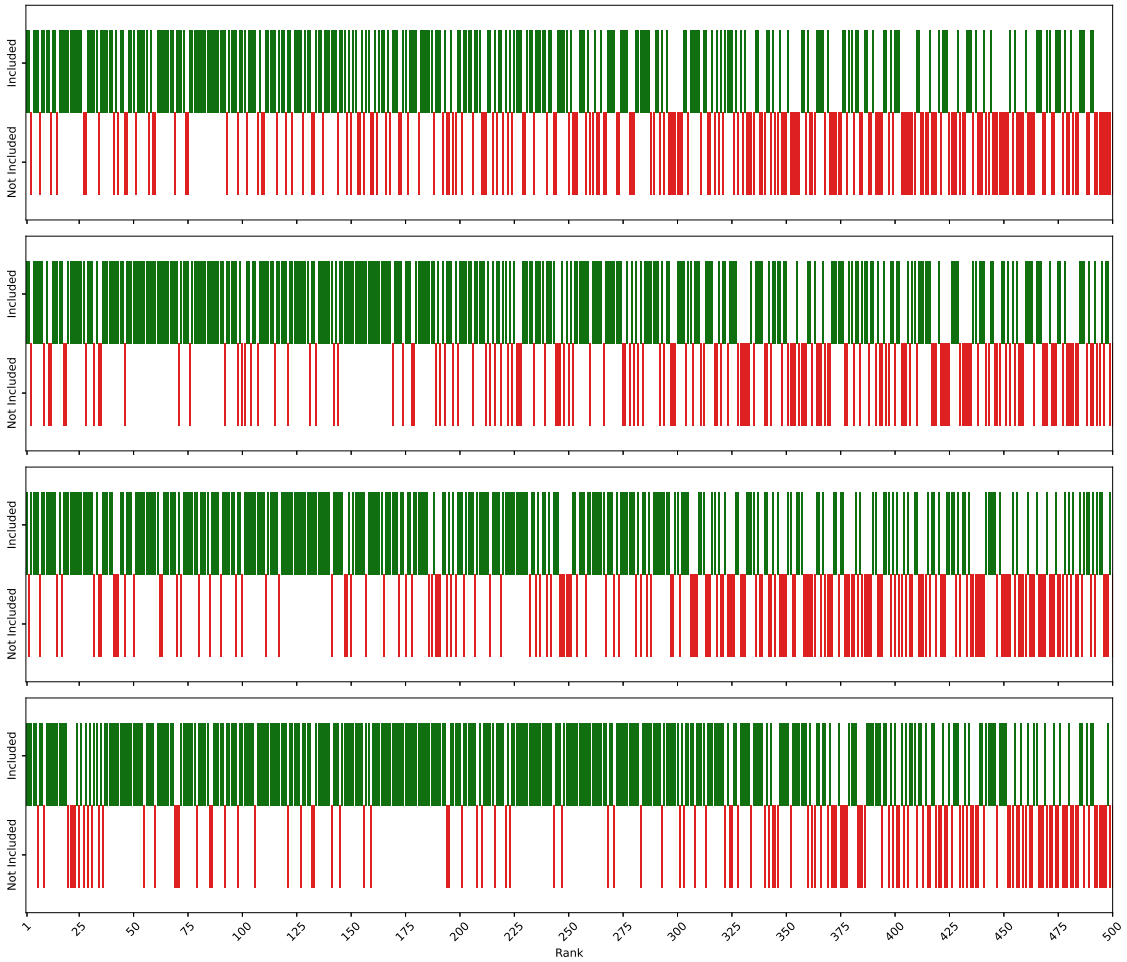
681 of these patent families are within the target set (57%). For each matched family, we keep the most similar document, i.e. the one with the best rank.

The matching quality overall is high. For all four papers, it holds the target patents are ranked highly. The median rank of related patents fluctuates between 398 and 442. The average rank fluctuates between 783 and 860. For comparison, if all 681 matched target patents were ranked sequentially (the best case possible), the median and mean would equal 341.

