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

Abstract

We identify a similarity model suitable for working with patents and scientific publications at the same time. As transformer-based language and similarity models are trained on a single corpus (such as patents), they often perform poorly on other corpora (such as publications). Especially in the patent-publication nexus, researchers are interested in linking semantically similar documents. In a horse race-style evaluation involving four models, we find that the SPECTER model fine-tuned on patents performs best. We name this model Pat-SPECTER and test it in two real-world applications.

1 Introduction

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 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.

To overcome this problem, we present a transformer-based similarity model that can be used for patent-publication comparisons. We identify four suitable transformer models and test them rigorously in a horse race. These models have SPECTER, developed by Devlin et al. (2019) for work with publications, and PaECTER, developed by Ghosh et al. (2024) for work with patents, as well as derivations from these two models. We benchmark the models using standard test metrics on small datasets derived from Patent Paper citations by Marx and Fuegi (2022) and Marx (2023).

The winning model, Pat-SPECTER, is a derivative of the SPECTER model fine-tuned on patents, more specifically on the training dataset for the PaECTER. To show its strength, we apply the model to two real-world scenarios, a prior art search akin to patent examination and the prediction of patent paper pairs, where the goal is to find relevant publications among thousands or even millions candidates. We utilize an ElasticSearch database in the Logic Mill search system (Erhardt et al., 2022) to identify (approximate) nearest neighbors among all of PATSTAT and OpenAlex.

2 Literature

In the patent domain, citations to scientific literature are used to link patented inventions to their scientific underpinnings. 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.

Poege et al. (2019) show that highly cited scientific publications often serve as the foundation for particularly impactful patented inventions, highlighting the critical role of science in driving technological innovation. They 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.

However, the use of explicit citations has serious disadvantages, even if indirect links are taken into account. Citations are rare, and they citations may be selected strategically or opportunistically (Jaffe & Rassenfosse, 2017).

As an alternative to citations, scholars have investigated the possibility of using text as input data and comparing two documents to gauge their similarity. When vectors represent individual documents, vector operations (such as Cosine distance, Manhattan distance, Euclidean distance or others) yield similarity estimates.

In his dissertation, 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).

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.

Recent advances in Natural Language Processing, especially the Word2vec (Mikolov et al., 2013), and Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), have made it possible to non-computer scientists to work with textual data while retaining full syntactial information.

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. 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 &

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

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.

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 (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.

As the language specific to patents likely differs from that relevant to scientific publications, which may impact performance, Ghosh et al. (2024) recently develop the PaECTER model. It was trained in a similar way as the SPECTER as it leverages credible citation information between patents. Credible citation information refers to citations added by examiners at the EPO.

3 Finding a cross-corpus language model

3.1 Experimental setup

Our goal is to assess which similarity models trace knowledge flows from science to technology best. We compare three similarity models against each other: PaECTER, Pat-SPECTER, and Pub-PaECTER. For comparison, we include prior models with a similar focus, namely BERT, SciBERT, BERT for Patents, SPECTER, and SPECTER2.

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) is pre-trained on newspaper articles and Wikipedia and focuses on the 40k most important tokens. Since its release, BERT has demonstrated outstanding performance across various domains, including semantic search, text classification, and natural language understanding.

SciBERT (Beltagy, Lo, & Cohan, 2019) is a BERT model specific to scientific publications. It has learned the 40k most important tokens used in scientific publications. In the patent domain, Google's BERT for patents (Srebrovic & Yonamine, 2020) was trained on 3 million granted USPTO patents.

The SPECTER model (Cohan et al., 2020) (and its successor SPECTER2) are extensions of the SciBERT model. Their task is to effectively identify similar documents, a task the aforementioned models struggle with since their task is to predict masked tokens. that applies to the aforementioned models concern to a large extent. SPECTER/SPECTER2 overcome this shortcoming through citation-informed learning. More specifically, SciBERT was fine-tuned via contrastive learning between two publications, an actually cited publication and an uncited random publication.

