An Augmented Large Language Model for Patent Acceptance Prediction

Anh Ta [1] **Erin McGowan** [2] **Maksat Kuanyshbay** [3]

Abstract

Patent acceptance prediction is a relatively new application for Large Language Models (LLM). When making patent acceptance prediction, patent examiner might look at certain sections, paragraphs, or the whole patent documentation rather than paying attention to individual words. Building LLM which captures the hierarchical relationship and inter-attention between individual sections or sentences of a patent's documentation helps improve patent acceptance prediction. Our main contribution includes proposing PatentLLM - an augmented hierarchical Large Language Model leveraging Sentence-BERT and SciBERT for patent acceptance prediction with improved accuracy results compared to other well-known traditional pretrained language models benchmark by better capturing inter-sentence attention and the semantic structure of the entire patent set of claims documentation as well as reducing the impact of pretrained model's token limitations. As most patents are of scientific content, we integrated SciBERT word embedding model into our augmented hierarchical LLM structure. We also explored the usage of patent's metadata with respect to our model's performance. Our model PatentLLM, which employs this novel approach within the realm of patent classification, could help pre-screen patents for acceptance prediction or extends to other long form scientific texts for classification tasks. The code for this project can be found here: https://github.com/egm68/ML-Final_Project

Keywords: Patent Classification, Large Language Model, BERT, SciBERT, Hierarchical Transformers

[1]Department of Computer Science, New York University Courant School of Mathematical Sciences [2]Department of Computer Science and Engineering, New York University Tandon School of Engineering [3]Department of Mathematics, New York University Courant School of Mathematical Sciences.

1. Introduction

Previous approaches aiming to improve patent acceptance prediction utilize several BERT-based models (Devlin et al., 2018) to extract information from the patent documentation (Suzgun et al.) (Kearns et al.). However, previous approaches' shortcomings include 512 tokens limitations, lack of metadata usage, and inadequate training methodology. BERT-based models (Devlin et al., 2018) in previous approach is limited to only 512 tokens per patent (Suzgun et al., 2022) (Kearns et al.), potentially leaving out other valuable information. Furthermore, the original paper did not leverage the metadata variables included in the Harvard USPTO patent dataset (Suzgun et al., 2022); newer approach only utilized the examiner's ID number and year in which the patent was filed (Kearns et al.). In addition, the original study only trained models on a single popular International Patent Classification (IPC) subclass at a time with niche technical information (Suzgun et al., 2022).

Long documents contain internal hierarchical structure involving sections, paragraphs, and etc. (Ruan et al.). The "Claims" sections of patents exhibit this characteristic where each numbered sentence forms together to produce meaningful evidence for patent acceptance decision. Examples of leveraging hierarchical structure in Natural Language Processing (NLP) include encoding sentences into document vectors (Yang et al., 2016) and introducing inter-paragraph attention for text summarization (Liu & Lapata, 2019).

By incorporating inter-sentence attention and leveraging the high scientific content within each patent, we proposed PatentLLM - an augmented hierarchical LLM integrating Sentence-BERT and SciBERT for patent acceptance prediction. PatentLLM achieved the highest accuracy performance compared to other well-known LLMs in predicting patent acceptance. Our main contributions include:

- Overcome the 512 tokens limitation of the previous approaches by using a hierarchical transformer architecture to capture the semantic structure of the entire patent's set of claims documentation
- Incorporate significantly more metadata by utilizing 6 metadata variables (filing date, foreign/domestic, IPC

subclass, entity size, AIA first to file, and examiner art unit)

- Familiarize our model with a wider breadth of domain-specific knowledge because it utilizes SciBERT word embedding model trained on all IPC subclasses.

