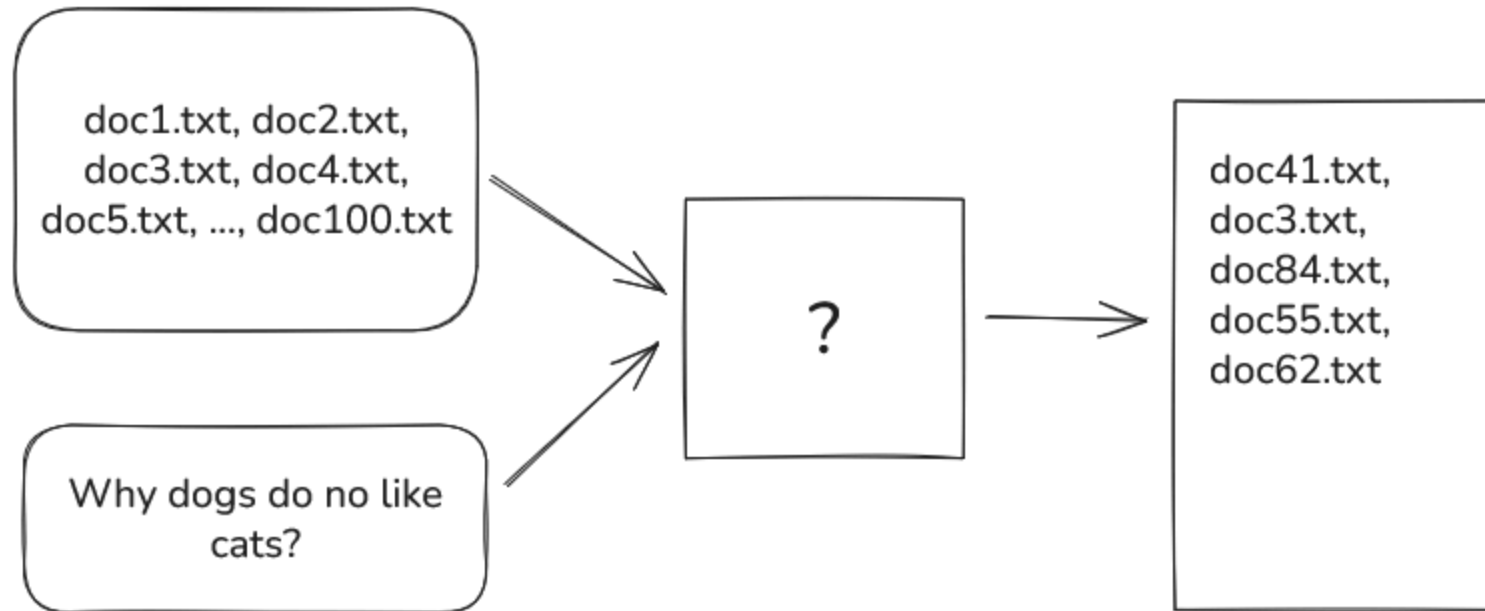
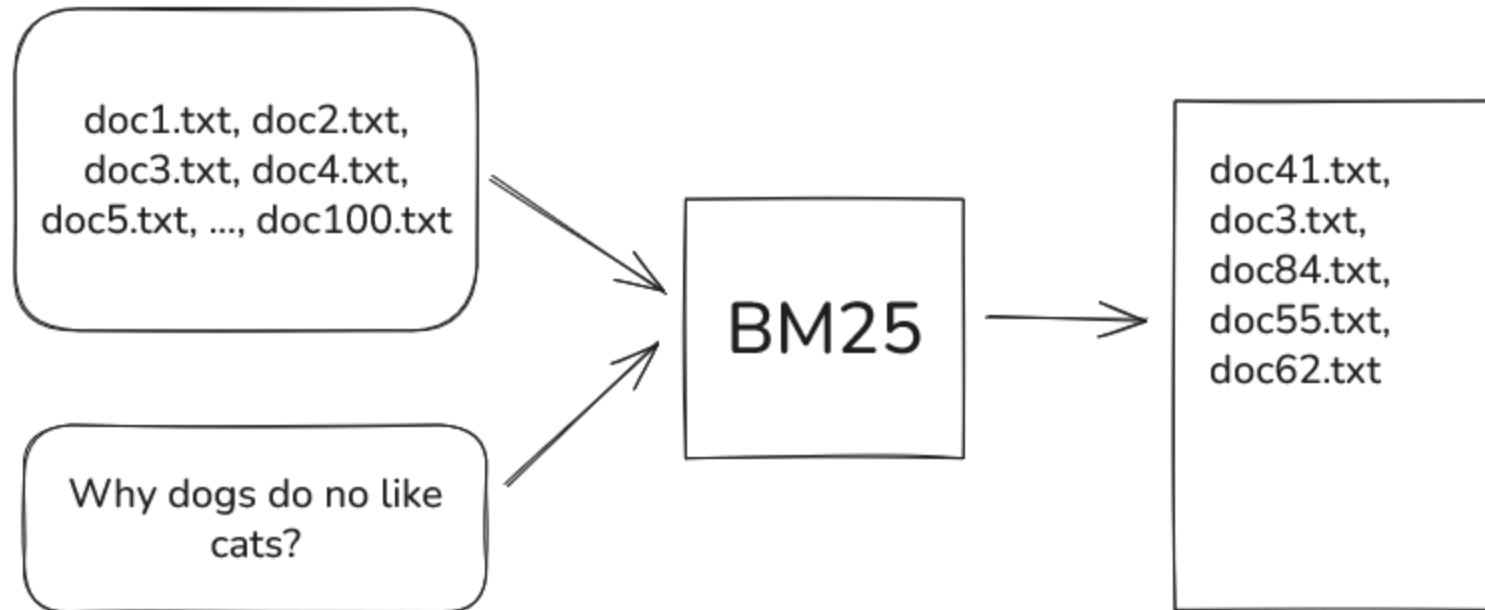