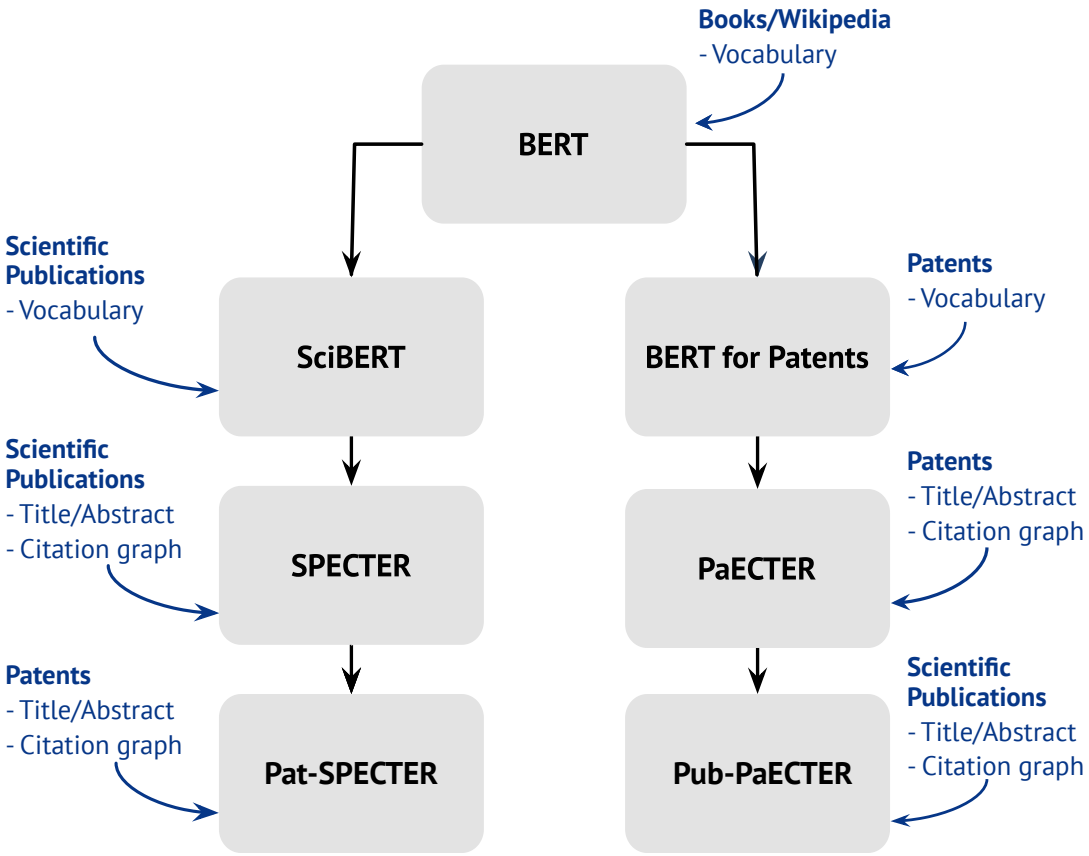
PaECTER (Ghosh et al., 2024) is the equivalent of SPECTER, but for the patent domain. The authors used certain patents cited in EP patents, since at the EPO

references to other patents are solely added by examiners. The training dataset consisted of 300,000 English-language patent families, including applications filed with the European Patent Office (EPO) from 1985 to 2022.

Pat-SPECTER and Pub-PaECTER are derivatives of SPECTER and PaECTER respectively. Pat-SPECTER is the original SPECTER model fine-tuned on the training data set for the PaECTER, and the Pub-PaECTER is the original PaECTER fined-tuned on the training data set of the SPECTER. While Pat-SPECTER is described in (Ghosh et al., 2024), Pub-PaECTER was specifically designed for this project.

Figure 1 illustrates how the models relate to each other.

Figure 1: Overview of the models used in the comparisons



3.2 Data

For our evaluations, we use the *Reliance on Science* dataset provided by Marx and Fuegi (2020, 2022). In particular, we look at Patent Paper Pairs and Patent-Paper Citations. Both of these data sets are rigorously curated, lending a high degree of reliability and applicability.

