Towards Comprehensive Patent Approval Predictions: Beyond Traditional Document Classification

Anonymous EMNLP submission

Abstract

Predicting the approval chance of a patent application is a challenging problem involving multiple facets. The most crucial facet is arguably the novelty — *35 U.S. Code § 102* rejects more recent applications that have very similar prior arts. Such novelty evaluations differ the patent approval prediction from conventional document classification — Successful patent applications may share similar writing patterns; however, too-similar newer applications would receive the opposite label, thus confusing standard document classifiers (e.g., BERT). To address this issue, we propose a novel framework AISeer that unifies the document classifier with handcrafted features, particularly time-dependent novelty scores. Specifically, we formulate the novelty scores by comparing each application with millions of prior arts using a hybrid of efficient filters and a neural bi-encoder. Moreover, we impose a new regularization term into the classification objective to enforce the monotonic change of approval prediction w.r.t. novelty scores. From extensive experiments on the large-scale USPTO dataset, we find that our time-dependent novelty features offer a boost on top of the document classifier. Also, our monotonic regularization, while shrinking the search space, can drive the optimizer to better local optima, yielding empirical performance gains. Ex-post analysis of prediction scores further confirms that the document classifier and handcrafted features capture distinct sets of learning information.

1 Introduction

Securing patent approvals offers a major shot in the arm to inventors and innovators in the knowledge economy, increasing the chances of obtaining angel and venture capital investments. However, the process of getting a patent approved can cost applicants tens of thousands of dollars in payments to law firms who claim to be helpful in understanding what gets approved and improving the odds of success of a patent application. Algorithmic approaches to aid in the patent evaluation process can potentially save precious time and resources for applicants including inventors and lawyers during the patent application phase, as well as benefit patent examiners in government patent offices around the world who could use the tool to accelerate and improve the review process (Ebrahim, 2018).

The approval of a *patent application* is determined necessarily and sufficiently by the approval of *application claims*. Patent laws define individual claims as the subject matter of *inventions* (*35 U.S. Code* §112), on which "patentability" is defined (*35 U.S. Code §* 101, 102, and 103). Application claims prescribe the particular scopes of legal protection that the applicant is seeking and are the eventual objects for investigation under legal disputes or transfer of commercial rights. Patent examiners from the U.S. Patent and Trademark Office (USPTO) will make decisions on each application claim individually and independently with other sections as supporting materials. Therefore we focus on claim texts and use the term "*patent approval*" informally and interchangeably referring to "*claims approval*." In particular, we primarily consider *35 U.S. Code § 102*, assessing the *novelty* of application claims.

To the best of our knowledge, we are the first to predict patent approval, which is as an extremely challenging problem. Patent documents are legal, technical, often vague, abstract and difficult to parse, with writing conventions different from typical articles (Singer and Smith, 1967). Although AI/ML approaches are often discussed in the patent domain (Aristodemou and Tietze, 2018) such as in the area of information retrieval (Kang et al., 2007; Fujii, 2007; Shalaby and Zadrozny, 2019), applications of deep NLP methods are mostly concerned with classifying the content domains of patents (Verberne et al., 2010; D'hondt et al., 2013; Hu et al., 2016; Lee and Hsiang, 2019). In addition, extant literature usually explore approved

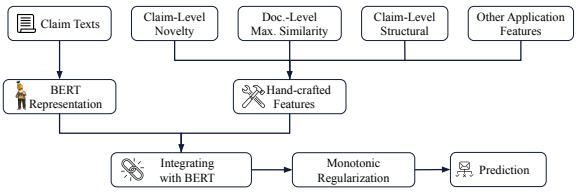


Figure 1: An overview of our proposed AISeer.

patents rather than applications (Balsmeier et al., 2018). Even to simply classify the topics of approved patents, state-of-the-art document classifiers can only achieve an accuracy of about 69.3% (only 2.2% over RoBerta) (Zaheer et al., 2020).

Compared with topic classification, our patent approval prediction task is much more challenging for these document classifiers, because the patent examination process tends to suffer from subjectivity and inconsistencies.[1] To mitigate the issue, we first develop several handcrafted features based on domain knowledge for use alongside the language model to provide contexts and control.

The time-dependent nature of the novelty also makes traditional document classifiers not suitable here, because they typically assume that similar instances belong to the same label. Rejections of claims by *35 U.S. Code § 102* require examiners to *cite* prior approved patent claims, *prior arts*, as evidence.[2] USPTO receives thousands of applications a week; thus a novel application at one time may be dramatically different in the assessment of novelty. This means that a classifier can pick up a positive label from an earlier approved application but receives a negative label from a similar but no longer novel application sometime later. Such conflicting information can confuse the classifier and undermine its performance.

