

Multi-Hop Fact Verification System

Ashwin Satish

Daniel Xiong

Roshan Chauhan

[Presentation Video](#)

The Misinformation Problem

- Misinformation spreads faster than the truth
 - (Best example: rumors about an individual or celebrity)
- Traditional fact-checkers → usually do single source verification
- Real world claims → need multiple sources / pieces of evidence
- Example
 - Lewis Hamilton is retiring from F1
 - No basis for claim, based on rumors from press and media
- Multi-hop reasoning needed to separate facts from hype

Our Goal

- End to end pipeline to verify claims
 - Evidence chains
- Input: Claim (text statement)
- Output:
 - SUPPORTED
 - REFUTED
- Key challenge: Connecting evidence across multiple Wikipedia documents
- Example output:
 - claim
 - true_label
 - predicted_label
 - confidence
 - retrieved_docs

Dataset: HoVeR

- 18,171 training claims
- 4,000 dev claims (2,000 SUPPORTED, 2,000 NOT_SUPPORTED)
- 4,000 test claims (No true labels)
- Evidence source: 5.5 Million preprocessed Wikipedia articles
- Why HoVer?
 - Designed specifically for multi-hop reasoning (Multiple evidence pieces)

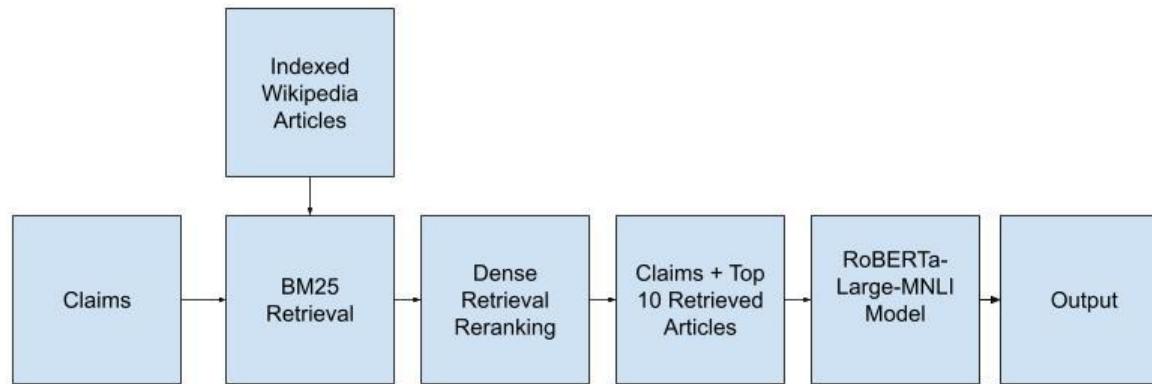
BM25

- Keyword matching algorithm
- Uses TF-IDF scoring with length normalization
- Fast retrieval
- Finds exact word matches
- Misses paraphrases/synonyms

Wikipedia Data Indexing using BM25

- Built search infrastructure for entire Wikipedia corpus
- Worked with HotpotQA Wikipedia dump (5.4M articles, ~30GB)
- Pre-processes and organizes articles for fast retrieval using elasticsearch
- Creates inverted index mapping words to documents
- Final indexed database: ~8GB, achieving 58% recall@100
- Enables millisecond-speed searches across millions of articles

