

| Document | Count (Number of Terms Containing It) |
|----------|---------------------------------------|
| 1 | 2 |
| 2 | 1 |
| 3 | 3 |
| 4 | 2 |
| 5 | 1 |
| 8 | 4 |
| 16 | 5 |
| 17 | 1 |
| 18 | 1 |
| 32 | 3 |
| 33 | 1 |
| 40 | 1 |
| 44 | 1 |
| 64 | 3 |
| 120 | 2 |
| 128 | 1 |

3. Filter Documents with At Least 3 Query Terms:

- Retain only documents that appear in **3 or more** of the query terms' postings lists.

| Document | Count |
|----------|-------|
| 3 | 3 |
| 8 | 4 |
| 16 | 5 |
| 32 | 3 |

| Document | Count |
|----------|-------|
| 64 | 3 |

Relevant Documents: [3, 8, 16, 32, 64]

B

Champion Lists: Concept and Extension

What Are Champion Lists?

Champion Lists are an optimization technique used in Information Retrieval to efficiently retrieve the most relevant documents for a query. Instead of scanning all the documents in the postings lists for each term, a precomputed subset of the most important documents for each term is maintained. These subsets are called "Champion Lists."

Champion Lists leverage term weighting measures like **TF-IDF** to rank documents for each term during preprocessing. At query time, only the documents in the Champion Lists are considered for ranking, drastically reducing computation time.

Steps to Build Champion Lists

- 1. Compute Weights:**
 - Assign a weight (e.g., TF-IDF) to each document for a term based on its importance in the document.
- 2. Rank Documents:**
 - For each term, rank all documents based on their weights.
- 3. Retain Top-k Documents:**
 - Retain the top kk documents for each term. These top-ranked documents form the Champion List for that term.

How Champion Lists Work

- At query time, retrieve the Champion Lists for all query terms.
 - Use these reduced lists to compute intersections or rank documents, rather than scanning the full postings lists.
-

Advantages of Champion Lists

1. Efficiency:

- The size of postings lists to scan is reduced significantly, making query processing faster.

2. Focus on Relevant Documents:

- Only the most relevant documents for each term are considered, leading to better precision.

Extension to Avoid Small Result Sets

While Champion Lists optimize efficiency, they can result in small result sets if:

1. kk (the number of documents in the Champion List) is too small.
2. Few query terms are matched across documents.

Solution: Combined Champion Lists or Dynamic Expansion

1. Expand Champion Lists Dynamically:

- Use a primary kk -size Champion List but include additional documents ranked just below the top kk threshold. This ensures the result set isn't too small while still focusing on relevant documents.

2. Include Global Best Documents:

- Maintain a **secondary Champion List** of globally important documents (e.g., based on overall TF-IDF scores) for each term. These documents are added to the result set if the primary Champion Lists yield too few results.

3. Query-Specific Adjustments:

- Dynamically adjust kk based on the query. For instance, increase kk when queries have fewer terms or when the query terms are rare.

4. Intersection and Union Strategies:

- Use **intersection** to focus on common documents across query terms but add documents from **unions** of individual Champion Lists when intersections are too small.

Example of Extension

- For a query like "Sunny Berlin girl," suppose Champion Lists for the terms are:

- **Sunny:** Top 5 documents → [4, 8, 16, 32, 64]
- **Berlin:** Top 5 documents → [16, 18, 17, 32, 64]
- **Girl:** Top 5 documents → [1, 3, 4, 8, 16]

Issue: Intersection gives only one document ([16]).

Solution:

1. Dynamically expand kk to 10 for all terms or fetch documents ranked 6–10.
2. Include additional documents from the union of Champion Lists: [1, 3, 4, 8, 16, 17, 18, 32, 64].

Advantages of the Extension

- Ensures sufficient documents are retrieved, even for small queries or sparse postings.
- Retains the efficiency of Champion Lists by keeping the primary result set focused on relevant documents.