[Re]PatentSBERTa: A Deep NLP based Hybrid Model for Patent Distance and Classification using Augmented SBERT

Gamunnarbi Park
Department of Data Science
University of SeoulTech
nuripeace99@ds.seoultech.ac.kr

Nayun Kim
Department of Data Science
University of SeoulTech
nayun3773@ds.seoultech.ac.kr

Nyamsuren Undram
Department of Data Science
University of SeoulTech
undram@ds.seoultech.ac.kr

Abstract

The PatentSBERTa model, based on Sentence-BERT, calculates patent similarities and offers superior classification. This study replicates the Augmented SBERT model from "PatentSBERTa: A Deep NLP based Hybrid Model for Patent Distance and Classification using Augmented SBERT," applying it to additional patent data for performance assessment. We verify the reproducibility of the PatentSBERTa model and gain an understanding of its structure. We also evaluate the model's performance on new data for similarity measurement and classification.

Scope of Reproducibility

The paper addresses BERT's high computational cost, limitations with patent data, and multi-label classification challenges. Solutions include using SBERT for efficiency, Augmented SBERT for specialization, and a hybrid model for better interpretability.

Methodology

We used the author's code and replicated the study using PatentView's data for comparison with the original paper's data. Each model took approximately 10 hours to run. The PatentsView database tables can be bulk downloaded as individual tab-delimited files.

Results

The results of the paper and our results showed a difference of 48.08 percentage points in F1 score at the class level. This refutes the claim that the hybrid model proposed in the paper does not perform well in patent classification.

What was easy

We were able to obtain the processed data from the case study in the paper. The methodology was explained in detail and was easy to understand.

What was difficult

The data collection process was omitted, leading to difficulties in gathering and preprocessing data for comparison. This process was time-consuming, and modifications to the code were necessary.

33rd Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

1 Introduction

Deep learning models automatically analyze and understand the textual content of patents, enabling more accurate and efficient patent information analysis. The PatentSBERTa model presents an efficient approach to calculate the technical similarity between patents using textual data based on Sentence-BERT and provides automated multi-class patent classification with higher performance than existing models.

In this study, we aim to verify the performance and applicability of the deep learning-based patent information analysis model by reproducing and analyzing the Augmented SBERT-based model proposed in the paper PatentSBERTa: A Deep NLP-based Hybrid Model for Patent Distance and Classification using Augmented SBERT. We also aim to verify its applicability to various real-world problems such as technology trend analysis and patent search by applying the reproduced model to real patent data.

We expect to achieve the following expected outcomes through this study. First, we can verify the reproducibility of the PatentSBERTa model and improve our understanding of the model structure and learning process. Second, we can apply the reproduced model to real patent data to evaluate the model performance and verify its utility in the field of patent distance and classification.

2 Scope of reproducibility

PatentSBERTa: A Deep NLP based Hybrid Model for Patent Distance and Classification using Augmented SBERT The problems raised in the paper are the following three.

- LLM models such as existing BERT are computationally expensive and difficult to deploy on a large scale
- Models trained on plain text data have limitations on specialized domain data such as patents
- Difficulty in handling the multi-label, multi-class nature of patent classification tasks

Therefore, to address these issues, we proposed the following.

- Measure similarity between patents using Sentence-BERT (SBERT) model for computational efficiency
- Apply Augmented SBERT method for a language model specialized for patent domain
- Proposed a hybrid model that combines transformers and traditional machine learning models
- Provided interpretability of prediction results with a K-Nearest Neighbors (KNN) based model

3 Methodology

We adopted the approach from the paper by utilizing the provided GitHub repository. The original paper used the PatentsView dataset, so we employed the same dataset to verify reproducibility. Subsequently, we attempted to compare the results using a different dataset.

