Patent Quality Prediction by Ensemble Modeling with Attention-Based Models

Nobu Yamaguchi and Meng-Hsien Lin
School of Information
University of California, Berkeley
{nobu.yamaguchi, lin_menghsien}@berkeley.edu

Abstract

Patents are important to drive the technological change. However, some patents are more influential than others, and would be beneficial if we can recognize these patents early on to focus our effort and investment on these patents. Traditionally the patent's influence is measured by the number of times the patent was cited, but this measurement is available when a patent is granted. Therefore, we deploy a patent quality prediction model by capturing the semantic structure in patent text data (ie. abstract, title, and claims), which allows us to derive patent quality measurement immediately when patents are granted. In addition, we explore different neural network and attention-based models, and formulate an Ensemble model including BERT and Hierarchical Attention Network which achieves the best performance. Our major contributions include: (1) examining and comparing various Deep Learning models to assess patent quality with NLP approach and build an Ensemble model specific for this purpose. (2) showing that the patent abstract alone does not appear to be sufficient for accurate quality assessment. Patent claims, however, serve as a better indicator to the patent quality measurement.

1 Introduction

Patent quality measurement is useful for businesses or investors to gauge the growth potential of a product or market that is related to the new patent. A high quality patent can drive meaningful long-term research, guide the firm level strategic decision, and lead to industry-wise expansion. However, it is challenging to identify a good quality patent in a timely manner among the tremendous volume of patents granted each year. In the US alone, there are over 600,000 patents filed, and close to 400,000 patents granted in 2019[1].

Traditionally, the patent quality is estimated by using the count of forward citations of the patent along with several other citation-based statistics. However, the forward citation count is not available at the time of patent granted, and it takes several years for a patent to aggregate sufficient forward citation. For instance, the Organization for Economic Co-operation and Development (OECD) aggregates the number of citations received within 5 years after publication as one input to derive the patent quality index, a number ranging from 0 to 1, the higher the better quality. (Note: below we use "quality index" and "quality score" interchangeably. In the Data section, we provide more detail about the quality index by OECD.) The 5 years requirement makes the quality score not readily available for investors or businesses of interest to identify valuable research or investment opportunity. Therefore, we build NLP models using several deep learning architectures to

[1] https://www.uspto.gov/web/offices/ac/ido/oeip/taf/us_stat.htm
[2] https://www.epo.org/news-events/press/background/epo.html

predict the patent quality score formulated by OECD[1], and predict whether the patent quality score is high based on the patent's semantic structure which is immediately available once a patent is filed. This approach captures the semantic signal available from the patent text data, which may provide more qualitative measurement that is hard to capture by quantitative measurement such as citation count.

We explored different deep learning models to tackle this prediction problem, including BERT which derives bidirectional encoder representation of its word sequences, which is capable of capturing context from longer sequences, appear to be a promising choice for our task. Therefore, in our work we experiment with various BERT models, and compare it with other baseline models in the later section. Also, we explored Hierarchical Attention Network Model[2], which has a hierarchical structure that mirrors the hierarchical structure of documents.

2 Background

There was a previous paper notifying the shortcoming of the citation-based measurement, and pointed out this purely quantitative measurement neglects the semantic information available from technology description [3]. It proposes to use NLP approach to create text embedding based on the patent abstract, and it calls this embedding a "signature" specific to each patent. It uses this signature to assess similarity between patents using nearest neighbor approximation, and further uses it to estimate the patent quality based on technological distance.

There were also other works that propose semantic approaches to study various aspects of patent information such as to construct patent map using text mining involving the use of WordNet to compute a similarity measurement which improves patent distribution visualization and assist preventing patent infringement [4], and another paper constructed a technology semantic network (TechNet) by utilizing terms extraction and word embedding model to derive the vector representations of these terms [5]. However, these do not directly serve to predict the patent quality or potential influence. Also, it did not deploy the seq2seq model or more advanced attention based model.