2. Background and Related Works

2.1. The Harvard USPTO Patent Dataset

In 2022, Suzgun et. al. introduced the Harvard USPTO Patent Dataset (HUPD) (Suzgun et al., 2022). The HUPD contains 4,518,263 utility patent applications filed to the United States Patent and Trademark Office (USPTO) between January 2004 and December 2018. Each patent is stored in JSON format and contains the following sections in its main text: title, abstract, claims, background, summary, and description. Each patent is also accompanied by 36 additional metadata variables including but not limited to examiner full name and ID number, various dates (filing date, patent issue date, and date published among others), various subject matter classification labels (IPC labels, CPC labels, USPC labels), various unique identifiers (WIPO number, PGPub number), and inventor descriptors such as foreign/domestic or entity size. Most crucial to our analysis is the "decision" label, which include patent's class: "ACCEPTED," "REJECTED," "PENDING," "CONT-ACCEPTED," "CONT-REJECTED," and "CONT-PENDING" (applications with parent filings in the USPTO continuity data files were marked with the prefix "CONT").

The HUPD is distinct from other NLP patent datasets such as CLEF-IP 2011 (Piroi et al., 2011), USPTO-2M (Li et al., 2018), and BIGPATENT (Sharma et al., 2019) because it includes patent applications, not just granted patents. This allows us to evaluate what makes a patent suitable for acceptance by looking at both successful and failed applications. The HUPD also includes multiple classes of rich textual and structural information present in patent applications (where other patent datasets may only contain patent abstracts or descriptions). The HUPD also obtained patent information directly from the USPTO rather than from Google's Patent search. Together, this dataset allows us to utilize the most accurate patent information for a wider variety of NLP tasks, particularly those that involve semantic meaning of the entire patent as opposed to 1-2 shorter sections.

2.2. Prior Uses of the HUPD

In their paper introducing the HUPD, Suzgun et. al. conducted binary patent acceptance prediction experiments by training a Bernoulli Naive Bayes classifier, Multinoulli Naive Bayes classifier, a logistic regression model, and a convolutional neural network from scratch as well as fine-tuning DistilBERT (Sanh et al., 2019), BERT (Devlin et al., 2018), and RoBERTa (Liu et al., 2019) models. All of their models were trained and evaluated on the patent applications filed to the USPTO between January 2011 and December 2016. Each of these models was trained in eight different ways, on either the abstract or claims section of patents from one of the four most popular IPC subclasses. The models trained on the claims sections generally yielded the best performance. Hence, we chose to focus on the claims section for our experiments. None of the individual accuracy scores from these original experiments went beyond 64%.

Kearns et. al. also investigated the topic of patent acceptance prediction using the HUPD (Kearns et al.). In addition to training a logistic regression model, a Bernoulli Naive Bayes model, and fine-tuning DistilBERT (Sanh et al., 2019) and RoBERTa (Liu et al., 2019) models, Kearns et. al. also fine-tuned an augmented DistilBERT model wherein they concatenated embeddings for patent examiner ID and filing year to the claims section of each patent prior to classification. The most accurate model in this study was the Bernoulli Naive Bayes model, which was able to achieve 64.37% accuracy.

We identified three main shortcomings of these previous approaches. For one, both the original paper from Suzgun et. al. and the study from Kearns et. al. only used BERT models that could only read 512 tokens from each patent (Suzgun et al., 2022) (Kearns et al.). The average word count per patent in our training and test set was around 900 words, which was 1.75x the tokens limit of a typical BERT-based model. Furthermore, the previous studies either did not leverage the metadata variables included in the Harvard USPTO patent dataset (Suzgun et al., 2022) or only utilized the examiner's ID number and year in which the patent was filed (Kearns et al.). Finally, the original study only trained models on a single popular IPC subclass at a time (Suzgun et al., 2022). The IPC subclasses are a subject matter classification system for patents. For instance, the IPC subclass "G06F" denotes patents related to Electric Digital Data Processing (Suzgun et al., 2022). Thus, models trained on a single IPC subclass would only be trained on a very specific niche of technical information.

3. Methods

Sentence-BERT is a LLM which derives semantically meaningful sentence embeddings using siamese and triplet network structures to reduce computational overhead (Reimers & Gurevych, 2019). SciBERT is a BERT-based LLM (Devlin et al., 2018) which was pretrained on large multi-domain corpus of scientific texts for downstream scientific NLP tasks (Beltagy et al., 2019).