The patent to paper citations data set is a patent data set with all the cited scientific publications for each patent. 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.

The patent-paper-pairs data set contains around 548k pairs with 310k unique USPTO patents and 336k unique publications. 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 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.

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.

For our comparisons we need textual data, notably title and abstracts. For publications, we use the OpenAlex (Priem, Piwowar, & Orr, 2022) database. OpenAlex is a comprehensive, community-curated database of scholarly works, authors, institutions, and more. For patents, we use PATSTAT 2023 Spring version. PATSTAT (Patent Statistical Database) is a comprehensive database produced by the European Patent Office (EPO).

Our dataset consists of 1,000 randomly selected triplets: Every record contains a focal EPO patent (A1, A2 or B1), five cited publications (positives), and 25 publications (negatives). We sampled the negatives randomly from OpenAlex, provided their abstract is in English.

To compare the models, we use ranking metrics. For each test:

- Embeddings are generated for all 31 documents in a triplet.
- The cosine distance metric is calculated between the focal patent and the other 30 documents using the embeddings.
- The documents are then ranked from most similar to least similar based on the cosine distance.
- This process is repeated for each of the 1,000 triplets.

Ideally, the positive documents should be ranked in the top five positions. The metrics used (RFR, MAP, and MRR@10) provide different perspectives on the ranking results.

The results in Table 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 5 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.

Table 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
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

Notes: "Avg. RFR" is the average rank of the first relevant (i.e., actually cited) publication. "MAP" is the mean-average precision, which takes into account the precision and the recall at every position where a relevant item appears. "MRR@10" is the mean reciprocal rank of the first relevant publication within the 10 closest publications. "CLS" and "Mean" refer to different ways we compute embeddings for a text of multiple sentences: "CLS" concatenates all sentences into one sentence while "Mean" takes the average of the sentences' embeddings.

3.3 Conclusion

The evaluation suggests that Pat-SPECTER is the most suitable model for cross-corpus comparison involving scientific publications and patents. 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.

4 Applications

To make the model more palpable, we test it on three applications.

The goal of the first application is to separate "Patent-Paper-Pairs" (PPPs) from other pairs of publications and patents.

The second evaluation is analogous to the prior art search by patent offices. Prior art search is a crucial task in the patent examination process, as it aims to find existing patents, publications, and other relevant articles that might challenge the novelty of the patent application under prosecution.

The final evaluation applies the PPPs in a broader context. For each patent in the PPP dataset, we retrieve the most similar documents from millions of publications in OpenAlex and assess where the actual PPPs rank within this list.

4.1 Separating Patent Paper Pairs from other Pairs

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 logistic regression models to predict 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.

4.1.1 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. **??**). 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.

4.1.2 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 **??**). For the random data set, we add random publications to the patents of PPP and NPL. Publications must have a title and an abstract in English.

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.

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.

