Patentability: Improving Acceptance Prediction of US Patent Applications using Ensemble Modeling

Stanford CS224N Custom Project

Tommy Bruzzese
Department of Symbolic Systems
Stanford University
tbru@stanford.edu

Alex Lerner
Department of Statistics
Stanford University
alerner1@stanford.edu

Oscar O'Rahilly
Department of Computer Science
Stanford University
oscarfco@stanford.edu

Abstract

Patents are an essential part of modern innovation, allowing for certification and protection when an idea is useful and new. However, what determines a successful patent application in the United States has not been thoroughly explored on a mass-scale by machine learning. Suzgun et al. have presented the task of binary acceptance prediction, where the model predicts whether applications were accepted or rejected. In our paper, we re-establish these results on baseline models, and then improve on them with original ensemble models to determine an application's utility and novelty. Namely, by first ensembling models that look at different parts of a patent (Abstract, Claims) and also ensembling different types of models (Naive Bayes, DistilBERT), we can better model an application's utility and achieve 62.65% accuracy on acceptance prediction. Additionally, we confirm patterns from Suzgun et al. that Transformer models are not able to significantly outperform Naive Bayes in the prediction task, and discuss hypotheses why. Finally, we provide a visual understanding of how our models focus on various parts of text in its prediction using integrated-gradient analysis. With our improvements to the acceptance prediction task and deepened understanding of a model's focusing, we grow the information on determining what makes a patent application successful, which is useful especially for patent office efficiency and for groups that are underrepresented in the patent domain.

1 Key Information to include

- TA Mentor: Michihiro Yasunaga
- External Mentor: Mirac Suzgun

2 Introduction

With around 650,000 patent applications submitted in fiscal year 2021 alone, patents are important indicators of the shifting regulatory and competitive landscape of new technological innovations [1]. They are a powerful and direct source for analyzing emerging technologies, idea generation, and economic growth. However, in its annual accountability report, the U.S. Patent and Trademark Office (USPTO) has shared concerns that such high levels of applications are negatively affecting its average time until first action [1], and researchers have opined that patent offices are "in dire need for automation" [2].

Furthermore, given patents' complexity and the limited number of expert examiners trained to review applications, processes surrounding patents are often "inefficient and imprecise" [3]. Notably, acceptances are not uniformly distributed across different sectors of the US. Only 10% of US patent inventors are women [4], and there is evidence that female applicants and mirco entities (independent applicants, small companies, non-profits) are less likely to be approved than male applicants and entities with over 500 employees [4, 5]. Interestingly, such unequal distributions are thought to come from human bias after implicitly inferring genders from applicant names [4], and knowing larger company names better, giving room for machines to improve equality by not learning or acting upon inferred biased name associations.

In our work, we seek to reliably predict the success of a US patent application and produce new information on what make an application successful. By doing so, we aim to improve these current issues and make US patent applications more accessible. Given especially that only around 50% of applications are currently granted [6], having the ability to accurately pre-approve patents could reduce strain on the USPTO, provide a less biased review of an application, and give underrepresented groups confidence in their work before submission. Across the globe, patent offices spend more than $10 billion annually in operational costs, highlighting how these efficiency gains could also have large economic benefits [7, 8].

Prior models have not performed exceptionally well on the classification task, likely due to the complex writing, specialized "legalese" and domain terminology, and long length of patents. In particular, the Abstract and Claims sections (the two most significant portion of an application) average 132 and 1271 tokens, respectively [9]. Given that BERT models are only able to train on 512 tokens at a time, prior work has often limited itself to only training one section of an application in isolation. We improve upon this work by introducing two novel model architectures. First we ensemble models to look at both the Abstract and Claims section before making a prediction decision. Next, we ensemble together different types of models (Naive Bayes, DistilBERT) to more richly represent an application. Our ensemble models seek to outperform prior state-of-the-art on prediction.

