Case Similarity Detection in Legal Research

Stas Susha
Computer Science, HIT
propertysetter@gmail.com

June 04, 2025

Abstract

Legal professionals depend on precedents, yet retrieving relevant cases from large judicial datasets remains challenging—particularly in non-English contexts like the Israeli Supreme Court. This study presents a Legal Case Similarity Detection (LCSD) system that leverages semantic embeddings with citation-based features. From a larger corpus of 46,000 decisions, a subset of 100 cases was translated, summarized, and represented using both dense semantic vectors and sparse citation vectors. These representations are fused using a weighted hybrid scoring function that integrates cosine and Jaccard similarity. A Retrieval-Augmented Generation (RAG) component enables natural language interaction through a custom agent that answers legal queries. Evaluation on a manually labeled set shows that the hybrid approach achieved 0.83 Recall@3 and 0.63 Precision@5 on average, outperforming purely semantic or citation-bases methods in some cases. This research highlights the potential of LLM-driven hybrid retrieval systems to enhance legal research in complex judicial environments.

Keywords Legal research, Israeli Supreme Court Database, Legal case similarity, Artificial Intelligence (AI), Natural Language Processing (NLP), Large Language Model (LLM), Retrieval Augmented Generation (RAG), Hybrid search

1 Introduction

Precedence is a cornerstone of the legal system, ensuring consistency and fairness in judicial decisions [1], [2]. By adhering to established precedents, courts uphold stability and predictability, which are essential for maintaining the rule of law. This practice, known as *stare decisis* (Latin for "to stand by things decided"), guides the courts to make informed decisions based on prior rulings.

In legal research, the efficient retrieval and analysis of relevant precedents are crucial for legal professionals who rely on past cases to inform their strategies and predict outcomes. However, the vast volume of legal documents and complexity of legal language present significant challenges. Leveraging Machine Learning (ML) and Natural Language Processing (NLP) techniques can automate the identification and retrieval of similar legal cases. By employing sophisticated similarity measures, these systems can quickly find contextually relevant precedents, enhancing the efficiency and accuracy of legal research.

This study utilizes the dataset described in Sela and Schler's 2024 publication, "*The Israeli Supreme Court Database 1997-2022*"[3], which provides a rich collection of the Israeli Supreme Court decisions and extensive metadata. Using this comprehensive dataset, we aim to create a Legal Case Similarity Detection (LCSD) to assist jurists in their research and decision-making processes. Our approach combines Large Language Models (LLMs) with a hybrid search system that integrates retrieval-augmented generation (RAG) using dense vectors and citation-based references from law and other court cases (sparse vectors). This hybrid model aims to efficiently retrieve similar legal cases by considering both semantic content and citation relationship between documents. We will delve into the preprocessing of legal text written in Hebrew, the extraction of meaningful features and the application of the similarity metrics.

While prior works have explored case similarity detection in various jurisdictions, this research focuses specifically on Israeli Supreme Court cases, offering a novel approach for Hebrew-language legal texts. By addressing the challenges in developing the LCSD, this article contributes to the broader understanding of integrating Artificial Intelligence (AI) into legal domain research. It demonstrates the importance of efficient and accurate retrieval of legal precedents, which supports consistency, stability and transparency in judicial processes. Our work highlights AI's potential to transform Israeli legal research, making it more accessible and reliable for legal professionals.

2 Related work

We assume familiarity with AI techniques such as NLP, deep learning and LLMs. Therefore, we focus on methodologies applied to detecting similarity between the legal cases[4], [5], including traditional text-based methods, deep learning, knowledge graphs and transformer-based models.

A. Traditional Text Based Approaches

Early methods for case similarity detection relied heavily on text-based techniques such as Bag-of-Words (BOW), TF-IDF and cosine similarity[6], [7], [8], [9]. While straightforward, these methods often struggle with the complexity and variability of legal language. For example, Thenmozhi et al. 2017[6] used cosine similarity with lexical features, concepts (nouns) and relations (verbs) extracted from legal texts. Their approach utilized TF-IDF scores or Word2Vec embeddings and calculating cosine similarity to rank the prior curt case based on their relevance to current case.

Aryal et al. 2019 [7] presented a new similarity measure called SP (Simple Probabilistic), which uses probabilistic approach to assess similarity based on term occurrences, improving consistency over traditional methods. Similarly, Kumar et al. 2011 [8] evaluated multiple similarity measures in the Indian Supreme Court dataset and found that models using legal terms instead of general terms performed better.

B. Neural Networks and NLP

The advent of neural networks and NLP brought more sophisticated approaches, particularly transformer-based models. Mandal et al. 2017[10] applied topic modeling and neural network-based embeddings (Word2Vec and Doc2Vec) to Indian Supreme Court cases, using cosine similarity for relevance scoring. These models better capture the semantic context of legal text.

The study by Raphael Souza 2023 leveraged BERT-based models to analyze similarities between legal court documents[11]. The usage of deep learning and knowledge graphs may provide a structured way to incorporate legal knowledge constructed from the court cases documentation into the graphs, as explored by Jaspreet Singh Dhani et al. 2024 [12] and D. Cavar et al. 2018 [13].

C. Hybrid Approaches

Hybrid methods combine traditional techniques and advanced models to provide more robust solutions for case similarity detection[14], [15], [16], [17]. For instance, Bhattacharya et al. 2022 highlighted the importance of integrating citation networks with text embeddings to improve similarity detection[15]. This hybrid approach addresses the limitations of purely text-based methods by incorporating the relational context of legal documents. Similarly, Chavan et al. [16] presented CaseRex, a hybrid system combining Doc2Vec with citation network, based on hypergraphs, to capture both direct and indirect relationship between case.