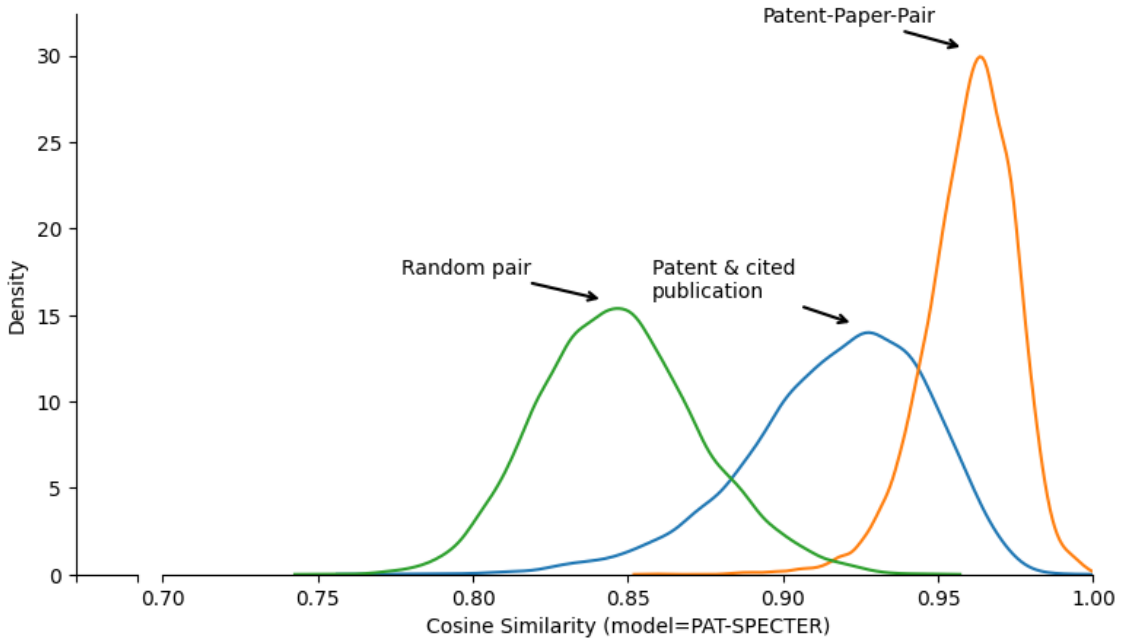
The next step is the creation of dense numerical representations (embeddings). We use our models Pat-SPECTER and PaECTER. SPECTER and SPECTER2-base are used as baselines.

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.

After this, we calculate the cosine similarity between the embeddings of the pairs. 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

Figure 2: KDE of cosine distribution PAT-SPECTER



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.

4.1.3 Results

The results show that there is a separation between Random, NPL, and PPP pairs (KDE plots Figure 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 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 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 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 **??**). 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 2.