It is regarded that "the claims – together with the prior art – provide the most useful and critical information about the patentability of an invention," [9]. By creating our new ensemble models, we blend Abstract and Claims for the first time in acceptance classification and show that our approach can better model a patent's utility and novelty. Additionally, we provide discussion using integrated-gradient analysis about what our acceptance classifier models are learning to weight heavily. Lastly, we analyze hypotheses from prior literature about why Naive Bayes models have similar performance to more complex and rich Transformers.

3 Related Work

Natural Language Processing (NLP) has had a significant impact in summarizing, determining the novelty, and analyzing the semantics of patents. Prior work has focused almost exclusively on IPC (International Patent Classification) code classification of patents, retrieval of similar patents, and Claims generation [2, 3, 7, 10, 11]. Much of this work has been done alongside patent offices, and this domain of research is marked by its exciting collaboration between industry, government, and academia. In fact, to help with AI-related learning of patents, USPTO, the European Patent Office (EPO), and World Intellectual Property Office (WIPO) have all released their own corpora of titles, abstracts, and class labels for millions of applications [2]. Given the variety of datasets and scale of prior projects, direct comparison between prior NLP patent work is often difficult [12] and the domain lacks an agreed-upon task like the ImageNet challenge.

Suzgun et al. have begun to centralize the field and push it forward by creating the largest and most rich dataset of US patent applications to date, collecting more than 4.5 patent applications submitted to USPTO from 2004–2018. Previous datasets have often focused on granted patents which have been useful for some NLP tasks, however Suzgun et al. present the Harvard USPTO Patent Dataset (HUPD) that instead contains all submitted *applications* during these years. Comparing granted patents (which tend to have their Claims narrowed since the initial application [13]) versus rejected patent applications was not an equal comparison. By having the accepted and rejected applications as they were first filed, the HUPD has opened up various new NLP tasks for the first time, namely acceptance prediction.

The data in HUPD is also "clean data" because it aggregates data directly from the USPTO, as opposed to other datasets that use Google's Patent search, which may contain incorrectly machine-translated text from other languages into English and text scraped from images. Additionally, while previous datasets, such as CLEF-IP 2011 [14],USPTO-2m [15], and BIGPATENT [16] included only the Description and Abstract of a patent, the HUPD also contains 34 metadata fields for each application, including filing date, fine-grained IPC codes, and examiner information, allowing more freedom in what a model can be trained on.

Prior literature has explored the difficulties of working with patent data, which is often more complex, contextual, and technical than other natural language. Given that patents must contain a novel idea, and because novel ideas are often written about in a novel way [7], it can be hard to generalize embedded information because terms and sentences are often used in ways that they never have been before. There has been a shift and increased interest in applying deep-learning NLP methods, including Transformers and patent-specific trained BERT models [7]. Specifically, there is work building patent-specific pre-trained word embeddings to emulate expert examiner knowledge [12] and also work in custom tokenization that better preserves a word [7]. Despite its unique complexity, patent data is used already in large-scale NLP corpora. In fact, the most-frequent source of text in Colossal Clean Crawled Corpus (C4) dataset is "patents.google.com," by number of tokens [17]. Thus, by studying and modeling patent data, we look forward to exploring how patent-specific architecture will continue to grow, and improve performance on tasks within — and also beyond — the patent domain.

Our work focuses specifically on the new acceptance prediction task created by Suzgun et al. Benchmarks for the task have been established after training Naive Bayes classifiers and various fine-tuned BERT models. Prior work has focused on training and validating only on one IPC subclasses at a time or universally across all subclasses, and we take up the latter task. We seek to expand the limitations of prior work, which has focused on only using one portion of a patent text (i.e., only the Claims section) to determine the acceptability of the entire application. Additionally, despite running many Transformer models, baselines have often been set using a Bernoulli Naive Bayes model. It has been hypothesized that the BERT models might have been unable to go beyond word-level extraction, and therefore ultimately mirrored the behavior and performance of the Naive Bayes classifiers [9].

