UMICORE OLEN, INTERNSHIP

Research and Planning

Bachelor in Applied Computer Science.

Emmanuel Akpandara

Academiejaar 2024-2025

Campus : Geel

LID VAN ASSOCIATIE KU LEUVEN

THOMAS MORE

Table of Contents

1. Introduction

This project focuses on the automation of patent classification for battery-related patents using advanced machine learning models, particularly large language models (LLMs). Battery patents contain intricate technical details and a specialized vocabulary, making their classification a complex task. Traditional methods, such as bag-of-words (BoW), fail to capture the full depth and semantic context of patent language. The goal of this project is to leverage LLMs like PatentBERT and BatteryBert to automate and enhance the classification process, improving accuracy, efficiency, and scalability compared to the previous system.

2. Problem Statement

Battery patents are rich with technical terminology, and existing manual and rule-based classification systems struggle to scale or adequately handle such complexity. Previous approaches, such as using the Bag of Words (BoW) model for text classification, were used to categorize patents based on keyword frequency. While functional, these methods lack the ability to understand the semantic meaning of the text, which limits their performance. This project aims to build a more sophisticated, automated classification system using LLMs to capture deeper semantics and automate patent categorization, with the goal of improving the accuracy and efficiency of the process.

3. Objectives

- To develop an LLM-based classification model capable of automatically categorizing battery-related patents.
- To improve the existing classification pipeline by incorporating advanced machine learning techniques and semantic embeddings.
- To enhance the model's performance by experimenting with different architectures and data augmentation strategies.
- To analyse the effectiveness of using pre-trained models like PatentBERT and other LLMs in a domain-specific context.
- To track experiment results, model performance, and hyperparameters using MLflow.
- To document and share insights and improvements over the previous classification approach (BoW).

4. Background and Literature Review

In preparation for the implementation, I studied several research papers to deepen my understanding of patent classification, particularly those leveraging deep learning techniques. Research in this area shows that traditional methods like BoW or TF-IDF are insufficient for the high-level understanding required in patent classification. Recent works highlight the advantages of using embeddings from pre-trained language models

(such as BERT) for text classification tasks. Notably, I came across several papers that explore fine-tuning models such as PatentBERT and BatteryBERT, which are specifically tuned to handle domain-specific vocabularies and technical language.

Additionally, I explored a Meta's Llama 3.1 Model approach for embeddings, which I thought would perform better than the BERT models, however, I couldn't correctly implement it as I encountered some technical issues, so it wasn't fully explored.

5. Previous Approach

Prior to the implementation of my approach, the classification system employed the Bag of Words (BoW) technique. The BoW method relies on counting the frequency of words within a document, ignoring word order and context. While BoW is simple and computationally efficient, it fails to capture the deeper meaning of patent language, which is often technical and domain-specific. This limitation motivated my decision to explore more advanced models like BERT, which can understand contextual relationships between words and generate richer semantic representations.

6. Methodology:

6.1 Data Collection and Preprocessing

The dataset used in this project consists of patent documents related to battery technologies. The patents were sourced from publicly available patent databases. Preprocessing steps included:

- **Text cleaning**: Removing irrelevant content after data retrieval from api and any non-patent-related metadata or non English words.
- **Specified Fields**: Selecting specific fields to be used in training data
- **Stopword removal**: Eliminating common words that do not contribute to the semantic meaning.
- **Contextual Translation (Data Augmentation)**: Translating parts of training data foreign language to english
- **Tokenization**: Splitting the text into smaller units (tokens) for processing by the LLM.
- **Weighted Sampling**: Applying a random weighted sampler for equal class representation in training due to imbalanced dataset.

6.2 Model Selection and Fine-Tuning

I initially experimented with pre-trained LLMs like PatentBERT, BatteryBert and BioBert. The PatentBert model is fine-tuned specifically for patent text and was chosen for its potential to capture the nuanced language found in patent documents. Training and testing was also done on BatteryBert and BioBert because after analysing the training data, these models' corpus suited the training data in my understanding. I fine-tuned PatentBERT by first freezing the pre-trained layers and training only the classifier layer. This was done to retain the semantic knowledge captured by the model during pre-training and minimize the risk of overfitting. More experiments on the encoder layers were conducted to for more findings.

6.3 Data Augmentation

To combat the limited size of the dataset, I employed a data augmentation strategy. I experimented with generating synthetic data using embeddings from BERT to create new, semantically similar patent texts. This was done by taking existing patents and perturbing parts of the text that was in a foreign language to create English variations. This approach, however, had mixed results, and I eventually decided to focus more on improving the base model's performance before revisiting data augmentation.

6.4 Evaluation and Experimentation

Several different experiments were run to test and improve the model. I experimented with varying hyperparameters such as learning rate, batch size, and the number of epochs. Additionally, I tried fine-tuning all layers of the pre-trained model after initially training the classifier alone then conducting more experiments on the encoder layers to see how it affects training performance. Throughout this process, I tracked all experiment results using **MLflow**, which allowed me to compare performance across different configurations and monitor key metrics such as accuracy, loss, precision, recall, and F1-score.

I also compared the performance of the LLM-based model with the previous BoW-based approach. The LLM approach demonstrated slightly better performance, suggesting that capturing semantic embeddings was somewhat effective than relying on simple word counts.

6.5 Model Evaluation

The final model's performance was evaluated using a test set and several key metrics:

- **Accuracy**: The overall percentage of correctly classified patents.
- **Precision, Recall, and F1-score**: These metrics provided insight into the model's ability to classify patents accurately across different categories.
- **Confusion Matrix**: To understand where the model was making errors and refine its predictions.

6.6 Implementation of the Automated Pipeline

This step was initially part of my project plan but unfortunately I could not explore further due to insufficient time and a stronger focus on improving the previous parts.

7. Tools and Technologies

- **PatentBERT & BERT**: Used as base models for semantic understanding of patent text.
- **PyTorch**: The primary deep learning framework for model implementation.
- **Hugging Face**: For utilizing pre-trained transformer models.
- **MLflow**: For tracking experiments, logging metrics, and managing model versions.
- **Databricks**: For distributed computing and data management.
- **Python**: For implementing the model and automation pipeline.
- **Tensorflow**: This was used alongside Pytorch for testing but later discontinued and chose Pytorch solely.

8. Expected Outcomes

- An automated patent classification system capable of accurately categorizing battery patents.
- Enhanced semantic understanding through the use of embeddings from advanced models like PatentBERT.
- Improved classification accuracy compared to the previous BoW approach.

9. Risks and Challenges

- **Data Quality and Representation**: Ensuring the dataset accurately represents the variety of battery patents.
- **Overfitting**: Balancing model complexity and generalization to avoid overfitting to the training data.
- **Dataset Imbalance**: Making sure that a certain class is not biased during training due to more appearance of that class.
- **Method of Finetuning**: Exploring and finding the optimum method for finetuning
- **Large Text**: Difficulty in capturing context over extended text. Effective approaches to be taken.
- **Complexity of Patent Language**: Understanding the highly technical and domain-specific language in patents remains a challenge for model performance.