
Stage 7 — Semantic Search: Conceptual Overview

1️⃣ What Stage 6 (Current Engine) Does

- Your **Stage 6 engine** is a **text-based search engine**:
    - Converts words → **term IDs** via the **Lexicon**.
    - Finds matching documents using the **inverted index**.
    - Scores and ranks documents using **BM25** (TF-IDF-based ranking).
    - Retrieves snippets.
- Limitation: **only exact word matches matter**.
    - Example: Searching `"automobile"` won't find `"car"` because they are different words.

2️⃣ What Stage 7 Adds

- **Semantic understanding**: engine now understands **word meaning**, not just exact words.
- Uses **pre-trained embeddings** (GloVe/Word2Vec) to map words into **vectors in a 50–300 dimensional space**.
    - Words with similar meanings → vectors close to each other.
    - Example: `"car"` and `"automobile"` now have **similar vectors**.

3️⃣ How It Works (Step-by-Step)

Step 1 — Embedding Words

- Every word in your lexicon gets a **vector representation**.
- Think of it as each word being a **point in a multi-dimensional space**.

"car" → [0.12, -0.03, ..., 0.56]
"automobile" → [0.11, -0.02, ..., 0.55]

Step 2 — Represent Documents

- Each document is represented as a **weighted combination of its words** (TF-IDF weights).
- Result: **document vector** that summarizes its semantic meaning.

Doc 1: "fast car rental" → combined vector
Doc 2: "rent an automobile quickly" → combined vector

Step 3 — Represent Queries

- Query is also converted into a **vector** using the same embedding + TF-IDF weighting.

Query: "automobile hire" → query vector

Step 4 — Compare Query and Documents

- Compute **cosine similarity** between query vector and **top BM25 document vectors**.
- Cosine similarity tells us **how close the meaning of the document is to the query**, even if exact words don't match.

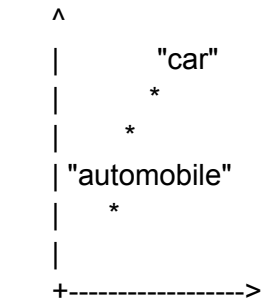
cosine_similarity(query_vector, doc_vector)

Step 5 — Re-rank Results

- Use **semantic similarity** to reorder the documents.
- Now your results include documents that **match meaning, not just words**.

4️⃣ Visual Intuition

Word Vectors Space:



Doc 1: "rent car" → vector near "car"
Doc 2: "rent automobile" → vector near "automobile" (~close to Doc1)
Query: "hire automobile" → vector close to both

- Even if Stage 6 wouldn't match `"hire automobile"` to `"rent car"`, Stage 7 **semantic search fixes it**.

5️⃣ How Stage 7 Enhances Stage 6

Feature	Stage 6 (BM25)	Stage 7 (Semantic)	Benefit
Match Type	Exact words only	Meaning of words	Finds synonyms & related terms
Query Understanding	Bag-of-words	TF-IDF weighted vector	Captures context
Ranking	BM25	BM25 + Semantic	More relevant & intuitive results
Snippets	Exact words	Semantic info	Better snippet relevance
Example	"car" finds docs with "car"	"car" also finds docs with "automobile"	Users find what they **mean**, not just what they type

---

✅ **Key Takeaway**: Stage 7 **makes your search smarter**, bridging the gap between **literal word matching** and **real understanding** of document meaning — all without using ML models or transformers, just embeddings.