In view of these challenges, it has often been natural in other domains to integrate — or ensemble — multiple models together. Work has been done to apply a voting algorithm to combine multiple classifiers from multiple datasets [18]. The voting approach is often the most common approach given its simplicity and impact, and we build upon this ensembling work in this paper [19].

4 Approach

4.1 Ensemble Modeling

The primary approach we introduce is using deep-learning models to ensemble together various baseline models in patent acceptance prediction. Specifically we ensemble models trained on different sections of an application as well as different types of models.

To address the concerns of Suzgun et al. on the limitations of looking at once section of an application at a time, we chose to ensemble together models trained on the Abstract section with models trained on the Claims section. This is a novel contribution that has not been done before in acceptance prediction, and required coding of specific ensembling architecture. Ensembling individual models is preferred to training one model on both sections simultaneously, given that Abstract and Claims section make up an average of 1403 tokens when combined [9]. This is much larger than many Transformer models can handle. Additionally, the various sections of a patent have meaningfully different linguistic structures [7], given that the Abstract is a brief gist of the problem and the invention's use, whereas the Claims section is a much more thorough technical description [9, 20]. Training models on each section — and then ensembling them — thus should help better preserve the unique nuances of a section.

In classification, performance also depends extremely on how data is represented [21]. Thus we also ensemble together multiple types of models to help create a richer representation of the patent data and take full advantage of various classifiers. In prior literature, both Bernoulli Naive Bayes

and fine-tuned DistilBERT models have set baselines when trained on the Claims or on the Abstract. Thus we continue to use these two models [9].

First, the Naive Bayes Model makes several assumptions about text, particularly that tokens are independent. As its namesake suggest, the model employs Bayes' rule to quantity the probability of patent acceptance:

$$p(Acceptance|w_1, ..., w_{|V|}) \propto p(Acceptance) \prod_{i=1}^{|V|} p(w_i|Acceptance) \tag{1}$$

where each w_i is a token, and V is the vocabulary set. As we can see, the Naive Bayes makes the linguistically incorrect assumption that the probability of a token appearing is independent of the other tokens, however it is included given that it has set the state-of-the-art performance.

We use a fine-tuned DistilBERT model, created by HuggingFace's Sanh et al [22], which was another top-performing model [9]. DistilBERT largely contains the same architecture of the BERT model, with several important distinctions. First, a major motivation for the DistilBERT model was to reduce its size, and it resulted in a 40% decrease compared to BERT. This allows it to be 60% faster to pretrain, while being able to retain 97% of the language understanding of the BERT model [22]. DistilBERT reduced its size by halving the number of layers and introducing distillation to mimic BERT model.

The concept of knowledge distillation [23] was thus integral, and is best explained using the analogy of a student and teacher: A compact model (student) is trained to replicate the larger model (teacher). The loss function in the DistilBERT model is a linear combination of the supervised training loss, a cosine embedding loss, and distillation loss, L, which we define below:

$$L_{CE} = \sum_i t_i * \log(s_i) \tag{2}$$

where t_i is a probability estimated by the BERT teacher distribution, resulting in a strong training signal.

For our ensemble models, we present two different, novel architectures:

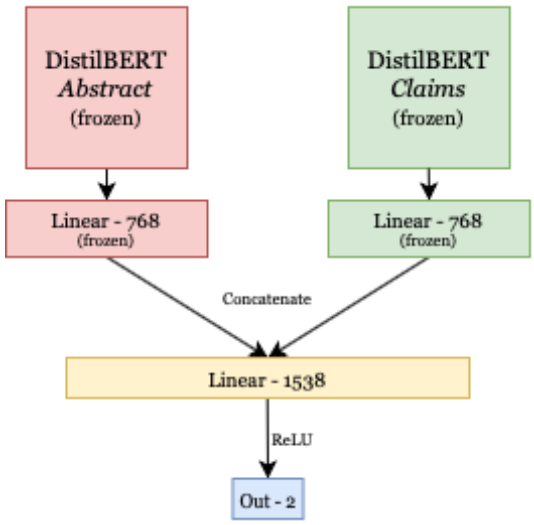

Figure 1: Ensemble Model 1: DistilBERT on Multiple Patent Sections, with frozen weights and concatenation

