PATENT APPROVAL PREDICTION
CSCI 567 - MACHINE LEARNING

Ryan Lee[*], **Eshan Bhargava**[*], **Sathya Raminani, Edgar Prieto**
Viterbi School of Engineering - Department of Computer Science
University of Southern California
Los Angeles, CA 90007, USA
{ryantlee,ebhargav,raminani,ejprieto}@usc.edu

ABSTRACT

Despite the large investments companies make to file patents, there are no predictors of patent outcome that account for both the text of the patent and the attributes of the patent. In this **application** project, we evaluate how well transformer-based classifiers, such as BERT, RoBERTa and BigBird, forecast patent outcomes on a claims text dataset of 113,281 patents. Separately, we assess predicting using Support Vector Machines (SVMs), logistic regression, and XGBoost with the numerical attributes of 1 million patents. For the text-based predictions, BigBird, designed for long-sequence lengths, far outperformed the alternative models, achieving an F1 score of 0.88 and an accuracy of 84%. On predictions with patent attributes, XGBoost was the best performing model with an F1 score of 0.94 and accuracy of 83%.[0]

1 INTRODUCTION

1.1 OBJECTIVE

The objective of this project is to develop a reliable method for predicting whether a given patent will be approved by the US patent office (USPTO). This is a classification task with features based on the legal text of the patent as well as numerical patent attributes, such as the success rate of the patent filer or agent. Since patents are only granted if a described invention is novel and non-obvious, the text-based model can serve as a proxy for assessing the novelty of the inventor's concept in the given technical domain. Additionally, in this project, we generate numerical feature data from a large set of patents and use traditional machine learning methods not limited to Support Vector Machines (SVM), Logistic Regression, and XGBoost to predict the outcome of a patent's approval.

1.2 IMPORTANCE OF PROJECT

The rejection of a patent's approval can severely hinder a companies' ability to protect their intellectual property and profit off of their innovation. Therefore, it is crucial that care is taken when writing a patent to maximize its approval odds. However, even some of the most reputable patent agencies are susceptible to errors that could result in the rejection of an otherwise novel idea.

This project aims to provide a means for predicting the outcome of a patent based on the analysis of patent text and numerical metadata features. The primary objective is to assist companies in creating patents with a high probability of approval. This will enable companies to refine and revise their patents until they achieve a level of confidence in their likely acceptance by the USPTO prior to submission.

[*]These authors contributed equally to this work.
[0]project repository: https://github.com/ml-ryanlee/patent-predict.git

2 RELATED WORK

2.1 RELATED WORK ON NUMERICAL MODELS FOR PATENT CLASSIFICATION

Joachims (1998) reported that using SVMs with Radial Basis Function (RBF) kernels led to an average accuracy of 0.86 in text classification tasks, specifically in categorizing text. This result underscores that SVMs consistently deliver strong performance in text categorization tasks, often surpassing the efficacy of most existing methods. Picca & Gay-Crosier (2021) used a mixed dataset consisting of textual, boolean, and numerical features to classify parts of text into 35 different categories. This approach, particularly the use of numerical features like part length and character ratios, can be adapted for analyzing patents based on word length and sentence complexity. The research presented in Marco et al. (2016) is especially pertinent for our numerical classification task. Their approach, using metrics such as the length and count of independent claims, directly aligns with methodology of numerical analysis of patent texts.

2.2 RELATED WORK ON TEXT BASED CLASSIFIERS

The BERT model, as detailed by Devlin et al. (2019) was previously used by Lee & Hsiang (2020) for the related task of classifying a patent's technical field by the claims text. Due to the highly technical nature of patent text, we hypothesized that BERT's capabilities may not be suitable for our task, as the dataset we use would generate a substantial amount of out-of-vocabulary data, which BERT has not been pretrained to handle. As a result, we looked into using RoBERTa, a variation of BERT which is designed to effectively manage out-of-vocabulary words (Liu et al. (2019)).