Luis et al. 2024 [17] explore the effectiveness of the hybrid search combining sparse (BM25, SPLADE) and dense (DPR, ColBERT) models focusing on French language legal texts, demonstrating that hybrid models outperform standalone approaches when carefully tuned.

D. Domain Specific Models

Specialized models trained on legal corpora have significantly improved similarity detection. LegalBERT[18], is a publicly available LLM trained on a diverse set of legal texts, including EU, UK and US court cases. These domain specific embeddings enhance the performance of similarity measures by capturing domain-specific semantics.

Other models such as Law2Vec [19] (trained on ~123K English documents), BERT_LF 2022 [20] (trained on Chinese BERT) and InternLM-Law 2024 [21] (fine-tuned for Chinese legal domain), further improve text similarity detection.

E. RAG and Hybrid Search

Recent advancements in Retrieval-Augmented Generation (RAG) have focused on improving retrieval accuracy and efficiency by integrating semantic search techniques and hybrid retrieval approaches. Mahboub et al. (2024)[22] demonstrated how deep learning-based encoders can improve semantic search in RAG systems, particularly for Arabic language processing. Juvekar and Purwar (2024)[23] introduced COS-Mix, which fuses cosine similarity and distance measures with BM25 to improve retrieval efficiency, especially in sparse data settings. Sawarkar et al. (2024)[24] introduced *Blended RAG*, which combines dense vector retrieval with sparse encoder indexes, setting new benchmarks for datasets like NQ and TREC-COVID. It's particularly useful for systems like legal case retrieval, where both semantic meaning and citation relationships are crucial.

2.1 Trends and Trade-offs

The trend in recent research has been towards leveraging deep learning and domain-specific embeddings to improve similarity detection accuracy. The integration of knowledge graphs and citation networks has also gained traction, providing a more holistic view of legal documents.

However, these advancements come with trade-offs regarding the best approach to similarity detection **[4], [5]**. Traditional text-based methods, while simpler, often fail to capture the full context of legal cases. On the other hand, network-based methods can be more complex to explain and not always provide better solution to the end user. Hybrid approaches attempt to balance these issues but can be computationally intensive. Furthermore, domain-specific models face limitations when adapting to languages or jurisdictions not covered by the training data.

2.2 Existing Gaps

A. *Data Volume* and Quality
Many existing datasets are limited in scope, often covering only a subset of cases or lacking detailed metadata. In contrast, the Israeli Supreme Court Database[3], that includes 249,115 cases with full-text decisions and 135 metadata variables, providing a rich and comprehensive dataset for analysis.

B. Location and language
There is a limited availability of high-quality datasets for training NLP and ML models in the legal domain, particularly for non-English languages. The dataset used in this study is designed to support the development of NLP models for Hebrew legal texts, addressing a critical gap in the field and facilitating the advancement of AI-based applications for legal research in Israel.

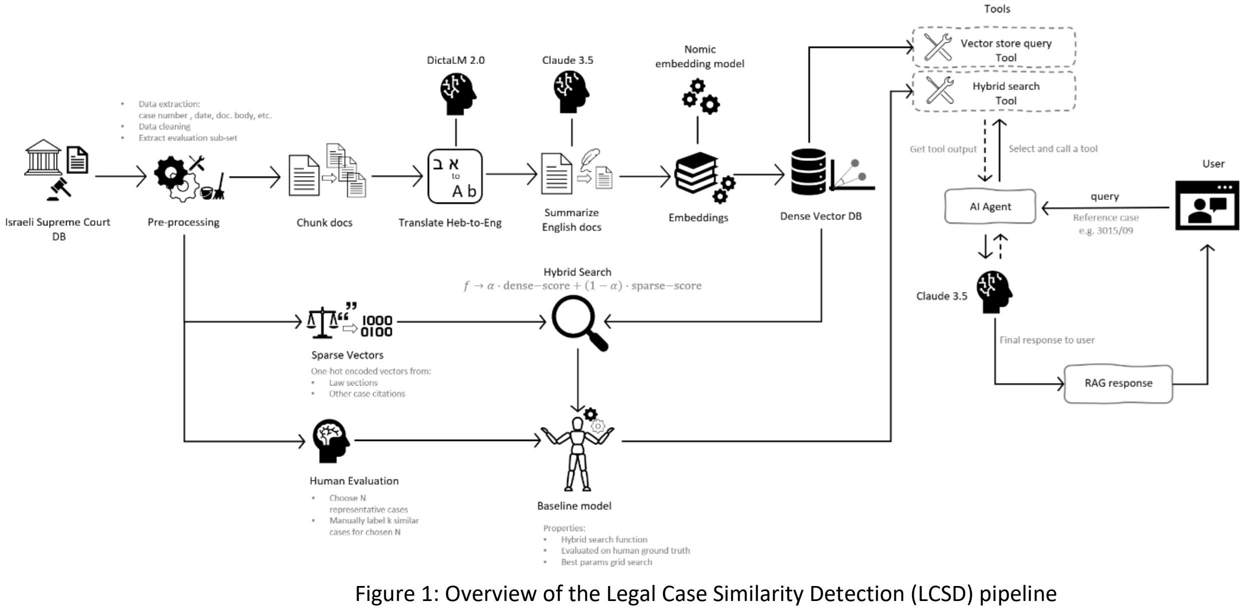


Figure 1: Overview of the Legal Case Similarity Detection (LCSD) pipeline

3 Methodology

This section outlines the pipeline used in developing the LCSD, highlighting the data preparation, modeling techniques, and retrieval strategy. The system was implemented in Python, leveraging tools such as LangChain for LLM orchestration, Ollama for LLM deployment on local machine, AWS Bedrock for accessing Anthropic Claude 3.5, and FAISS for vector-based retrieval. The project also uses standard math and data science libraries such as pandas, numpy, matplotlib, scikit-learn, etc.