Ensemble Model 1, DistilBERT on Multiple Patent Sections For our first ensemble model, we take two DistilBERT models, one trained on the Abstract and one trained on the Claims section. We strip the final classification layer of each individual model, concatenate the final two linear layers and

append a final classification layer. We freeze the weights on all prior layers for the ensemble and learn the weights for the linear layer using stochastic gradient descent.

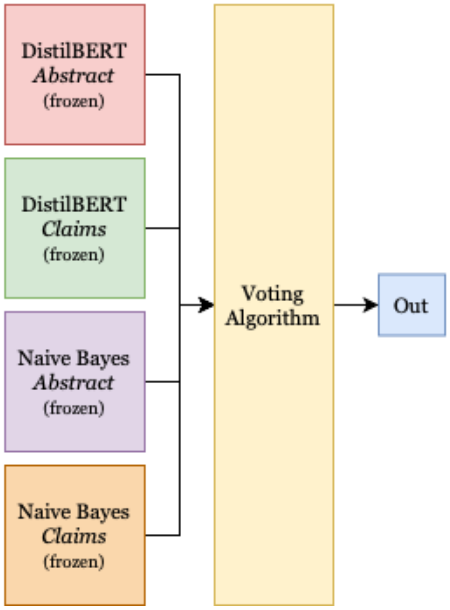

Figure 2: Ensemble Model 2, Naive Bayes + DistilBERT: Multiple Sections and Models, with a voting algorithm

Ensemble 2: For our second ensemble model, we take four trained models: two DistilBERTs trained on Abstract and Claims respectively, and two Naive Bayes trained on Abstract and Claims respectively. Unlike Ensemble 1, we freeze the weights of all layers of each model, and use two different Voting Algorithms to determine the final classification. In the case of any ties, we randomly pick the prediction.

1. **Naive voting algorithm:** The first voting algorithm simply returns the mode classification of the four models. In the case where two models classify a patent as accepted and two classify it as rejected, we return a random classification.

2. **Weighted voting algorithm:** The second voting algorithm weights the importance of each model, assigning higher importance to models with higher stand-alone classification accuracies. To calculate these weights we simply divide each model's accuracy by the accuracy for the DistilBERT Abstract model. The following weights for [DistilBERT-Abstract, DistilBERT-Claims, NB-Abstract, NB-Claims] are thus obtained **[1.029, 1.060, 1.0, 1.016**].

4.2 Saliency maps and model interpretation

There is a strong desire, especially in a task with societal implications like ours, to better understand how models are making their decisions. Thus our secondary approach and contribution to the prediction acceptance task is expanding upon model interpretability work by Suzgun et al. Prior work specifically explored integrated-gradient analysis of classifier models in the IPC Classification task. We originally contribute similar integrated-gradient analysis instead for acceptance prediction classifiers.

We decided to focus on Abstract sections because they are succinct and full summaries of the patent, and do not have to be truncated to be analyzed. Additionally, we decided to focus our attention on our DistilBERT models like prior work [9].

Using the Captum AI package, we wanted to get a visual understanding of attribution in our models. As such, we chose a popular method established by Sundararajan et al, known as Integrated Gradients. [24] Using Integrated Gradients is advantageous because it does not require us to make any modification to our model.

