PatentPal: Summarising and Retrieving Patents

Deepi Garg, Himanshi Mathur, Vanshika Goel, Vibhu Agrawal
Indraprastha Institute of Information Technology, Delhi, India
Okhla Phase 3, Delhi
India
{deepi18389, himanshi18037, vanshika18202, vibhu18116}@iiitd.ac.in

ABSTRACT

With the advent of technology, inventions and innovations are on a rise. To protect their interests and gain recognition, inventors file patents. However, before filing a patent, it is essential to have a brief summary of all the patents filed in the domain so far, in order to ensure that there is no overlapping patent filed. With the great influx of patents every year, it is essential to have a fast retrieval and summarising system for the same. Hence, we propose, PatentPal - which can summarise the patents filed, create a summary for a new patent and find similar patents, given another patent or appropriate keywords. Pegasus from Google and BM25 have been used for summarisation and retrieval respectively. The codebase can be found on `https://github.com/vibhu18116/IR-Project-Group23-PatentPal`

1. INTRODUCTION

Inventorship is an art, proving you smart
But so many patents, where to start?

In the present world, the information is available at a click away. The resources thus provided and otherwise allow exploration at a large scale, resulting into innovations and inventions. The inventions need to be patented to claim the work, get recognition and obtain some materialistic rewards. However, it is essential to maintain uniqueness of the work done and carry forward the previous literature to ensure that the efforts made are in the best of "human interests" and to prevent "reinvention of wheel".

The process of filing a patent is a long and tedious one. One has to lookup if a similar work already exists, then draft a patent proposal and attach a summary to it. However, due to plethora of patents filed per year and the patents in force as illustrated in Figure 2, it is very difficult to look at every patent, and it is better to have a comprehensive summary rather than having to read the whole of it. The ratio of design patent applications filed, and accepted through various years is shown in Figure 3. It demonstrates a lot of patents are not accepted, where one of the possible reasons could be that the same or very similar work already exists. Similar statistics exist across other domains as well. Hence, it is critical to have an efficient retrieval system for existing patents.

Figure 1: Depicting problems faced by inventors and researchers while filing patents

Keeping such a scenario in mind, we intend to develop a system which can provide comprehensive summaries of the existing patents and to summarise any new patent based on the learning of present corpus. The system would also find similar patents already filed, given another patent, query keywords or a general query.

1.1 Major Contributions

We worked to create a platform for summary and retrieval of patents. The contributions of the system are as enumerated.

1. We have built an intuitive and easy to use platform to retrieve patent documents based on user queries and also provide summaries for input patents.
2. There is no existing UI based platform to generate summaries of patent documents.
3. We combined the extractive and abstractive summary generation approaches by feeding the output of the LSTM summariser to PEGASUS to generate more meaningful summaries.
4. There has been no prior retrieval work performed for the Big Patent Dataset.

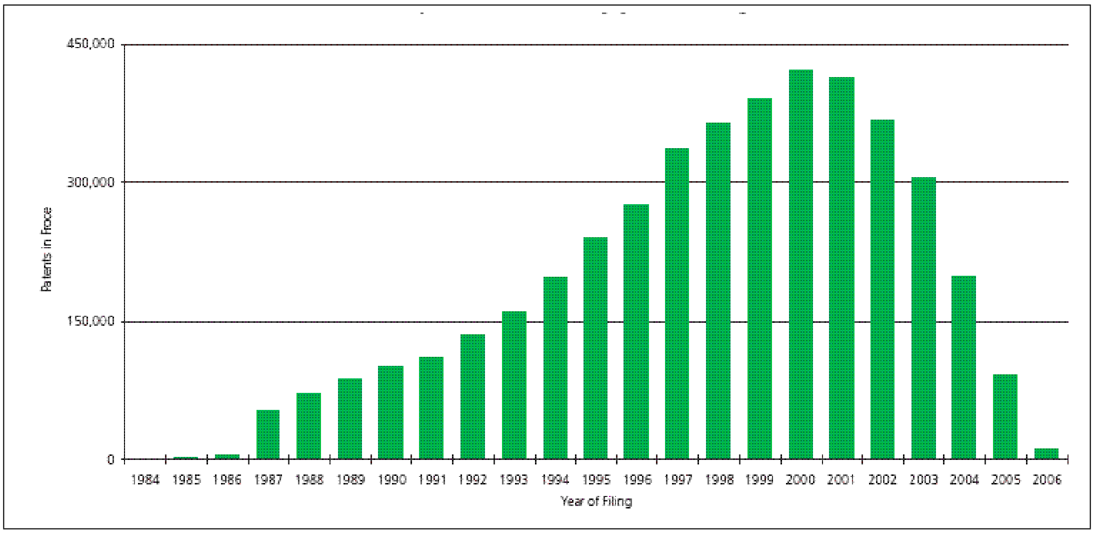

Figure 2: Number of patents in force by year of filing, 2006. The graph does not contain data for the Japan Patent office and the State Intellectual Property Office of China. Source: WIPO Statistics Database (`https://www.wipo.int/ipstats/en/statistics/patents/wipo_pub_931.html`)

