Hellman Oded 205434590
Stav Cohen 316492776

Dry exercise

Question 1:

Introduction:

Assume the following docs:

Doc 1: I am a scientist, and I am currently enrolled to IR course. I am a student at Reichman University
Doc 2: I was a student and scientist.

We want to find sentences which contains the words “I” and “scientist” in the same sentence and no more than two words apart

Solution:

We’ll setup the following positional index structure for each word:

```
[
doc1:
    [
        sentence1: [position1, position2 …];
        sentence2: [position1, position2…];
        …
    ];
doc2:
    [
        sentence1: [position1, position2 …];
        sentence2: [position1, position2…];
        ...
    ];
]
```

For the word “I”, the following will be the index structure:

```
[
doc1:
    [
        sentence1: [1, 6];
```

sentence2: [1];
];

doc2:
[
sentence1: [1];
];
]

For the word “scientist”, the following will be the index structure:

[
doc1:
[
sentence1: [4];
sentence2: [];
];
doc2:
[
sentence1: [6];
];
]

Explanation:

Now we’ll return the documents that matches the following criterias:

1. In at least one of the sentences, both “I” and “scientist” have a position (not an empty list)
2. In the same sentence, if the word “I” has **m** occurrences, and the word “scientist” has **n** occurrences. Then we have at least one position **i** (for each position i in m) and one position **j** (for each position j in n) such that: **0 < j - i < =3**

Therefore, **doc1** is returned, since in sentence1, “I” has a position 1, and “scientist” has a position 4, and 0 < 4 - 1 <= 3

Hellman Oded 205434590
Stav Cohen 316492776

Question 2:

To build a search engine for malware, we can use an index concept that supports multi-field searches tailored to the characteristics of malware. Each malware object will be indexed by its **type**, **author**, and **description**. The index would consist of separate fields, each dedicated to a specific attribute of the malware.

For the **type** field, the index would store terms such as "virus," "ransomware, "worm," etc.. These categories allow the search engine to search for malware based on its type. For example, a query like "Retrieve all ransomware" would rely on this field to find relevant matches.

The **author** field is designed to store information about the group's creator. This enables users to search for malware created by specific authors, such as "Oded Hellman" or "Stav Cohen." For instance, a query might aim to find all malware linked to a known attacking group or hacker.

The **description** field is the most versatile and contains textual information about the malware (similar to a regular doc posting list), probably including its behavior, attack vectors, or intended targets. This field supports full-text search capabilities, allowing users to find malware based on keywords or phrases. For example, the description might include "Linux server" or "targeting education institutions.".

Additionally, metadata such as discovery date, threat severity, or targeted systems could be indexed to provide advanced filtering options. Combining these fields ensures the index can handle a wide range of queries effectively.

Hellman Oded 205434590
Stav Cohen 316492776

Example:

Field	Term	Posting List (Document IDs)
Type	ransomware	[1, 5, 8]
	virus	[2, 3, 7]
	worm	[4, 6]
Author	Oded H.	[1, 3, 6]
	Stav C.	[2, 4, 5]
Description	Windows	[1, 2, 3, 5]
	Linux	[1, 2, 4]
	server	[2, 4]

Example Query:

- `"virus by Oded H. targeting Linux servers"`

Execution Steps:

- Retrieve matching documents from the **Type** field for "virus" → [2, 3, 7].
- Retrieve documents from the **Author** field for "Oded H." → [1, 3, 6].
- Retrieve documents from the **Description** field for "Linux" & "servers" → [1, 2, 4].
- Combine results using boolean operators (e.g., `AND`, `OR`, `NOT`) to refine the final result.