Pipeline



Dense Retrieval Reranking

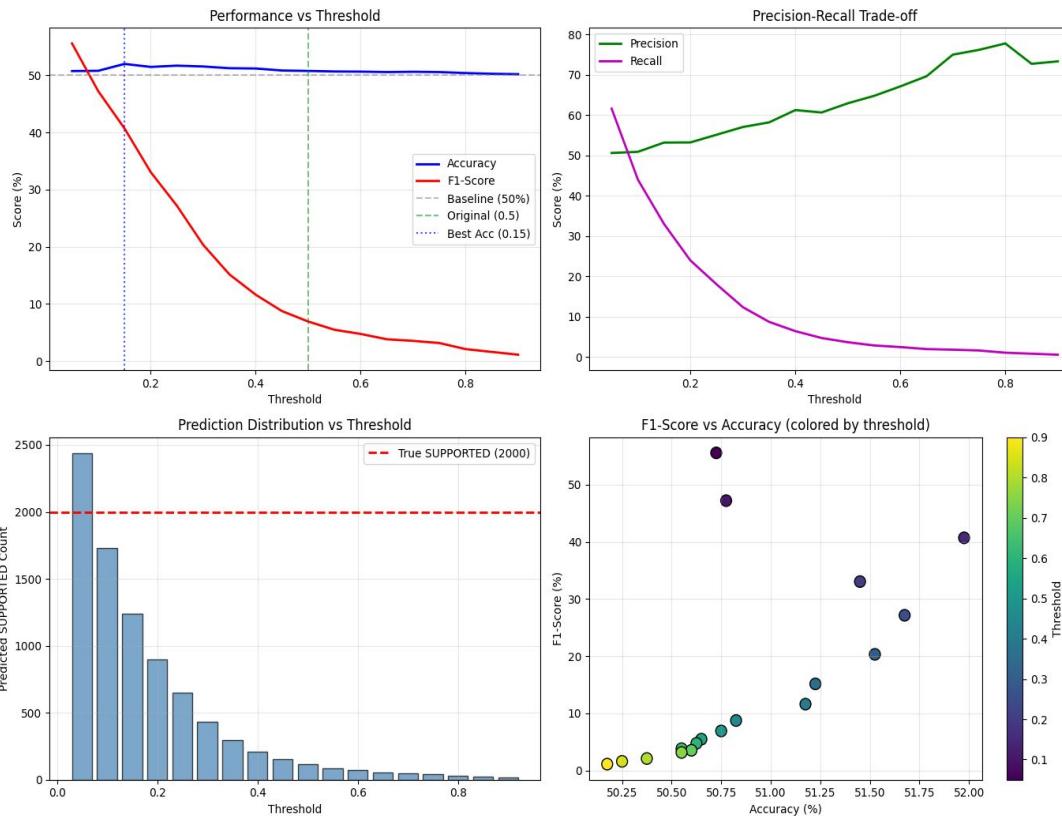
- Neural embedding-based
 - Encodes text into vectors
- Semantic similarity matching
 - Understands meaning beyond keywords
- Pre-trained transformer model
- Slower but more accurate
- Improves top-10 precision: Recall@10 improved from 37.6% to 46.9% for training data

RoBERTa-large-MNLI Model

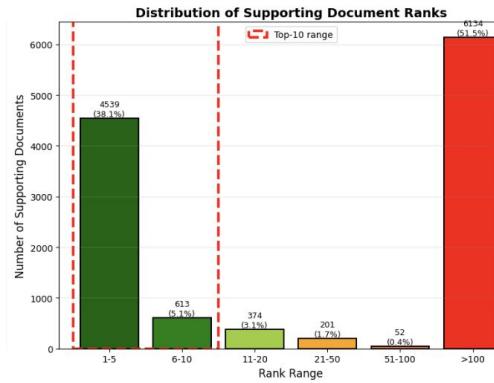
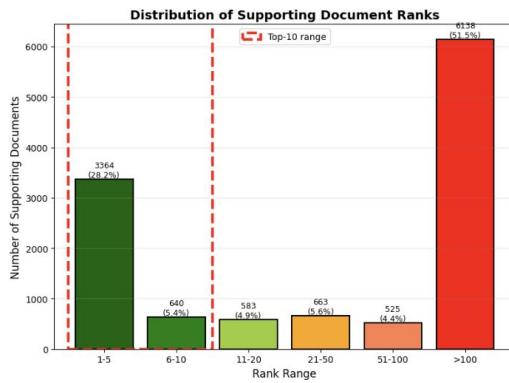
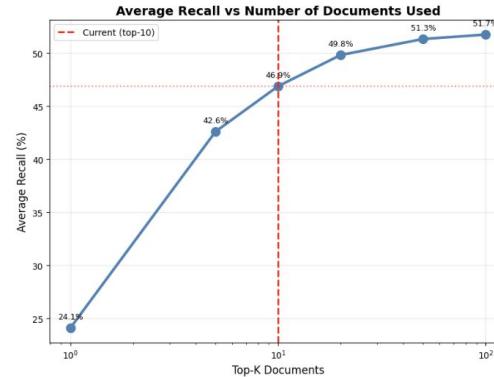
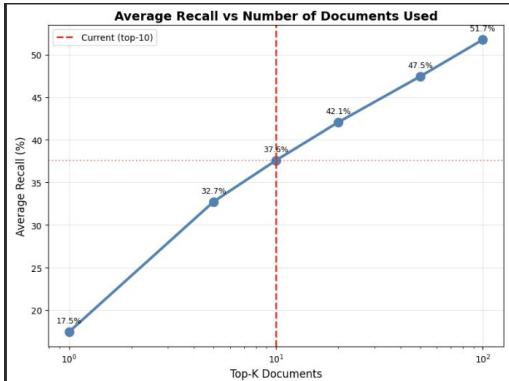
- Pre-trained version of RoBERTa model
- Useful for NLI (Natural Language Inference)
 - Determining relationship between two texts
- Robust and able to be used on text it wasn't pre-trained on, like our dataset
- Returns a confidence score rather than a binary output
- Takes top 10 articles per claim and then aggregates the top 5 scoring sentences from them to verify each claim