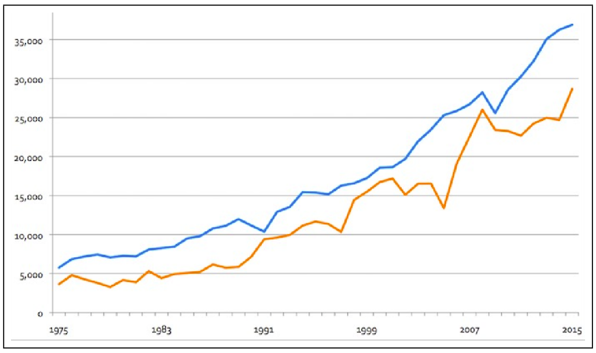

Figure 3: Number of design patent applications filed vs number of patents issued per year. Blue line indicates Applications Filed and orange line indicates Patents Issued (`https://www.ipwatchdog.com/2016/09/10/design-patents/id=72714/`)

2. LITERATURE REVIEW

The project is divided into two tasks: Summarisation and Retrieval. We review prioir work in each of these domains.

2.1 Summarisation of Patent Documents

Summarisation of documents differ from one domain to another domain. It is very difficult to have a general summariser that summarises legal as well as research papers well. Summarisation either works on word level or sentence level by finding their similarity and importance score in the document and corpus. Most of the work done so far focusses on word level. [1]

Trappey et al presented a 2 step approach to summarise patent documents. The first step extracts key words and phrases using two extraction algorithms. The first algorithm uses the TF-IDF concept clustering approach and the second one uses a specific ontology embedded in the system to calculate the frequency of mapping words. The second step involves summary generation which includes key phrase extraction, the creation of the paragraph and key word frequency matrix, the computation of paragraph importance, document concept clustering, and the building of the visualization summary tree. [2]

Hu et al proposed "patent keyword extraction algorithm (PKEA)" to extract keywords from patent documents for summary generation. The method applies a skip-gram model to the pre-trained word embeddings to obtain a table, which is then used to generate the centroid vector using k-means algorithm. The cosine similarity between the centroids and candidate keywords is calculated to extract the top n important keywords. [3]

A graph-based sentence ranking method was put forth by Yeh to produce document summaries. The document is modelled into a network of sentences connected on the basis their lexical overlap. A feature profile is created using three surface level features, namely centroid, position and first sentence overlap. The sentence similarity network and feature profile are used to rank the sentences by applying the spreading activation theory (Quillian, 1968). The summary is generated by adding the top ranking non-redundant sentences, which are then ordered chronologically. [4]

2.2 Retrieval of Related Patents

The quick retrieval of the documents is also paramount to the success of project. Sharma et al applied a semantic expansion technique to enhance the patent search process. This technique incorporates external sources for expansion. The method first filters the abstracts on the basis of IPC using TF-IDF and generates a keyword vector from the abstracts. It then expands the queries and constructs the vector space using external sources like WordNet and Wikipedia. It finally calculates the document similarity using cosine similarity and extended Jaccard Coefficient. [5]

Helmers et al used features like high dimensional sparse bag-of-words (BOW) vectors with tf-idf, neural network language models (NNLM), word2vec (combined with BOW vectors) or doc2vec instead of the keyword approach to find the cosine similarity between two patent documents. [6]

2.3 Evaluation Metrics

The summaries can be generated using extractive or abstractive method. Extractive method tries to figure out most important sentences and use them as it is in the summary. On the other hand, abstractive method tries to summarise the content by paraphrasing and retaining the original context. Here we intend to use abstractive method for summarisation as the dataset has well written human summaries associated with each patent which would act like gold-standard summaries. To evaluate the summaries produced by the model, we would use Rouge [7] scores, which stands for Recall-Oriented Understudy for Gisting Evaluation. Its goal is to provide a measure of quality of an automatically generated summaries in comparison against a reference summary produced by humans.

To evaluate ranking and viability of the retrieved documents, similarity measure with thresholds is used. Precision, Recall and F1 score can be used to check the correctness and accuracy of the retrieved documents, however they require a labelled set with queries. Alternative methods like cosine score, tf-idf scores and jaccard scores can also be used to judge the level of document similarity.