Due to the limitations of the BERT model's input sequence length of 512 tokens as well as the high scientific con-

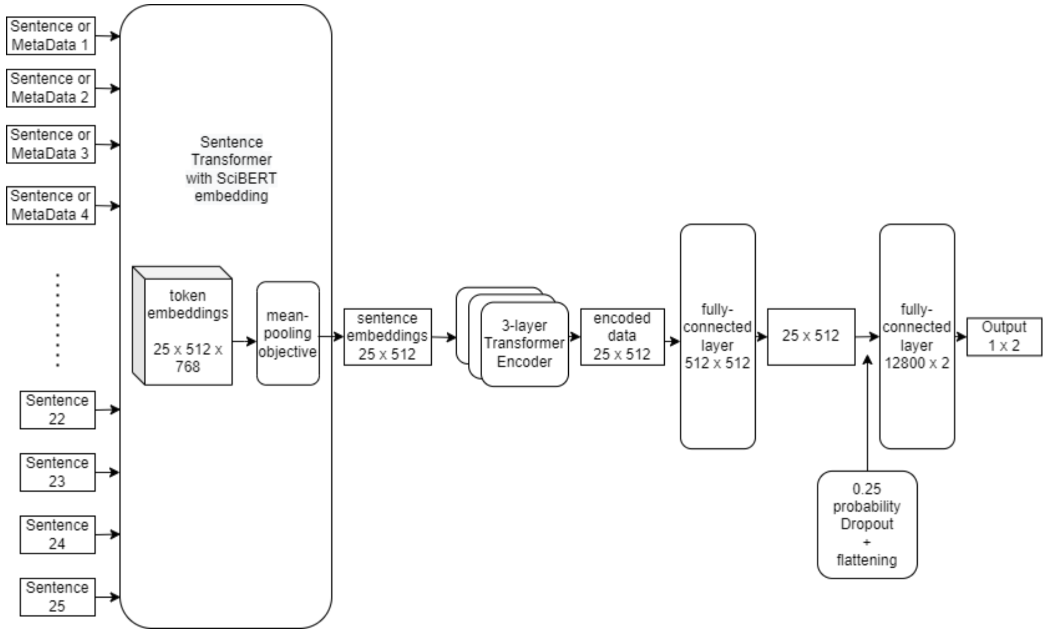

Figure 1. Proposed PatentLLM Architecture.

tent of the patent documentation, we decided to utilize Sentence-BERT to produce sentence embeddings with SciBERT word embedding model and obtain inter-sentence attention through transformer encoder layers. The model's architecture is shown in Figure 1 above.

3.1. Input Data

To leverage the metadata in the HUPD, we trained two models - one with the metadata and one without. Our model without the metadata was trained to read 25 sentences per patent. We observed that most of the patents include no more than 25 sentences as observed in Figure 2. Hence, by training on 25 sentences per patent, we have captured most of the sentences per patent. The other model with metadata was trained on 6 metadata variables and 25 sentences.

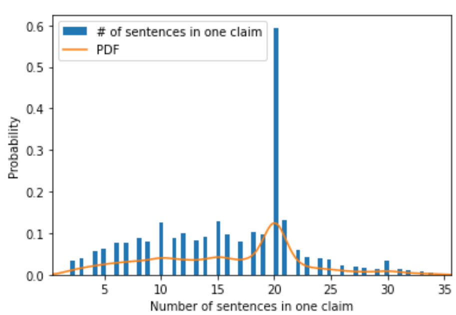

Figure 2. Sentences distribution in one claim

3.2. Sentence Embedding

We utilized SentenceTransformer from the `sentence_transformers` library, which is a Python framework for state-of-the-art sentence embeddings based on Sentence-BERT.

During the first phase, each sentence passed through the Sentence Transformer and each of the 512 tokens within these sentences were mapped into a vector of length 768. Therefore, 25 sentences were mapped into a 3-dimensional matrix of shape $25 \times 512 \times 768$, where 25 corresponds to the number of sentences read, 512 corresponds to the number of tokens (words or sub-words) within each sentence and 768 corresponds to the size of a word or sub-word embedding. During the second phase, the 3-dimensional vector was reduced to a 2-dimensional vector by "mean pooling" over all token embeddings; all 768 elements within a token vector were "pooled" into a scalar with value equal to the mean of the vector divided by the square root of the input length. After those two phases, Sentence-BERT output a 2-dimensional matrix of sentence embeddings with the shape 25×512 representing the sentence embeddings for the 25 sentences.