Two complementary modeling approaches are used: semantic embeddings derived from language model summaries (dense vectors), and citation-based feature vector capturing legal references (sparse vectors). These representations are fused in a hybrid scoring function that balances the semantic meaning and legal structure for improved retrieval performance. A high-level overview of LCSD pipeline is shown in **Error! Reference source not found.**.

3.1 Data Source

The project uses the Israeli Supreme Court Database (1997-2022), a comprehensive dataset of over 249,000 legal cases with associated metadata and full text-decisions. From the full dataset we've selected 46,109 civil and criminal appeals submitted to the Supreme Court of Israel.

3.2 Preprocessing and Sampling

Due to a big number of 135 features and complex legal case document structure, preprocessing was a critical step in preparing the dataset for downstream tasks. It included:

- Column translation: Hebrew column names were mapped to standardized English labels using a predefined translation dictionary. This was done both to enable structured analysis and to improve readability when displaying and interacting with dataframes in Jupyter notebook.
- Text cleaning: The document body field was cleaned using regex to remove HTML tags, special characters and URLs.
- Filtering: documents with empty or missing content, likely because of the confidential information, were excluded
- Case sampling: From the dataset of ~46K records, a random subset of 100 legal cases was selected. This sample served as the working dataset for both development and performance evaluation, managing the trade-off between dataset scope and computational feasibility.

3.3 Translation and Summarization

To support semantic similarity retrieval, legal documents were first translated from Hebrew to English using DictaLM 2.0. This step was necessary due to empirical findings – embeddings generated directly from Hebrew performed poorly in the FAISS vector store, while English embeddings showed more coherent semantic vectors and better retrieval results.

The translated texts were then summarized using Anthropic Claude 3.5 via AWS Bedrock, producing concise structured representation reducing complexity and preserving essential information.

3.4 Vector Representation

Each legal case was represented using two vector formats to support hybrid similarity retrieval:

- Dense semantic vectors: English case summary was embedded using the Nomic text embedding model, which generates 768-dimensional vectors optimized for fast semantic search. To ensure compatibility with cosine similarity, the vectors were normalized to unit length before being inserted into FAISS vector index. These embeddings capture the semantic meaning of each legal case.
- Sparse citations vectors: separately, sparse binary vectors were created based on legal citations found, using regex, in the original Hebrew documents – such as references to laws and case numbers.

These vectors encode the relational legal information that is often missed in semantic models.

3.5 Hybrid Similarity Scoring

A hybrid scoring function integrates both semantic and citation-based features:

$$\text{Hybrid score} = \alpha \cdot \text{Dense score} + (1 - \alpha) \cdot \text{Sparse score}^{\gamma}$$

Where:

- α: weight parameter $\alpha \epsilon [0, 1]$ - balances the dense vs. sparse contribution
- γ: exponent scale-up to adjust the impact of sparse score (since sparse score values can be very low)
- Dense score: calculated using cosine similarity
- Sparse score: calculated using Jaccard similarity on binary vectors

Dense vector database: for efficient semantic retrieval, all normalized dense vectors were stored in a FAISS-based index, which support fast top-k nearest neighbor search using cosine distance.

Sparse matching: citation-based sparse vectors were compared using Jaccard similarity, computed separately and optionally scaled-up using power transformation to emphasize weak but meaningful citation overlaps (since spare score value can be relatively small value comparing to dense).

$$\text{Jaccard similarity}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Where:

- A, B: binary vectors, i – index in vector
- $A_i \cap B_i = 1$: intersection of citations (shared citations - same law or precedent)
- $A_i \cup B_i = 1$: union or total active citations (at least one cites the law or precedent)

3.6 RAG-based Information Retrieval

To improve interpretability and provide user friendly legal case exploration, we have included a Retrieval Augmented Generation (RAG) component. This module extends the hybrid search by enabling question answering over retrieved cases using a conversational AI agent.

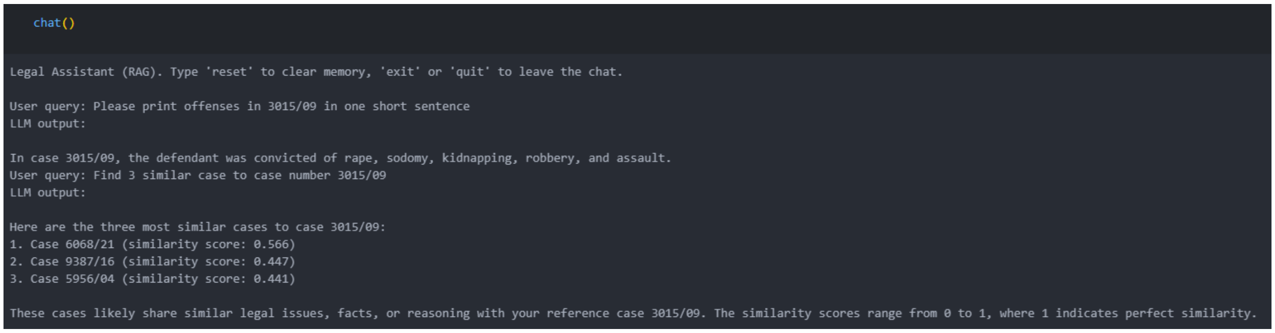
The RAG was built using LangChain a custom agent equipped with three tools:

- Hybrid search Tool-A: retrieves top-k (k provided by user) similar legal cases using the hybrid search function defined in section 3.5.
- Vector store query Tool-B: used to answer generic questions for a query provided by a user. It generates natural language responses based on the retrieved summaries from FAISS database. The responses produced by Claude 3.5 via AWS Bedrock. In practice Tool-B consists of two functions with same purpose: handling generic and case-specific queries, using document metadata to access directly the requested case content.