There is a paper that applies supervised machine learning models such as Decision Tree and Random Forecast as well as Deep Learning model to predict whether a patent would be within the Top 50% and Top 1% most cited patent [6], but it uses non-textual features such as number of backward citations, claims, and inventors. And in our model, we would like to use only text data to predict the overall patent quality with more advanced NLP algorithms.

We have seen one paper that uses the BERT model for patent classification, and achieve the new state-of-the-art performance [7]. However, we have not yet seen a paper to assess the patent quality directly using patent text data utilizing BERT or other attention based models. Therefore, our work will experiment with the BERT model and Hierarchical Attention Network model to predict the patent quality based purely on patent text data.

3 Data

In terms of patent text data, we leverage the Google Patents Public Dataset on BigQuery. We retrieve different text components (ie. patent abstract, title, and claims) from US patents from year 2010 to 2014, and experimented to train with different text components for our models. Furthermore, we retrieve our label data from OECD (OECD Patent Quality Indicators database, January 2020), which conducts patent quality indicators using several research-based approaches. There are several indicators, and we are particularly interested in the "quality_index_4" score, which is a composite measurement taking into account the number of forward citations (up to 5 years after publication), patent family size, number of claims, and the number of IPC technology classes among its forward citations.

We merged the two datasets based on the unique publication number, and resulted in a dataset of about 1.1 million patents. First, we used 5,000 records datasets (80/20% for Train/ Dev) and trained models with different text components to test their usefulness to predict if the patent quality is high.

After the initial training, we picked our top performing models to run with a larger dataset with 50,000 records datasets (70/20/10% for Train/ Dev/ Test). We converted our label data "quality_index_4" to a binary indicator using 0.3 as a threshold which roughly splits to two classes of similar size.

4 Methods and Experimental Setup

The objective of our work is to formulate the best method to achieve the highest precision for the reason illustrated in section 5. Therefore, we explored the following models, and experimented with different parameters and with three different inputs: Claims Only (Table 1), Abstracts Plus Title (Table 1 in Appendix), and Abstracts Only. Since the abstracts alone performed similarly or worse than the other two methods, so we did not present their results here. Following are more details about each model we experimented. We will compare their results and present insights observed in the Result section. The GloVe embedding used in our models are 100 dimensions and pre-trained by 6 billion token corpus. Another model design choice we made is we used 2 sigmoid neurons in the output layer to predict the probability of each class separately because we intend to average multiple models to form Ensemble Model and we feel it is more precise to average each class separately.

4.1 TF-IDF + Naive Bayes

Since nobody has done this prediction, we used the proportion of the majority class as one baseline. Additionally, we implemented the TF-IDF+Naive Bayes model as a second baseline. The model is shown in Appendix Figure 1.

4.2 GloVe + Convolutional Neural Network (CNN)

We implemented Convolutional Neural Network [8] as our second model. We experimented with various parameters and settled for 5 different filter sizes with 50 filters of each filter size. In general, the models accuracy perform similarly between different filter size configurations, but it tends to have larger variance between epochs if the filter sizes are larger apart from each other.

4.3 GloVe + Hierarchical Long Short Term Memory (H-LSTM)

We implemented H-LSTM as the third model. H-LSTM can capture the context. We stored up to 30 sentences and 30 words per sentence (900 tokens), because this combination shows best accuracy when exploring various combinations to check differences in performance.

4.4 GloVe + Hierarchical Attention Network (HATT)

We implemented the Hierarchical Attention Networks model [2] as our fourth model. The differences between H-LSTM and HATT are (i) two attention layers which only HATT has and (ii) GRU-based sequence encoder [9]. The GRU uses a gating mechanism to track the state of sequences without using separate memory cells. Attention layers tend to select qualitatively informative words and sentences. This model is shown in Appendix Figure 2.