3.3. Transformer Encoder

To capture inter-sentence attention, PatentLLM passed sentence embeddings into 3 layers of the Transformer Encoder, which is the optimum number of layers for our experiments. For this, we utilized the built-in `torch.nn.TransformerEncoderLayer` function with 8 heads for multi-head attention, which is based on the Vaswani et. al. 2017 paper "Attention is all you need" (Vaswani et al., 2017). The output matrix for this step is a matrix with similar size as the previous step.

3.4. Fully Connected Layers and Loss Evaluation

The shape of the first fully connected layer is 512×512, which maps the input matrix to a matrix of the same size. Between the fully connected layers, a dropout layer was applied to randomly zero some of the elements of the input matrix with the probability of 25% to prevent over-fitting. The model flattened the matrix of shape 25×512 into a vector of length 12800 to eventually pass the vector through the second fully connected layer of shape 12800×2, which output the desired binary classification vector of length 2. The output of the vector was then compared with $[0\ 1]$ and $[1\ 0]$ vectors corresponding to acceptance and rejection, respectively, in the loss evaluation part. We utilized `torch.nn.BCEWithLogitsLoss` as our loss function, which is a sigmoid layer combined with BCELoss into one layer for a better numerical stability. The following is the equation for BCEWithLogitsLoss:

$$l(x, y) = L = \{l_1, ..., l_N\}^T$$

where

$$l_n = -w_n[y_n \cdot \log \sigma(x_n) + (1 - y_n) \cdot \log(1 - \sigma(x_n))]$$

and x, y are input and target vectors, respectively.

4. Experiments

4.1. Sampling the Dataset

We randomly sampled 25,000 patents filed in 2016 from the Harvard USPTO Patent Dataset, splitting the data into a training set containing 20,000 patents and a test set containing 5,000 patents. To avoid class imbalance, we ensured that both the training and test sets contained an even split of patents from the two classes: "ACCEPTED" and "REJECTED." We did not use the "CONT-ACCEPTED" and "CONT-REJECTED" classes and thereby focused only on patents who did not have parent filings in the USPTO continuity data files. Thus, our training set contained 10,000 accepted patents and 10,000 rejected patents while our test set contained 2,500 accepted patents and 2,500 rejected patents.Some patents had conflicting "decision" (from the metadata file), "decision as of 2020" (from the metadata file), and "decision" (from the patent JSONs themselves) classifications. We took the "decision" status from inside each patent JSON file to be the ground truth decision, as this was the decision used by the `load_dataset` builder function created by Suzgun et. al. (Suzgun et al.).

In addition to the decision classification, each patent in our sample of the dataset contained several metadata columns. We discarded any unique identification numbers (i.e. "earliest_pgpub_number" and "wipo_pub_number") or information that would make the decision status of the patent blatantly obvious (i.e. "appl_status_dec," or application status description). We also disregarded metadata variables that had the same value for every patent in our sample (i.e. "application_invention_type" was "utility" for all patents in our sample). Finally, we disregarded metadata variables that only applied to decision classes we were not using (i.e. "continuation," which denotes if a patent is "a second application for the same invention claimed in a prior non-provisional application and filed before the first application becomes abandoned or patented" (Suzgun et al., 2022); we only considered non-continuation patents). Below is our analysis of the impact of the following metadata variables on patent acceptance rate: "filing_date," "examiner_id," "ipc_label", "foreign," "small_entity_indicator," and "aia_first_to_file." These terms are defined in the Glossary section of the Appendix.