5 Experiments

5.1 Data

As discussed, we are using the Harvard USPTO Patent Dataset (HUPD). We chose this dataset for its extensiveness in year range and because it includes 34 metadata fields for each application. The dataset comes in the form of JSON files, and contain several fields for each patent, including patent number, IPC code, decision, Abstract, Claims, Summary, Description, and Title. However, we note that this dataset is especially difficult to use given its incredible size of 367 GB. Given we are first external users because it has yet to be released publicly, we worked with our mentor — the author of the dataset — to create solutions to problems we faced in loading, processing, and training with the dataset that can be re-used for future groups.

We chose to focus our efforts on building a universal classifier across all patents in all IPC subclasses. Given the size of such a dataset, we followed similar techniques from Suzgun et al. who only chose to train and run inference on applications from January 2011 to December 2016. To make our project more manageable, we chose to filter down even more to applications only from January 2011 to December 2013, and re-run the baseline models in this new subset as we will describe. This allowed us to make progress on the project given data and time constraints.

Like prior work in the acceptance prediction task, we chose to remove pending applications and those that were continued from prior applications. By specifically removing continued parent/child applications, *CONT-applications* in HUPD terminology, we avoid duplicate documents and keep our setup simple. Future work may instead use these continued applications to study changes in applications for the same invention over time, but this is out of scope for our paper.

To address the imbalance between accepted and rejected applications, we utilized prior work's code to use a weighted random sampler with a fixed seed to select samples for the training and validation sets, with a roughly 90-10 split (see Table 1 for a breakdown of the examples). In the validation set, like prior work, we ensured that there was an equal split of accepted and rejected applications to ensure that the true baseline accuracy would be 50%.

	Accepted	Rejected	Total
Training	379408	210811	**590219**
Validation	37175	37175	**74350**
Total	416583	247986	**664569**

Table 1: Breakdown of training and validation examples in our 2011-2013 subset. Note the equal distribution between accepted and rejected in the validation set.

Lastly, as mentioned earlier, because the Claims section averages 1271 tokens and BERT models can only take in 512 tokens, we follow prior work's decision to truncate the Claims section after 512 tokens. This technique is grounded in the semantics of patent applications, because while patents can have multiple claims, examiners tend to focus primarily of the first claim listed given it is often the most illustrative of a patentable idea [13].

5.2 Evaluation method

Given our validation set was balanced between accepted and rejected applications like prior work, we are using accuracy of predictions as our core evaluation method of our models. Additionally, we qualitatively analyze our saliency maps.

5.3 Experimental details

Naive Bayes: For our two Bernoulli Naive Bayes models, we trained for 5 epochs, with $alpha = 1.0$. It took an hour to process all 664569 applications and split them into our training and validation sets.

The Claims model took 10 minutes to tokenize the data, an hour to train, and another hour to run inference; the Abstract model took 5 minutes to tokenize the data, a half hour to train, and another half hour to run inference.

Ensemble Model 1: For this model we finteuned our linear classifier for 3 epochs. We used a learning rate of 0.001 and a batch size of 64. With these hyperparemters we saw a steady reduction in loss which plateaued towards the end of the 3rd epoch. The training time for this model was roughly 4.5 hours.

Ensemble Model 2: As we had trained all of the models in this ensemble model we did not perform any further training. We simply ran inference for all four models on the same validation set and applied the two voting algorithms discussed earlier.

5.4 Results

Model	BernNB *A	BernNB *C	DistilBERT *A	DistilBERT *C	Ensemble 1	Ensemble 2 (Naive)	Ensemble 2 (Weighted)
Accuracy	60.33	62.17	58.63	59.59	60.87	60.77	**62.65**

Table 2: The accuracies for universal IPC subclass acceptance classifier models in our 2011-2013 subset, with the top accuracy coming from Ensemble 2 (Weighted). Note that the *A and *C distinctions refer to models trained only on the Abstract and Claims sections, respectively.

