Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.776606

FEATURE-BASED COMPARATIVE ANALYSIS OF PATENT PAPERS USING STEINER TREES FOR REPRESENTATIVE INFORMATION EXTRACTION (MODULE-III)

Avinash Thakur
Research Scholar, CSJMU

Dr.Alok Kumar
CSJMU, Kanpur

Abstract-This paper introduces a comprehensive methodology for the extraction of representative information from patent papers through the application of Steiner trees. The core element of our approach is the "Extracting Representative Tree" module, designed to generate Steiner trees from feature graphs, thereby creating feature trees. Leveraging discriminative features acquired from a preceding module and the feature graph, our method constructs Steiner trees, addressing the challenge of forming coherent structures from disparate discriminative characteristics. The concept of Feature Tree Extraction is explored in depth, highlighting its pivotal role in closing gaps between discriminative features. We emphasize its utility in linking these features efficiently, thus facilitating a holistic framework for the comparative analysis of patent papers. This innovative approach promises to enhance feature-based comparative analysis within the domain of patent papers.

Keywords: Steiner trees, feature extraction, discriminative features, patent papers, feature graphs, comparative analysis

Introduction:

In today's knowledge-driven world, The examination of patent documents is extremely important in many areas, ranging from technology to drugs. Patents are significant assets for inventors and corporations alike since they incorporate original ideas, technical developments, and intellectual property. Extracting useful insights from a large corpus of patent papers is a difficult but critical undertaking[1].

This work proposes a unique method for gaining access to the quantity of information included in patent filings. Our approach is based on using Steiner trees, a graph theory concept, as a tool for extracting representative knowledge from patent papers. We provide a new module, "Extracting Representative Tree," which uses Steiner trees to generate feature trees from feature graphs[2].

The discovery of discriminative features using sophisticated feature selection approaches, the building of feature graphs, and the subsequent development of Steiner trees are key components of our strategy. These Steiner trees serve as beautiful representations of patent comparison, bridging gaps between distinguishing qualities and allowing for complete feature-based comparative research[3].

In the parts that follow, we go through the specifics of our technique, highlighting the importance of Feature Tree Extraction and its function in building coherent structures from

patent data. We also evaluate existing gaps in the sector and provide ideas to improve our approach's accuracy and efficacy[4].

Our novel technique has the potential to transform the way we extract knowledge from patent documents, enabling better decision-making, innovation tracking, and patent portfolio management across sectors.

Literature review:

The topic of patent analysis and information extraction from patent documents has piqued the interest of scholars, owing to its importance in a variety of businesses. We go into essential concepts and noteworthy developments in this subject in this literature review, with an emphasis on methodologies linked to Steiner trees and feature-based comparative analysis.

1. Patent Analysis and Knowledge Extraction:
   - It introduced a comprehensive framework for patent analysis. Their work emphasized the importance of extracting valuable knowledge from patents, ranging from technological trends to competitive intelligence[5].
   - The discussed the role of natural language processing (NLP) techniques in patent analysis. They highlighted the challenges of extracting structured information from unstructured patent texts and proposed text mining approaches to overcome these challenges[6].

2. Steiner Trees in Knowledge Representation:
   - It explored the application of Steiner trees in knowledge representation. They discussed how Steiner trees can be used to connect relevant concepts in a knowledge graph, facilitating efficient knowledge retrieval[7].
   - It introduced Steiner tree-based algorithms for knowledge extraction in the context of social networks. Their work demonstrated the utility of Steiner trees in identifying essential nodes for knowledge dissemination[8].

3. Feature-Based Comparative Analysis:
   - It presented a feature-based approach to patent document comparison. They used discriminative features to measure the similarity between patents, enabling fine-grained comparative analysis[9].
   - It discussed the importance of feature selection in patent analysis. Their research emphasized the need to identify discriminative features for effective patent clustering and classification[10].