Figure 7.10 visualizes these green and red bars, where each bar indicates a patent family ordered by similarity, starting with the most similar patent family. For each of the seed documents, the most similar families are depicted in green, meaning they are in the target set.

An evaluation needs to take into account similar patents not in the target set, as well as target patents not among the most similar documents. It is important to note that the target list is not a gold standard but instead the result of a semi-

Figure 7.10: Status of 500 most similar patent families, CRISPR/Cas9 landscaping



Notes: Figures show which of the 500 most similar patent families are included in the database for each of the four source publications. Green bars indicate the x-th most similar patent family is in the target set. Red means they are not in the target set.

manual search. Upon our own manual inspection, several consistently highly ranked patents deem natural candidates to be included in the analysis. Vice versa, there are patents that are perhaps mistakenly included in the patent analysis and/or the number is not correctly stated in the supplementary information.

7.3.2 mRNA Vaccine

Another tangible application is the search for documents related to mRNA. In 2023, Katalin Karikó and Drew Weissman were jointly awarded the Nobel Prize in Medicine for their work on the development of mRNA. It became a vital asset in the development of a speedy vaccine against COVID-19. Again, we extract five relevant publications by the Nobel laureates from the Advanced information by Karlsson Hedestam and Sandberg (2023), who are members of Nobel Assembly at Karolinska Institute (see Table S1). A sixth early publication from 2005 described as "breakthrough" is omitted because the distance to technology must be higher and fewer results are to be expected.

The set of related patents stems from the supplementary material Martin and Lowery (2020). They identify 113 patents with application dates between 1 January 2011 and 1 April 2020 in the Derwent Innovation database. The authors employ a keyword search strategy with multiple synonyms for mRNA across title, description or claims, and require that the term "vaccin*" occurs as well. All of these are documents issued by either USPTO, EPO, or WIPO, which was a requirement by the authors, meaning all patents are present in Logic Mill.

Since the target set is much smaller, we search for no more than the closest 500 patent documents for each seed document, but with the requirement that the term "vaccin*" is present in the title. The latter condition is due to the fact that mRNA is relatively broad in its application, and it reflects the search strategy of Martin and Lowery (2020) (according to whose search strategy the term "vaccin" may occur in the description or claims section as well). We only include patent families with the earliest filing date on or past 1 January 2010. In total, we find 718 different families.

Out of the 113 patent families, our search yields 45 of them (39%). Their average rank fluctuates between 79 and 94. The median rank fluctuates between 63 and 102.

We depict the matching quality in Figure 7.11, which again shows the 500 most similar patent families for each of the seed documents. Given the much smaller and narrower target set, we expect many more red bars. The matched target is predominantly in the left half.

While the matching rate is lower than for the CRISPR/Cas9 example, the most similar patents are still consistently highly ranked in terms of similarity. Like Martin and Lowery (2020) we restrict ourselves to vaccine-related patents, however, we cannot extend the search beyond patent titles. We are likely to exclude relevant patents because of this. On the other hand, manual inspection of similar but unmatched patents suggests again that our search yields candidates for inclusion.