When a user provides a query (e.g., “Find 3 similar cases to case number 3015/09”), the system performs:

- Agent calls a tool (in our case Tool-A) with appropriate arguments (k=3, doc_id=3015/09). If Tool-B called the user query embedded using Nomic text embedding and passed to FAISS for document retrieval.
- Context creation – the agent takes the tool output and combines it with another context (optional).
- LLM answer generation – produces final response to the user

We implemented a command-line interactive chat to support multi-turn queries, with memory management and traceable outputs. See example in **Error! Reference source not found.**.



```
   chat()

Legal Assistant (RAG). Type 'reset' to clear memory, 'exit' or 'quit' to leave the chat.

User query: Please print offenses in 3015/09 in one short sentence
LLM output:

In case 3015/09, the defendant was convicted of rape, sodomy, kidnapping, robbery, and assault.
User query: Find 3 similar case to case number 3015/09
LLM output:

Here are the three most similar cases to case 3015/09:
1. Case 6068/21 (similarity score: 0.566)
2. Case 9387/16 (similarity score: 0.447)
3. Case 5956/04 (similarity score: 0.441)

These cases likely share similar legal issues, facts, or reasoning with your reference case 3015/09. The similarity scores range from 0 to 1, where 1 indicates perfect similarity.
```

Figure 2: RAG based QA example

4 Experiments and Evaluation

4.1 Evaluation metrics

A ground truth evaluation set was manually (non-expert) constructed by selecting 5 representative legal cases across different domains (e.g., driving offense, family law, fraud, terrorism or assault, robbery). For each, the top 3 most similar cases were identified based on manual reading, topic modeling, citation extraction and analysis, semantic review.

To enable better evaluation reliability, a parallel ground truth was generated using ChatGPT 4o, which retrieved its own top 3 similar cases for the same inputs. This dual ground truth - human and LLM-based - provides a more comprehensive comparison.

Two main metrics were used:

- Precision@k: proportion of top-k hybrid search retrieved cases that appear in the ground truth set

  $$\text{Precision@}k = \frac{|\text{Relevant cases in top-}k|}{k}$$

  Example: I have retrieved 5 documents, and 4 of them are relevant.

  $$\text{Precision@5} = \frac{4}{5} = 0.8$$

- Recall@3: how many of the top-3 human ground truth cases found using the hybrid search

  $$\text{Recall@3} = \frac{|top\text{-}k \text{ relevant cases found}|}{3}$$

  Example: I have retrieved [a, b, c, d, e], and ground truth [a, d, e]. All 3 appear in ground truth.

  $$\text{Recall@3} = \frac{3}{3} = 1.0$$

4.2 Experiment Results

We have executed a grid search across three hybrid search parameters α – dense vs. sparse weight, γ – power transformation applied to Jaccard score, and k – number of top retrieved cases (3, 5, 10).

We first identified the globally best performing configuration by averaging recall@3 and precision@k scores across all test cases and tools (either human or ChatGTP4o). This configuration was selected based on the optimal values of the hybrid parameters ($\alpha,\ \gamma$) that yielded the highest average scores (Figure 3, Figure 4)

Best configurations:

Metric	Alpha	Sparse Exp	k	Score
Recall@3	0.3	0.01	10	0.83
Precision@k	0.7	0.5	3	0.63

- Recall: average results indicate that, in most cases, semantic search provides stronger score support. In $k = 3$, citations caused degradation in results. On the other hand, in $k = 5$ and $k = 10$ the citations-based features demonstrated an added value, particularly when using $\alpha = 0.3$ or 0.5 in combination with a high scale-up factor of Jaccard score ($\gamma = 0.01$). In other cases, relying solely on citations (sparse vectors) resulted in poor performance.

- Precision: like in Recall average score results, we observe importance of the citations only in $k = 5$ and $k = 10$, with $\alpha \leq 0.5$, especially with high (low value) scale-up factor of Jaccard score ($\gamma = 0.01$). In other cases, using citation features alone led to poor performance.

Table 1 and Figure 5 present the Recall@3 scores for each test case and both tools (ChatGPT 4o, Human). For each value of top-k (3, 5, 10), we report the average, best, and worst recall achieved across all configurations. Table 2 and Figure 6**Error! Reference source not found.** provide the corresponding summary for Precision@k metric.

- Average – the mean recall and precision across all configurations. It reflects the expected performance of the system if the hybrid search parameters were selected randomly
- Best – maximum score per case and tool, represents the upper bound of system performance, assuming the best hybrid search parameters were selected for each case
- Worst – the minimum score observed per test case

4.3 Observations

- In both recall and precision on average the human annotation obtained higher score comparing to ChatGPT 4o, but the gap varies by case. Note: annotation done by non-professionals and may not represent the real-life use case.
- Case 9376/00 achieved near-perfect score for both tools - the legal cases are indeed very close semantically and rely on the same law.
- Case 5668/13 showed low recall and precision scores in some cases even 0.0, meaning that certain parameter configuration failed completely to retrieve any similar cases.
- The performance spread best vs worst was larger in ChatGPT 4o, probably more sensitive to parameter tuning.