2.4 Novelty of Work

Most models which rely on a keyword-based search for summarisation as well as retrieval often produce suboptimal results. Bag of Words can fetch out patents with exact similar words without any regard of their meaning in the required context. We club the approaches to obtain a context sensitive summary for the documents which would also retain words important to the flow and meaning of legal sense. Compiling it, we embed them into a web-platform capable of summarising the patent documents well.

To the best of our knowledge, there is no existing platform to aid with writing summaries for the patents. Most existing platforms retrieve similar documents and present the human written summary from the same.

2.5 Dataset

Summarisation is a not so straightforward task as evaluation of summaries require comparison from well written human summaries. BigPatent [8] consists of 1.3 million records of U.S. patent documents along with human written abstractive summaries. The data has been collected from Google Patents Public Datasets using BigQuery. It contains patents filed after 1971 across nine different technological areas. We shall use each patent's abstract as the gold standard summary and its description as the input. The dataset is publicly available on `https://evasharma.github.io/bigpatent/`.

Compared to existing summarization datasets, BigPatent has the following properties:

- Summaries contain richer discourse structure with more recurring entities
- Salient content is evenly distributed in the input
- Lesser and shorter extractive fragments present in the summaries

We also used BigPatent dataset for our work.

3. METHODOLOGY

3.1 Data Exploration

The first step after obtaining the dataset was to understand it before proceeding to adopt summarization techniques and indexing. We plotted Word Clouds of summary and description for 1000 documents and found results as shown in Figure 5 and Figure 6. Figure 5 has word cloud for abstract and 5 for the description of the chosen patents. There are a lot of common words between the two plots like "device", "portion", "include", "invention", "system", "body", "embodiment", etc. Figure 7 shows the distribution of the number of common words between abstract and description, which accounts for almost 30-60 words.

To see the compression ratio achieved by the summaries in the domain of patents, we plotted the number of words in the abstract to the number of words in the description ratio. We also did the same analysis on the sentence level. Most abstracts were compressed to a ratio of 1-5% in terms of the number of sentences and 2-6% in terms of the number of words. Figure 8 shows the ratio plot for sentences and Figure 9 shows the ratio plot for words.

To see the correlation between the article and summary length, we again plotted the regression curve between abstract and description as shown in Figure 10. We see the slope of the regression line is positive but the r-squared value is meagre 0.02% indicating that the relationship between the sentence length in abstract and description is very low.

The majority of descriptions have a length of about 200 sentences and abstracts have a length of about 2-3 sentences which is in accordance with the heuristic ratio found above. On average, an English sentence has 15-20 words including all the stopwords and supporting words. This makes about 35 words out of a 60-word abstract come straightforwardly from the description, showing scope for extractive summarization models to maybe perform well.
We constructed baseline models for summarization and retrieval of patent documents. Summarisation can be performed in an extractive or abstractive sense. For retrieval, we need to index the documents, preprocess the documents and queries and then build a logic for reteiving relevant documents.