5.4.1 Baselines

Given that we are using a smaller subset of years in our data and also expanding it to a universal classifier, we first chose to reestablish the baseline Bernoulli Naive Bayes models. We trained a simple Naive Bayes model like Suzgun et al. and evaluated on our subset and received similar baselines around 60.33% when trained on Abstract alone and 62.17% when trained on Claims alone. It is thus worthwhile to note that even when we generalize the Naive Bayes to be a universal classifier across all IPC subclasses, that it performs roughly similarly to Naive Bayes models trained on specific IPC subclasses from prior work [9].

5.4.2 Ensemble Model 1

We were very impressed with the performance of this model. The classification accuracy of 60.87 outperformed both DistilBERT models. This supports our prior belief that access to more information in the patent leads to a better understanding of its classification. This form of ensembling could be crucial in develpoing a robust classification system. By combining more parts of the patent in a similar way we could train a model that is able to look at and weight the importance of all the information just like a human patent examiner would.

5.4.3 Ensemble Model 2

The results from Ensemble 2 were unsurprisingly the best we found. By combining all the models together it would make sense that we could capture more representative capacity. What was surprising, however, was how much of a difference a small amount of weighting could do in overall accuracy which we see when comparing the Naive and Weighted approaches. We conclude that this type of ensembling should be further explored, as it has a lot of potential for growth for even higher accuracy.

6 Analysis

6.1 Saliency maps and model interpretation

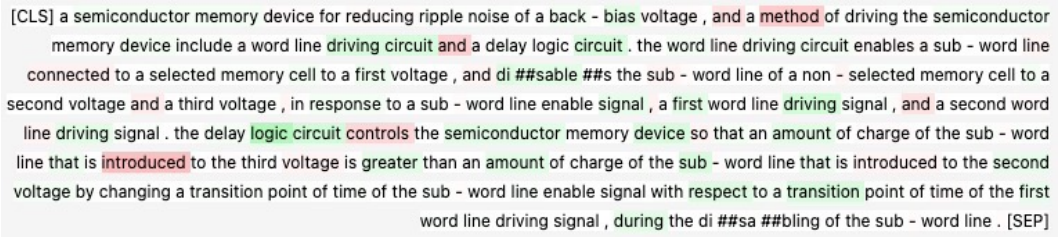

Figure 3: Saliency example of the Abstract section of an accepted H01L (Semiconductor Devices) subclass application, with tokens' impact highlighted. This was correctly predicted as being accepted.

Looking at Figure 3's saliency analysis of an accepted application, the DistilBERT model found that technical words such as *circuit*, *semiconductor*, *device*, and *logic* had strong contributions to a positive classification. Words that introduced novelty such as *first* also had a strong effect on positive classification. Note that certain words such as *method*, *introduced*, and *controls* were penalized. This is possibly due to the fact that these words are often in many patent applications, even somewhat generic, and thus the DistilBERT may have penalized them. We see similar patterns hold across more examples, that can be found in Appendix A.

Prior literature has looked at a similar saliency analysis for the related patent task of IPC classification. It found that the tokens that the model prefers are, as one would assume, the tokens most indicative of the technology area [9], with words like *device*, *semiconductor*, *memory*, and *data* being the most popular tokens it gave attention to when predicting the H01L subclass label. It is worthwhile to note that in our prediction task, some of these same tokens (*semiconductor*, *device*) are also what the model is giving the most positive attention to for its acceptance prediction. Given that human examiners specialize in reviewing applications of certain IPC codes [25], we hypothesize that a human examiner is likely to approve applications that use more of the technical terms relevant to the domain. That is, we hypothesize that technical terms are positively focused for acceptance prediction because they are a good indicator of how well a patent fits into its related area, a relevant criteria for acceptance; our initial findings here motivate further work.