4.4 Parameter Impact and Trade-off

- Alpha (α) – balances semantic vs citation weight. Moderate values 0.3-0.5 yield best trade-off. High values, such $\alpha = 1.0$, ignore citation and focus on language models embeddings.
- Sparse exponent (γ) – since all citations (law sections and case references) were combined into a single sparse vector, the resulting Jaccard similarity scores were very low. To enhance the impact of sparse vectors in the hybrid search calculation, we applied power transformation. This had noticeable effect only when the exponent was highly scaled up (i.e., lower γ values such as $0.1 - 0.01$, resulted in stronger amplification).
- k – the k itself is not tunable parameter but rather an indicator of the hybrid search performance with different top-k values. Increasing k boosts the Recall but lowers the Precision. Higher hybrid score in small k value indicates better quality.
- Our recommendations for parameter tuning in different use cases:

Use case	Parameter set
Max recall	$\alpha = 0.3$, $\gamma = 0.01$
Max precision	$\alpha = 0.7$, $\gamma = 0.5$
Balanced	$\alpha = 0.5$, $\gamma = 0.1$

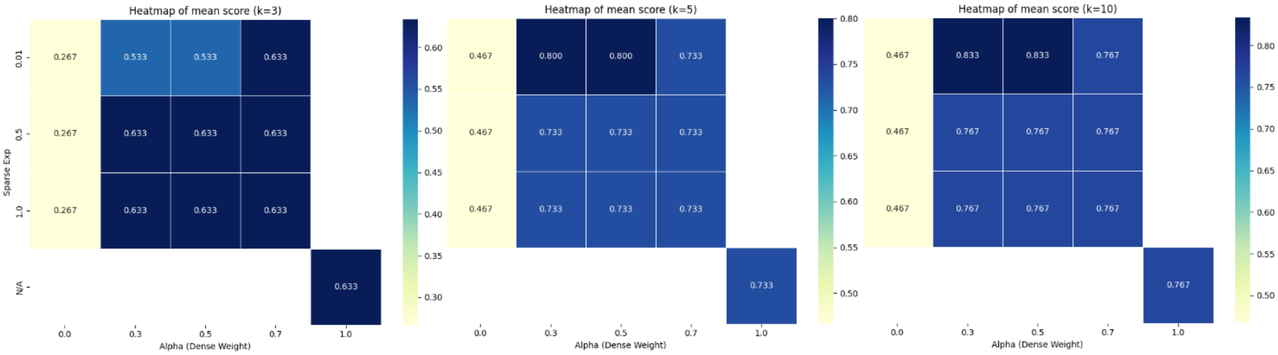


Figure 3: Recall@3 average score vs alpha and gamma

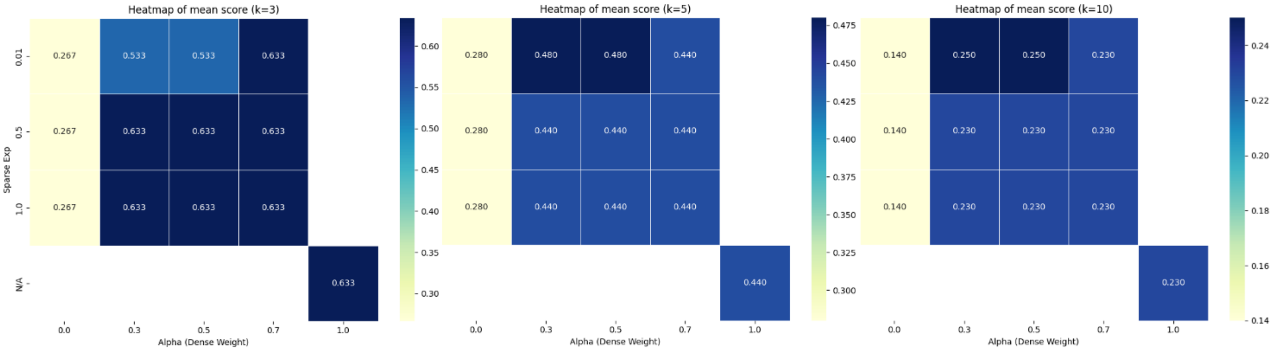
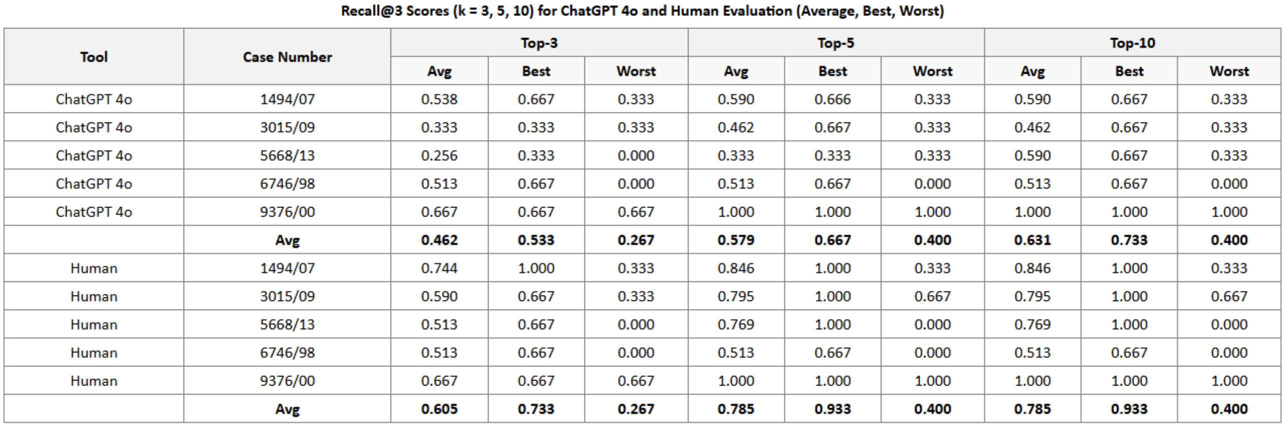


Figure 4: Precision@k average score vs alpha vs gamma

Table 1: Recall@3 (k=3, 5, 10) average, best and worst hybrid score results



Recall@3 Scores (k = 3, 5, 10) for ChatGPT 4o and Human Evaluation (Average, Best, Worst)