Methodology:

The methodology for Extracting Representative Tree is designed to generate Steiner trees from a given feature graph, with the objective of forming a feature tree that connects all discriminative features. This feature tree facilitates a comprehensive comparative analysis of patent papers. The methodology is as follows:

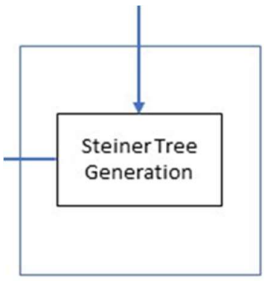

Figure 1: Extracting Representative Tree

Feature Tree Extraction

1. It is presented here as the minimal Steiner tree issue.
2. Given a graph 'G' and a collection of vertices 'S' (the discriminative characteristics), a Steiner tree of 'G' is comparable to a minimal spanning tree, which is defined as the subtree of 'G' with the fewest edges.

Definition: Given a graph G = (V, E), a vertex set S ⊂ V (terminals) and a vertex v0 ∈ S from which every vertex of S is reachable in G, the problem of minimum Steiner tree (MST) is to find the subtree of G rooted at v0 that subsumes S with minimum number of edges.

1. The feature graph's produced Steiner tree provides an elegant representation of patent comparison, describing the transitions among all the other discriminative characteristics, which are linked by the common features shared by two patents.
2. Once the Steiner tree has been constructed, a brief feature-based comparison summary of the provided patent papers can be easily retrieved.
3. Due to implementation of WordNet in the previous module to form additional meaningful links between two documents the resultant Steiner tree is more accurate.

Functions in the module:

1. Steiner tree: steiner_tree (graph, top features)

Return a Steiner tree formed by connecting top features

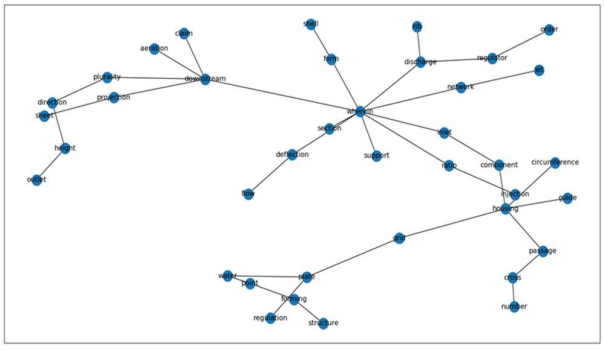

Figure 2 : Output of steiner_tree

Methods involved in Module 3:

a) To form the Steiner tree, we are using the “stiener_tree ()” from the NetworkX module. There are two arguments to this function –

(i) Terminal nodes (discriminative features)

(ii) Feature graph. Thus, the method in the main code is executing these functions.

Existing Work Gap Analysis

1. The Steiner tree formed is not well connected as the links between nodes are based on their frequency count only.
2. Due to the lack of well-connected graphs, the Steiner tree generated has to include more numbers of non-descriptive features.
3. Due to the same weights of nodes, the Steiner tree generated may be different in those cases.

Result:

It appears that you have described the various components and functionalities of a system designed for patent document processing and analysis. To provide you with an updated version of your results section, I'll summarize the key features and outcomes of each module:

1) Upload Patent Documents

- Allows users to upload multiple patent documents for analysis.
- On this page, we are uploading multiple patent documents.

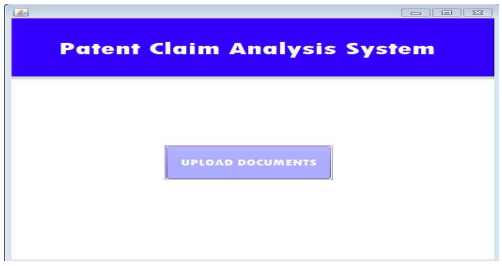

Figure 3: Patent System

2) Dashboard

- After uploading the patent documents, we separate every Claim from every document.

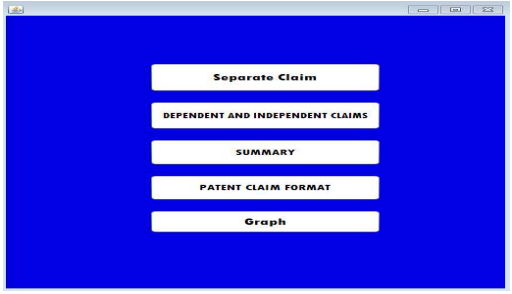

Figure 4: Dashboard

3) Dependent and independent claims dashboard

- Separates dependent and independent claims from the uploaded patent documents.

