CIS5300 Milestone 3 Context-Aware Legal Information Retrieval:

Hybrid Lexical + Semantic Retrieval

Team Members:

- John Harshith Kavuturu (67055516)
- Navya Saxena (52078368)
- Aniket Ghorpade (52552619)
- Pavithra Manikandan (38819531)

Project Mentor - Zhitong Chen

The Problem

Illustrative Example =

What Problem Are We Solving?

- Ex. Query: "Does the Agreement indicate that the Receiving Party has no rights to Confidential Information?"
- Task: Find the exact passage in a 50-page legal contract that answers this question
- Challenge: Attorneys currently spend hours manually searching through contracts
- Our Goal: Automate this retrieval process using NLP

Formal Problem Definition

- Input: A natural language query Q (often a yes/no question about contract terms)
- Corpus: A collection of legal document passages D = {d_1, d_2, ..., d□}
- Output: Ranked list of top-K passages R = [d_1, d_2, ..., d□] most relevant to Q
- Objective: Maximize retrieval quality measured by Recall@K, nDCG@K, and Span F1

Mathematical Formulation:

Given query q and corpus D, find: argmax_{R⊆D, |R|=K} Score(q, R)

Why We Chose This Project

- Real-World Impact: Legal professionals spend 20-30% of their time on document review
- Practical Application: Directly applicable to contract analysis, due diligence, and legal research
- Technical Challenge: Combines information retrieval, semantic understanding, and domain adaptation
- Research Opportunity: Legal domain is understudied compared to general NLP tasks
- Scalability: Can handle large document collections (thousands of contracts)

Connection to CIS5300 Course Material

- Information Retrieval: Learned TF-IDF, BM25, and ranking algorithms
- Neural Networks: Applied transformer-based models (Sentence-BERT) for semantic embeddings
- Evaluation Metrics: Implemented IR metrics (Recall@K, nDCG@K) and QA metrics (Span F1)
- Hybrid Systems: Combined lexical (BM25) and semantic (SBERT) retrieval signals
- Ensemble Methods: Explored weighted fusion of complementary retrieval approaches
- New Learning: Gained experience with dense retrieval, embedding normalization, and score fusion strategies

What Data Do We Have?

- Benchmark: LegalBench-RAG (ContractNLI subset)
- Training Data: 100,000 question-answer pairs from legal contracts
- Test Set: 977 queries from ContractNLI
- Corpus: Legal contract documents from CUAD and ContractNLI
    - ContractNLI: 95 documents - CUAD: 462 documents - Privacy QA: 7 documents
- Total Passages: 563 passages (after 500-word chunking with 50% overlap)

Data Format:
- Queries: Natural language questions about contract terms
- Gold Answers: Exact text spans from contracts
- Documents: Full legal contract texts

Evaluating Performance

We use four complementary metrics:
1. Exact Match (EM): Does top-1 retrieved passage exactly match gold answer?
2. Span F1: Token-level overlap between predicted and gold passages
3. Recall@10: Fraction of gold answers found in top-10 results
4. nDCG@10: Ranking quality considering position of relevant items

Why Multiple Metrics?
- EM: Strict correctness - Span F1: Partial credit for overlap
- Recall@10: Coverage of relevant passages - nDCG@10: Ranking quality

Understanding Our Evaluation Metrics

Recall@10:
- Formula: |{gold_passages} ∩ {top_10_retrieved}| / |{gold_passages}|
- Interpretation: What fraction of correct answers did we find?
- Example: If 3 out of 5 gold answers are in top-10, Recall@10 = 0.6

nDCG@10:
- Formula: DCG@10 / IDCG@10
- DCG = Σ(relevance_i / log$_2$(i+1)) for positions 1-10
- Interpretation: How well are relevant items ranked?
- Higher positions get more weight (logarithmic discounting)

Span F1:
- Token-level precision and recall
- F1 = 2 × (Precision × Recall) / (Precision + Recall)
- Measures partial overlap, not just exact match

Simple Baseline Performance

TF-IDF (Simple Baseline):

- Exact Match: 0.0000
- Span F1: 0.2018
- Recall@10: 0.3090
- nDCG@10: 0.2204

Observations:

- TF-IDF serves as a lower bound
- Achieves ~31% recall, meaning it finds about 1 in 3 correct answers
- No exact matches (expected - chunks don't align with gold spans)
- Demonstrates task difficulty

Why TF-IDF?