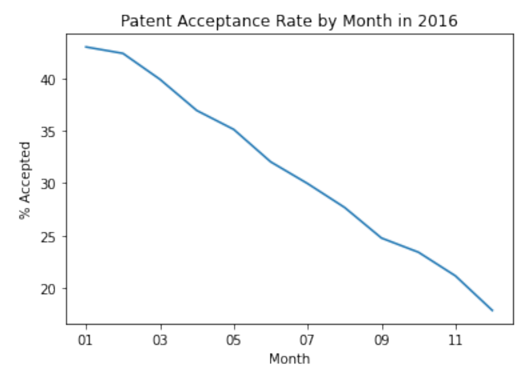

Figure 3. The percentage of patents filed in each month of 2016 with decision class "ACCEPTED."

Figure 3 shows that our sample's overall patent acceptance rate declines over the course of 2016.

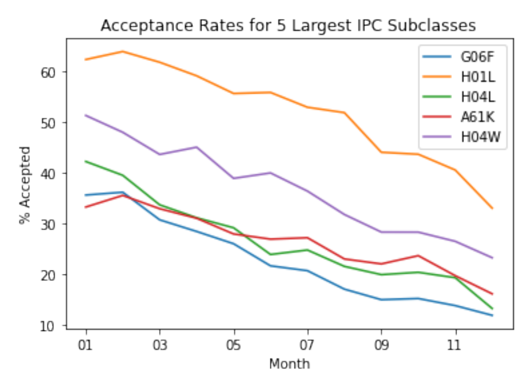

Figure 4. Patent acceptance rates by month for the five largest IPC subclasses in 2016.

Figure 4 shows that patent acceptance rates follow the same overall downward trend over the course of 2016 for all five IPC subclasses shown. However, patents that fall under some IPC subclasses (such as H01L) consistently have much higher acceptance rates than others (such as those in the G06F subclass).

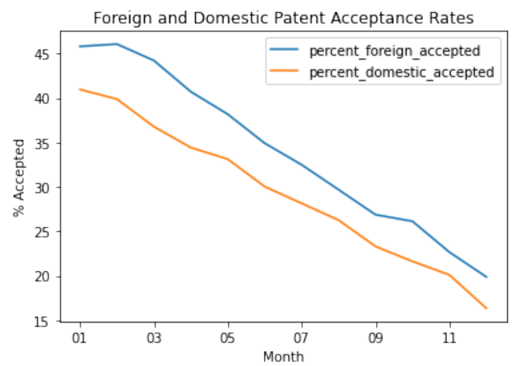

Figure 5. Percentage of patents filed by foreign and domestic inventors accepted in each month of 2016.

Figure 5 shows that across each month of 2016, patents originally filed in a country that is not the United States had a consistently higher acceptance rate.

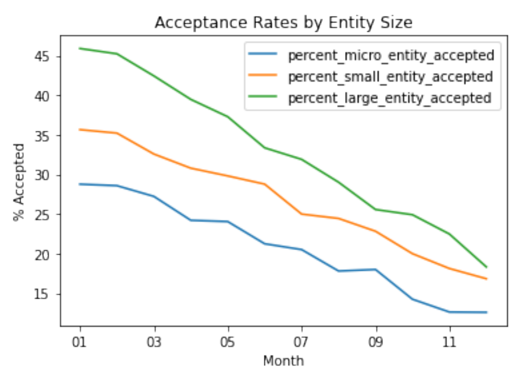

Figure 6. Percentage of patents filed by micro, small, and large entities accepted in each month of 2016.

Figure 6 shows that patents filed by large entities are consistently accepted at a higher rate than patents filed by micro and small entities. This is consistent with Suzgun et. al.'s finding that "small and micro entities (i.e., independent inventors, small companies, non-profit organizations) are less likely to have positive outcomes in patent obtaining than large entities (i.e. companies with more than 500 employees)" (Suzgun et al., 2022).

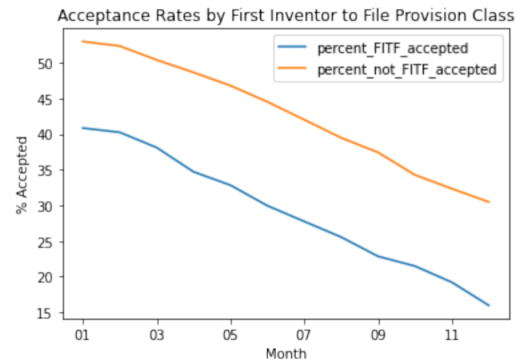

Figure 7. Acceptance rates of patents filed by inventors who meet the AIA's first-inventor-to-file eligibility requirements and those who do not in each month of 2016.