Figure 3: KDE of cosine distributions TEST and TRAIN non-normalized data

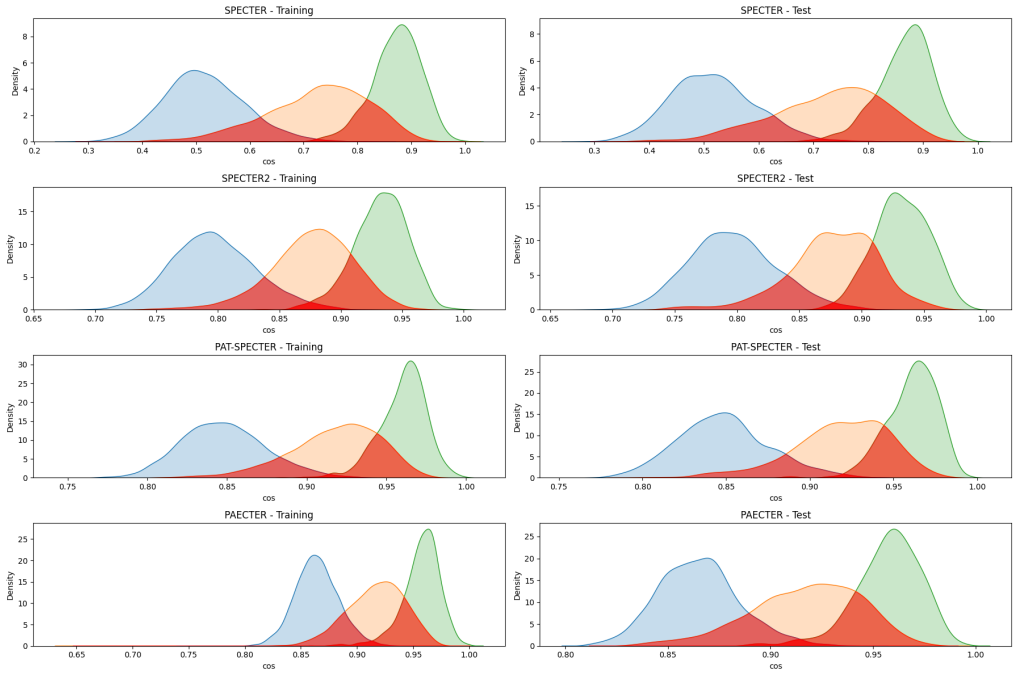
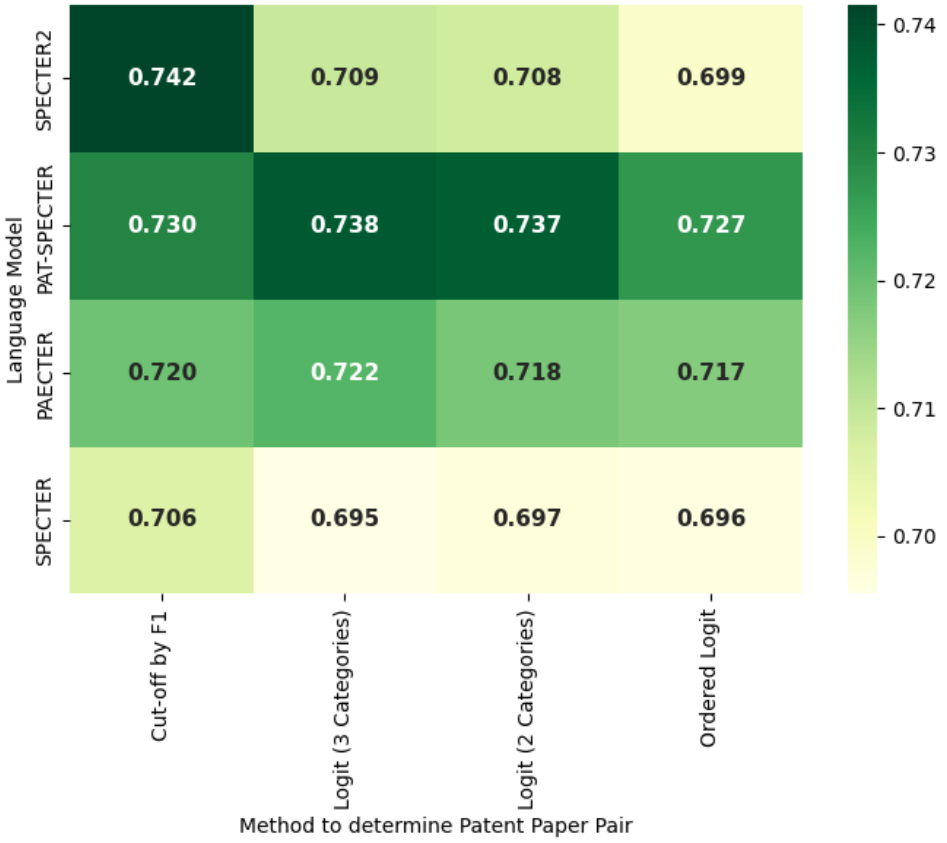


Table 2: Optimal F1 cut-off values for prediction of PPPs

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

PaECTER seems to be more sensitive to outliers in the dataset. The bump around $cos = 0.74$ in Figure 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 2).

Figure 4: Heatmap of the F1 Scores (with original cosine values)



4.1.4 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.