6.2 Comparison of DistilBERT vs. Naive Bayes

The fact that Naive Bayes outperformed the DistilBERT transformer model in universally classifying both Abstract and Claims is extremely surprising. A Transformer model should be able to learn long term sentence structures present within the complicated language of the patent. The fact it is outperformed by Naive Bayes, however, suggests that DistilBERT was not able to do this and also was not able to achieve as rich of a token level semantic understanding as Naive Bayes. One hypothesis we wanted to test is whether or not DistilBERT simply ended up learning a classifier that solely relied on isolated word-level tokens, much like Naive Bayes. To test this we compared all the prediction labels of the DisilBERT models with the prediction labels of our Naive Bayes models. Note that we did this on the same validation set. Surprisingly, we found that there was only a 65% similarity in prediction, meaning that the two models had learnt fairly different representations of the patent data. This motivates future work on why else DistilBERT may not be able to perform significantly better than Naive Bayes.

7 Conclusion

In conclusion, we have found that that ensemble modeling of multiple models is a lightweight addition, but gives notable improvement to the accuracy of our patent acceptance prediction, especially when compared to DistilBERT models. Ensembling models focused on different sections of the patent produces an architecture that looks at more information, just like a real patent examiner. Whereas combining the predictions of different model architectures like Naive Bayes and DistilBERT with a

weighted voting algorithm, is an extremely fast and reliable method to incorporate different semantic representations of the same patent.

Our work in interpreting the model was also very interesting as it confirmed our prior belief about certain keywords we know to be important in assessing a patent's overall decision. It also elucidates the underlying decision process of our models which is crucial if any tool like this is to be used alongside human professionals.

There are many promising avenues for future work. As beta-testers of this to-be open-sourced repository, we are encouraged that our contributions to test and patch the code will help grow research in this area by improving its tools. With improvements to tools and models, another relevant task is modeling a better way of finding all similarly-semantic patents. Like word-vector mapping, it would be extremely helpful if given a new application, one could find all the closely clustered patents with similar semantics. These prior art searches are one of the most time-consuming parts of an application, and thus the impact here is great. Lastly, this work can be broadened outside the patent domain. By encoding deep, technical computational understanding of the patent "legalese" language, future work can explore how well models can be extended to other technical legal domains where experts are also in short supply.

References

[1] USPTO. Fy 2021 performance and accountability report. Technical report, 2021.

[2] Ralf Krestel, Renukswamy Chikkamath, Christoph Hewel, and Julian Risch. A survey on deep learning for patent analysis. *World Patent Information*, 65:102035, 2021.

[3] Xiao-Lei Chu, Chao Ma, Jing Li, Bao-Liang Lu, Masao Utiyama, and Hitoshi Isahara. Large-scale patent classification with min-max modular support vector machines. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 3973–3980. IEEE, 2008.

[4] Kyle Jensen, Balázs Kovács, and Olav Sorenson. Gender differences in obtaining and maintaining patent rights. *Nature Biotechnology*, 36(4):307–309, 2018.

[5] Evelina Gavrilova and Steffen Juranek. Female inventors: The drivers of the gender patenting gap. *Available at SSRN 3828216*, 2021.

[6] USPTO. U.s. patent statistics. Technical report, 2020.

[7] J.Y. Rob Srebrovic. Leveraging the bert algorithm for patents with tensorflow and big query. Technical report, Google, 2020.

[8] USPTO. Intellectual property and the u.s. economy: 2016 update. Technical report, 2016.

[9] Mirac Suzgun, Luke Melas-Kyriazi, Suproteem Sarkar, Scott Duke Kominers, and Stuart M. Shieber. The Harvard USPTO Patent Dataset: A Large-Scale, Well-Structured, and Multi-Purpose Corpus of Patent Applications. (in review), 2022.

[10] Michael Freunek and André Bodmer. Bert based patent novelty search by training claims to their own description. *arXiv preprint arXiv:2103.01126*, 2021.

[11] Jieh-Sheng Lee and Jieh Hsiang. Patent claim generation by fine-tuning openai gpt-2. *World Patent Information*, 62:101983, 2020.