To address this challenge, we propose a novel framework AISeer as shown in Figure 1. We formulate a time-dependent novelty score for each patent claim with its semantic similarity against prior approved claims from *patent grants*, which are final versions of *approved* patents. Specifically, inside a comprehensive pool comprising millions of grants, we consider those approved before the filing date of the focal application and then measure the maximum semantic similarity score of the focal patent claim matched with all approved claims in the time-dependent sub-pool. To improve computing efficiency, we apply document-level filters to narrow the sub-pool for each claim. After integrating such similarity scores along with the handcrafted features on top of BERT, experiments on the large-scale USPTO dataset demonstrate significant performance gains over fine-tuning a standard BERT alone. Intuitively, with all else equal, a patent claim with a higher similarity score, i.e., semantically more similar to prior approved claims, should be less likely approved. Hence we propose to impose monotonic regularization on the novelty score so that the loss function has an additional term of the hinge loss to further penalize non-decreasing predictions in the similarity. This effectively restricts the search space for the optimizer to prediction mechanisms that are reasonably consistent with the novelty measure. From our experiments, this regularization can help the optimizer steer away from unfavorable local optima and further improve AUROC.

In summary, our contributions are as follows.

- We collect patent application data from several data sections of USPTO and integrate full texts, metadata, office actions, rejections and citations data into a massive dataset;
- We develop a series of handcrafted features to aid the prediction of *35 U.S. Code § 102* approval decisions. In particular, we design and analyze a time-dependent feature that measures the novelty of patent applications at the time of filing;
- We propose to incorporate the handcrafted features and impose monotonic regularization on the novelty features and verify the effectiveness of the methodology in predicting patent approvals.

Reproducibility. We will release the benchmark dataset and our code on GitHub.

2 Problem Formulation and Benchmark

In this section, we formally formulate the novelty-based patent approval problem. We describe the experiment setup, the dataset, and baseline results with common document classifiers.

2.1 Problem Formulation

Each patent applications A_k, $k \in \{1 \cdots M\}$, sorted by filing dates, comprises of a number of application claims. Given text representation $\mathbf{X}_i$, $i \in \{1 \cdots N\}$, of each application claim, there exist $\{i_k\}$, $k \in \{0 \cdots M\}$ such that claim representa-

[1] https://www.ipwatchdog.com/2018/10/31/visualizing-outcome-inconsistency-uspto/id=102810/

[2] More details about the examination process can be found in *Manual of Patent Examining Procedure* at https://www.uspto.gov/web/offices/pac/mpep/index.html.

Table 1: Dataset Statistics. The approval ratio is calculated based on *35 U.S. Code § 102* labels.

	Train	Validation	Test
Applications M	216,101	175,597	153,632
Claims N	3.90M	3.07M	2.58M
Approval %	80.65	80.16	81.68
Time range	04/16-02/17	03/17-10/17	11/17-06/19

tions $\{\mathbf{X}_{i_{k-1}} \cdots \mathbf{X}_{i_k}\}$ belong to patent application A_k. *35 U.S. Code § 102* based binary labels y_i indicate approval decisions derived from patent examination history where $y_i = 1$ indicates approvals. We would like to classify application claims according to approval labels.

2.2 Benchmark Dataset Preparation

Dataset Collection. USPTO provides public data arranged in separate sources, including application and grant full texts, application metadata, citations, office actions, and rejections. Patent grants are final versions of approved patent applications. Later we will utilize grants for constructing the application novelty feature. To extract labels and create handcrafted features, we utilize both the legacy data system for office actions, rejections and citations made between 2008 and mid-2017 (Lu et al., 2017), and newer v2 APIs that cover mid-2018 onward. For application metadata, we obtain bulk data from PEDS (Patent Examination Data System).[3] In order to match all the available labels, we obtain weekly bulk releases for of both utility patent applications and utility patent grants in XML format ranging between 2005 and 2019. In total, we extract 8.8 million patent applications and 3.7 million patent grants during the same time period whose texts are around 730 GiB.

Patent applications are usually required to be published within 4 months after filing. Yet only one version among possibly a number of revisions is published. Next we identify the office actions and rejections associated with the published version by matching the closest action dates with publication dates minus 4 months, so that correct labels can be obtained. We identify the labels associated with the published version, and we then merge the different sources of data by the application number and ingest them into a DBMS. This way, we allow a model to predict for any version of a patent application so that the attorneys and applicants can evaluate their chances for decision making. We find out around 900K applications under which all corresponding sections of data are available. Because of the data size and to control for computation times, we choose the most recent, around 500K applications for experiments.

Dataset Splits. We split the data into training data, validation data, and testing data by their filing dates. The more recent patent applications are chosen for testing. The size for final experimental data, including the abstract, claim texts, labels, and handcrafted features, is around 15 GiB. For more details, see Table 1. The dataset is highly imbalanced towards positive labels.

2.3 Common Document Classifier Benchmark

Common Document Classifiers. We mainly evaluate the following common document classifiers.