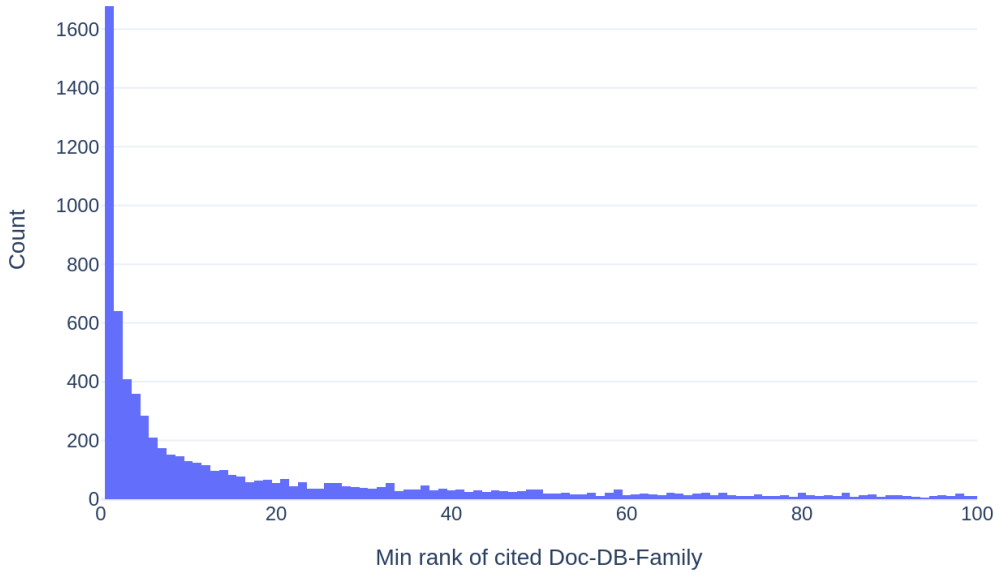
The F1 cutoff value can be directly used as a guide as to which documents are to be selected. In this case, the (Approximate) Nearest Neighbor search operates on one similarity metric, thus allowing for considerable speed, even while searching through millions of documents. After the initial selection, 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.

4.2 Prior Art Search

When performing a prior art search, patent examiners seek to find documents that help assess whether the patent application is novel or not. These are usually patents, but also scientific publications may be relevant. There we conduct two exercises, a patent to patent prior art search, and a patent to publication prior art search.

To perform these two tasks, we rely on the Logic Mill system, developed by the authors (Erhardt et al., 2022). Logic Mill is a scalable and openly accessible software system with the goal to identify semantically similar documents within either one domain-specific corpus or multi-domain corpora. Logic Mill provides embed-

Figure 5: 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).

dings for scientific publications (all of OpenAlex) and patent documents (all patent documents from the three jurisdictions EPO, USPTO and WIPO) as computed by Pat-SPECTER, provided they have an English abstract. As it uses ElasticSearch, we can leverage the approximate nearest-neighbor search algorithm to find the most similar documents with a very high probability for any query document within milliseconds.

4.2.1 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 (which at the EPO are exclusively added by examiners). 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 5).

Table 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 3 demonstrate the Mean Reciprocal Rank (MRR)

Table 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

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

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. Given the results of Table 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.

4.2.2 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, 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 are the same as in the previous exercise.

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 6, the distribution is not as steep as in the patent-to-patent case. Table 4 presents the MRR and MAP for various $k \in \{5, 10, 20, 50, 100\}$.

The results presented in Table 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

Figure 6: Distribution of the first ranked publication cited by the reference patent (N=3,740 of 10,000)

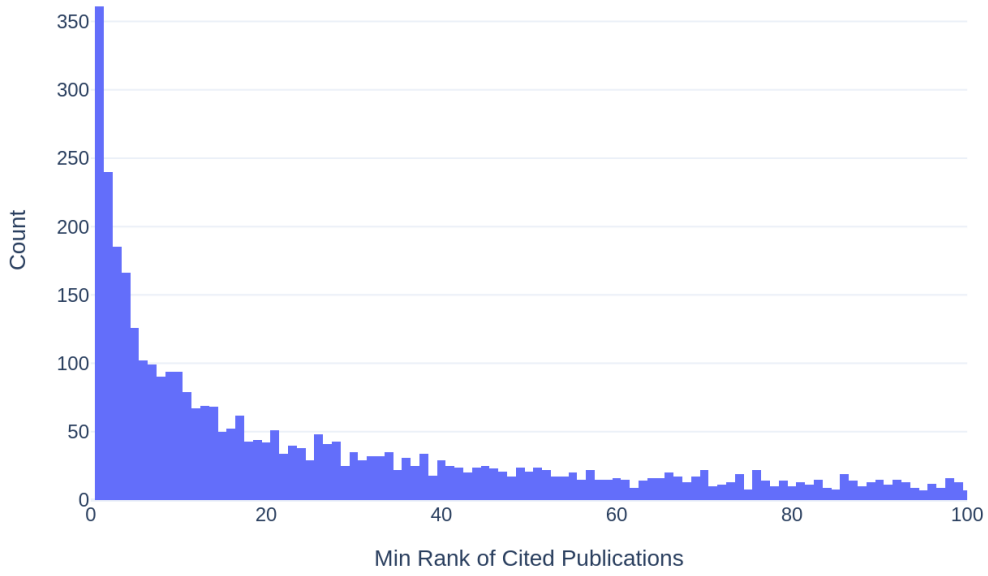


Table 4: MRR and MAP for patent-to-publication citations

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.