Figure 7 shows that patents filed by inventors who meet the AIA's first-to-file eligibility requirements are accepted at a lower rate than those filed by inventors who do not meet those eligibility criteria. We note here that the AIA first-inventor-to-file law introduced in 2013 is not as simple as "only the first inventor to file a patent for a given invention must be granted said patent" - there may be extenuating circumstances which can cause the patent to fall into a legal gray area (Quinn, 2018); even the "aia_first_to_file" metadata variable can take on the values "True," "False," or "Other." While the complexities of intellectual property law are outside the scope of this project, we do see a clear correlation between the value of this metadata variable and patent acceptance rate within our sample.

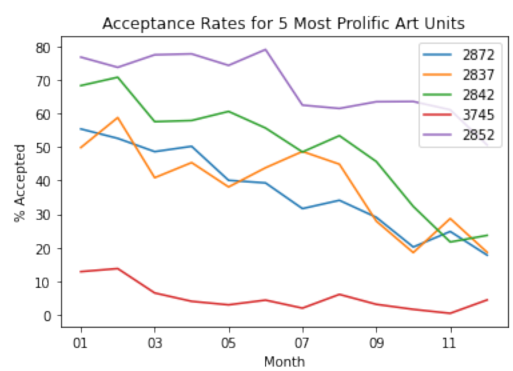

Figure 8. Patent acceptance rates by month for the five examiner art units that processed the most patents in 2016.

Figure 8 shows the acceptance rates for the five most prolific art units (that is, the art units that examined the most patents) in 2016. We see that most of these art units follow the general trend of patent acceptance rate decreasing from the beginning to the end of the year. There is a stark difference, however, between the percentage of patents they each accept, with art unit 3745 accepting only 5 to 15% of patents they examined in any given month while art unit 2852 accepted as many as 80% of patents examined in some months. This is consistent with previous findings which indicate that certain art units accept patents at a much higher rate than others (Mann, 2022; Underhill, 2015).

Overall, our analysis provided reasonable justification for integrating these six metadata variables into PatentLLM.

4.2. Implementation Details

For our baseline models, we trained four other well-known Large Language Models, which include BERT, DistilBERT, Longformer, and SciBERT. We trained each model on 10 epochs on a single NVIDIA A100 GPU. We used an Adam optimizer with a learning rate of 2e-5 as suggested by the original BERT paper (Devlin et al., 2018) with a default epsilon of 1e-8, which is optimal for our purposes. We observed that the baseline over fitting our training set quickly. Hence, a 25% probability dropout layer was also added to prevent over fitting. At the conclusion of each training epoch, we tested the model's accuracy against the test set and recorded an associated F-1 score. We implemented flat accuracy, which maps the target vectors of size 2 into 0s and 1s using argmax function and evaluated the accuracy comparing to the ground truth values.

5. Results and Discussion

We defined accuracy as the highest accuracy achieved by each model during the 10 training epochs. The F-1 score shown in our result corresponds to the F-1 score associated with the highest accuracy score. Initially, given that the model had not seen the test set yet and the test set was balanced, each model's accuracy was around 50% (essentially a coin flip). After each model was trained on the claims section, all models experienced increases in accuracy. **Our model had the overall highest accuracy of 65.1% relative to other baseline models, beating the average and the median benchmark group by close to 1%**. Regarding the baseline models, Longformer had the higher accuracy of 64.6% compared to other models such as BERT (64.2%), DistilBERT(63.8%), and SciBERT(64.1%). This result confirms our hypothesis that an augmented LLM via a hierarchical structure was able to capture the semantics of the entire patent documentation better than traditional LLM for this acceptance classification task. The addition of metadata, although may seem informative, did not improve the overall accuracy of our model. However, adding metadata did help to improve our model's F-1 score by 2.2%. We also note that PatentLLM (utilizing SciBERT word embedding model) outperformed SciBERT by 1%.