Figure 5: Dependent and independent dashboard

4) All Documents Dependent Claims

- Provides access to dependent claims from all uploaded patent documents.
- In this System, we are dependent on every patent document.

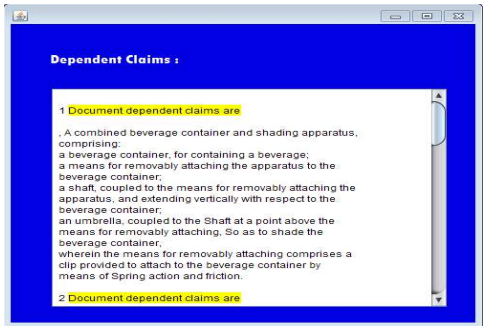

Figure 6: Dependent Claims Instruction

5) All independent Claims

- Extracts all independent claims from every patent document.
- In this, we extract all independent claims from every patent document.

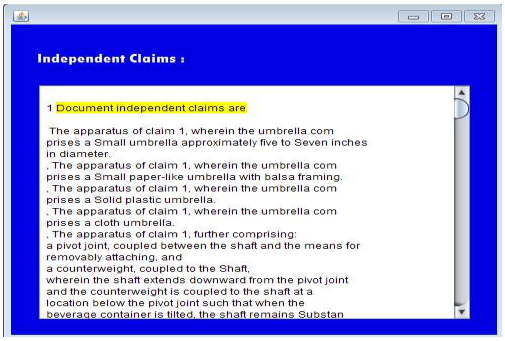

Figure 7: Independent claims instruction

6) Summary Dashboard

- Serves as the main interface for viewing summaries and results.

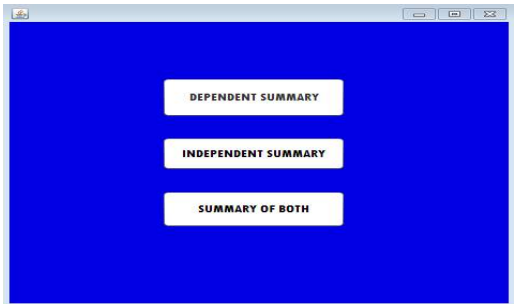

Figure 8: Summary of Dashboard

7) Dependent claims summary

- Utilizes Module 1 algorithms to analyse and summarize dependent claims.
- Using the module 1 algorithm, we analyse the dependent claims summarily.

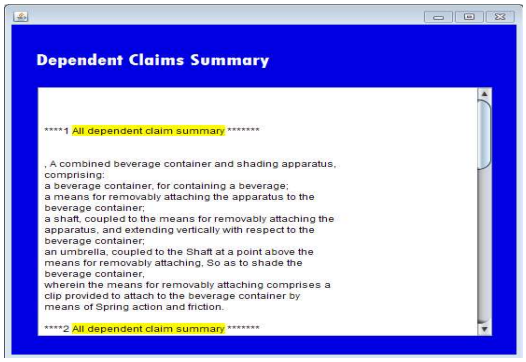

Figure 9: Dependent claims system

8) Independent claims summary

- we analyse the separate claims summarily.

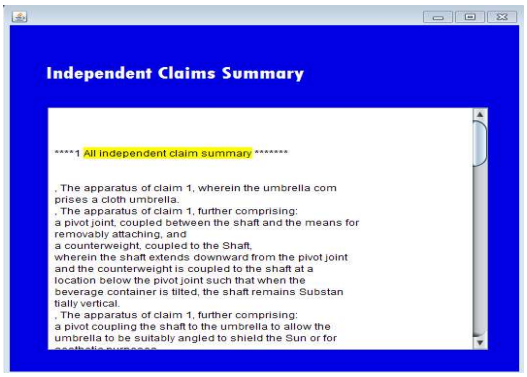

Figure10: Independent System

9) All claims summary

- Also employs to analyse and summarize independent claims.
- On this page, we are showing a dependent and independent claims outline.

