ETH zürich

Information Retrieval Spring 2024

Exercise Session Week 3

Last week questions

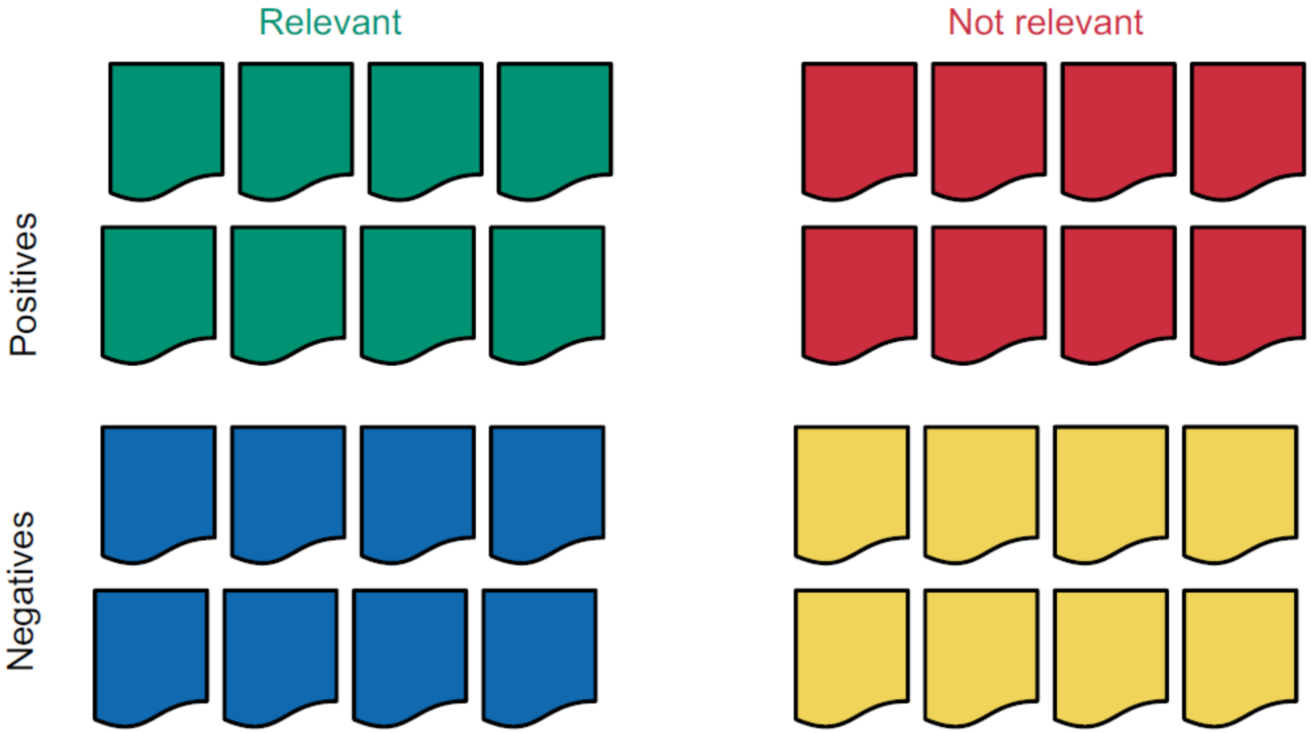
- Files should still be saved on the server after Jupyter backend shuts down after 120 minutes
- Exam seems to not contain coding element (but can change – confirm with Ghislain towards end of semester)
- Ungraded quizzes have unlimited attempts.
- 3 graded quizzes likely have only one attempt with some time limit (TBD). But reviews should be available after closing.

Lecture last week: Boolean queries

- Grepping
  - Shortcomings?
- Inclusion, occurrence, order
- Set vs. Bag vs. List
- Incidence matrix
  - Boolean queries
  - Shortcomings?