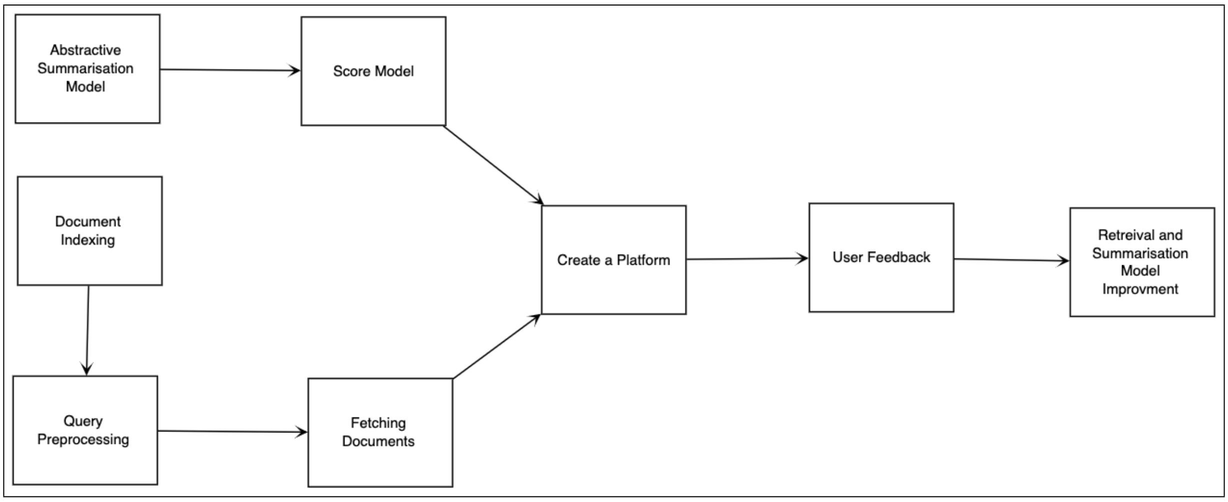

Figure 4: Proposed Methodology

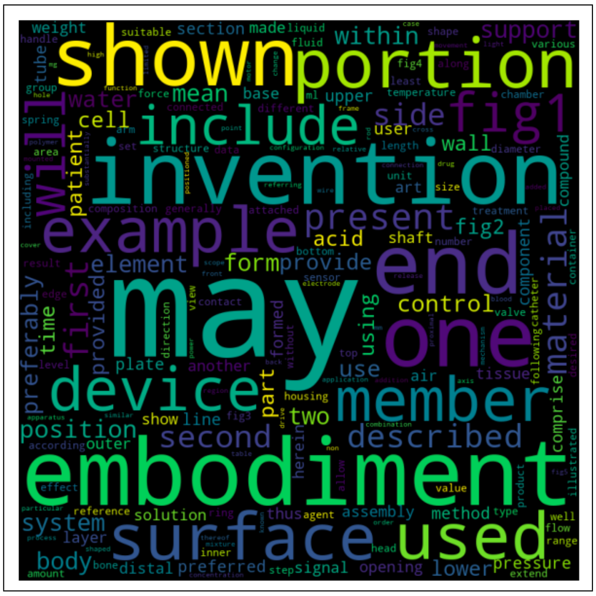

Figure 5: Wordcloud for abstract

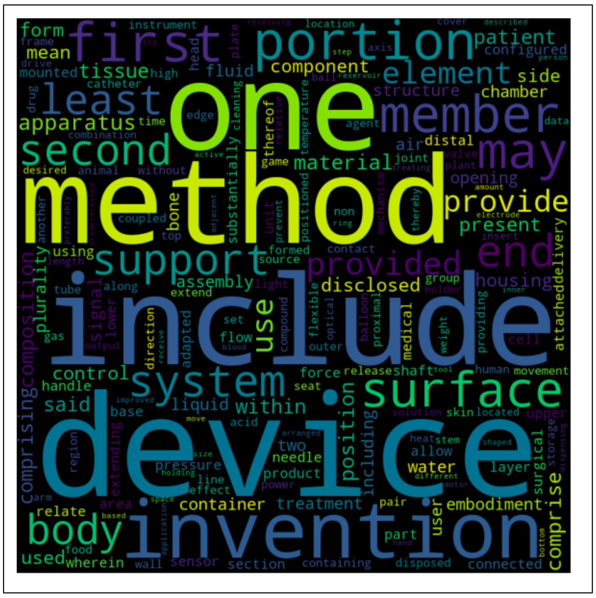

Figure 6: Wordcloud for description

3.2 Summarisation Models

Extractive summary figures out the key phrases and tokens and constructs a summary using them. The summary constructed has words and phrases from the description in verbatim. The following are the baseline models used for extractive summaries:

3.2.1 LSTM (Long Short Term Memory)

LSTM is an artificial recurrent neural network (RNN) architecture that has feedback connections [9]. Two LSTMs: A unidirectional LSTM model with 25 neurons and a bidirectional LSTM model with 25 neurons were trained. The embeddings for sentences were obtained using BERT provided by Google.

3.2.2 TextRank

TextRank is a graph-based ranking model for text processing that can be used in order to find most relevant sentences in text and to find keywords [10]. The found sentences are then appended to form the final summary. A limit of maximum of 15 phrases and 5 sentences was imposed on summary.

3.2.3 Gensim

Gensim is an NLP processing library that provides tools for summarization using the PageRank algorithm [11].