4.5 Pre-trained BERT + CNN

For the fifth model, we used a Pre-trained BERT-Base model to create embedding layers without fine-tuning, and added 3 convolutional filter with 100 filters each before the max pooling layer. Since BERT is limited to 512 tokens, and the length of patent claims are average to around 1,000 words. Therefore, for both BERT models in section 4.5 and 4.6, we experimented with two versions, one to train with the first 510 tokens, and another to train with the last 510 tokens. Their results are shown in Table 1.

4.6 Fine-Tuning BERT

For the sixth model, we fine-tuned the BERT-Base embedding layer with one feedforward neural network with size of 768 as illustrated in Appendix Figure 3. We used a GPU machine to train our BERT model which improves the efficiency about 10x faster . We set the epoch to 2 since for both 5,000 dataset and 50,000 dataset, it shows overfitting starting the 3rd epoch. Note that we also experimented with

Distil-BERT (train with first 510 words). Their results are shown in Table 1.

4.7 Ensemble

For the final model, we ensembled previous models since we believe a diverse set of models make better decisions compared to individual models which have some level of biases.

We can achieve this with ensemble learning which combines the prediction from multiple models to: (i) lower error (ii) avoid overfit, and (iii) reduce bias/variance errors. We chose an averaging method where we take an average of the predicted class probabilities for each model to make the prediction. We approach finding the right ensemble by a 3-step greedy approach: Firstly, computing the Top 3 Ensemble model (in terms of precision) by iterating every possible combination of all the model outputs derived from 5,000 dataset using either claims or abstract+title (Results in Table 2). Secondly, for all the models included in the Top 3 Ensemble, we ran with a 50,000 dataset intending to draw our conclusion with better confidence (Results in Table 3). Lastly, we again iterate through all possible combinations among these selected models to get our best Ensemble Model (Results in Table 4). We believe this 3-step process can better reveal the best model with less overfitting. See Figure 1 for our final ensemble model illustration.

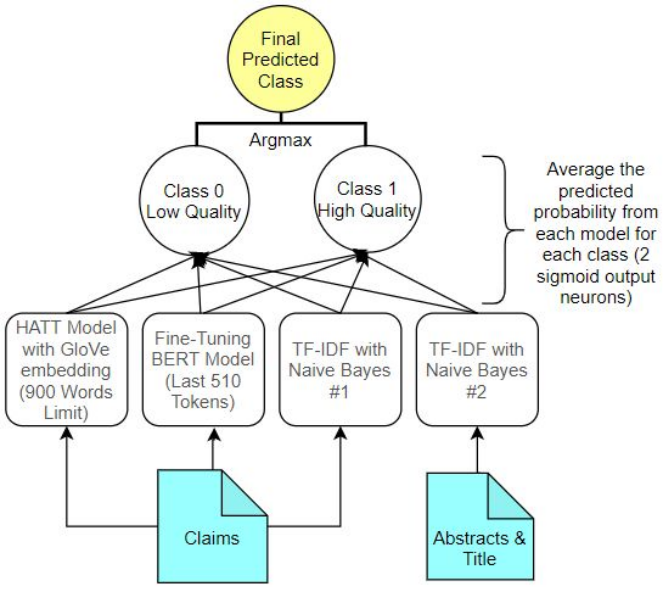

Figure 1: Final Ensemble Model

5 Results and Discussion

In our analysis, we monitor both accuracy and precision metric because of the following reasons to solve the business problem:

(i) Accuracy: to maximize both true positives and true negatives so that we can show more appropriate patents to the users.

(ii) Precision: to minimize the false positives so that the users do not need to read unnecessary patents.

However, when choosing the final model for business purpose, we use precision as the determining metric because businesses usually are limited in the resource and time available, so being able to focus our effort with better precision is more important.

5.1 Results

We compared models trained with different text components. and claim is the clear winner. The results with 5,000 datasets with claims are shown in Table 1. Comparing our main metric Precision, the HATT achieves the best accuracy (ie. 0.699) and precision (ie. 0.693) among all models.