- **Log. Reg.** refers to logistics regression using TF-TDF features.
- **Text-CNN** (Kim, 2014) with GloVe (Pennington et al., 2014) embeddings as the input. Adam optimizer with learning rate 0.001. 10 epochs' run; batch size as 1024;
- **LSTM** (Hochreiter and Schmidhuber, 1997) with GloVe embeddings as the input. AdamW optimizer with learning rate 0.005 and 10 epochs' run; batch size as 1024;
- **BERT** (Devlin et al., 2018) fine-tuning. AdamW optimizer with learning rate 5e-5 as the optimizer. The number of fine-tuning epochs as 5; batch size as 256. This is the the same model as in the state-of-the-art model, **PatentBERT**, in patent content classification (Lee and Hsiang, 2019) with a different set of hyper-parameters and balanced class weights. The original PatentBERT model is designed for a different task, and the experimental setting is not suitable for predicting patent approvals, hence we make the tweaks.

In all of the models, we impose class weights in the loss functions inversely proportional to the number of class instances, such that two classes are treated equally by the optimizer. For the details, please refer to Section 3.1. The neural models are trained with text inputs processed at a maximum length of 128 tokens per claim and on a single GPU.

Evaluation Metrics. Given the imbalanced nature of our dataset, we adopt both the Area Under the Curve for the ROC plot (Fawcett, 2004) (AUROC) and macro F1 score as our evaluation metrics. With AUROC, the predicting performance of the minority class could be taken into consideration with

[3] https://ped.uspto.gov/peds/

Table 2: Benchmark Results of Common Document Classifiers.

	AUROC %	Macro F1 %
Random Guess	50.00	50.00
Predicting All "1"	50.00	44.96
Log. Reg. (Tf-Idf)	58.94	54.54
TextCNN (GloVe)	59.70	55.58
LSTM (GloVe)	61.68	**56.95**
BERT (PatentBERT)	**61.79**	56.51

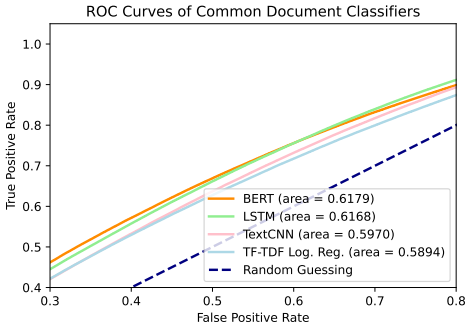


Figure 2: ROC Curves for Common Document Classifiers. BERT and LSTM are arguably the most effective ones.

a similar weight as for the majority class (in our case, positive class). Moreover, the probability-based metric can provide more detailed insights into model performances. Therefore, we choose **AUROC** as our **main metric**. The **macro F1** score is a direct average of F1 scores of both the positive class and the negative class and provides an alternative balanced view of both classes' performances. We treat it as a **secondary metric**. We compute the maximum macro F1 score (Lipton et al., 2014) by varying the decision threshold for each model. Other traditional measures focused on the positive class performance such as accuracy and recall have little practical implications due to data imbalance.

Benchmark Results. Table 2 shows common document classifiers' performance with some naive predictions as references. Results of neural models are reported with the median metrics among several runs with different optimizer random states. Figure 2 further visualizes more details of the ROC curves of these models. One can find that BERT and LSTM are arguably the most effective ones. Therefore, we will focus on BERT and LSTM for further comparisons.

3 Our AISeer Framework

Our AISeer framework unifies the document classifier, handcrafted features and monotonic regularization, as shown in Figure 1. It is compatible with

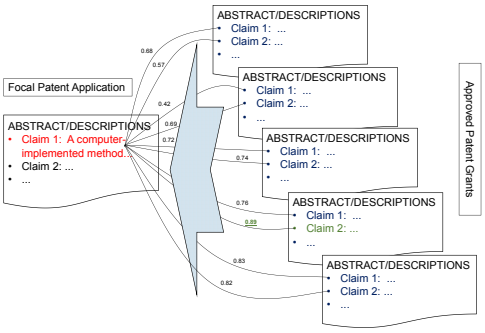


Figure 3: Illustration of Novelty Feature Construction.

almost all document classifiers. In this paper, we choose BERT as the base document classifier to demonstrate the effects as it is widely adopted and also performing well in our benchmark evaluations. After each application claim text is run through the BERT model, the output representation is concatenated with the corresponding handcrafted features. Our handcrafted features include a time-dependent claim-level novelty score, claim-level structural features, document-level similarity scores, and other application metadata features. We further impose a monotonic regularization on the impact of the claim-level novelty score so that the loss function has an additional term of the hinge loss.

3.1 Base Document Classifier

For the self-containess, we briefly introduce how we use BERT in AISeer. We first utilize BERT to transform the i-th application claim to a text representation $\mathbf{X_i}$ in batches of a size N_b, which is then passed to a linear layer to obtain the prediction through a softmax layer.

Approvals (i.e., $y_i = 1$) are much more popular than rejections (refer to Table 1), so the vanilla training will bias the model towards approvals. Therefore, we adopt a weighted loss for training: $\mathcal{L} = \sum_i -w_{y_i} \left(y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)\right)$ where w_{y_i} denotes the fixed weights of the two classes, which is inversely proportional to the number of instances from the corresponding class, balancing the training weights of the two classes.