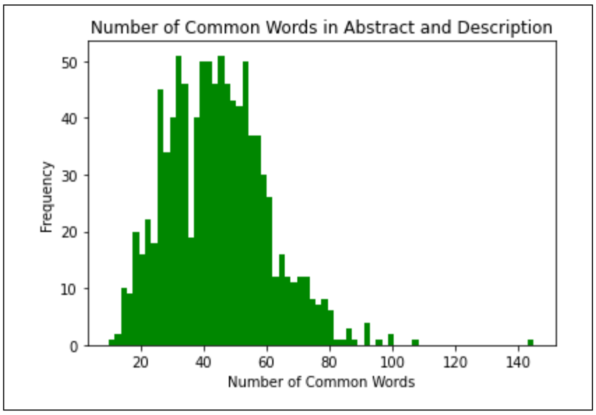

Figure 7: Number of common words between abstract and description

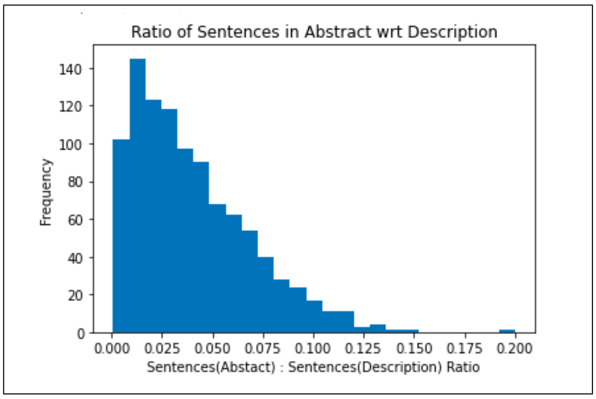

Figure 8: Ratio of abstract length and description length in terms of number of sentences

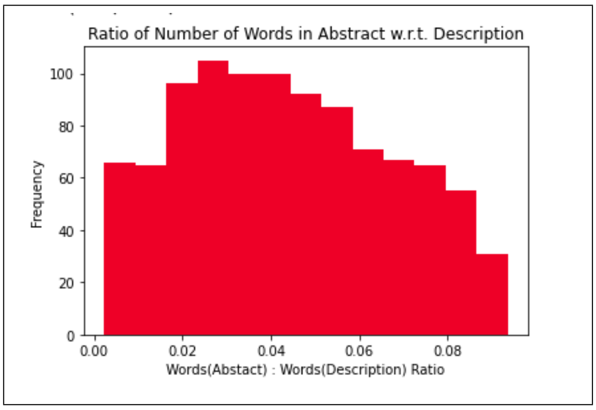

Figure 9: Ratio of abstract length and description length in terms of number of words

3.2.4 Using Cosine Similarity

The cosine similarity metric measures the similarity between two non-zero vectors of an inner product space by measuring the cosine of the angle between them. The similar vectors

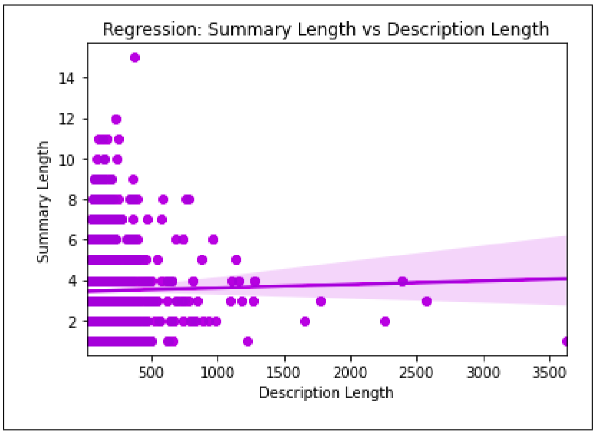

Figure 10: Regression between number of sentences in description and abstract

have the product closer to 0 and for different vectors, the product increases towards 1 [12]. BetterNLP pre-trained model for summarization was used along with cosine similarity for this task. It ranks sentences on the basis of cosine similarity and then picks the top ranking sentences from there to form the summary.

3.2.5 Using TF-IDF Ranking Method

TF-IDF (Term Frequency Inverse Document Frequency) finds out the rank of the sentences using the frequency with which the terms occur in it and occur in general in the corpus [13]. BetterNLP pre-trained model for summarization was used along with TF-IDF similarity for this task. It ranks sentences on the basis of TF-IDF similarity and then picks the top ranking sentences from there to form the summary.

The abstractive summary does not pick words and phrases directly from the text but rather paraphrases the text. It uses a different set of vocabulary to construct the final summary. The following model was used for the abstractive summary.

3.2.6 PEGASUS

PEGASUS is a pre-trained large Transformer-based encoder-decoder model[14]. In PEGASUS, important sentences are removed/masked from an input document and are generated together as one output sequence from the remaining sentences. It has been trained on different datasets. In specific, the model used here was trained on the big-patent dataset.

3.2.7 L-PEG

We tried another architecture, to combine the extractive and abstractive summarisation techniques. We trained an LSTM of 25 neurons and fed its output to Pegasus as input. The idea was to etract the key phrases and allow Pegasus to mask only those phrases, before outputting the final summary. The architecture is illustrated in Figure **??**.

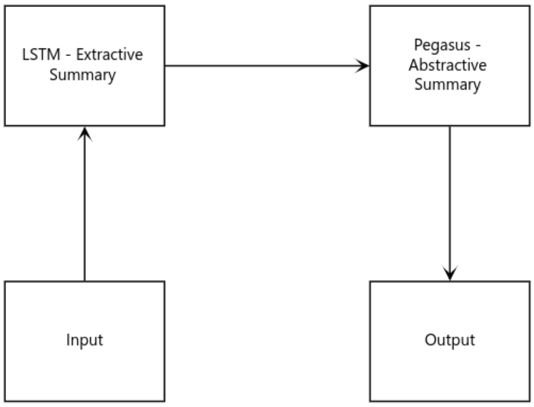

Figure 11: L-PEG: Architecture

The state of the art methods for summarisation on BigPatent Dataset include: Transformer-Base and Pegasus (base and large) models. The best ROUGE score obtained is by Pegasus for BigPatent dataset, which is 53.63.