Method	Dataset	Acc	Precision	Recall
1. Majority Class (MC)	5,000	0.595	0.000	0.000
2. TF-IDF + Naive Bayes	5,000	0.618	0.632	0.136
3. GloVe + CNN	5,000	0.625	0.549	0.417
4. GloVe + H-LSTM	5,000	0.681	0.623	0.538
5. GloVe + HATT	5,000	**0.699**	**0.693**	0.462
6. GloVe + HATT (remove stopwords)	5,000	0.688	0.630	0.540
7. Pre-trained BERT + CNN (first 510 words)	5,000	0.604	0.514	0.412
8. Pre-trained BERT + CNN (last 510 words)	5,000	0.606	0.667	0.054
9. Fine-Tuning BERT (first 510 words)	5,000	0.628	0.577	0.306
10. Fine-Tuning BERT (last 510 words)	5,000	0.689	0.636	**0.543**
11. Fine-Tuning Distil-BERT (first 510 words)	5,000	0.627	0.603	0.232

Table 1: Model Evaluation with **2010 to 2014** Patent Data (Input: **Patent Claims**. Label: **Binarized Quality Index**)

Next, we explored ensemble models with all model combinations as illustrated in section 4.7. Table 2 shows the top 3 ensembles by Precision.

Method *	Dataset	Acc	Precision	Recall
2 ** + 5 + 8 + 10	5,000	**0.665**	**0.830**	**0.217**
2 + 8 + 10	5,000	0.622	0.830	0.084
2 + 5 + 8	5,000	0.651	0.826	0.175

Table 2: First-Pass Top 3 Ensemble by Precision.
* Refer to Table 1 for mapping
** This 2 is TF-IDF trained with Abstract, all other models in the Ensembles are trained with Claims

Comparing our main metric Precision, the above Top 1 Ensemble shows the highest precision of 0.830, which is 0.137 higher than the standalone HATT model, which demonstrated the power of ensemble. However, we want to verify the results with larger dataset, so based on above ensemble results, we select the following models to run with 50,000 dataset as shown in Table 3.

Method	Input	Acc	Precision	Recall
Majority Class (MC)	Claim	0.590	0.000	0.000
TF-IDF + Naive Bayes	Claim	0.655	0.634	0.374
TF-IDF + Naive Bayes	Abstract	0.628	0.587	0.313
GloVe + H-LSTM	Claim	**0.693**	0.659	0.522
GloVe + HATT	Claim	0.692	0.685	0.463
Pre-trained BERT + CNN	Claim	0.606	0.517	**0.578**
Fine-Tuning BERT	Claim	0.697	**0.712**	0.440

Table 3: Model Evaluation with **2010 to 2014** Patent includes **50,000** data points (Input: **Claims**. Label: **Binarized Quality Index**)

Lastly, as illustrated in section 4.7, we do the second-pass of ensemble evaluation on all above selected models with 50,000 dataset.

Method *	Dataset	Acc	Precision	Recall
2** + 2 + 5 + 10	50,000	0.698	**0.745**	0.402
2** + 2 + 4 + 10	50,000	**0.706**	0.740	**0.436**
2 + 5 + 10	50,000	0.700	0.739	0.418

Table 4: Second-Pass Top 3 Ensemble by Precision.
* Refer to Table 1 for mapping
** This 2 is TF-IDF trained with Abstract, all other models in the Ensembles are trained with Claims

From Table 4, our Top 1 Ensemble Model includes TF-IDF, Fine-Tuning BERT and HATT as detailed in Figure 1. We then use this Ensemble to evaluate our final test set with 5,000 patents, presented in Table 5 along with our 2 best performing standalone models and the 2 baselines.