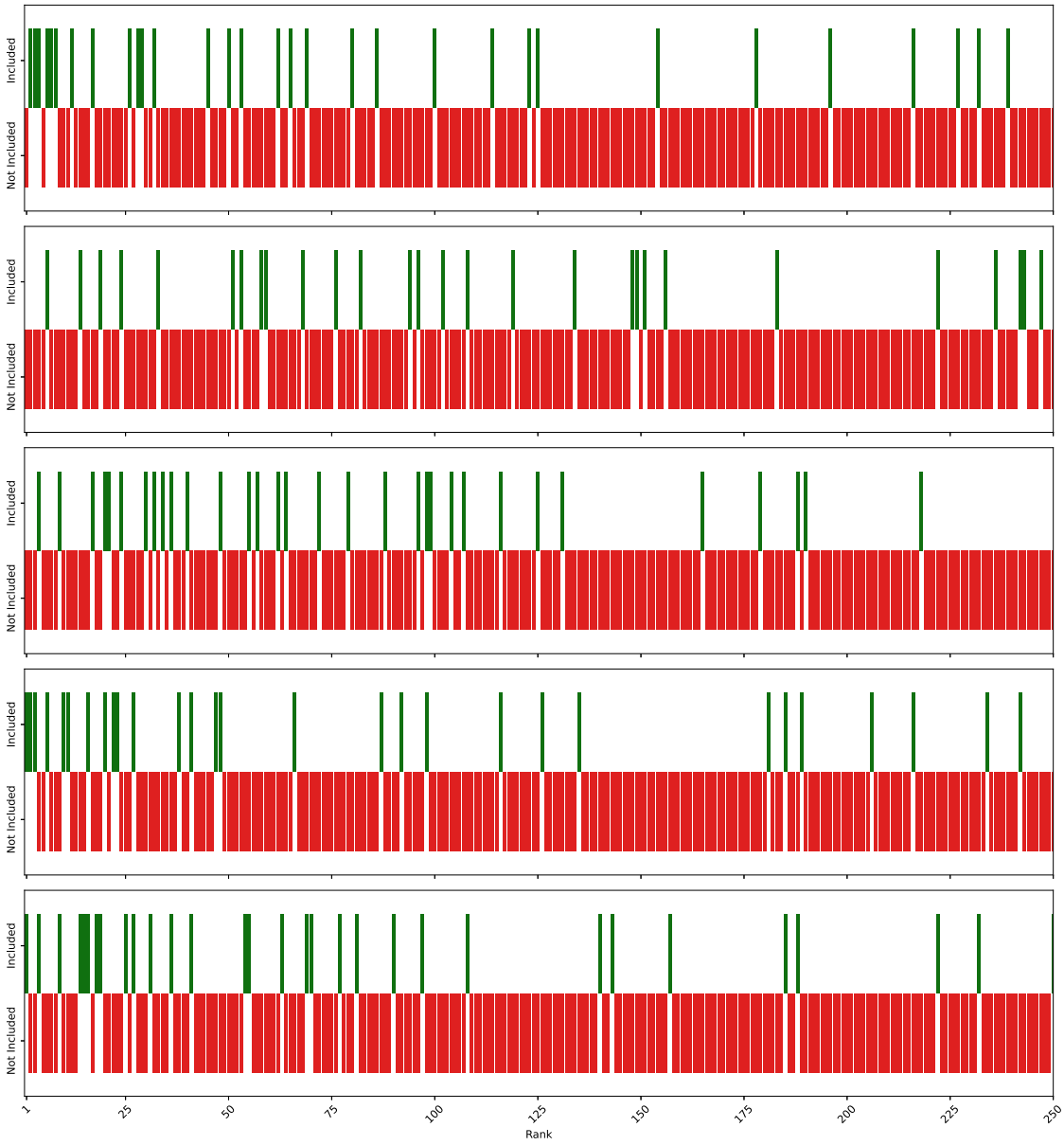
7.4 Conclusion

The prediction of Patent Paper Pairs, the prior art search, and the patent landscaping all demonstrated the capability of the Pat-SPECTER model in conjunction with the Logic Mill system.

In testing, we were faced with imperfect data sources and a lack of gold standards. For instance, in many cases of the PPC dataset, documents are cited for reasons other than semantic relevance.

Yet we conclude that the performance is more than satisfying, especially given the vast amount of documents to be digested.

Figure 7.11: Status of 500 most similar patent families, mRNA landscaping



Notes: Figures show which of the 500 most similar patent families are included in the database for each of the four source publications. Green bars indicate the x-th most similar patent family is in the target set. Red means they are not in the target set. There are 113 target patent families as identified by Martin and Lowery (2020).

Chapter 8

Conclusion

This research project aimed to develop and evaluate advanced machine learning models for tracing (potential) knowledge flows from scientific publications to patents. The primary objective was to leverage state-of-the-art document similarity models to predict citations and identify relevant connections between patents and scientific literature. Throughout this project, three distinct models were created and rigorously tested: Pat-SPECTER, PaECTER, and Pub-PaECTER.