[12] Julian Risch and Ralf Krestel. Learning patent speak: Investigating domain-specific word embeddings. In *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, pages 63–68. IEEE, 2018.

[13] Otto Stegmaier. Measuring patent claim breadth using google patents public datasets. Technical report, Google, 2018.

[14] Florina Piroi, Mihai Lupu, Allan Hanbury, and Veronika Zenz. Clef-ip 2011: Retrieval in the intellectual property domain. 01 2011.

[15] Shaobo Li, Jie Hu, Yuxin Cui, and Jianjun Hu. DeepPatent: patent classification with convolutional neural networks and word embedding. *Scientometrics*, 117(2):721–744, November 2018.

[16] Eva Sharma, Chen Li, and Lu Wang. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy, July 2019. Association for Computational Linguistics.

[17] Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus, 2021.

[18] Sung-Bae Cho and Hong-Hee Won. Machine learning in dna microarray analysis for cancer classification. In *Proceedings of the First Asia-Pacific Bioinformatics Conference on Bioinformatics 2003-Volume 19*, pages 189–198, 2003.

[19] Yawen Xiao, Jun Wu, Zongli Lin, and Xiaodong Zhao. A deep learning-based multi-model ensemble method for cancer prediction. *Computer methods and programs in biomedicine*, 153:1–9, 2018.

[20] United States Patent and Trademark Office.

[21] Zhiquan Qi, Bo Wang, Yingjie Tian, and Peng Zhang. When ensemble learning meets deep learning: a new deep support vector machine for classification. *Knowledge-Based Systems*, 107:54–60, 2016.

[22] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter, 2019.

[23] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[24] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *CoRR*, abs/1703.01365, 2017.

[25] AW Charles, Nicholas A Pairolero, and Mike HM Teodorescu. Examination incentives, learning, and patent office outcomes: The use of examiner's amendments at the uspto. *Research Policy*, 50(10):104360, 2021.

A Appendix

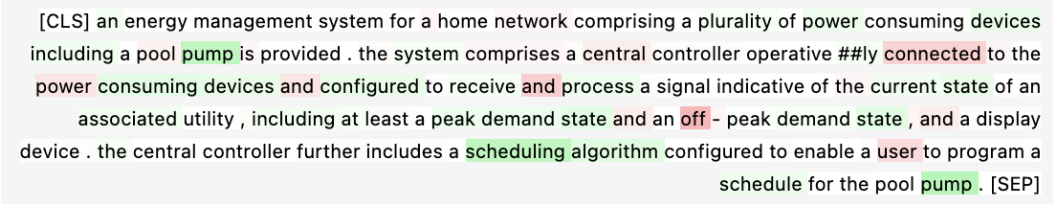

Figure 4: Saliency example of the Abstract section of an accepted G06F (Electrical Digital Data Processing) subclass application, with tokens' impact highlighted. This was correctly predicted as being accepted. Note the similar trend of how technical phrases like *scheduling algorithm*, *pump*, *devices*, and *peak demand state* are preferred over other more commonplace words like *connected*, *user*, *and*, and *off*.

,

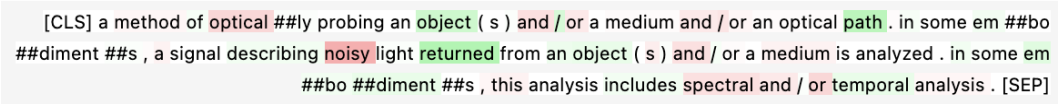

Figure 5: Saliency example of the Abstract section of a rejected H01L (Semiconductor Devices) subclass application, with tokens' impact highlighted. This was correctly predicted as being rejected. Note the lack of very technical words that one would expect in this subclass, and instead the presence of more commonplace words like *optical*, *noisy*, *or*, and *and* that the model associates negatively. Additionally, we can see how this Abstract perhaps presents less the technical use of an invention and more describes an upcoming analysis, which the model may be picking up on.

,