Method	Acc	Precision	Recall
Majority Class (MC)	0.599	0.000	0.000
TF-IDF + Naive Bayes	0.662	0.631	0.375
Ensemble Model	**0.698**	**0.735**	0.387
GloVe + HATT	0.695	0.706	0.410
Fine-Tuning BERT	0.698	0.700	**0.431**

Table 5: Final Test Results on Top Models (5,000 patents)

5.2 Insight from BERT Experimentation

Comparing the BERT models results in Table 1, we see Fine-Tuning BERT generally perform better than BERT+CNN without fine-tuning for both Claims and Abstract+Title. This signals Fine-Tuning BERT can capture additional signals from the patent data and improve the Pre-Trained BERT, allowing the model to benefit from transfer learning. Furthermore, comparing training with the last 510 tokens instead of the first 510 tokens, it shows good improvement for Fine-Tuning BERT, which is an indication that the concluding claims tend to serve a better indicator to patent quality than the initial claims.

We also experimented with distil-BERT which trains with less parameters, and the performance is similar to the Fine-Tuning BERT, slightly better precision and worse recall. It trains faster so if we are short on time and computing resources, we may choose this method.

One more insight for Fine-Tuning BERT is it shows significant improvement when increasing from 5,000 to 50,000 datasets, and achieves performance similar to the HATT model. Also, when trained with Abstract+Title (average 130 word count), Fine-Tuning BERT outperforms HATT as shown in Appendix Table 1. This implies that BERT has

greater potential than HATT if we can train with a larger dataset or if it can process more words at once.

5.3 Insight from Hierarchical Model Results

Hierarchical Models (H-LSTM and HATT) show better accuracy and precision than the TF-IDF model. The HATT model which has word- and sentence-level attention layers shows the best precision as a standalone model. This result indicates that the hierarchical model and attention layers can capture the input information more effectively.

Comparing the result of HATT models with and without stopwords, removing stopwords slightly decreased the accuracy, indicating that stopwords contribute to capture the sentence level context.

The difference between H-LSTM and HATT appears in the balance of false positives and false negatives. Therefore, HATT shows higher Precision and lower Recall than H-LSTM.

Also, Hierarchical Models have strengths in the training time. The training time is much faster than BERT models. Furthermore, Hierarchical Models show high accuracy and precision even with the smaller dataset. However, the accuracy and precision do not improve with the larger dataset.

5.4 Error Analysis

We analyzed the result from the two perspectives.

First, Figure 2 shows the relationship between quality index score and prediction result. Although our model predicts low quality patents very well, it cannot predict high quality patents well. This is why we have a low Recall score.

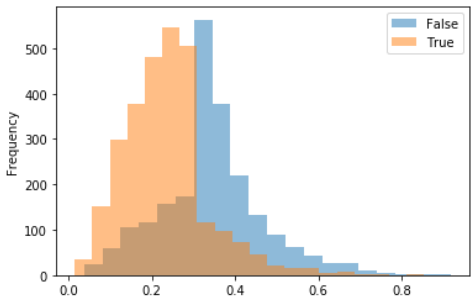

Figure 2: Prediction Results and Quality Score(0 to 1)

Next, Figure 3 shows the relationship between number of tokens and prediction results. We see better results under 1,000 words, indicating a limitation of capturing the large number of words as our model can store up to 900 words.

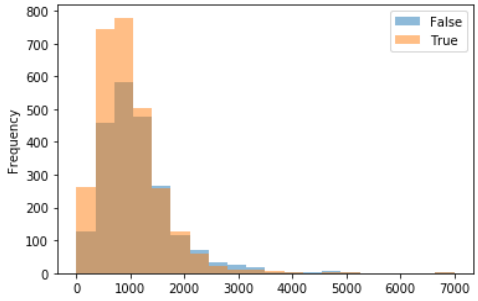

Figure 3: Prediction Results and Number of Words

6 Conclusion & Future Work

Patents are driving the technological innovation and fuel for industry expansion. The motivation of our work is to build a model to predict patent quality based on readily available patent text data so that we can identify patents with large potential for investment or business strategic decision as soon as they are published.