3.2 Claim-Level Novelty Feature $\mathbf{N_{s,claim}}$

The backbone of the novelty feature is the *time-dependent* claim-level maximum similarity score.

We first index all patent grants with ElasticSearch (NV). Given a patent application and a claim under it, we first take advantage of its fast BM25-based document-level fuzzy matches to obtain 5 most similar grant documents to the focal

application document as a first-stage pre-filter. To account for time-dependence, each focal application is matched against a sub-pool of patent grants which are time-stamped to be approved strictly before the filing date of the focal application. In application level matching, all document sections are considered, including the abstract, summary of invention, details of invention of all claims.

Among all claims under the top-5 matched grants, we then find the most similar one to the focal claim using sentence-transformer (Reimers and Gurevych, 2019) with `stsb-roberta-large` pre-trained bi-encoder model. Base cross-encoder transformers such as BERT can lack in performance for pure semantic similarity tasks. Although certain cross-encoders have excellent semantic similarity performance, it can be computationally too demanding for our purpose since the scale of the claims in all patent grants is more than 100 million, and since each grant claim can be required to be paired many times with a focal application claim. The Elasticsearch-based pre-filter process also helps manage the computational need.

Figure 3 demonstrates how the time-dependent novelty feature is generated — the application that the red-highlighted focal claim belongs to is first matched with 5 patent grants on the application level; then the focal claim is matched against every claim under the 5 matched grants to compute the semantic similarity score, before the most similar grant claim is identified. Our experiments confirm that the claim-level maximum similarity score, as expected, is negatively correlated with *35 U.S. Code § 102* labels, as shown in Figure 4.

3.3 Application-Level Handcrafted Features

Application-Level Similarity. We consider the application-level maximum similarity score, denoted as $N_{s,doc}$, and mean similarity score generated by ElasticSearch (NV) as handcrafted features. These document-level scores measure how similar overall are the applications to the approved grants. The document-level similarity scores are positively correlated with *35 U.S. Code § 102* labels. We believe that they primarily capture the overall writing quality and structural resemblance. We will present further analyses and discussions in Section 4.2.

Features from Metadata. The USPTO dataset offers a rich collection of metadata about each patent application. We use the following two of them:

- *Patent Classification*: the USPC class designated for the applications. USPC is a system of classifying the subject matter of each patent application for recording, publication, and assignment purposes. Different classes of patents tend to have varying approval rates, as illustrated in Table 6 in the appendix.
- *Number of Applicant Cited References*: the number of citations of other patents or articles initiated by the applicant herself. In the patent domain, most *citations* are initiated by the examiners as "prior arts" to reject application claims. However, they can also be made by the applicant to demonstrate understanding of related work and claim contributions. The number of applicant-initiated citations is a signal of the effort and research the applicant puts in the application.

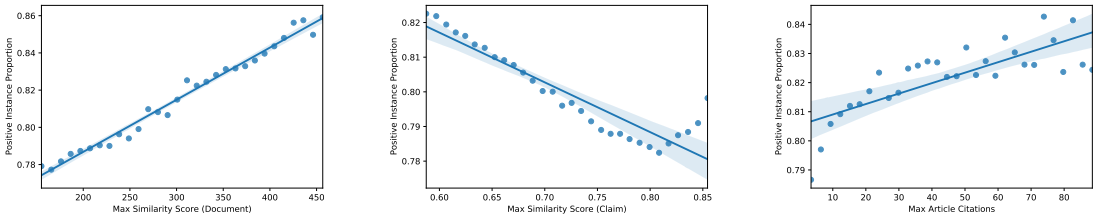
Other Application-Level Features are also considered for utility and writing as follows.

- *Max Citation*: based on ElasticSearch pre-filter, the maximum number of total citations among the top 5 most similar patent grant documents to the focal patent application.
- *Max Article Citation* is similar to the above. It refers to the maximum number of citations which are research articles (not other patents) in top matched grants.
- *Lexical Diversity*: the richness in the vocabulary of the abstract of the patent application.

3.4 Claim-Level Structural Features

We consider two indicators on how each claim is specified.

- *Component*: indicator on whether the application claim is describing the components of a system (e.g., a machine, a process, a compound). Other claims may describe the properties or utility of particular components. This is identifiable by the transitional phrases used in the claim.
- *Transitional Phrase*: indicator on whether a component claim is *open*, *closed*, or *half-open*, which is determined by which transitional phrase is used. Openness or closedness regulates the scope of legal IP protection the applicant enjoys once the patent is approved. Often it is a strategic choice by the applicant and the attorney. If a claim is *open*, indicated by transitional phrases "comprising" and legal synonyms, any additional components later added to the system are also protected, in contrast to closed claims . Open claims are, in turn, more difficult to be approved. These particular language phenomena are well-



(a) $N_{s,doc}$ (b) $N_{s,claim}$ (c) Max Article Citation

Figure 4: Handcrafted Features vs. Proportions of Positive 102-Labels. Features are grouped into bins for 10-90 percentile against mean positive label proportions.