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.

4.3 Searching Patent Paper Pairs in the patent-publication universe

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 4.2. 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).

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 8. 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: Empirical Cumulative Distribution of the rank for the matched PPPs

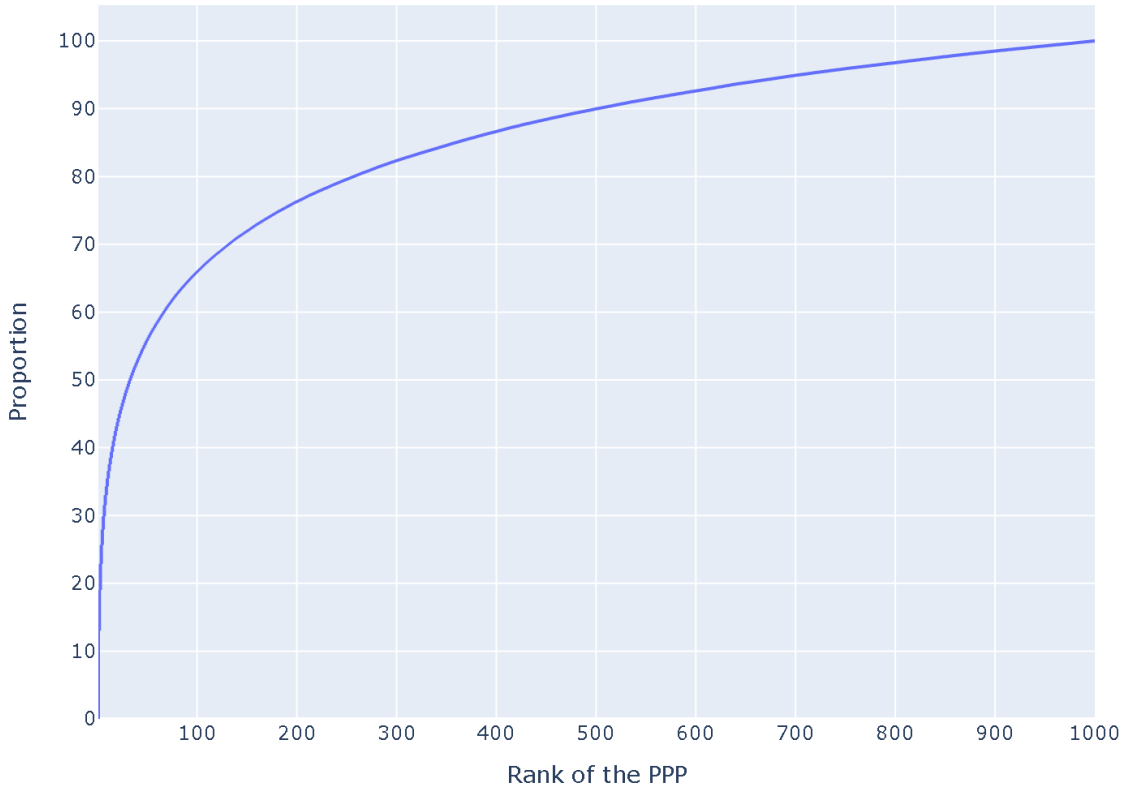
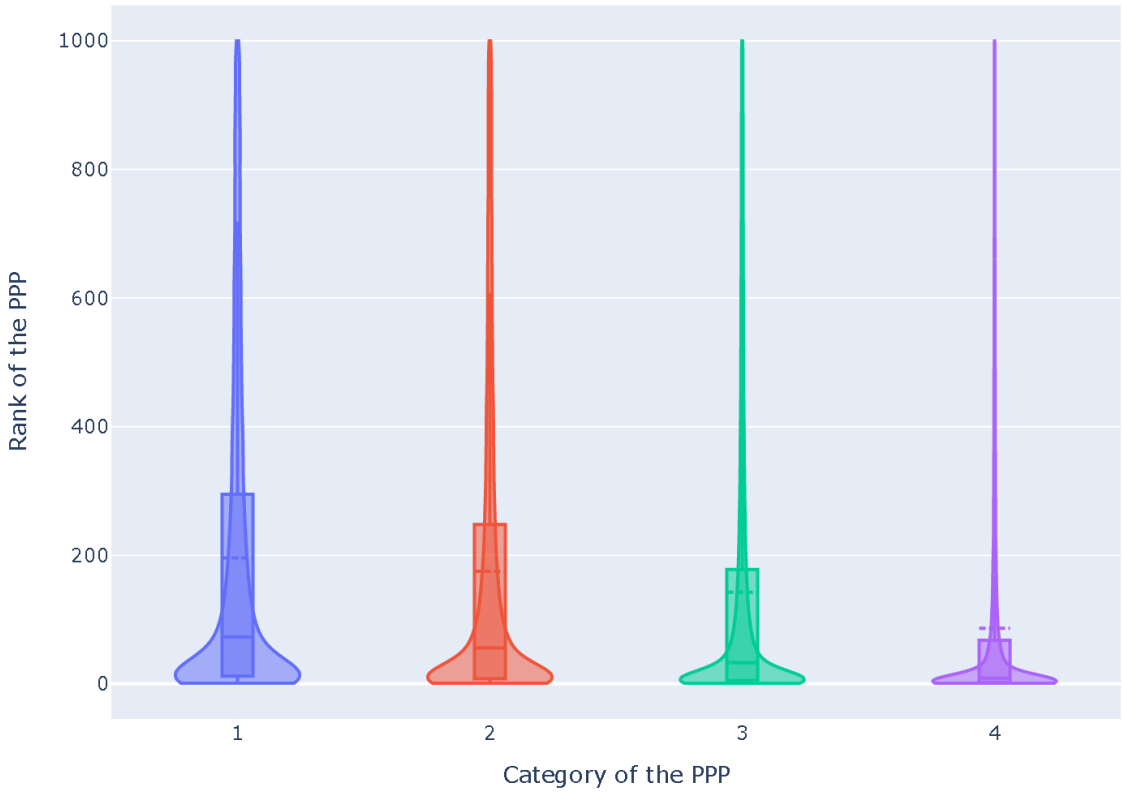