3.3 Retrieval Models

The Big Patent dataset contains US patent applications filed under Cooperative Patent Classification Code (CPC). The 9 CPC categories are: A (Human Necessities), B (Performing Operations; Transporting), C (Chemistry; Metallurgy), D (Textiles; Paper), E (Fixed Constructions), F (Mechanical Engineering; Lightning; Heating; Weapons; Blasting), G (Physics), H (Electricity), and Y (General tagging of new or cross-sectional technology).

In order to perform patent retrieval on our dataset, we first tried text classification in order to classify the input queries into a CPC class and then use retrieval models to retrieve top similar patents in the predicted class so as to reduce the search space and get faster results.

3.3.1 Text Classification

A TF-IDF feature vector of size 10000 was used to train multiple machine learning models such as Gaussian/ Multinomial Naive Bayes, Support Vector Machine, random Forest and Multi Layer Perceptron. However, the text classification models did not yield satisfactory results and thus retrieval was performed on patents of all classes for each query.

3.3.2 Boolean Retrieval Models

Basic patent retrieval was performed using the boolean AND query model on an inverted index created on the patent database.
Another retrieval model used positional bigram matching between the query and patent documents to obtain results. A positional index was created for this model.

3.3.3 Probabilistic Model: BM25

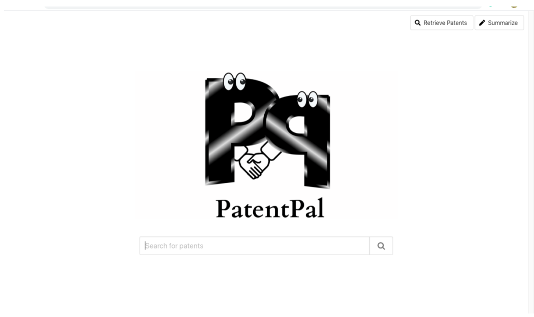

Figure 12: Home Page of PatentPal

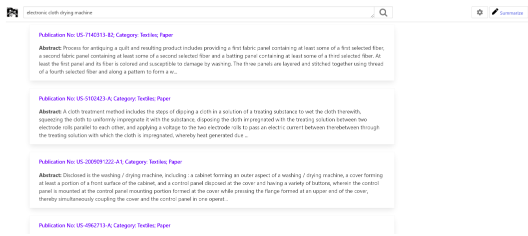

Figure 13: Retrieval Page of PatentPal

The Okapi BM25 [1] model based on probabilistic retrieval framework was used to perform ranked retrieval on the patent documents. The BM25 model is a bag-of-words type retrieval system that uses the TF-IDF features of the documents and performs term frequency scaling and document length normalisation so as to ensure that the ranking is not affected by the document length.

3.4 Platform

We created an interactive platform with options for search and summarisation of patents. The search functionality works on BM25 model and the summarisation works on pretrained PEGASUS model by Google. The different cases handled are discussed in evaluation section.
The frontend code has been developed using HTML, CSS, Javascript, Bulma, and Python-Flask. We have used pickled models for the sake of quick retrievals and summaries. A database has not been used due to compute intensive nature of the problem. Screenshots of different pages follow in Figure 12 - 14.

4. EVALUATION

We evaluated the three dimensions we worked on: Summary, Retrieval, and the platform we created.

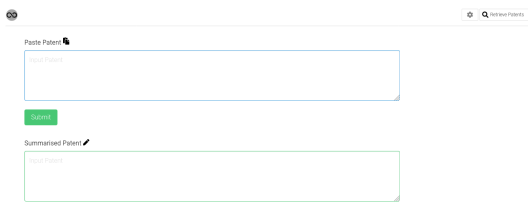

Figure 14: Summary Page of PatentPal

4.1 Summarisation Models

The summaries generated were scored using ROUGE score. ROUGE-1, ROUGE-2 and ROUGE-L were used to check correctness of the generated summaries against the human written GOLD summaries available as abstract along with descriptions.

ROUGE-1 matches the number of unigrams between the two summaries. ROUGE-2 similarly matches the number of bigrams. ROUGE-L finds the ratio of Longest Common Subsequence Length between the two summaries generated.

The results obtained for different models are summarised in Table 1.