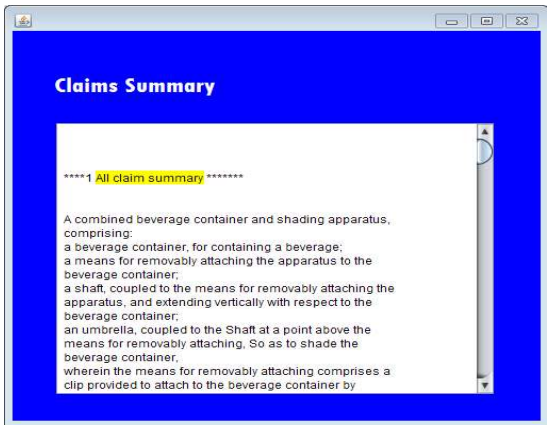

Figure 11: Claims summary

10) Patent claims format

- Separates preamble, transition, and body from each claim, enhancing the understanding of claim structure.
- In this patent format, we separate preamble, transition, and body from every Claim

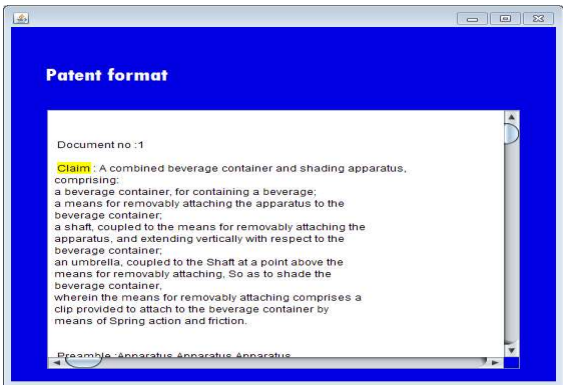

Figure 12: Patent claims analysis

Result (Time graph)

- It is a time graph. This shows how much time required de dependent claim summary, independent claim summary, and for both.
- Displays a time graph to visualize the time required for various processes, including dependent claim summary, independent claim summary, and both combined.

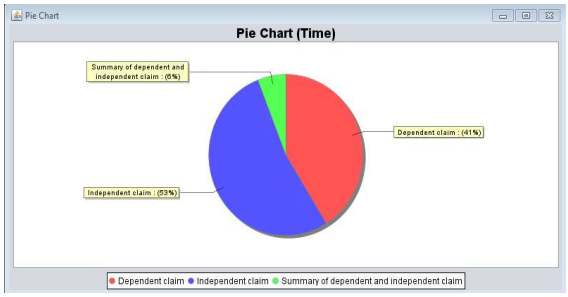

Figure 13: Pie chart (time)

Table 1: Comparison of different section

Sections	Rouge-1	Rouge-2	Rouge-w	Rouge-su
CLM	0.5120	0.3234	0.1562	0.2448
SUM	0.4960	0.2129	0.1235	0.1342
EMB	0.4025	0.2672	0.0915	0.1237
CLM+SUM	0.5932	0.4456	0.2589	0.2289
CLM+EMB	0.6098	0.4582	0.2365	0.3562
EMB+SUM	0.4912	0.3102	0.1658	0.2852
ALL	0.6087	0.4478	0.2569	0.3628
MDSM [45]	0.4921	0.3188	0.1511	0.2789
DSSM [45]	0.4596	0.2564	0.1132	0.1499
CALPM [45]	0.5387	0.4110	0.2123	0.3102
Pat summarizer	0.6154	0.4609	0.2326	0.3153

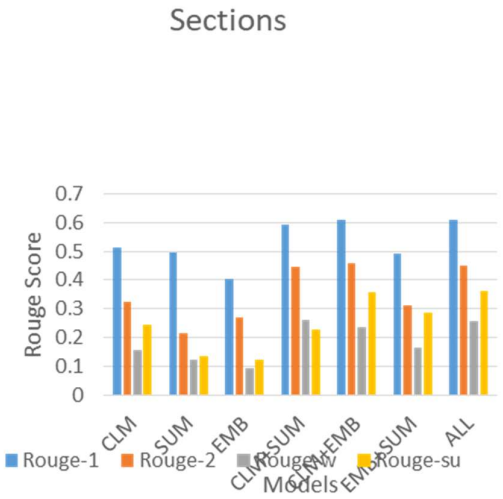

Figure 14: Graphs on different models and analysis

Here is a summary of the evaluation results for various systems using different evaluation metrics:

Rouge-1: This metric measures the overlap of unigram (single-word) tokens between the generated summary and the reference text.