Tool	Case Number	Top-3			Top-5			Top-10		
		Avg	Best	Worst	Avg	Best	Worst	Avg	Best	Worst
ChatGPT 4o	1494/07	0.538	0.667	0.333	0.590	0.666	0.333	0.590	0.667	0.333
ChatGPT 4o	3015/09	0.333	0.333	0.333	0.462	0.667	0.333	0.462	0.667	0.333
ChatGPT 4o	5668/13	0.256	0.333	0.000	0.333	0.333	0.333	0.590	0.667	0.333
ChatGPT 4o	6746/98	0.513	0.667	0.000	0.513	0.667	0.000	0.513	0.667	0.000
ChatGPT 4o	9376/00	0.667	0.667	0.667	1.000	1.000	1.000	1.000	1.000	1.000
	Avg	**0.462**	**0.533**	**0.267**	**0.579**	**0.667**	**0.400**	**0.631**	**0.733**	**0.400**
Human	1494/07	0.744	1.000	0.333	0.846	1.000	0.333	0.846	1.000	0.333
Human	3015/09	0.590	0.667	0.333	0.795	1.000	0.667	0.795	1.000	0.667
Human	5668/13	0.513	0.667	0.000	0.769	1.000	0.000	0.769	1.000	0.000
Human	6746/98	0.513	0.667	0.000	0.513	0.667	0.000	0.513	0.667	0.000
Human	9376/00	0.667	0.667	0.667	1.000	1.000	1.000	1.000	1.000	1.000
	Avg	**0.605**	**0.733**	**0.267**	**0.785**	**0.933**	**0.400**	**0.785**	**0.933**	**0.400**

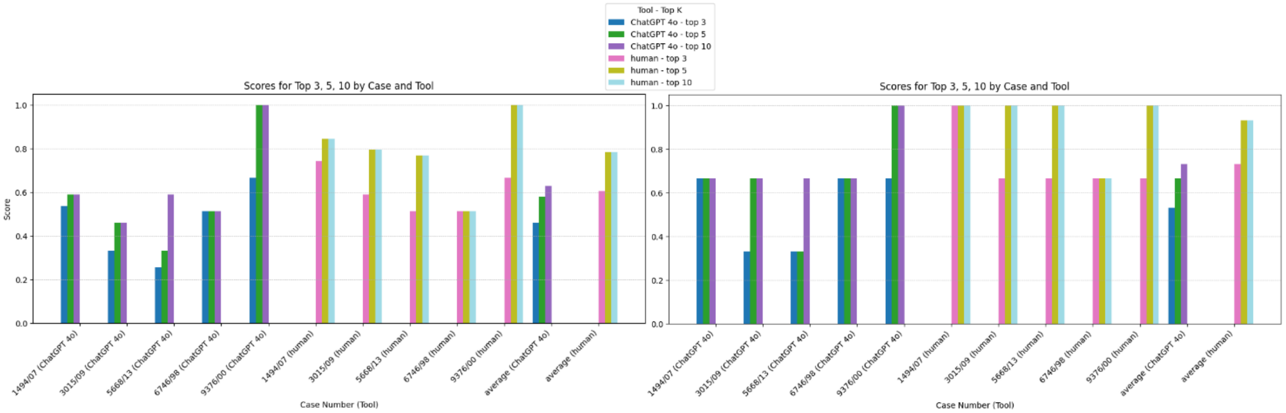
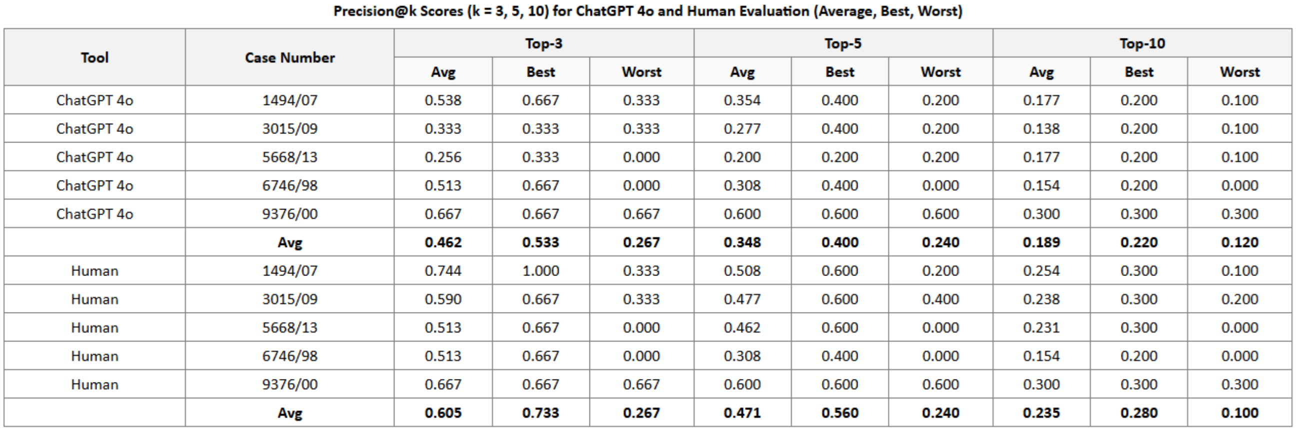


Figure 5: Recall@3 hybrid score results average (left) vs best (right)

Table 2: Precision@k (k=3,5,10) average, best and worst hybrid results



Precision@k Scores (k = 3, 5, 10) for ChatGPT 4o and Human Evaluation (Average, Best, Worst)