Figure 8: Distribution of the rank for the matched PPPs in different categories



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

5 Conclusion

Fine-tuned from the SPECTER2 model, Pat-SPECTER demonstrated robust performance in predicting citations from patents to scientific publications and from patents to patents. This reflects its potential for enhancing the search for prior art and identifying technological advancements rooted in scientific research.

In the real-world application of Pat-SPECTER, the model was subjected to prior art tests to evaluate its practical utility in identifying relevant prior art documents within a large dataset. This involved using Pat-SPECTER to search extensive databases for patents and scientific publications that closely matched specific patent documents. The model's performance was assessed based on its ability to rank the most relevant prior art at the top of the search results. The tests demonstrated that Pat-SPECTER could effectively identify pertinent prior art.

The most severe limitation stems from lack and incompleteness of our data sources. OpenAlex 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.[2] While many of them are not in English, a great deal of them belong to Springer journals which does not share abstracts with OpenAlex.

[2]It should be noted that "Works" included documents published by journals which are not commonly referred to as scientific publications. front matters, back matters, table of contents, advertisements, etc.

6 Appendix: Statistical Significance Test Pat-Specter

Table 5: 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 1.
* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

References

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.

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.

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.

Marx, Matt (2023). *From Patent-Paper Citations to Patent-Paper Pairs*. en. Cambridge, MA, USA.

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.

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.

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.