Framework :

- Data processing
- augmented SBERT
- Hybrid Model (AugSBERT + KNN)

3.1 Model descriptions

To predict the class and subclass of a patent, we used Augmented SBERT and KNN algorithms to find the top N claims with high semantic similarity to a claim and predict the subclass label based on the K nearest neighbors.
We used a semi-supervised approach as a data augmentation method for training SBERT, and a cross-encoder was used to augment the training data to tune SBERT. The trained model is called Augmented SBERT. framework following:

-

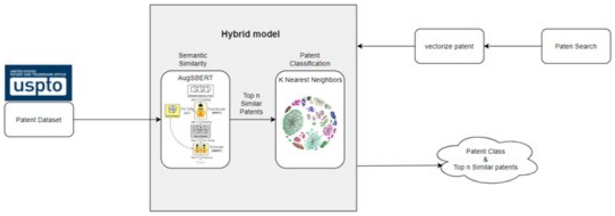
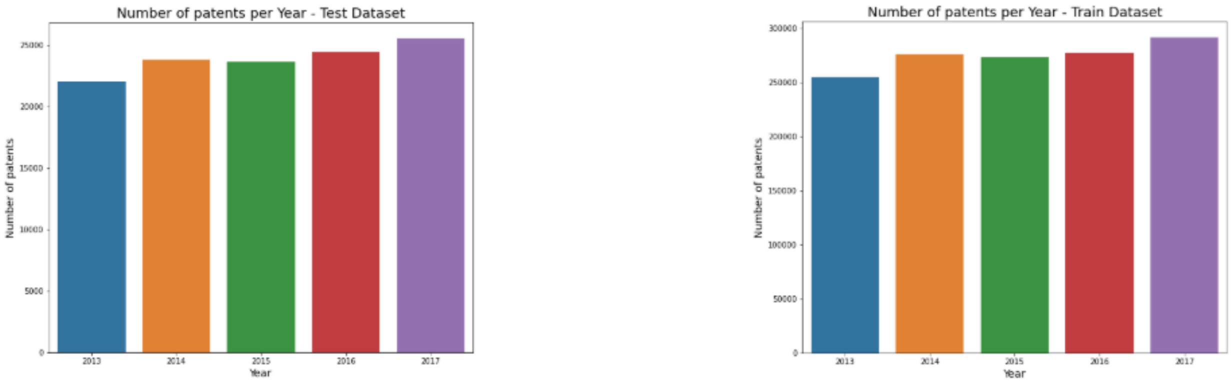


Figure 1: Framework

3.2 Datasets

In this study, utilized all patents between 2013-2017 which have at least one claim on the Google patent public datasets

- DataSource : USPTO
- collect period : 2013-2017 (year)
- Condition : at least one claim on the Google patent public dataset on Bigquery



(a) data - paper

(b) additional data - eco

Figure 2: Comparison of CPC data from the paper and additional eco data

3.3 Hyperparameters

```
# Augmented SBERT Data Loader
{   'batch_size': 16,
    'sampler': 'torch.utils.data.sampler.RandomSampler',
    'batch_sampler': 'torch.utils.data.sampler.BatchSampler'
}

# Augmented SBERT Loss
{   "epochs": 1,
    "evaluation_steps": 0,
    "evaluator": "NoneType",
    "max_grad_norm": 1,
    "optimizer_class": "<class 'transformers.optimization.AdamW'>",
    "optimizer_params": {"lr": 2e-05},
    "scheduler": "WarmupLinear",
    "steps_per_epoch": null,
    "warmup_steps": 100,
    "weight_decay": 0.01
}

# Full architecture (Hybrid Model)
```

Figure 3: prediction result

```
SentenceTransformer((0): Transformer({
                                    'max_seq_length': 512,
                                    'do_lower_case': False}) with Transformer model: MPNetModel
                    (1): Pooling({
                                    'word_embedding_dimension': 768,
                                    'pooling_mode_cls_token': True,
                                    'pooling_mode_mean_tokens': False,
                                    'pooling_mode_max_tokens': False,
                                    'pooling_mode_mean_sqrt_len_tokens': False
                                    }
                                 ))
```

Listing 1: Augmented SBERT & Hybrid Model hyperparameter

3.4 Experimental setup and code

- F1-score : skip samples with all zero true and predict labels

$$F1 = \frac{2 \times \sum \text{True Positives}}{\sum \text{True Labels} + \sum \text{Predicted Labels}}$$