known in the IP communities and sometimes referred to as "patentese" (Singer and Smith, 1967). The patent examination manual explicitly discusses the transitional phrases with case laws.[4]

3.5 Integrating with BERT

Now let $\mathbf{H_i}$ denote other handcrafted features in addition to $N_{s,claim}$ and $N_{s,doc}$. Figure 4 demonstrates the correlations between some representative handcrafted features and the positive label. Let $\mathbf{Z}_i = \mathbf{X_i} \cup \mathbf{H_i} \cup N_{s,claim} \cup N_{s,doc} \cup \{1\}$, $\forall i \in \{1, ..., N_b\}$. Note that X_i is the representation for the claim and that the document or application-level handcrafted features will be augmented to each claim. The concatenated Z_i will pass through the linear and the softmax layer.

3.6 Monotonic Regularization

Mathematically, we would like to restrict the search space upon $N_{s,claim}$, regularizing predictions to be decreasing in it. Let $\tilde{\mathbf{Z}}_\mathbf{i}$ denote all other inputs except $N_{s,claim}$. We would like to manipulate the input such that inconsistency with the monotonicity in $N_{s,claim}$ is represented. For a positive constant $C(0 < C < 1)$ let $N'_{s,claim} = CN_{s,claim}$, let $\mathbf{Z}'_\mathbf{i} = \tilde{\mathbf{Z}}_\mathbf{i} \cup N'_{s,claim}$. Given log-likelihood with respect to $\mathbf{Z_i}$,

$$F(\mathbf{Z_i}) = y_i \log \hat{y}_i(\mathbf{Z_i}) + (1 - y_i) \log(1 - \hat{y}_i(\mathbf{Z_i})),$$

we shall constrain $F(\mathbf{Z_i}) < F(\mathbf{Z}'_\mathbf{i})$. To implement it, we shall impose a hinge loss penalty whenever $F(\mathbf{Z_i}) > F(\mathbf{Z}'_\mathbf{i})$ and return 0 when otherwise. Therefore, the final objective function becomes:

$$\mathcal{O} = \mathcal{L} + \lambda \sum_i \max\left\{0, F(\mathbf{Z_i}) > F(\mathbf{Z}'_\mathbf{i})\right\},$$

where λ determines the regularization strength.

[4]Refer to patent glossary https://www.uspto.gov/learning-and-resources/glossary and examination manual https://www.uspto.gov/web/offices/pac/mpep/s2111.html#d0e200824

4 Experiments

We mainly compare **AISeer** with two models, **BERT** and **LSTM**, as they are the best common document classifiers from our benchmark results. For ablation study purpose, we also compare with **Log. Reg. Feat. Only**, a logistics regression model with handcrafted features only, and **AISeer w/o Regu.**, which is a BERT model integrated with our handcrafted features but not regularized by our monotonic constraints. AISeer is trained with the same set of hyper-parameters as BERT: maximum length for the tokenizer as 128, the number of fine-tuning epochs as 5; batch size as 256; AdamW with learning rate being 5e-5 as the optimizer. The monotonic regularization parameters C is $\frac{1}{2}$ and λ is 5e-4.

4.1 Empirical Results

With the setup above, we conducted several empirical tests. The results are shown below in Table 3.

Baseline BERT model gives decent AUC (ROC) and macro F1. The introduction of handcrafted novelty feature along with other computed ones together helps both the metric dimensions: AISeer boosts AUROC by around 2.5% percent and macro F1 by around 1% compared to the best common document classifiers. Figure 5 shows the AUROC improvement originates consistently from the entire spectrum of prediction scores. The reported numbers are medium results from multiple runs with the setup. We find the pattern of the results consistent.

Aforementioned in the introduction, wven to simply classify the topics of approved patents, state-of-the-art document classifiers can only achieve an accuracy of about 69.3% (only 2.2% over RoBerta) (Zaheer et al., 2020). Therefore we believe a 2.5% performance margin is substantial for the patent domain given its difficulty, especially given the relatively low dimensionality of hand-

Table 3: Evaluation Results of AISeer, Compared Methods, and Ablations.

	AUROC%	**Macro F1%**
LSTM (GloVe)	61.68	56.95
BERT (patentBERT)	61.79	56.51
AISeer	**64.14**	**57.92**
Log. Reg. Feat. Only	60.45	55.47
AISeer w/o Regu.	63.71	57.73

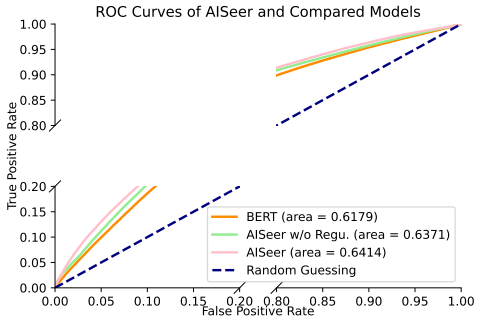


Figure 5: ROC Curves for AISeer and Compared Models. Only the two ends are visualized because we want to zoom in enough while saving space.

crafted features compared to BERT.