Table 1. Experiment Results

	Accuracy	F1 Score	Precision	Recall
Our Model	**65.1%**	64.1%	65.9%	62.4%
Our Model w/ metadata	**65.1%**	66.3%	64.1%	68.6%
BERT	64.2%	62.2%	65.8%	59.0%
DistilBERT	63.8%	**69.4%**	60.0%	**82.2%**
Longformer	64.6%	67.4%	62.4%	73.4%
SciBERT	64.1%	61.1%	**66.6%**	56.6%

Our main limitations are resource constraints (20,000 training and 5,000 test set versus around 4,518,263 outstanding patents (Suzgun et al.) or around 0.55% of the outstanding U.S. patent universe). Results could likely be further enhanced with more training data.

A commercial application of this model could be as a prescreen tool for patent acceptance decision for individual or corporate patent filers or the patent office.

6. Conclusion

By observing that a patent's claims following a hierarchical structure where each numbered sentence together forms a meaningful claim and recognizing the high scientific contents within each patent, we proposed an augmented hierarchical LLM, which outperforms other currently well-know and popular LLMs such as BERT, DistilBERT, SciBERT, and Longformer.

Future works could extend our hierarchical LLM architecture to other long-form scientific text with bullet points, numbered claims, or paragraph structure, leverage our model on patents from other countries, or utilize our architecture on other NLP tasks such as scientific document classification.

7. Acknowledgements

We would like to thank Professor Rajesh Ranganath and Teaching Assistant Wanqian Yang for their support and guidance of this project. This project was conducted as a part of the curriculum for the Machine Learning (CSCI-GA.2565-001) course at the New York University Courant Institute of Mathematical Sciences.

References

Beltagy, I., Lo, K., and Cohan, A. Scibert: A pretrained language model for scientific text, 2019. URL https://aclanthology.org/D19-1371.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL https://arxiv.org/abs/1810.04805.

IPC. Guide to the international patent classification (2022) - wipo, 2022. URL https://www.wipo.int/edocs/pubdocs/en/wipo-guide-ipc-2022-en-guide-to-the-international-patent-classification-2022.pdf.

Kearns, R., Khosla, S., and Wittenbrink, B. A nlp approach to understanding patent acceptance criteria. URL https://benjaminhwittenbrink.github.io/assets/docs/bw_patents.pdf.

Li, S., Hu, J., Cui, Y., and Hu, J. Deeppatent: Patent classification with convolutional neural networks and word embedding. *Scientometrics*, 117(2):721–744, nov 2018. ISSN 0138-9130. doi: 10.1007/s11192-018-2905-5. URL https://doi.org/10.1007/s11192-018-2905-5.

Liu, Y. and Lapata, M. Hierarchical transformers for multi-document summarization, 2019. URL https://arxiv.org/abs/1905.13164.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

Mann, C. How classification works at the uspto: Targeted drafting to influence prosecution outcomes, Nov 2022. URL https://www.lexisnexisip.com/resources/how-classification-works-at-the-uspto/.

Piroi, F., Lupu, M., Hanbury, A., and Zenz, V. Clef-ip 2011: Retrieval in the intellectual property domain. In *Conference and Labs of the Evaluation Forum*, 2011.

Quinn, G. A simple guide to the aia oddities: First to file - ipwatchdog.com: Patents amp; intellectual property law, Dec 2018. URL https://ipwatchdog.com/2013/09/11/a-simple-guide-to-the-aia-oddities-first-to-file/id=45104/.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL https://aclanthology.org/D19-1410.

Ruan, Q., Ostendorff, M., and Rehm, G. Histruct+: Improving extractive text summarization with hierarchical structure information. URL https://arxiv.org/pdf/2203.09629.pdf.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2019. URL https://arxiv.org/abs/1910.01108.

Sharma, E., Li, C., and Wang, L. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2204–2213, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1212. URL https://aclanthology.org/P19-1212.