Pegasus Large is the state of the art model which prvides a ROUGE scores (Recall) as 53.63, 33.16, 42.25 for ROUGE-1 Recall, ROUGE-2 Recall and ROUGE-L Recall. The Rouge-1 Recall score we obtained for PEGASUS was slightly better but that could be because of the smaller dataset used due to limited computational power available.

PEGASUS - the abstractive model performs the best. It has the highest ROUGE scores. It probably happens because the pre-trained model used has been trained on Big-PATENT dataset itself. Also, the gold summaries that are being compared against have been written by humans who use the abstractive method by paraphrasing the main content.

The L-PEG model provided significantly better results as compared to other extractive models, but the results were not comparable to pre-trained PEGASUS. Hence we integrated the pre-trained Pegasus large model in the platform created. However, one limitation that PEGASUS has is huge memory requirements and longer processing time due to deep learning approach.

4.2 Retrieval Models

Text classification models were used to classify the queries into CPC codes. Patent retrieval was performed using boolean and probabilistic models.

4.2.1 Text Classification

The performance of the text classification models was evaluated using the accuracy metric.

The models were unable to perform well on the data as can be seen from the accuracy scores. A possible reason could be because the documents in different classes contained similar words.

4.2.2 Retrieval Models

Since the Big Patent Dataset does not contain annotations for query-document relevance, a precise evaluation metric like DCG or precision/ recall could not be used, and hence we used Cosine Similarity between the queries and Documents to evaluate the performance of the retrieval models.

The results show that BM25 yields a better average cosine similarity, as boolean retrieval methods lead to document famine as they search for exact terms.

4.3 Platform Created

To handle varied inputs by user, we performed certain input validations and checks. For retrieval, we used the following quality checks for the query:

1. We performed a basic spell check to correct the spelling errors in input query.
2. For a query consisting of only stopwords or punctuation marks, we do not return any records.
3. We preprocess the query with same steps as the documents when indexing was done.
4. The system does not crash on any small or very large queries and considers all terms for large queries.
5. Even for a very irrelevant query, we try to get top 10 documents based on BM-25 ranking.

For summary, the following checkpoints were used:

1. A blank input is not accepted.
2. An input of less than 100 words is not considered and a flash message is displayed.
3. Even if an irrelevant input is given, the best possible summary produced by the model trained on big-patent dataset is returned.

5. CONCLUSION

We created a platform to summarise and retrieve patents. We also tried and compiled code for various indexing and abstractive and extractive summarisation methods. Additionally, we tried to create a model working on the best of both worlds: abstractive and extractive summary methods. However, we can conclude that abstractive methods based on deep learning definitely perform better than extractive methods, but need to be optimised more for performance and time gain.

There were certain challenges like a computation heavy problem but limited computational resources available. Another challenge was lack of literature in retrieval for patents specifically. Also, there was no annotated data available for query and document relevance due to which metrics like nDCG, precision, recall could not be used. For text classification, low accuracies were obtained due to which we could not reduce search space. Primary reason could be code generated queries which might not be representative of the actual human needs.

The project has scope for further improvement. The PEGASUS model used takes about 14GB RAM and 1 minute to return an output. Its memory requirements and time could be optimised. Apart from that, the score it attains is also only about 0.53. This could also be improved. We can also generate a dataset of actaul queries using logs and

Model	ROUGE-1		ROUGE-2		ROUGE-L	
Name	Recall	F1	Recall	F1	Recall	F1
Extractive Summrisation						
LSTM	36.11	33.64	9.70	8.95	21.2	19.68
Bi-LSTM	36.37	33.58	9.90	9.12	21.53	19.76
TextRank	24.57	31.66	6.77	8.83	13.85	17.97
Gensim	20.96	25.47	4.73	5.83	12.93	15.67
Cos score	24.54	30.47	7.37	9.16	14.48	18.00
TF-IDF	25.66	30.39	6.30	7.56	14.14	16.79
Abstractive Summrisation						
L-PEG	49.14	27.10	15.64	8.20	34.29	18.52
PEGASUS	57.72	36.41	25.43	15.73	41.32	25.73

Table 1: ROUGE scores for different summarisation models

Model	Train Accuracy	Test Accuracy
Gaussian NB	0.4869	0.2695
Multinomial NB	0.5478	0.3913
SGD	0.4347	0.2956
Linear SVC	0.5652	0.3217
Random Forest	0.05217	0.0782
SVM	0.5304	0.3130
MLP	0.4695	0.2086

Table 2: Performances of Text Classification Models

Model	Average Cosine Similarity for 50 queries
Boolean AND query Retrieval	0.0182315
Positional Bigram retrieval	0.000201
BM25	0.0882715

Table 3: Performances of Text Classification Models

then annotate the data to use more sophisticated models of machine learning and deep learning to improve the retrieval scores.