To resolve the text input limitations of BERT and RoBERTa, we looked into both abstractive and extractive summarization in order to capture full-context of a passage, rather than the text being abruptly cut off due to token length. BART, as described by Zaheer et al. (2020), uses a de-noising autoencoder trained to generate summaries. We experimented generating these summaries as a pre-processing step prior to classification with BERT and RoBERTa. As an alternative to generated summaries, we also investigated keyword and key sentence extraction with the TextRank algorithm Mihalcea & Tarau (2004), which ranks sentences by most words in common with other sentences.

There has been no prior work on patent classification that has explored using transformers designed for long sequences such as BigBird, as described by Zaheer et al. (2020). However, we find that such transformers perform best on the patent text dataset. We also looked into using LongFormer described by Beltagy et al. (2020) as alternative long sequence classifier.

3 PROBLEM FORMULATION

3.1 CLASSIFICATION TASK

Our task is to use information about a patent application, such as the written text and the patent's class, to predict whether the patent will be granted. We separate the features we predict with into text data and numerical data. Our initial approach is to train classifiers on either text or data, rather than combining the two types, to understand whether text or numerical data has more prediction power. To assess classifier performance on an imbalanced dataset, we used F1, precision, recall, as well as accuracy.

3.2 DATASETS

Two datasets from USPTO were used to extract the numerical and text features. The numerical features and the patent status, i.e. issued or abandoned, were extracted from the Patent Examination Research Dataset (PatEx), containing quantitative information of about 13 million patents from 1985 to 2015. We did not directly use the PatEx dataset for predictions. Instead we extracted numerical features based on the patent's profile which we call the metadata features. For example, patent agent success rate was calculated for every example by counting the number of patents granted prior and dividing by the number of patents submitted by the same agent in the PatEx dataset.

We extracted 113,281 patents and claims from 2005 which we used as our text feature dataset (100K dataset). Although it was also possible to also extract the technical description from the patent as text data, we chose to exclusively use the claims section of the patent. As background, the claims text is what is evaluated by the patent office and what is used to argue legal coverage in patent litigation. While the claims text was used primarily as the text dataset, other numerical features were generated with the claims, such as word count, number of unique words, and sentence count.

4 METHODOLOGY

4.1 TEXT DATASET ANALYSIS

Text data was extracted from the publicly available datasets from the USPTO, from early 2005. We focus on predicting based on the claims text, or the legal language used to claim patent coverage, rather than the technical description in the patent. The primary challenge with the patent text dataset was the amount of text per example, and the variability in the patent topics. As shown in Figure 1 below, a majority of patents have around 900 words in the claims text, with some patents having as many as 5000 words in their claims.

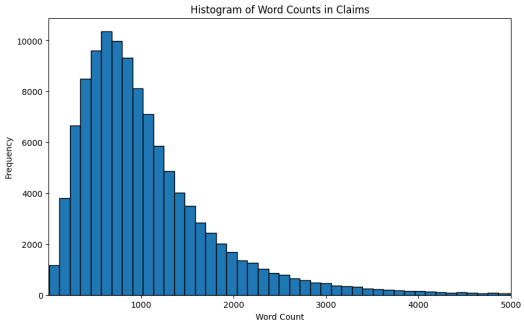

Figure 1: In 2005 patent text dataset, most patents have claims around 900 words

4.2 TEXT BASED CLASSIFIERS

We initially fine-tuned transformer based classifiers by directly inputting the patent text dataset. First, we fine-tuned a pretrained BERT model from the Hugging Face repository Wolf et al. (2020). We also fine-tuned RoBERTa, a variant of BERT generally recognized to improve upon BERT's performance across various tasks. These improvements are attributed primarily to RoBERTa's dynamic masking technique, which allows for different masked versions of sentences. As a baseline to our fine-tuned models, we used OpenAI's GPT-3.5 API for zero-shot classification on the patent text dataset.