We explore different models intending to predict patent quality based on purely text data using advanced NLP models, including CNN, H-LSTM, HATT, Pre-trained BERT with CNN, and Fine-Tuning BERT on a subset of patent text data from year 2010 to 2014. Among our models, ensemble learning shows the highest precision and accuracy. The components of the ensemble model include Fine-Tuning BERT and HATT, which show high performance as a standalone model. These models contribute to improving precision and accuracy. To answer our business problem, we conclude that this ensemble model is the best.

We used both 1 year data (Result in Appendix Table 2), and 5 year data to predict the patent quality. Both have similar trends. Therefore, we assume that we can use the past data to predict the quality of a new patent.

For this paper, we chose 100 dimensions and pre-trained by 6 billion token GloVe for embedding. Future work can be done with different dimensions of GloVe corpus, or different types of the BERT model such as Longformer, to capture longer documents. Also, future work can be done with the BERT+HATT model. Furthermore, as we see the

improvement on the Fine-Tuning BERT model, the model can be improved by using the larger dataset.

References

[1] Squicciarini, Mariagrazia, Hélène Dernis, and Chiara Criscuolo. "Measuring patent quality." (2013).

[2] Yang, Zichao, et al. "Hierarchical attention networks for document classification." Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. 2016.

[3] Hain, Daniel, et al. "Text-based Technological Signatures and Similarities: How to create them and what to do with them." arXiv preprint arXiv:2003.12303 (2020).

[4] Wang, Hei Chia, Yung Chang Chi, and Ping Lun Hsin. "Constructing patent maps using text mining to sustainably detect potential technological opportunities." Sustainability 10.10 (2018): 3729.

[5] Sarica, Serhad, Jianxi Luo, and Kristin L. Wood. "TechNet: Technology semantic network based on patent data." Expert Systems with Applications 142 (2020): 112995.

[6] Hain, Daniel, and Roman Jurowetzki. "Introduction to Rare-Event Predictive Modeling for Inferential Statisticians--A Hands-On Application in the Prediction of Breakthrough Patents." arXiv preprint arXiv:2003.13441 (2020).

[7] Lee, Jieh-Sheng, and Jieh Hsiang. "Patentbert: Patent classification with fine-tuning a pre-trained bert model." arXiv preprint arXiv:1906.02124 (2019).

[8] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

[9] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

Appendix

Method	Dataset size	Acc	Precision	Recall
Majority Class (MC)	5,000	0.595	0.000	0.000
TF-IDF + Naive Bayes	5,000	0.600	0.532	0.104
GloVe + CNN	5,000	0.589	0.489	0.316
GloVe + H-LSTM	5,000	0.594	0.498	**0.402**
GloVe + HATT	5,000	0.570	0.464	0.400
Pre-trained BERT + CNN	5,000	0.603	0.531	0.170
Fine-Tuning BERT	5,000	**0.616**	**0.538**	0.363

Table 1 : Model Evaluation with **2010 to 2014** Patent Data (Input: **Patent Title + Abstract**. Label: **Binarized Quality Index**)

Method	Dataset size	Acc	+/- from MC
Majority Class (MC)	5,000	0.576	-
TF-IDF + Naive Bayes	5,000	0.598	+0.022
GloVe + CNN	5,000	0.620	+0.044
GloVe + H-LSTM	5,000	0.676	+0.100
GloVe + HATT	5,000	**0.685**	**+0.109**
GloVe + HATT (remove stopwords)	5,000	0.673	+0.097
Pre-trained BERT + CNN	5,000	0.614	+0.038
Fine-Tuning BERT	5,000	0.649	+0.073

Table 2: Model Evaluation with single year **2010** Patent Data (Input: **Patent Claims**. Label: **Binarized Quality Index**)