Log. Reg. Feat. Only result in the lower half of Table 3 indicate the necessity of a language model. Neither a language model only nor handcrafted features only could yield satisfactory performance. Also , AISeer w/o Regu. result shows that adding monotonic regularization on the novelty feature is effective. The relatively small performance gain over monotonic regularization may be attributed to the compromised precision of the novelty feature due to the use of the ElasticSearch pre-filter for the sake of computational costs. We would like to emphasize that the combination methodology has great potential.

In the next section, we will discuss in depth what the model framework has learned from the language model, the handcrafted features, and monotonic regularization, respectively.

4.2 Ex-Post Analysis for AISeer

Learning from Handcrafted Features.

Empirical results in Table 3 demonstrate that handcrafted features improve on best common document classifiers by about 2%. One may ask whether the handcrafted features have contributed significantly given the moderate improvement. Granted, application full texts may also contain signals for the patent class and applicant efforts that may partially reflect handcrafted features and the document classifier such as BERT may pick up.

Table 4: Regression Analysis of Prediction Scores on Handcrafted Features.

	BERT	AISeer w/o Regu.	AISeer
No. of Applicant Cited Refs	-3.5e-06*** (9e-7)	-8.2e-06*** (1e-6)	4.3e-06*** (8e-7)
Transitional Phrase - Open	-0.045*** (0.000)	-0.037*** (0.000)	-0.067*** (0.000)
Transitional Phrase - Closed	-0.015*** (0.000)	-0.022*** (0.000)	2e-4 (0.000)
Max Article Citations	1.9e-5*** (7e-7)	2.5e-5*** (7e-7)	3.2e-5*** (5e-7)
$N_{s,doc}$	2e-4*** (6e-7)	4e-4*** (6e-7)	2e-4*** (4e-7)
$\mathbf{N_{s,claim}}$	-0.18*** (0.001)	-0.17*** (0.001)	-0.21*** (0.001)
$\mathbf{R^2}$	0.085	0.125	0.189

Notes: HC1 heteroskedasticity-robust standard errors used. Not all regressors shown. *** 1% significance level.

To shed light on how AISeer learns from handcrafted features, we run linear regressions for the model prediction scores on handcrafted features for interpretable insights and present statistical results, as shown in Table4. Even prediction scores under BERT are significant in all handcrafted features, showing that BERT does learn knowledge overlapping with the handcrafted features to some extent. However, low R^2's indicate that knowledge from the deep neural model and knowledge from handcrafted features are quite distinct.

The dramatic R^2 increase from 0.085 to 0.189 shows that AISeer captures handcrafted features much more effectively than BERT. About 19% of knowledge of AISeer corresponds to handcrafted features, a 10% increase over BERT. Also, AISeer corrects incorrect coefficient signs from BERT. Intuitively, the chance of approval shall increase with in the number of applicant cited references. However, BERT is negatively correlated with it statistically significantly. Under AISeer, the direction of the effect is revered to match intuitions.

Learning from Monotonic Regularization. According to Table 4, our claim-level novelty feature $N_{s,claim}$ has the most significant impact. The use of monotonic regularization alone boosts the R^2 significantly, indicating that the approach also helps the model learn from handcrafted features overall.

We also evaluate the Spearman correlation coefficients of the probability prediction scores produced by the models with the claim-level novelty feature Pearson correlations with the document-level simi-

Table 5: $N_{s,doc}$, $N_{s,claim}$ vs. Prediction Scores Correlations.

	BERT	AISeer w/o Regu.	AISeer
$N_{s,doc}$ (Pearson)	0.128	0.238	0.180
$N_{s,claim}$ (Spearman)	-0.0788	-0.0230	-0.103

larity score. Spearman correlations measure how monotone two variables are correlated. According to Table 5, first we can confirm that applying monotonic regularization significantly pushes the prediction scores to be more monotonically decreasing in the core novelty feature – the Spearman correlation shifts from -0.0230 to -0.103. However, compared to the BERT, the regularization effect is less prominent. Observe that adding handcrafted features will actually steer the monotonicity into the opposite direction. Our regularized AISeer model manages to both benefit from the novelty feature and incorporates knowledge from other handcrafted features.

We believe the novelty feature should be only considered under contexts and will not perform well on it's own. First, novelty can be a subjective concept and may vary according to different types of claims, openness of claims, the department (category), etc. Second, novelty as practically measured by dis-similarity, can be easily achieved by poorly written random content, thus structural or overall similarity is also important. However, the observations indicate that there are potential conflicts between the novelty feature and other handcrafted features. While the latter helps with prediction performance on their own and provide contexts for the novelty feature thus imperative, it will also attenuate the effects of the regularized novelty feature. We leave this challenge for future work.