- Classic IR baseline, no training required
- Fast and interpretable
- Sets a reasonable floor for comparison

What Have Others Tried?

1. Lexical Retrieval (BM25):
- Robertson & Zaragoza (2009): Probabilistic ranking function
- Still state-of-the-art for keyword matching
- Used in production systems (Elasticsearch, Lucene)

2. Dense Retrieval:
- Karpukhin et al. (2020): Dense Passage Retrieval (DPR)
- Uses bi-encoder architecture (query + passage encoders)
- Achieves strong results on open-domain QA

3. Hybrid Approaches:
- Xiong et al. (2020): Combined sparse and dense retrieval
- Khattab & Zaharia (2020): ColBERT - late interaction model
- Recent trend: Fusion of lexical + semantic signals

Key Finding: Hybrid methods often outperform individual approaches

Strong Baseline: BM25 Implementation

What is BM25?

- Probabilistic ranking function
- Improves upon TF-IDF with term frequency saturation and length normalization
- Formula: $BM25(Q,D) = \Sigma\ IDF(q_i) \times [f(q_i,D) \times (k_1+1)] / [f(q_i,D) + k_1 \times (1-b + b\times|D|/avgdl)]$

Our Implementation:

- Parameters: $k_1 = 1.5$, $b = 0.75$ (standard values)
- Tokenization: NLTK with stopword removal
- Performance: Recall@10 = 0.5137, nDCG@10 = 0.4445

Why BM25?

- Proven effectiveness in IR systems
- No training required
- Strong baseline for legal domain

Semantic Baseline: Sentence-BERT

What is Sentence-BERT?

- Bi-encoder architecture that maps sentences to dense vectors
- Uses siamese BERT networks (Reimers & Gurevych, 2019)
- Similarity computed via cosine similarity or dot product

Our Implementation:

- Model: `sentence-transformers/all-mpnet-base-v2`
- Embedding dimension: 768
- Normalization: L2-normalized embeddings
- Performance: Recall@10 = 0.4317, nDCG@10 = 0.3232

Why Sentence-BERT?

- Captures semantic relationships and paraphrases
- Pretrained on large-scale data (NLI, STS)
- Widely used in production RAG systems

Why Hybrid Retrieval?

The Problem:

- BM25 excels at exact term matching but misses paraphrases
- Sentence-BERT captures semantics but misses rare legal terms
- Neither alone is sufficient

The Solution:

- Combine both approaches through score fusion
- Leverage complementary strengths:
    - BM25: Exact legal terminology matching
    - Sentence-BERT: Semantic similarity and paraphrases

Hypothesis:

Hybrid approach will outperform individual methods by combining lexical and semantic signals

How Our Hybrid System Works

Step 1: Indexing

- Index each 500-word chunk with both BM25 and Sentence-BERT
- BM25: Token-based index
- Sentence-BERT: Dense embedding vectors (768-dim)

Step 2: Query Processing

- Compute BM25 scores for all passages
- Compute Sentence-BERT cosine similarities
- Normalize both score distributions to [0,1]

Step 3: Fusion

- Weighted combination: $S\_hybrid = 0.55 \times S\_BM25 + 0.45 \times S\_SBERT$
- Weights tuned on validation set
- Consider top-100 from each retriever (fusion depth)

Step 4: Ranking

- Sort by fused scores
- Return top-K passages

Implementation Specifications

Fusion Strategy:

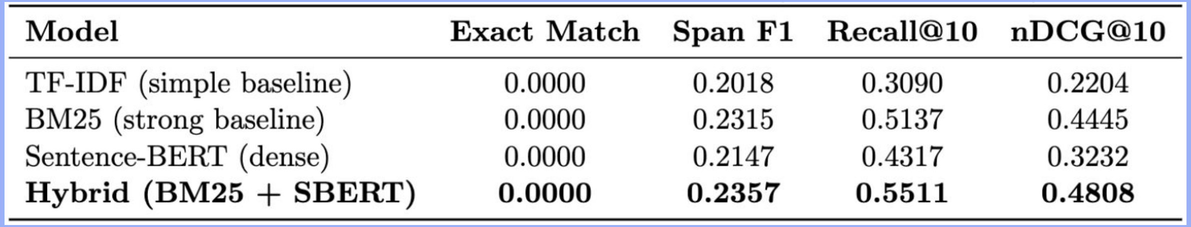
- Method: Weighted combination of normalized scores
- Weights: BM25 = 0.55, Sentence-BERT = 0.45
- Normalization: Min-max scaling to [0,1]

Parameters:

- Fusion depth: Top-100 candidates per retriever
- BM25: $k_1 = 1.5$, $b = 0.75$
- Sentence-BERT: all-mpnet-base-v2, batch_size = 32
- Chunk size: 500 words with 50% overlap

Training-Free:

- No fine-tuning required
- Uses pretrained Sentence-BERT
- Efficient and scalable



Model	Exact Match	Span F1	Recall@10	nDCG@10
TF-IDF (simple baseline)	0.0000	0.2018	0.3090	0.2204
BM25 (strong baseline)	0.0000	0.2315	0.5137	0.4445
Sentence-BERT (dense)	0.0000	0.2147	0.4317	0.3232
Hybrid (BM25 + SBERT)	**0.0000**	**0.2357**	**0.5511**	**0.4808**

What Do These Results Mean?

Hybrid Advantages:

- 7.3% relative improvement in Recall@10 over BM25 (0.5511 vs 0.5137)
- 27.8% relative improvement over Sentence-BERT alone (0.5511 vs 0.4317)
- Better ranking quality (nDCG@10 = 0.4808)

Why It Works:

- BM25 catches exact legal terms (e.g., "Confidential Information", "Receiving Party")
- Sentence-BERT captures paraphrases (e.g., "no rights" vs "lacks authority")
- Fusion combines both signals effectively

Limitations:

- Exact Match = 0.0 (gold spans don't align with 500-word chunks)
- Future work: Passage trimming, cross-encoder re-ranking

What We Learned

Technical Insights:

- Combining complementary retrieval signals improves performance
- Lexical matching remains crucial for domain-specific terms
- Semantic embeddings help with paraphrases and conceptual similarity
- Weighted fusion is simple but effective

Course Connections:

- Information retrieval fundamentals (TF-IDF, BM25)
- Neural embeddings and transformers (Sentence-BERT)
- Evaluation metrics (Recall@K, nDCG@K)
- Ensemble methods and hybrid systems

Practical Impact:

- Demonstrates feasibility of automated legal document retrieval
- Provides foundation for production legal tech systems

Next Steps (Milestone 4)

Planned Extensions:

1. Cross-encoder re-ranking: Two-stage pipeline (retrieve → re-rank)
2. Domain fine-tuning: Legal-domain Sentence-BERT adaptation
3. Scaling: FAISS for ANN indexing on larger corpora
4. Advanced fusion: Learned fusion weights, query-dependent fusion

Expected Improvements:

- Address Exact Match limitation
- Scale to a larger CUAD corpus
- Further improve Recall@10 and nDCG@10

Where to Find Our Work

Repository: https://github.com/JohnHarshith/CIS5300-F2025-Project

THANK YOU!