6. REFERENCES

[1] Pirmin Lemberger. Deep learning models for automatic summarization. *arXiv preprint arXiv:2005.11988*, 2020.

[2] Amy Trappey, Charles Trappey, and Chun-Yi Wu. Automatic patent document summarization for collaborative knowledge systems and services. *Journal of Systems Science and Systems Engineering*, 18:71–94, 03 2009.

[3] Jie Hu, Shaobo Li, Yong Yao, Liya Yu, Yang Guanci, and Jianjun Hu. Patent keyword extraction algorithm based on distributed representation for patent classification. *Entropy*, 20:104, 02 2018.

[4] Jen-Yuan Yeh, Hao-Ren Ke, and Wei-Pang Yang. ispreadrank: Ranking sentences for extraction-based summarization using feature weight propagation in the sentence similarity network. *Expert Systems with Applications*, 35:1451–1462, 10 2008.

[5] Pawan Sharma, Rashmi Tripathi, and R.C. Tripathi. Finding similar patents through semantic query expansion. *Procedia Computer Science*, 54:390–395, 12 2015.

[6] Lea Helmers, Franziska Horn, Franziska Biegler, Tim Oppermann, and Klaus-Robert Müller. Automating the search for a patent's prior art with a full text similarity search. *PLOS ONE*, 14:e0212103, 03 2019.

[7] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. page 10, 01 2004.

[8] Eva Sharma, Chen Li, and Lu Wang. Bigpatent: A large-scale dataset for abstractive and coherent summarization. pages 2204–2213, 01 2019.

[9] Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. 2014.

[10] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

[11] Keyvan Khosrovian, Dietmar Pfahl, and Vahid Garousi. Gensim 2.0: A customizable process simulation model for software process evaluation. In *International conference on software process*, pages 294–306. Springer, 2008.

[12] Faisal Rahutomo, Teruaki Kitasuka, and Masayoshi Aritsugi. Semantic cosine similarity. In *The 7th International Student Conference on Advanced Science and Technology ICAST*, volume 4, page 1, 2012.

[13] Cheng-Hui Huang, Jian Yin, and Fang Hou. A text similarity measurement combining word semantic information with tf-idf method. *Jisuanji Xuebao(Chinese Journal of Computers)*, 34(5):856–864, 2011.

[14] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.

7. APPENDIX

Summary Examples

Golden Summary

a fitting for inserting a needle or catheter into a blood vessel and for positionally stabilizing the inserted needle or catheter in situ includes separate needle holder and stabilizer elements . a retainer member is secured at its proximal end to the holder and extends in spaced relation along the holder to a distal end . the stabilizer has a bi - wing shape with a slot contoured to receive the retainer member in longitudinally slidable relation at a location below a holder supporting region on the top surface of the stabilizer . the wings can be flexed upward and toward one another to positively grip the holder for insertion of the needle or catheter . when unflexed , the stabilizer slot constrains the retainer element which in turn holds the stabilizer in the space between the retainer and the holder . removal of the needle is achieved by withdrawing the retainer longitudinally from the stabilizer slot .

PEGASUS Summary
A needle support includes a holder member for supporting a needle and a retainer member extending between the holder member and the rearward end of the needle. The retainer member is resiliently urged against the holder member so as to preclude mutual axial displacement of the needle and holder member during insertion of the needle into a blood vessel.

LSTM Summary
holder member 11 may be molded about the contained rearward end of needle 12 or may be otherwise secured thereto in any conventional manner so as to preclude mutual axial displacement of needle 12 and holder 11 during insertion of the needle into a blood vessel in the manner described herein . portions of bottom surface 22 are provided with a suitable adhesive coating , as is conventional for such applications , so that the bottom surface can adhere to a patient &# 39 ; s skin for initial positional stabilization after insertion of needle 12 into a blood vessel . in order to withdraw the needle from the patient &# 39 ; s blood vessel , the surgical tape is first removed .', 'a fitting for inserting a needle or catheter into a blood vessel and for positionally stabilizing the inserted needle or catheter in situ includes separate needle holder and stabilizer elements . a retainer member is secured at its proximal end to the holder and extends in spaced relation along the holder to a distal end . the stabilizer has a bi - wing shape with a slot contoured to receive the retainer member in longitudinally slidable relation at a location below a holder supporting region on the top surface of the stabilizer . the wings can be flexed upward and toward one another to positively grip the holder for insertion of the needle or catheter . when unflexed , the stabilizer slot constrains the retainer element which in turn holds the stabilizer in the space between the retainer and the holder . removal of the needle is achieved by withdrawing the retainer longitudinally from the stabilizer slot .