5 Related Work

To our knowledge, our work is the first in predicting patent approvals according to the examination procedures at the government patent office. Few extant researches attempt to predict decisions in office. (Winer, 2017) studies PTAB (Patent Trial and Appeal Board) hearing decisions at USPTO. Other related work addresses patent quality in a general and broad sense (Wu et al., 2016). More broadly in the IP/patent domain, although AI/ML applications have been often advocated (Ebrahim, 2018), studied (for a review see (Aristodemou and Tietze, 2018)) or implemented in practice (Lu et al., 2017), most work focus on determining patent content classes to save manpower or concern only with patent grants rather than applications (Verberne et al., 2010; D'hondt et al., 2013; Hu et al., 2016; Balsmeier et al., 2018; Lee and Hsiang, 2019). Recent studies (Hsu et al., 2020) emerge aiming at predicting patent transfers and the economic value.

Other streams of related work include those exploring patent similarity. Our approach of constructing the novelty feature with a state-of-the-art neural bi-encoder (Reimers and Gurevych, 2019) is significantly more advanced than relatively rudimentary approaches in the extant literature, such as text matching and frequency-based methods (Younge and Kuhn, 2016; Arts et al., 2018; Shahmirzadi et al., 2019). Studies on semantic analysis and representation of technology (Kim et al., 2016; Strumsky and Lobo, 2015) based on patent data are also related.

6 Conclusions and Future Work

In this paper, we tackle the challenging problem of predicting patent approval decisions as per *35 U.S. Code § 102*, namely the novelty-based decisions. We have prepared a large-scale benchmark dataset by consolidating different data sources from USPTO. From the evaluations of the popular document classifiers, BERT and LSTM are arguably the most effective ones. We identify the time-dependent challenge of the novelty judgement, and therefore propose AISeer, a novel framework going beyond the traditional document classifiers. Specifically, we construct a claim-level core novelty feature along with several other handcrafted features and apply them on top of the pre-trained BERT model. We further propose to add the monotonic regularization on the core novelty feature to resolve the potential label conflicts caused by the mechanism of the patent examination process. Experimental results have verified the superiority of AISeer and also the effectiveness of introducing novelty features and monotonic regularization.

We believe that our work is beneficial to various parties, including patent applicants, attorneys, examiners and regulators. While the advantages of our regularization methodology are significant, there is still room for potential metric improvements, thus further developing the work will yield opportunities for promising future research and greater contributions to the communities. In future, it is also important to extend the scope from claims to the other sections in the patent applications. Model interpretability is another direction that worthies exploring.

References

Leonidas Aristodemou and Frank Tietze. 2018. The state-of-the-art on intellectual property analytics (ipa): A literature review on artificial intelligence, machine learning and deep learning methods for analysing intellectual property (ip) data. *World Patent Information*, 55:37–51.

Sam Arts, Bruno Cassiman, and Juan Carlos Gomez. 2018. Text matching to measure patent similarity. *Strategic Management Journal*, 39(1):62–84.

Benjamin Balsmeier, Mohamad Assaf, Tyler Chesebro, Gabe Fierro, Kevin Johnson, Scott Johnson, Guan-Cheng Li, Sonja Lück, Doug O'Reagan, Bill Yeh, et al. 2018. Machine learning and natural language processing on the patent corpus: Data, tools, and new measures. *Journal of Economics & Management Strategy*, 27(3):535–553.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Eva D'hondt, Suzan Verberne, Cornelis Koster, and Lou Boves. 2013. Text representations for patent classification. *Computational Linguistics*, 39(3):755–775.

Tabrez Y Ebrahim. 2018. Automation & predictive analytics in patent prosecution: Uspto implications & policy. *Ga. St. UL Rev.*, 35:1185.

Tom Fawcett. 2004. Roc graphs: Notes and practical considerations for researchers.

Atsushi Fujii. 2007. Enhancing patent retrieval by citation analysis. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 793–794.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Po-Hsuan Hsu, Dokyun Lee, Prasanna Tambe, and David H Hsu. 2020. Deep learning, text, and patent valuation. *Text, and Patent Valuation (November 16, 2020)*.

Mengke Hu, David Cinciruk, and John MacLaren Walsh. 2016. Improving automated patent claim parsing: Dataset, system, and experiments. *arXiv preprint arXiv:1605.01744*.

In-Su Kang, Seung-Hoon Na, Jungi Kim, and Jong-Hyeok Lee. 2007. Cluster-based patent retrieval. *Information processing & management*, 43(5):1173–1182.

Daniel Kim, Daniel Burkhardt Cerigo, Hawoong Jeong, and Hyejin Youn. 2016. Technological novelty profile and invention's future impact. *EPJ Data Science*, 5(1):1–15.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Jieh-Sheng Lee and Jieh Hsiang. 2019. Patentbert: Patent classification with fine-tuning a pre-trained bert model. *arXiv preprint arXiv:1906.02124*.

Zachary C Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. 2014. Optimal thresholding of classifiers to maximize f1 measure. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 225–239. Springer.

Qiang Lu, Amanda Myers, and Scott Beliveau. 2017. Uspto patent prosecution research data: Unlocking office action traits.

Elastic NV. The heart of the free and open elastic stack.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Omid Shahmirzadi, Adam Lugowski, and Kenneth Younge. 2019. Text similarity in vector space models: a comparative study. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 659–666. IEEE.