Pat-SPECTER: Fine-tuned from the SPECTER2 model, Pat-SPECTER demonstrated robust performance in predicting citations from patents to scientific publications and from patents to patents. Pat-SPECTER is the main output of the project, as it won the horse race against the other two models. Its ability to accurately trace potential knowledge flows between these domains showcases its potential for practical applications in patent analysis and innovation tracking.

PaECTER: Fine-tuned from the BERT for Patents model and specifically designed for the patent domain, it outperformed all other models in predicting citations among patents. This indicates that a domain-specific vocabulary and fine-tuning on patent citations significantly enhance prediction accuracy, making PaECTER a valuable tool for patent analysis. In the public domain, the PaECTER is currently the state-of-the-art model for patent similarity models. For this reason, the model represents a significant stand-alone contribution.

Pub-PaECTER: Fine-tuned from the PaECTER model on scientific publications and aimed at predicting citations within scientific articles and from patents to scientific publications. However, it performed poorly in comparison to the other models (even compared to its foundational models), suggesting that the nuances of scientific literature require a different approach than what was applied to the patent domain.

The models were applied in several realistic scenarios to evaluate their practical utility. Notably, the prediction of Patent-Paper Pairs (PPPs) in a larger setting was explored, demonstrating that the models could effectively identify relevant documents from a vast pool of candidates.

We also identified some limitations of these models, especially when it comes to identifying relevant scientific articles for a given patent. This was demonstrated in the patent-to-paper evaluations and especially in the mRNA and CRISPR/Cas9 patent landscaping exercise. However, an important reason for these shortcomings is the incompleteness of the underlying databases. We would expect model performance to improve considerably once these database restrictions have been amended.

Furthermore, non-patent literature frequently cites foundational scientific concepts and methodologies, which may not exhibit sufficient semantic similarity to appear prominently in search results. Here, further investigations are needed.

Future research could focus on enhancing the models' scalability and exploring additional features, such as the use of metadata (e.g., author names, dates, and CPC classes), to boost prediction accuracy. PaECTER, its training dataset and Pat-SPECTER and are freely available for other researchers to use or improve.[1] Furthermore, extending the training data to include more diverse datasets and the evaluation of real-world applications will help in refining the models and expanding their utility. In conclusion, this project successfully demonstrated the potential of state-of-the-art Natural Language Processing in tracing (potential) knowledge flows between patents and scientific publications. The developed models, particularly Pat-SPECTER and PaECTER, show promise in enhancing patent analysis and fostering innovation by accurately predicting relevant citations and connections across domains.

[1] See https://huggingface.co/mpi-inno-comp.

Bibliography

Ahmadpoor, Mohammad and Benjamin F. Jones (2017). "The dual frontier: Patented inventions and prior scientific advance". en. In: *Science* 357.6351, pp. 583–587. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aam9527.

Beltagy, Iz, Kyle Lo, and Arman Cohan (2019). "SciBERT: A Pretrained Language Model for Scientific Text". en. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 3613–3618. DOI: 10.18653/v1/D19-1371.

Cohan, Arman et al. (2020). "SPECTER: Document-level Representation Learning using Citation-informed Transformers". en. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 2270–2282. DOI: 10.18653/v1/2020.acl-main.207.

Devlin, Jacob et al. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". en. In: *Proceedings of the 2019 Conference of the North*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

Erhardt, Sebastian et al. (2022). *Logic Mill – A Knowledge Navigation System*. arXiv:2301.00200 [cs]. DOI: 10.48550/arXiv.2301.00200.

Ghosh, Mainak et al. (2024). *PaECTER: Patent-level Representation Learning using Citation-informed Transformers*. Version Number: 1. DOI: 10.48550/ARXIV.2402.19411.

Gustafsson, Claes (2020). *A Tool for Genome Editing*. Tech. rep. The Royal Swedish Academy of Sciences.

Jaffe, Adam B. and Gaétan de Rassenfosse (2017). "Patent citation data in social science research: Overview and best practices". en. In: *Journal of the Association for Information Science and Technology* 68.6, pp. 1360–1374. ISSN: 23301635. DOI: 10.1002/asi.23731.