Concerns about the truncation of the text input due to the memory/token limitation of BERT and RoBERTa led us to explore summarization and long-sequence transformers. We employed a technique called abstractive summarization with Bidirectional and Auto-Regressive Transformer (BART) to summarize the claims text, as shown in Figure 2. This allowed us to reduce the number of tokens of the input as well as potentially capture more of the meaningful context in the patent claims.

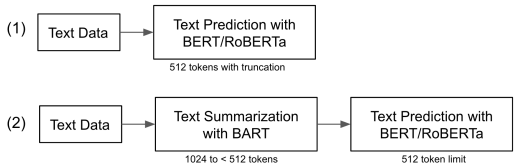

Figure 2: BART was trialed as pre-processing step to classification with BERT and RoBERTa

Finally, we fine-tuned long-sequence transformers such as BigBird and LongFormer for the patent classification task. Both BigBird and LongFormer models have attention mechanisms which scale linearly with sequence length, enabling much longer inputs, such as our patent claims text. Of the two, BigBird is the larger model to train, with sparse random, window, and global attention mechanisms.

4.3 Numerical Data Analysis

We separately trained classifiers with the metadata features such as examiner experience, technical class saturation, and patent filer experience. These features were calculated based on the USPTO PatEx dataset which contains information patent characteristics from 1984 to 2014. Figure 1 shows a feature example before normalization and standardization.

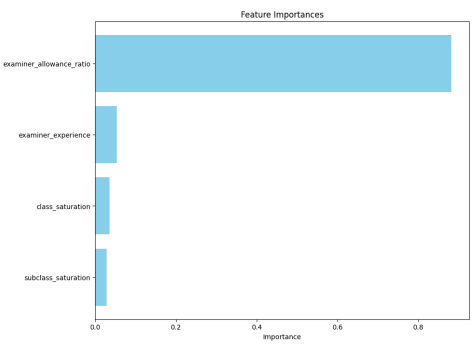

Figure 3: Metadata feature importance extracted from XGBoost model

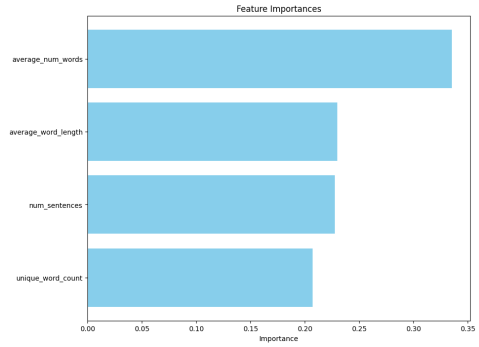

Figure 4: Linguistic feature importance extracted from XGBoost model

We also experimented with a new set of numerical features that were centered around the linguistic characteristics of the patent claim text. We hypothesized that factors like the average number of words per sentence and the number of unique words could affect the readability and complexity of a patent claim, which could in turn affect the likelihood of the examiner issuing the patent. We extracted four numerical features from the claims text: the average number of words per sentence, average number of letters in each word, number of sentences, and number of unique words per patent.

4.4 Numerical Classifiers

Preliminary results for the metadata feature classifiers were obtained using the last 1,000,000 training examples from the numerical dataset, utilizing XGBoost, SVM, and Logistic Regression models. Results for the linguistic feature classifiers were obtained by extracting numerical features from the 100K text dataset and training on them. The models selected included XGBoost, SVM, Logistic Regression, and Random Forest.

5 Results and Discussion

5.1 Patent Prediction on Text

Table 1 presents the outcomes of using transformers on a curated set of 10,000 samples from the USPTO dataset. For this purpose, we utilized four model architectures: BERT, RoBERTa, BigBird, and GPT 3.5 (zero-shot).

The initial result showed us that RoBERTa performed similarly to BERT. We hypothesized that the limited performance stemmed from the necessary truncation of lengthy patent claims, leading to predictions based on incomplete context. RoBERTa and BERT have comparable performance, showing that the patent text data is complex enough to forego RoBERTa's general performance improvements.