Tool	Case Number	Top-3			Top-5			Top-10		
		Avg	Best	Worst	Avg	Best	Worst	Avg	Best	Worst
ChatGPT 4o	1494/07	0.538	0.667	0.333	0.354	0.400	0.200	0.177	0.200	0.100
ChatGPT 4o	3015/09	0.333	0.333	0.333	0.277	0.400	0.200	0.138	0.200	0.100
ChatGPT 4o	5668/13	0.256	0.333	0.000	0.200	0.200	0.200	0.177	0.200	0.100
ChatGPT 4o	6746/98	0.513	0.667	0.000	0.308	0.400	0.000	0.154	0.200	0.000
ChatGPT 4o	9376/00	0.667	0.667	0.667	0.600	0.600	0.600	0.300	0.300	0.300
	Avg	**0.462**	**0.533**	**0.267**	**0.348**	**0.400**	**0.240**	**0.189**	**0.220**	**0.120**
Human	1494/07	0.744	1.000	0.333	0.508	0.600	0.200	0.254	0.300	0.100
Human	3015/09	0.590	0.667	0.333	0.477	0.600	0.400	0.238	0.300	0.200
Human	5668/13	0.513	0.667	0.000	0.462	0.600	0.000	0.231	0.300	0.000
Human	6746/98	0.513	0.667	0.000	0.308	0.400	0.000	0.154	0.200	0.000
Human	9376/00	0.667	0.667	0.667	0.600	0.600	0.600	0.300	0.300	0.300
	Avg	**0.605**	**0.733**	**0.267**	**0.471**	**0.560**	**0.240**	**0.235**	**0.280**	**0.100**

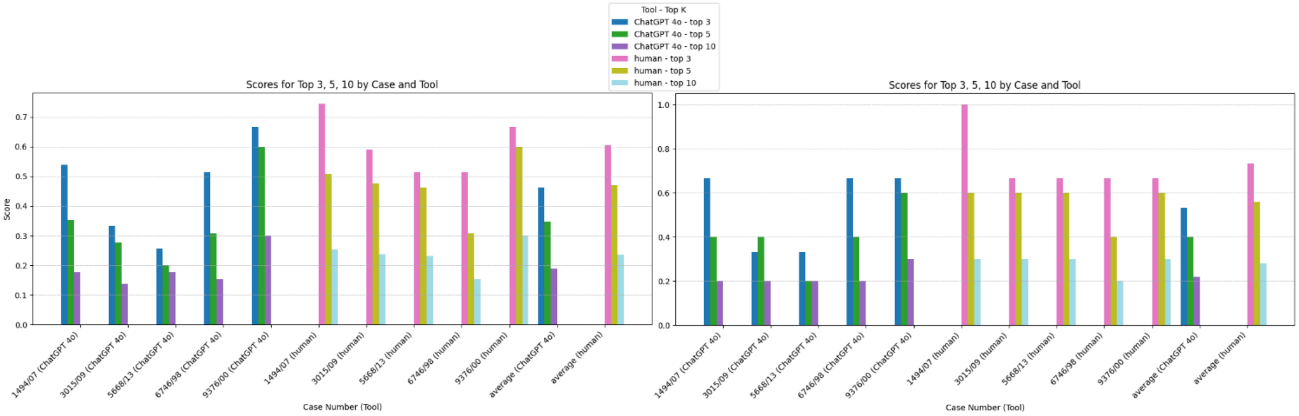


Figure 6: Precision@k hybrid score results average (left) vs best (right)

5 Future Work

- Sparse vectors redefinition – future work should separate law citations and legal case references into distinct sparse vectors. This will allow more accurate similarity scoring by distinguishing between legal norms and precedents.
- Enhanced evaluation set – evaluation was done on 100 cases due to computational limits and was annotated manually by a non-legal professional (article author). A broader, expert review is needed to better reflect real-world performance, especially for rear or hard to match legal cases.
- Adaptive hybrid scoring – some domains may benefit more from citations, others from semantics. Future models could learn optimal weighting dynamically bases on defined criteria (e.g., case domain or case type).
- Improved summarization and translation – domain tuned models for legal Hebrew to English translation and summarization could preserve critical legal nuances better.
- User feedback – expanding beyond a command line chat to a user-friendly web interface with feedback features could provide a user (legal professionals) continuous system improvement.

6 Conclusion

This study presents an approach to Legal Case Similarity Detection using a hybrid search system that combines retrieval-augmented generation (RAG) with citation-based sparse vectors and dense vectors for semantic search. By leveraging the Israeli Supreme Court Database[3], we aim to demonstrate the potential of AI and NLP techniques in enhancing the efficiency and accuracy of legal research. This work not only addresses the challenges of analyzing Hebrew-language legal texts but also contributes to the broader application of AI in the legal field, offering new possibilities for improving case retrieval systems and supporting the consistency and transparency of judicial decision-making.

References

[1] G. Goldstein, “Precedent in Law,” *Michigan Law Review*, vol. 87, no. 6, pp. 1705–1711, May 1989, doi: 10.2307/1289280.

[2] S. Lewis, “Precedent and the Rule of Law,” *Oxf. J. Leg. Stud.*, vol. 41, no. 4, pp. 873–898, Dec. 2021, doi: 10.1093/ojls/gqab007.

[3] A. S. Jonathan Schler, “The Israeli Supreme Court Database: A Comprehensive Resource for Legal Research (1997-2022),” p. 14, Nov. 2024.

[4] A. Trivedi, A. Trivedi, S. Varshney, V. Joshipura, R. Mehta, and J. Dhanani, “Similarity Analysis of Legal Documents: A Survey,” in *ICT Analysis and Applications*, S. Fong, N. Dey, and A. Joshi, Eds., Singapore: Springer, 2021, pp. 497–506. doi: 10.1007/978-981-15-8354-4_49.