Suzgun, M., Melas-Kyriazi, L., Sarkar, S. K., Kominers, S. D., and Shieber, S. M. Hupd.py · hupd/hupd at main. URL https://huggingface.co/datasets/HUPD/hupd/blob/main/hupd.py.

Suzgun, M., Melas-Kyriazi, L., Sarkar, S. K., Kominers, S. D., and Shieber, S. M. The harvard uspto patent dataset: A large-scale, well-structured, and multi-purpose corpus of patent applications, 2022. URL https://arxiv.org/abs/2207.04043.

Underhill, A. These are the 20 hardest and easiest art units - ipwatchdog.com: Patents amp; intellectual property law, May 2015. URL https://ipwatchdog.com/2015/05/21/hardest-easiest-art-units/id=57864/.

USPTO. First inventor to file (fitf) resources, Feb 2016. URL https://www.uspto.gov/patents/first-inventor-file-fitf-resources#:~:text=The%20first%20inventor%20to%20file,that%20impact%20patent%20prosecution%20directly.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., and Polosukhin, I. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. Hierarchical attention networks for document classification, 2016. URL https://aclanthology.org/N16-1174.

A. Appendix

A.1. Metadata Example

```
Filing Month: February
IPC Label:  B29C4514
Foreign Status:  True
Small Entity Indicator: MICRO
AIA First To File:  true
Examiner Art Unit:  2834
```

Figure 9. Example of Metadata values associated with a patent sample

A.2. Claims Section Example

"1. A method for manufacturing a motor bobbin around which a coil is to be wound, the method comprising forming the motor bobbin by injection molding wherein the insulating sheet and a core member are arranged in a cavity of an injection mold and a resin is injected into the cavity, wherein the motor bobbin comprises an insulating sheet and a molded resin article. 2. The manufacturing method according to claim 1, wherein the motor bobbin comprises a bobbin body portion which is at least partially made of the insulating sheet. 3. The manufacturing method according to claim 1, wherein the motor bobbin comprises a pair of the molded resin articles connected to both ends of the bobbin body portion. 4. The manufacturing method according claim 1, wherein the insulating sheet and the molded resin article are connected and fixed to each other without using an adhesive agent. 5. The manufacturing method according to claim 1, wherein a surface of the insulating sheet in contact with the molded resin article is impregnated with the molded resin article. 6. The manufacturing method according to claim 1, wherein the molded resin article has a groove for positioning a coil. 7. The manufacturing method according to claim 1, wherein the molded resin article and the insulating sheet are bonded to each other simultaneously with the molding of the molded resin article. 8. The manufacturing method according to claim 1, wherein the molded resin article is formed by using a polymer having an amide bond, and a surface of the insulating sheet in contact with the molded resin article is made of an aramid paper comprising an aramid fibrid and an aramid short fiber. 9. A motor bobbin which is fabricated by the manufacturing method according to claim 1. 10. A motor comprising a stator obtained by winding a coil around the motor bobbin according to claim 9. 11. A motor generator comprising a stator obtained by winding a coil around the motor bobbin according to claim 9. 12. A power generator comprising a stator obtained by winding a coil around the motor bobbin according to claim 9."

A.3. Glossary

- **"Filing_date"** is defined as "the date of receipt in the Office of an application which includes (1) a specification containing a description and, if the application is a non-provisional application, at least one claim, and (2) any required drawings" (Suzgun et al., 2022).
- **"Ipc_label"** is defined as the main subject matter classification given to a patent by the International Patent Classification. The first four characters of this label correspond to the label's IPC subclass (IPC, 2022).
- **"Foreign"** is a boolean variable which is True if the file was originally filed in a country that is not the United States and False otherwise.
- **"Small_entity_indicator"** denotes whether the size of the entity the inventor filing the patent belonged to was "Micro," "Small," or "Large."
- **"Aia_first_to_file"** denotes whether the patent meets the first-inventor-to-file eligibility requirements introduced by the America Invents Act which became effective in March 2013 (USPTO, 2016).
- **"Examiner_art_unit"** is a unique identification number given to each group of examiners, called an "art unit," which evaluate patents that fall under a group of subjects in which they have expertise.