Some entry title here with logo

What problem we are trying to solve ?



Our solution to the problem



How to use ?

Step1 : Build index

```
$ java -jar target/bm25.jar build \  
    -I=index.txt src/main/resources/documents
```

Step2 : Search using it

```
$ java -jar target/bm25.jar search \  
    index.txt does the bird purr like a cat?
```

How it works ? (1)

1. Read the content of files

```
"a cat is a feline and likes to eat bird",           // file1.txt  
"a dog is the human's best friend and likes to play", // file2.txt  
"a bird is a beautiful animal that can fly",         // file3.txt
```

How it works ? (2)

2. split them
3. avoid meaningless words (is/a/to/etc)
4. stem them (connections, connected, connecting -> connect)

```
[  
  ["cat", "felin", "like", "eat", "bird"],           // file1.txt  
  ["dog", "human", "best", "friend", "like", "plai"], // file2.txt  
  ["bird", "beauti", "anim", "can", "fly"]           // file3.txt  
]
```

How it works ? (3)

5. build vocabulary

```
corpus = [  
    ["cat", "felin", "like", "eat", "bird"],  
    ["dog", "human", "best", "friend", "like", "plai"],  
    ["bird", "beauti", "anim", "can", "fly"]  
]  
  
vocabulary = [  
    "like", "best", "plai", "can", "fly", "beauti",  
    "cat", "bird", "friend", "eat", "anim", "dog", "human", "felin"  
]
```

How it works ? (4)

6. For every token in every document compute BM25 scores

$$\log\left(\frac{N - df_t + 0.5}{df_t + 0.5} + 1\right) \cdot \frac{tf_{td}}{k_1} \cdot (1 - b + b \cdot \left(\frac{L_d}{L_{avg}}\right)) + tf_{td}$$

How it works ? (5)

6. Build document-term matrix with resulting BM25 scores

docIdx	like	best	plai	can	fly	beauti	cat	bird	friend	eat	anim	c
0	0.22	0	0	0	0	0	0.48	0.23	0	0.48	0	C
1	0.19	0.4	0.4	0	0	0	0	0	0.4	0	0	C
2	0	0	0	0.48	0.48	0.48	0	0.23	0	0	0.48	C

Thank you for attention