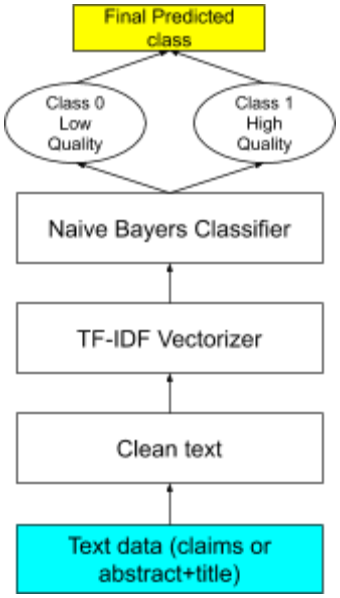

Figure 1: TF-IDF + Naive Bayes Model

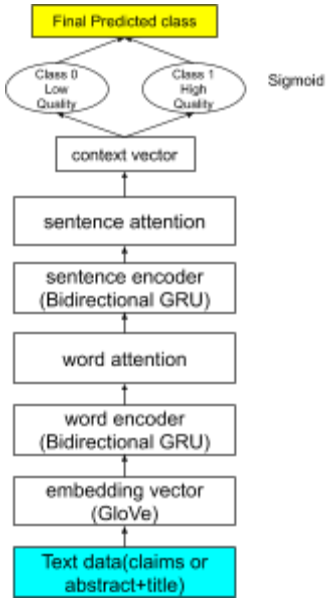

Figure 2:Hierarchical Attention Network Model

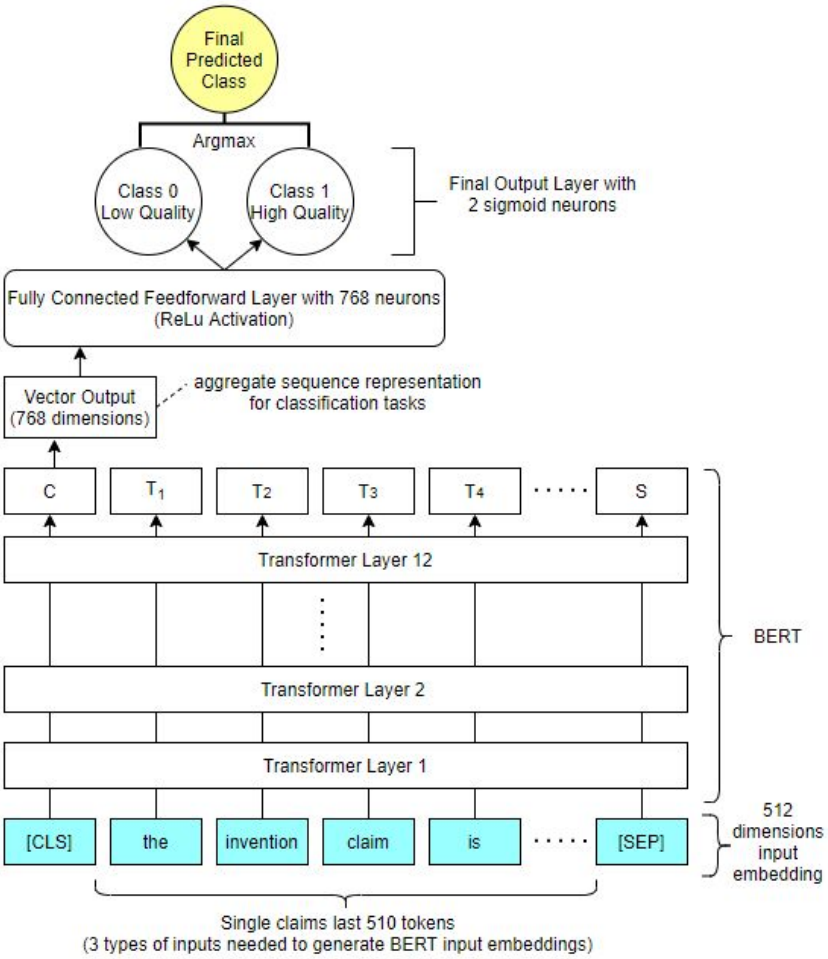

Figure 3: Fine-Tuning BERT Model