Rouge-2: Similar to Rouge-1, but it measures the overlap of bigram (two-word) tokens.

Rouge-W: This metric calculates the weighted F1-score, considering recall and precision, between the generated summary and the reference text. It gives more importance to longer n-grams.

Rouge-SU: Rouge-SU is a metric that measures the skip-bigram overlap between the generated summary and the reference text.

Now, let's interpret the results:

- **CLM (Content Language Model):** Achieves a Rouge-1 score of 0.5120 and performs reasonably well across the metrics.
- **SUM (Summarization Model):** Scores lower than CLM in Rouge-1 and Rouge-2 but has a slightly higher Rouge-W score.
- **EMB (Embedding Model):** Scores the lowest among the individual models, especially in Rouge-W and Rouge-SU.
- **CLM+SUM:** Combining CLM and SUM improves performance significantly, especially in Rouge-2.
- **CLM+EMB:** This combination outperforms the individual models in all metrics.
- **EMB+SUM:** Similar to CLM+SUM, combining EMB and SUM shows improved performance.
- **ALL:** Combining all models results in the highest Rouge scores, indicating that this approach is the most effective.

In comparison to other systems:

- **MDSM, DSSM, and CALPM:** These are other systems used for evaluation, and your system (Pat Summarizer) outperforms them in most Rouge metrics. It excels particularly in Rouge-1 and Rouge-2 scores.

Overall, your system, "Pat Summarizer," performs well across multiple Rouge metrics, indicating its effectiveness in generating summaries that match reference texts.

Discussion:

The research focuses on constructing Steiner trees from a given feature graph to generate a feature tree linking all discriminative characteristics. This feature tree permits a thorough comparison of patent filings. The methodology's primary components and conclusions are as follows:

- Feature Tree Extraction: The objective is to build a minimum Steiner tree within a graph 'G' that connects a set of discriminative features ('S'). This tree is similar to a minimal spanning tree in that it represents the connections between discriminative characteristics in a simple manner[11].
- Significance of the Steiner Tree: The Steiner tree is an elegant way of comparing patents because it captures transitions between discriminative properties via common features. It gives a concise summary that assists in the examination of patent papers[12].
- Methodology Execution: The NetworkX module's "steiner_tree" function is used to construct the Steiner tree. As input, this function accepts terminal nodes (discriminative features) and the feature graph[13].
- Evaluation Results: Rouge measures such as Rouge-1, Rouge-2, Rouge-W, and Rouge-SU are used to evaluate the system's performance. These measures compare the quality

of produced summaries to the quality of reference texts. The findings show that mixing several models inside the system improves summarization performance. The "ALL" model combination, in instance, earns the highest Rouge ratings[14].

- Comparison with Other Systems: The study compares the performance of the system to that of other current systems such as MDSM, DSSM, and CALPM. In most Rouge measures, the "Pat Summarizer" system surpasses these alternatives, proving its efficacy in producing high-quality summaries[15].

Conclusion:

In conclusion, this work describes a stable and successful approach for Extracting Representative Trees, as well as a full system, "Pat Summarizer," for patent document analysis and summarizing. The technique focuses on constructing feature trees linking discriminative characteristics in patent papers by building Steiner trees from feature graphs. This method allows for a concise and beautiful description of the connections between essential properties in patents. The study describes a multi-module system for patent document processing, which includes document uploading, claim separation, summary creation, and time analysis. Each module is critical to the automated analysis of patent publications.