Model	BERT	RoBERTa	BigBird	GPT 3.5
Accuracy	0.694	0.690	**0.823**	0.68
F1 Score	0.779	0.772	**0.878**	0.67
Precision	0.766	0.787	**0.828**	0.68
Recall	0.793	0.756	**0.935**	0.56

Table 1: Unshortened Claims (10K)

Moreover, using abstractive text summarization with BART in conjunction with BERT and RoBERTa did not yield substantial improvements in predictions, as evidenced by a mere 0.01% change in accuracy on the subset of 10,000 patents. This outcome suggests that BART effectively condensed key semantic information from the claims that was crucial for classification performance, resulting in only minimal changes in performance metrics. While this indicates progress in applying abstractive text summarization to our task, we believe that more advanced models might be capable of generating richer summaries, potentially leading to enhanced performance.

These initial transformer results prompted us to look into transformers that are better equipped to deal with long passages and documents, which led us to find the single, dominant model in our study, Big Bird. Training this model on the 10,000 patent subset, we were able to get a validation accuracy of 0.83 and an F1 score of 0.88, which vastly outperformed previous models. We then proceeded to fine-tune BigBird with the 100K, full patent text dataset, which pushed the model to not over-fit on the smaller subset. After further fine-tuning, BigBird ultimately achieved the same superior performance on the complete text dataset, with an **F1 score of 0.88** and an **84% accuracy** on 11,329 examples in the test dataset.[1]

The use of GPT-3.5 as a zero-shot classifier in our study has consistently resulted in the lowest performance across all recorded metrics. We hypothesize that fine-tuning this model could significantly enhance its performance, potentially making it at least comparable to, if not superior to, BERT and RoBERTa.

5.2 Patent Prediction on Metadata, Linguistic Features

The left table in Table 5.2 presents our findings from analyzing approximately 1,000,000 metadata samples. Remarkably, logistic regression, XGBoost, SVM, and a feed forward neural network all demonstrated equivalent performance. This similarity in outcomes reveals significant insights about the data, suggesting several possibilities. These include potential limitations such as insufficient data volume due to class imbalance for example, a lack of variability within the data, or the problem is simple enough so that these relatively basic classifiers yield the highest possible performance on this dataset. We hypothesize that this result is due to the nature of the numerical data because a highly flexible method, like a neural network has comparable performance to logistic regression.

Additionally, our initial attempts to enhance the performance of these traditional machine learning techniques through preliminary feature selection methods, like recursive feature elimination (RFE), coupled with data augmentation (SMOTE), did not yield successful results. RFE reduced the feature set to examiner allowance ratio alone, and performance decreased. Furthermore, an examination of the correlation matrix for the feature set revealed that correlations between variables primarily occur when they are correlated with examiner features. This strengthens the conclusion that the current set of metadata features exhibit slight inter-correlations, and incorporating each feature into the analysis is shown to enhance overall performance.[2] Therefore, the path forward may involve either deploying more advanced algorithms or enriching the set of metadata features to gain deeper insights.

In this respect, we decided to extract custom numerical features from the patent claims text. The right table in Table 5.2 shows the results of using traditional machine learning methods on this set of features. All of the models tested scored highly on F1-score and recall but more poorly on accuracy and precision. This is indicative of the fact that the model is classifying the vast majority of patents as accepted. The low accuracy and precision reflect the high level of false positives in the models' predictions. We estimate that there a few possible explanations for the high level of false positives.

[1]For BigBird fine tuning we split the dataset of 113,281 examples into 80/10/10 train/validate/test

[2]This correlation matrix can be seen in the codebase.