[5] P. Bhattacharya, K. Ghosh, A. Pal, and S. Ghosh, “Methods for Computing Legal Document Similarity: A Comparative Study,” Apr. 26, 2020, *arXiv*: arXiv:2004.12307. doi: 10.48550/arXiv.2004.12307.

[6] D. Thenmozhi, K. Kannan, and C. Aravindan, “A Text Similarity Approach for Precedence Retrieval from Legal Documents,” presented at the FIRE, 2017. [Online]. Available: https://www.semanticscholar.org/paper/A-Text-Similarity-Approach-for-Precedence-Retrieval-Thenmozhi-Kannan/8c2e41b3d7df5977ae0559dc49f8996394e8685a

[7] S. Aryal, K. M. Ting, T. Washio, and G. Haffari, “A new simple and effective measure for bag-of-word inter-document similarity measurement,” Feb. 09, 2019, *arXiv*: arXiv:1902.03402. doi: 10.48550/arXiv.1902.03402.

[8] S. Kumar, P. K. Reddy, V. B. Reddy, and A. Singh, “Similarity analysis of legal judgments,” in *Proceedings of the Fourth Annual ACM Bangalore Conference*, in COMPUTE ’11. New York, NY, USA: Association for Computing Machinery, Mar. 2011, pp. 1–4. doi: 10.1145/1980422.1980439.

[9] A. Huang, “Similarity Measures for Text Document Clustering,” in *New Zealand Computer Science Research Student Conference (NZCSRSC)*, 2008. Accessed: Feb. 21, 2025. [Online]. Available: https://www.semanticscholar.org/paper/Similarity-Measures-for-Text-Document-Clustering-Huang/b29bf8f6900b4b0258397f73957eabd1bb977ef4

[10] A. Mandal, R. Chaki, S. Saha, K. Ghosh, A. Pal, and S. Ghosh, “Measuring Similarity among Legal Court Case Documents,” in *Proceedings of the 10th Annual ACM India Compute Conference*, in Compute ’17. New York, NY, USA: Association for Computing Machinery, Nov. 2017, pp. 1–9. doi: 10.1145/3140107.3140119.

[11] R. S. de Oliveira and E. G. S. Nascimento, “Analysing similarities between legal court documents using natural language processing approaches based on Transformers,” May 11, 2023, *arXiv*: arXiv:2204.07182. doi: 10.48550/arXiv.2204.07182.

[12] J. S. Dhani, R. Bhatt, B. Ganesan, P. Sirohi, and V. Bhatnagar, “Similar Cases Recommendation using Legal Knowledge Graphs,” Mar. 02, 2024, *arXiv*: arXiv:2107.04771. doi: 10.48550/arXiv.2107.04771.

[13] D. Cavar, J. Herring, and A. Meyer, “Law Analysis using Deep NLP and Knowledge Graphs,” 2018. Accessed: Dec. 01, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Law-Analysis-using-Deep-NLP-and-Knowledge-Graphs-Cavar-Herring/c549f824620d041dbe490869bc1ebe78df9331e3

[14] “Application of citation network analysis for improved similarity index estimation of legal case documents : A study | IEEE Conference Publication | IEEE Xplore.” Accessed: Dec. 01, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/8249996

[15] P. Bhattacharya, K. Ghosh, A. Pal, and S. Ghosh, “Legal Case Document Similarity: You Need Both Network and Text,” Sep. 26, 2022, *arXiv*: arXiv:2209.12474. doi: 10.48550/arXiv.2209.12474.

[16] S. Chavan, J. Balasubramanian, J. Puro, M. Naik, and A. V. Nimkar, “Similarity Analysis of Legal Documents using Content and Network Based Approach,” in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Jul. 2020, pp. 1–7. doi: 10.1109/ICCCNT49239.2020.9225586.

[17] A. Louis, G. van Dijck, and G. Spanakis, “Know When to Fuse: Investigating Non-English Hybrid Retrieval in the Legal Domain,” Sep. 02, 2024, *arXiv*: arXiv:2409.01357. doi: 10.48550/arXiv.2409.01357.

[18] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, “LEGAL-BERT: The Muppets straight out of Law School,” Oct. 06, 2020, *arXiv*: arXiv:2010.02559. doi: 10.48550/arXiv.2010.02559.

[19] Ilias Chalkidis, *Law2Vec: Legal Word Embeddings*. [Online]. Available: http://archive.org/details/Law2Vec

[20] W. Hu *et al.*, “BERT_LF: A Similar Case Retrieval Method Based on Legal Facts,” *Wirel. Commun. Mob. Comput.*, vol. 2022, no. 1, p. 2511147, 2022, doi: 10.1155/2022/2511147.

[21] Z. Fei *et al.*, “InternLM-Law: An Open Source Chinese Legal Large Language Model,” Jun. 21, 2024, *arXiv*: arXiv:2406.14887. doi: 10.48550/arXiv.2406.14887.

[22] A. Mahboub, M. E. Za'ter, B. Al-Rfooh, Y. Estaitia, A. Jaljuli, and A. Hakouz, "Evaluation of Semantic Search and its Role in Retrieved-Augmented-Generation (RAG) for Arabic Language," May 30, 2024, *arXiv*: arXiv:2403.18350. doi: 10.48550/arXiv.2403.18350.

[23] K. Juvekar and A. Purwar, "COS-Mix: Cosine Similarity and Distance Fusion for Improved Information Retrieval," Jun. 02, 2024, *arXiv*: arXiv:2406.00638. doi: 10.48550/arXiv.2406.00638.

[24] K. Sawarkar, A. Mangal, and S. R. Solanki, "Blended RAG: Improving RAG (Retriever-Augmented Generation) Accuracy with Semantic Search and Hybrid Query-Based Retrievers," in *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, Aug. 2024, pp. 155–161. doi: 10.1109/MIPR62202.2024.00031.