Karlsson Hedestam, Gunilla B. and Rickard Sandberg (2023). *Discoveries concerning nucleoside base modifications that enabled the development of effective mRNA vaccines against COVID-19*. Advanced Information. The Nobel Assembly at Karolinska Institutet.

Kelly, Bryan et al. (2021). "Measuring Technological Innovation over the Long Run". en. In: *American Economic Review: Insights* 3.3, pp. 303–320. ISSN: 2640-205X, 2640-2068. DOI: 10.1257/aeri.20190499.

Lee, Jieh-Sheng and Jieh Hsiang (2020). "Patent classification by fine-tuning BERT language model". en. In: *World Patent Information* 61, p. 101965. ISSN: 01722190. DOI: 10.1016/j.wpi.2020.101965.

Loshchilov, Ilya and Frank Hutter (2019). "Decoupled Weight Decay Regularization". In: *Proceedings of the ICLR 2019*. New Orleans, LA. DOI: 10.48550/ARXIV.1711.05101.

Martin, Cecilia and Drew Lowery (2020). "mRNA vaccines: intellectual property landscape". en. In: *Nature Reviews Drug Discovery* 19.9, pp. 578–578. ISSN: 1474-1776, 1474-1784. DOI: 10.1038/d41573-020-00119-8.

Martin-Laffon, Jacqueline, Marcel Kuntz, and Agnès E. Ricroch (2019). "Worldwide CRISPR patent landscape shows strong geographical biases". en. In: *Nature Biotechnology* 37.6, pp. 613–620. ISSN: 1087-0156, 1546-1696. DOI: 10.1038/s41587-019-0138-7.

Marx, Matt and Aaron Fuegi (2020). "Reliance on science: Worldwide front-page patent citations to scientific articles". en. In: *Strategic Management Journal* 41.9, pp. 1572–1594. ISSN: 0143-2095, 1097-0266. DOI: 10.1002/smj.3145.

— (2022). "Reliance on science by inventors: Hybrid extraction of in-text patent-to-article citations". en. In: *Journal of Economics & Management Strategy* 31.2, pp. 369–392. ISSN: 1058-6407, 1530-9134. DOI: 10.1111/jems.12455.

Marx, Matt and Emma Scharfmann (2024). "Does Patenting Promote the Progress of Science?" In: *Unpublished manuscript*.

Mikolov, Tomas et al. (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv:1301.3781 [cs]. DOI: 10.48550/arXiv.1301.3781.

Natterer, Michael (2014). "Entwicklung, empirische Validierung und ökonomische Anwendung eines texbasierten Ähnlihckeitsmaßes". German. Inauguraldissertation. Munich: Ludwig Maximilians University.

Pedregosa, Fabian et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12.85, pp. 2825–2830. ISSN: 1533-7928.

Poege, Felix et al. (2019). "Science quality and the value of inventions". en. In: *Science Advances* 5.12, eaay7323. ISSN: 2375-2548. DOI: 10.1126/sciadv.aay7323.

Priem, Jason, Heather Piwowar, and Richard Orr (2022). *OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts*. Version Number: 2. DOI: 10.48550/ARXIV.2205.01833.

Reimers, Nils and Iryna Gurevych (2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Hong Kong, pp. 3982–3992.

Singh, Amanpreet et al. (2022). *SciRepEval: A Multi-Format Benchmark for Scientific Document Representations*. DOI: 10.48550/ARXIV.2211.13308.

Srebrovic, Rob and Jay Yonamine (2020). *Leveraging the BERT algorithm for Patents with TensorFlow and BigQuery*. Tech. rep. Google.

Vaswani, Ashish et al. (2017). "Attention Is All You Need". In: *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*. DOI: 10.5555/3295222.3295349.

Vowinckel, Konrad and Volker D. Hähnke (2023). "SEARCHFORMER: Semantic patent embeddings by siamese transformers for prior art search". en. In: *World Patent Information* 73, p. 102192. ISSN: 01722190. DOI: 10.1016/j.wpi.2023.102192.

Webb, Colin et al. (2005). "Analysing European and International Patent Citations: A Set of EPO Patent Database Building Blocks". In: *OECD Science, Technology and Industry Working Papers*.