The first is the unbalanced nature of the patent dataset. In the 100k dataset used for training and evaluation of the linguistic models, roughly two-thirds of the entries consisted of patents that had been issued. This could cause the model to be biased towards the positive class, resulting in the high level of true and false positives and low level of true and false negatives. We believe that this problem could be addressed by experimenting with re-sampling techniques to balance the training dataset or adjusting the decision threshold to lower the number of false positives. The results could also be explained by our feature engineering. The features we chose to represent the important linguistic characteristics of the patent claim text may not have captured enough valuable information about the data. In the future, we could experiment with an alternate set of features, possibly incorporating aspects of the patent other than the claim text. For instance, we could try to engineer features to represent the readability and specificity of the patent title or use the number of figures in the patent as a feature. Another likely cause for the performance of these models could simply be noisy data. In the process of separating the patent claim texts into sentences and eventually words, there was a trade-off between obtaining more data and making sure that data was clean. For example, when we were splitting the patent claim texts into sentences, there were many instances where the sentences were incomplete because of the splitting character being present in the sentence. Perhaps with a much lengthier data cleaning process, we'd be able to rectify more of these errors that could be adding noise to our linguistic numerical dataset.

Model	XGB	SVM	Log. Reg.	FFNN
Acc.	0.832	0.840	0.833	**0.866**
F1	**0.943**	0.908	0.875	0.925
Prec.	0.8739	0.869	**0.933**	0.893
Recall	0.934	0.951	0.903	**0.960**

Model	XGB	SVM	Log. Reg.	RF
Acc.	0.666	**0.673**	**0.673**	0.661
F1	0.793	0.804	**0.875**	0.791
Prec.	**0.679**	0.673	0.673	0.676
Recall	0.953	**1.000**	**1.000**	0.952

Table 2: The left table shows the Numerical Feature Results. The right table has results for Linguistic Numerical Features.

6 Conclusion and Future Work

In this study, we employed both transformers and traditional machine and deep learning classifiers for predicting patent outcomes. The accuracy achieved was notable, standing at 82 percent with text data and 86 percent with numerical data. A key finding was that the examiner allowance ratio significantly predicts patent acceptance. Furthermore, preliminary results suggest that the incorporation of a pretrained NLP tokenizer, combined with pre-summarization, could enhance model performance. There is also a promising potential for additional improvements through extended training on larger data sets.

In terms of the linguistic numerical data, we are interested in experimenting with alternative numerical features that better capture information about the patent claim text. We'd also like to broaden our approach to include other parts of the patent, like the patent figures and title, to create a richer feature set. Additional data cleaning and preprocessing may also help to improve performance by clearing up noise in our dataset.

However, our current analysis suggests that the key to improvement lies within the text data, rather than the numerical data. To achieve this, we aim to compress the size of the text data while still retaining its essential semantic content. This objective could be effectively addressed by investigating methods like extractive text summarization and dimensionality reduction.

Lastly, the limited availability of GPU resources emerged as a significant bottleneck in this project, particularly given the essential need for such resources in handling text data. Should access to these GPU resources be increased, our aim would be to extend the training duration and increase the data volume for the transformer models. Additionally, considering the rich structure of graphs, we are interested in exploring the potential of graphical natural language processing techniques.

REFERENCES

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. URL https://arxiv.org/abs/2004.05150.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol (eds.), *Machine Learning: ECML-98*, pp. 137–142, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. ISBN 978-3-540-69781-7.

Jieh-Sheng Lee and Jieh Hsiang. Patent classification by fine-tuning bert language model. *World Patent Information*, 61:101965, 2020. ISSN 0172-2190. doi: https://doi.org/10.1016/j.wpi.2020.101965. URL https://www.sciencedirect.com/science/article/pii/S0172219019300742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

Alan C. Marco, Joshua D. Sarnoff, and Charles deGrazia. Patent claims and patent scope. *USPTO Economic Working Paper*, 2016. doi: dx.doi.org/10.2139/ssrn.2844964.

Rada Mihalcea and Paul Tarau. Textrank: Bringing order into texts. In *Proceedings of EMNLP 2004*, pp. 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Davide Picca and Cyrille Gay-Crosier. An automatic partitioning of gutenberg.org texts. In *International Conference on Language, Data, and Knowledge*, 2021. URL https://api.semanticscholar.org/CorpusID:237357300.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.

Manzil Zaheer, Guru Guruganesh, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*, 2020. URL https://arxiv.org/abs/2007.14062.