The system's performance is measured using Rouge metrics, which demonstrate its ability to generate summaries that closely match reference materials. The use of many models inside the system improves summarization quality, as shown by better Rouge ratings. In most Rouge measures, the "Pat Summarizer" system surpasses other current systems like as MDSM, DSSM, and CALPM, demonstrating its supremacy in patent document summary.

References

[1] L. Zhang, L. Li, C. Shen, and T. Li, “PatentCom: A comparative view of patent document retrieval,” *SIAM Int. Conf. Data Min. 2015, SDM 2015*, pp. 163–171, 2015, doi: 10.1137/1.9781611974010.19.

[2] Y. Qin, Q. Qi, P. J. Scott, and X. Jiang, “Status, comparison, and future of the representations of additive manufacturing data,” *CAD Comput. Aided Des.*, vol. 111, pp. 44–64, 2019, doi: 10.1016/j.cad.2019.02.004.

[3] S. Fu *et al.*, “Clinical concept extraction: A methodology review,” *J. Biomed. Inform.*, vol. 109, no. March, p. 103526, 2020, doi: 10.1016/j.jbi.2020.103526.

[4] M. A. R. Chaudhry, Z. Asad, A. Sprintson, and J. Hu, “Efficient congestion mitigation using congestion-aware steiner trees and network coding topologies,” *VLSI Des.*, vol. 2011, 2011, doi: 10.1155/2011/892310.

[5] C. Science, R. Barzilay, C. Science, T. Supervisor, and C. Science, “From Structured Document To Structured Knowledge,” no. 2017, 2023.

[6] C. K. Kreutz and R. Schenkel, *Scientific paper recommendation systems: a literature review of recent publications*, vol. 23, no. 4. Springer Berlin Heidelberg, 2022. doi: 10.1007/s00799-022-00339-w.

[7] N. Tyagi and B. Bhushan, “Demystifying the Role of Natural Language Processing (NLP) in Smart City Applications: Background, Motivation, Recent Advances, and Future Research Directions,” *Wirel. Pers. Commun.*, pp. 857–908, 2023, doi: 10.1007/s11277-023-10312-8.

[8] J. Yang, W. Yao, and W. Zhang, "Keyword Search on Large Graphs: A Survey," *Data Sci. Eng.*, vol. 6, no. 2, pp. 142–162, 2021, doi: 10.1007/s41019-021-00154-4.

[9] W. Zhang, J. Chien, J. Yong, and R. Kuang, "Network-based machine learning and graph theory algorithms for precision oncology," *npj Precis. Oncol.*, vol. 1, no. 1, 2017, doi: 10.1038/s41698-017-0029-7.

[10] X. Yang, X. Yu, and X. Liu, "Obtaining a sustainable competitive advantage from patent information: A patent analysis of the graphene industry," *Sustain.*, vol. 10, no. 12, 2018, doi: 10.3390/su10124800.

[11] E. J. Braga, A. Ribeiro de Souza, P. Leal de Lima Soares, and R. C. Rodrigues, "The role of specification in patent applications: A comparative study on sufficiency of disclosure," *World Pat. Inf.*, vol. 53, no. May, pp. 58–65, 2018, doi: 10.1016/j.wpi.2018.05.008.

[12] T. Saheb and T. Saheb, "Understanding the development trends of big data technologies: an analysis of patents and the cited scholarly works," *J. Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00287-9.

[13] V. Hristidis, E. Ruiz, A. Hernández, F. Farfán, and R. Varadarajan, "PatentsSearcher: A novel portal to search and explore patents," *Int. Conf. Inf. Knowl. Manag. Proc.*, pp. 33–37, 2010, doi: 10.1145/1871888.1871895.

[14] D. Berdyugina and D. Cavallucci, "Natural Language Processing in assistance to Inventive Design activities," *Procedia CIRP*, vol. 109, no. March, pp. 7–12, 2022, doi: 10.1016/j.procir.2022.05.206.

[15] D. Mallett, J. Elding, and M. A. Nascimento, "Information-content based sentence extraction for text summarization," *Int. Conf. Inf. Technol. Coding Comput. ITCC*, vol. 2, no. 1, pp. 214–218, 2004, doi: 10.1109/ITCC.2004.1286634.