Walid Shalaby and Wlodek Zadrozny. 2019. Patent retrieval: a literature review. *Knowledge and Information Systems*, pages 1–30.

TER Singer and Julian F Smith. 1967. Patentese: A dialect of english? *Journal of Chemical Education*, 44(2):111.

Deborah Strumsky and José Lobo. 2015. Identifying the sources of technological novelty in the process of invention. *Research Policy*, 44(8):1445–1461.

Suzan Verberne, EKL D'hondt, NHJ Oostdijk, and Cornelis HA Koster. 2010. Quantifying the challenges in parsing patent claims.

David Winer. 2017. Predicting bad patents: Employing machine learning to predict post-grant review outcomes for us patents.

Jheng-Long Wu, Pei-Chann Chang, Cheng-Chin Tsao, and Chin-Yuan Fan. 2016. A patent quality analysis and classification system using self-organizing maps with support vector machine. *Applied soft computing*, 41:305–316.

Kenneth A Younge and Jeffrey M Kuhn. 2016. Patent-to-patent similarity: A vector space model. *Available at SSRN 2709238*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. In *NeurIPS*.

A An Example Patent Application

TITLE: Data labeling for deep-learning models

ABSTRACT: A first and second scoring endpoint with payload logging are deployed. At the second scoring endpoint, native data and a user-generated score for the native data are received, the native data is pre-processed into readable data for the deep-learning model, and the user-generated score and the readable data are output to the first scoring endpoint, which is associated directly with the deep-learning model...

BACKGROUND: The present disclosure relates generally to the field of deep-learning models, and more particularly to evaluating and providing feedback data for deep-learning models.

The evaluation and feedback data labeling for deep-learning models, where the pre-processing code is embedded in the model, can be difficult to execute and accurately assess...

SUMMARY: Disclosed herein are embodiments of a method, system, and computer program product for evaluating and providing feedback data for deep-learning models.

A method, system, and computer program product may manage deep-learning models. A first and a second scoring endpoint with payload logging are deployed for a deep-learning model. At the second scoring endpoint, native data and a user-generated score for the native data are received...

DETAILED DESCRIPTION: Aspects of the present disclosure relate to deep-learning models, and more particularly to evaluating and providing feedback data for deep-learning models. While the present disclosure is not necessarily limited to such applications, various aspects of the disclosure may be appreciated through a discussion of various examples using this context.

An understanding of the embodiments of the present disclosure may be aided by describing examples in the context of a neural networking environment. Such as examples are intended to be illustrative, and not limiting in any sense.

When black box (e.g., deep-learning) models include pre-processing of raw data, it can be difficult to accurately and efficiently evaluate and label feedback data for retraining purposes. Conventionally, machine learning deployment systems (e.g., deep-learning models) have difficulty when defining/extracting the logic used to transform the training data into the format used by the model 1, because the pre-processing steps (e.g., image transformation, text vectorization, etc.) are usually not included in the machine learning model definition...

CLAIMS:

Claim 1: A computer-implemented method for managing deep-learning, the method comprising: deploying a first and a second scoring endpoint with payload logging for a deep-learning model; receiving, at the second scoring endpoint, native data and a user-generated score for the native data; pre-processing, at the second scoring endpoint, the native data into readable data for the deep-learning model; outputting, from the second scoring endpoint to the first scoring endpoint, the user-generated score for the native data and the readable data, wherein the first scoring endpoint is associated directly with the deep-learning model; outputting, from the second scoring endpoint to a payload store, a raw payload, wherein the raw payload includes the native data; processing, at the first scoring endpoint and using the deep-learning model, the readable data and the user-generated score to output a transformed payload and a prediction, respectively, to the payload store; matching, at the payload store, the raw payload with the transformed payload and the prediction to produce a comprehensive data set; evaluating the comprehensive data set to describe a set of transformation parameters; and retraining the deep-learning model to account for the set of transformation parameters.

Claim 2: ...

Claim 3: ...

...

B Example Approval Rates across Common Patent Classes

Figure 6 demonstrates the variations of approval rates in different patent classess, ranging from 63.1% to 93.2%, indicating the inclusion of patent class feature is critical.

Table 6: Example Approval Rates across Common Classes.

USPC Code	Application Counts	Approval Rate	Description
716	4425	63.10%	COMPUTER-AIDED DESIGN AND ANALYSIS OF CIRCUITS AND SEMICONDUCTOR MASKS
362	28054	75.59%	ILLUMINATION
257	151435	80.43%	ACTIVE SOLID-STATE DEVICES (E.G.,TRANSISTORS, SOLID-STATE DIODES)
375	44245	89.10%	PULSE OR DIGITAL COMMUNICATIONS
718	6848	93.17%	ELECTRICAL COMPUTERS AND DIGITAL PROCESSING SYSTEMS: VIRTUAL MACHINE TASK OR PROCESS MANAGEMENT OR TASK MANAGEMENT/CONTROL