- Recall : Skips samples with all zero predicted labels

$$\text{Recall} = \frac{\sum \text{True Positives}}{\sum \text{Predicted Labels}}$$

- Precision : Skips samples with all zero true labels

$$\text{Precision} = \frac{\sum \text{True Positives}}{\sum \text{True Labels}}$$

- Accuracy : Calculates the proportion of correctly predicted labels

$$\text{Accuracy} = \frac{\sum \text{True Positives}}{\sum \text{True Positives} + \sum \text{False Positives} + \sum \text{False Negatives}}$$

- Hamming Loss : Counts mismatches between true and predicted labels

$$\text{Hamming Loss} = \frac{\text{Total Mismatches}}{\text{Number of Labels} \times \text{Number of Samples}}$$

4 Results

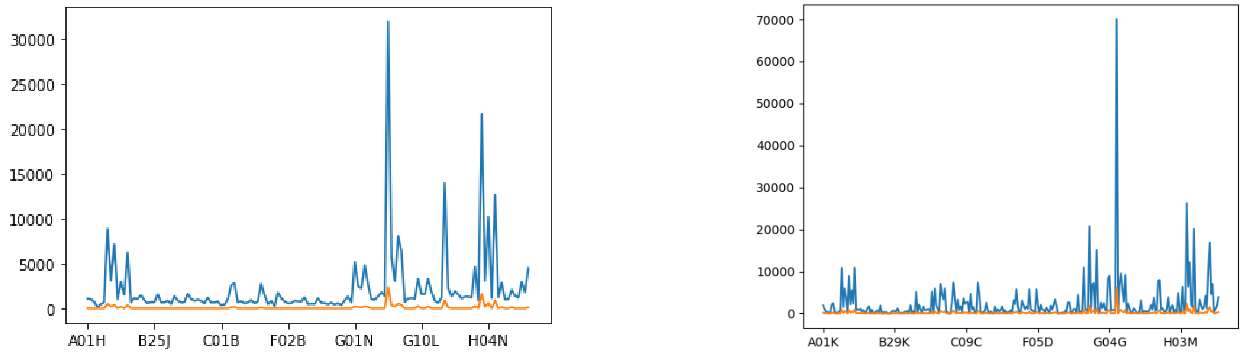
The results demonstrate that the model outperforms existing models at both the subclass and mainclass levels, and that the balance between precision and recall can be adjusted by selecting the appropriate 'K'. (See fig. 3)

4.1 Results reproducing original paper

First, we demonstrate the paper but only implemented AugmentedSBERT. We were unable to implement the KNN classification problem because there is a time gap between the data collection described in the paper and the present. Additionally, the data provided on the official GitHub is not suitable for solving the KNN problem.

4.2 Results beyond original paper

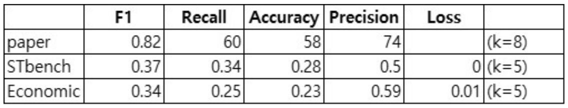
While this paper presents a well-explained model with promising performance, it lacks validation on KNN classification problems with the data. To address this, I will implement the unaddressed aspects using another bulk dataset from PatentView.



(a) data - STbench(paper)

(b) additional data - economic

Figure 4: Comparison of CPC data from the paper and additional eco data



	F1	Recall	Accuracy	Precision	Loss	
paper	0.82	60	58	74		(k=8)
STbench	0.37	0.34	0.28	0.5	0	(k=5)
Economic	0.34	0.25	0.23	0.59	0.01	(k=5)