Wolf, Thomas et al. (2019). *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. Version Number: 5. DOI: 10.48550/ARXIV.1910.03771.

Appendix A

A.1 Statistical Significance Test Pat-Specter

Table S1: Statistical Significance Testing of Rank-Aware Evaluation of Pat-SPECTER versus Different Models on a Cross-Corpus Dataset

	Avg. RFR	MAP	MRR@10
BERT	0.239***	-0.123***	-0.065***
	(0.035)	(0.008)	(0.007)
SciBERT	0.322***	-0.202***	-0.075***
	(0.035)	(0.008)	(0.007)
BERT for Patents	0.050	-0.054***	-0.011
	(0.035)	(0.008)	(0.007)
SPECTER	0.025	0.003	-0.003
	(0.035)	(0.008)	(0.007)
SPECTER2	0.057	-0.031***	-0.021**
	(0.035)	(0.008)	(0.007)
PaECTER	0.015	-0.017*	-0.005
	(0.035)	(0.008)	(0.007)
Pub-PaECTER	0.196***	-0.118***	-0.039***
	(0.035)	(0.008)	(0.007)
Constant	1.051***	0.914***	0.980***
	(0.025)	(0.006)	(0.005)
Adjusted R2	0.020	0.130	0.026
Observations	8,000	8,000	8,000

Notes: Standard errors in parentheses. Base level model is Pat-SPECTER. All models use their best pooling method, as derived from Table 6.1.
* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

A.2 Additional Tables and Figures

Table S2: CRISPR/Cas9-related publications

Title	Year	Journal
CRISPR RNA maturation by trans-encoded small RNA and host factor RNase III	2011	Nature
A programmable dual-RNA-guided DNA endonuclease in adaptive bacterial immunity	2012	Science
Cas9-crRNA ribonucleoprotein complex mediates specific DNA cleavage for adaptive immunity in bacteria	2012	PNAS
CRISPR-Cas9 Structures and Mechanisms	2017	Ann. Rev. Biophys.

Notes: Scientific publications included in the CRISRP/Cas9 case study. Own compilation based on Gustafsson, 2020.

Table S3: mRNA-related publications

Title	Year	Journal
Incorporation of pseudouridine into mRNA yields superior nonimmunogenic vector with increased translational capacity and biological stability	2008	Molecular Therapy
Incorporation of pseudouridine into mRNA enhances translation by diminishing PKR activation	2011	Nucleic Acids Research
Nucleoside modifications in RNA limit activation of 2-5-oligoadenylate synthetase and increase resistance to cleavage by RNase L	2011	Nucleic Acids Research
Generating the optimal mRNA for therapy: HPLC purification eliminates immune activation and improves translation of nucleoside-modified, protein-encoding mRNA	2011	Nucleic Acids Research
A Facile Method for the Removal of dsRNA Contaminant from In Vitro-Transcribed mRNA	2019	Molecular Therapy Nucleic Acids

Notes: Scientific publications included in the mRNA case study. Own compilation based on Martin and Lowery, 2020.

A.3 PaECTER Paper

PaECTER: Patent-level Representation Learning using Citation-informed Transformers.
Mainak Ghosh, Sebastian Erhardt, Michael E. Rose, Erik Buunk, Dietmar Harhoff (2024)
https://doi.org/10.48550/arXiv.2402.19411

A.4 Pat-SPECTER Model

Mainak Ghosh, Sebastian Erhardt, Michael E. Rose, Erik Buunk, Dietmar Harhoff (2024)
https://huggingface.co/mpi-inno-comp/pat_specter

A.5 PaECTER Model

Mainak Ghosh, Sebastian Erhardt, Michael E. Rose, Erik Buunk, Dietmar Harhoff (2024)
https://huggingface.co/mpi-inno-comp/paecter

A.6 Dataset

Mainak Ghosh, Sebastian Erhardt, Michael E. Rose, Erik Buunk, Dietmar Harhoff (2024)
https://huggingface.co/datasets/mpi-inno-comp/paecter_dataset