Evaluation Results | BM-25 → Evaluation

- **Retrieval Metrics:**
 - Recall@10: 33.76%
 - Recall@100: 51.7%
 - Gap: 18% of relevant docs ranked 11-100
- **Verification Results:**
 - Best Accuracy: ~51-52%
 - Best F1-Score: ~55%
 - Threshold: 0.05 optimal
- **Key Insight:** Precision-recall tradeoff at higher confidence thresholds limits performance
- **The Problem:** Poor retrieval = Poor verification



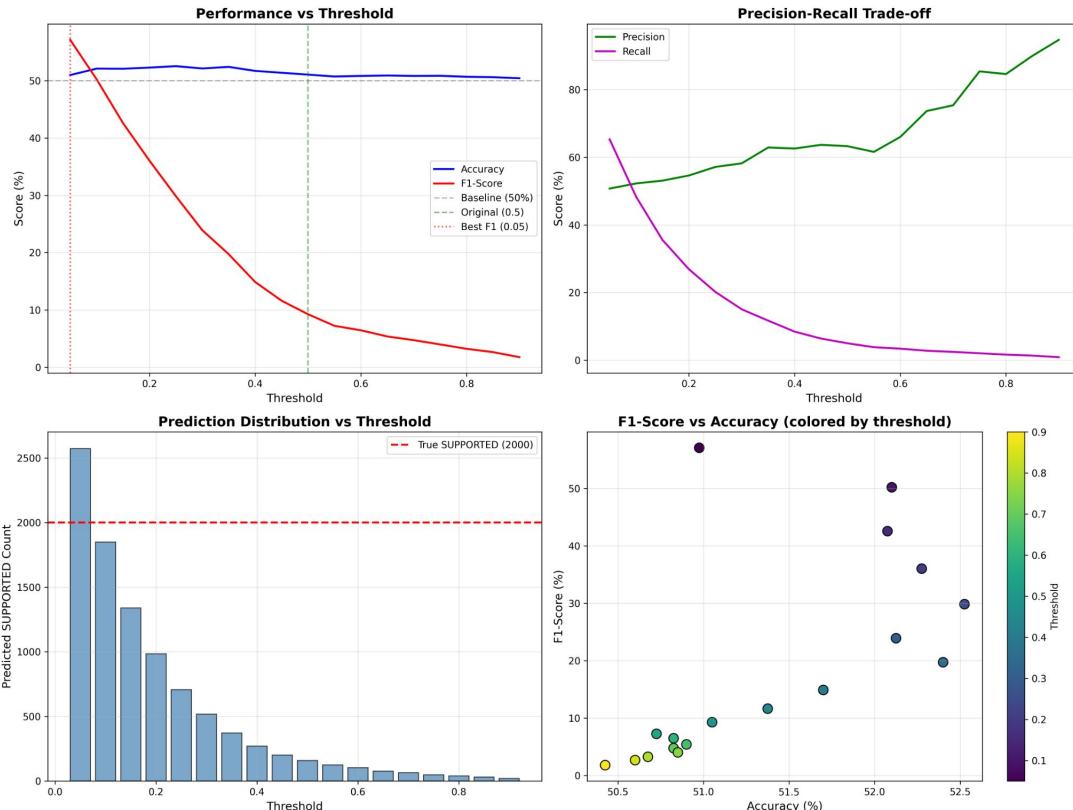
Dense Retrieval Results



Evaluation Results | BM25 → Dense Retrieval → Evaluation

- **Retrieval Metrics:**
 - Recall@10: 43.24% (+9.48pp from BM25)
 - Coverage@10: 83.97%
- **Verification Results:**
 - Best Accuracy: ~52->52.5%
 - Best F1-Score: ~55->57%
 - Threshold: 0.05 optimal
- **Improvements:**
 - Better retrieval → Better evidence for verification
 - Smoother precision-recall tradeoff
 - More balanced prediction distribution

Key Insight: Semantic understanding pushes relevant docs into top-10, improving downstream verification



Implementation Challenges

- 30GB Wikipedia dataset
 - Solution: Streaming processing with `json`
- Low Retrieval Coverage and Recall from BM25
 - Only 20% of documents used to support gold labels in Dev set found by our BM25 Retrieval
- HoVER paper uses multi-step reasoning: retrieve → read → retrieve again to find connected evidence chains
 - Hardware limitations prevented iterative retrieval
- Baseline BM25 achieves 52.6% accuracy; iterative dense methods reach ~75% but require significant GPU resources
 - Stuck with single-stage retrieval

Future Improvements

- Fine tune model on training data
- Implement Multi-class Classification by adding the class:NOT_ENOUGH_INFO
 - Currently 2 classes : SUPPORTED, REFUTED
- Implement iterative Dense Retrieval Neural Network instead of BM25