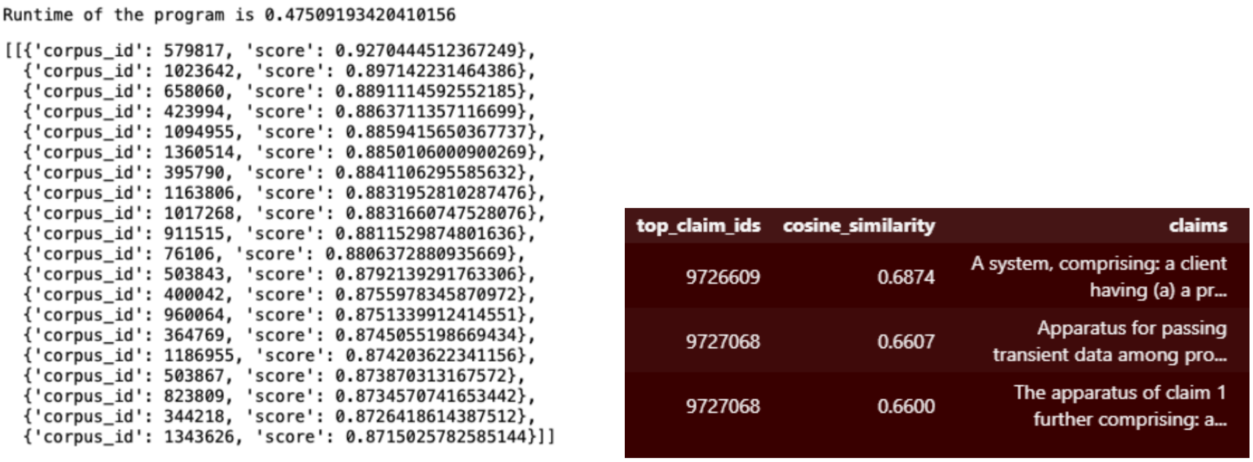
Figure 5: comparision

4.2.1 Augmented SBERT Similarity

Using the claim dataset from PATENTVIEW, train the model and calculate similarities for comparison.

4.2.2 AugBERT KNN Classification

Due to data size limitations, we used 1,000 claims. However, the paper demonstrated high accuracy with 1,400 data points. Despite this, the results of this experiment are as follows:



```
Runtime of the program is 0.47509193420410156

[[{'corpus_id': 579817, 'score': 0.9270444512367249},
  {'corpus_id': 1023642, 'score': 0.897142231464386},
  {'corpus_id': 658060, 'score': 0.8891114592552185},
  {'corpus_id': 423994, 'score': 0.8863711357116699},
  {'corpus_id': 1094955, 'score': 0.8859415650367737},
  {'corpus_id': 1360514, 'score': 0.8850106000900269},
  {'corpus_id': 395790, 'score': 0.8841106295585632},
  {'corpus_id': 1163806, 'score': 0.8831952810287476},
  {'corpus_id': 1017268, 'score': 0.8831660747528076},
  {'corpus_id': 911515, 'score': 0.8811529874801636},
  {'corpus_id': 76106, 'score': 0.8806372880935669},
  {'corpus_id': 503843, 'score': 0.8792139291763306},
  {'corpus_id': 400042, 'score': 0.8755978345870972},
  {'corpus_id': 960064, 'score': 0.8751339912414551},
  {'corpus_id': 364769, 'score': 0.8745055198669434},
  {'corpus_id': 1186955, 'score': 0.874203622341156},
  {'corpus_id': 503867, 'score': 0.873870313167572},
  {'corpus_id': 823809, 'score': 0.8734570741653442},
  {'corpus_id': 344218, 'score': 0.8726418614387512},
  {'corpus_id': 1343626, 'score': 0.8715025782585144}]]
```

top_claim_ids	cosine_similarity	claims
9726609	0.6874	A system, comprising: a client having (a) a pr...
9727068	0.6607	Apparatus for passing transient data among pro...
9727068	0.6600	The apparatus of claim 1 further comprising: a...

(a) data - STbench(paper)

(b) additional data - economic

Figure 6: Comparison of CPC data from the paper and additional eco data

References

Bekamiri, H., Hain, D. S., & Jurowetzki, R. (2021). PatentSBERTa: A Deep NLP based Hybrid Model for Patent Distance and Classification using Augmented SBERT. *arXiv preprint arXiv:2103.11933*.