Lecture last week: Boolean queries

- Precision/Recall

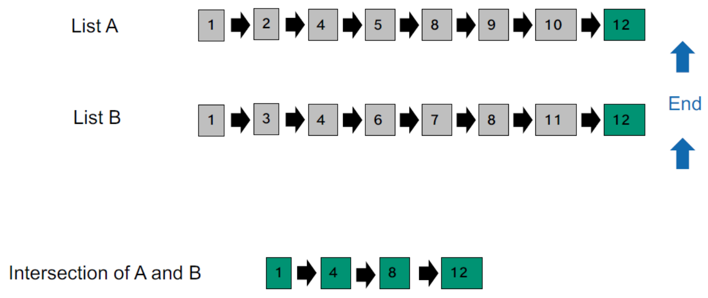


Lecture last week: Boolean queries

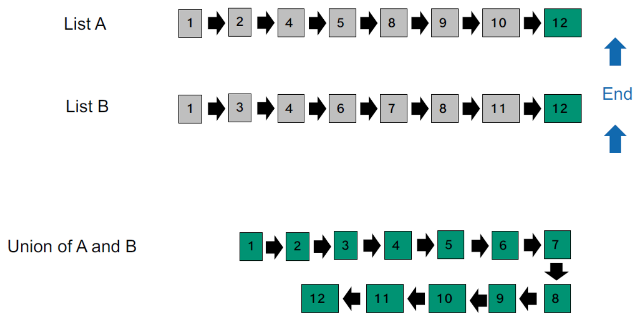
- Inverted index
  - Memorize terminology
- Boolean queries (AND, OR, NOT)
- Intersection & Union algorithms
- Optimizing

Terms		
	t	1, 4, 9, 10
	u	5, 6, 7
	v	2, 4, 6, 8, 10
	w	5
	x	1, 3, 4, 7
	y	5, 8, 10

Intersection algorithm



Union algorithm



Exercise 1: Boolean Retrieval

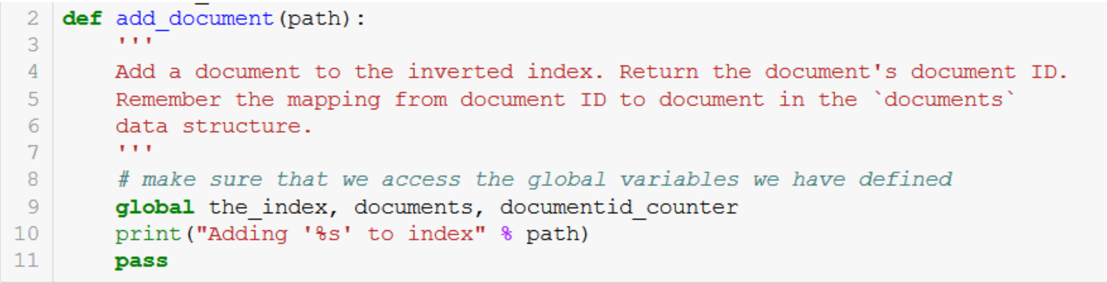
- Theoretical questions on Moodle

Exercise 1: Boolean Retrieval

- Practical part: Jupyter notebooks
  - Implement simple inverted index for Boolean Retrieval
- Already implemented for you:
  - Boolean query parser (abstract syntax tree)
  - Handles tokenizing of documents and simple preprocessing of them

Exercise 1: Boolean Retrieval

- Your part:
  - Add a document and its terms to the inverted index



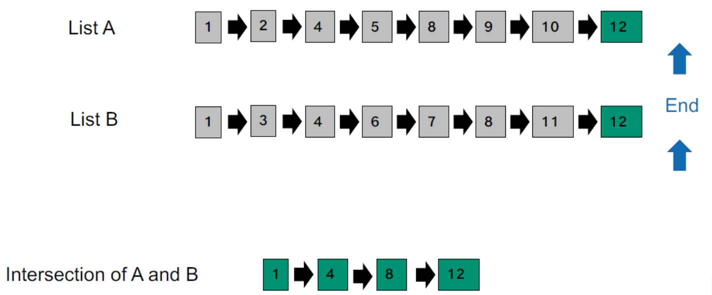
```python
def add_document(path):
    '''
    Add a document to the inverted index. Return the document's document ID.
    Remember the mapping from document ID to document in the `documents`
    data structure.
    '''
    # make sure that we access the global variables we have defined
    global the_index, documents, documentid_counter
    print("Adding '%s' to index" % path)
    pass
```

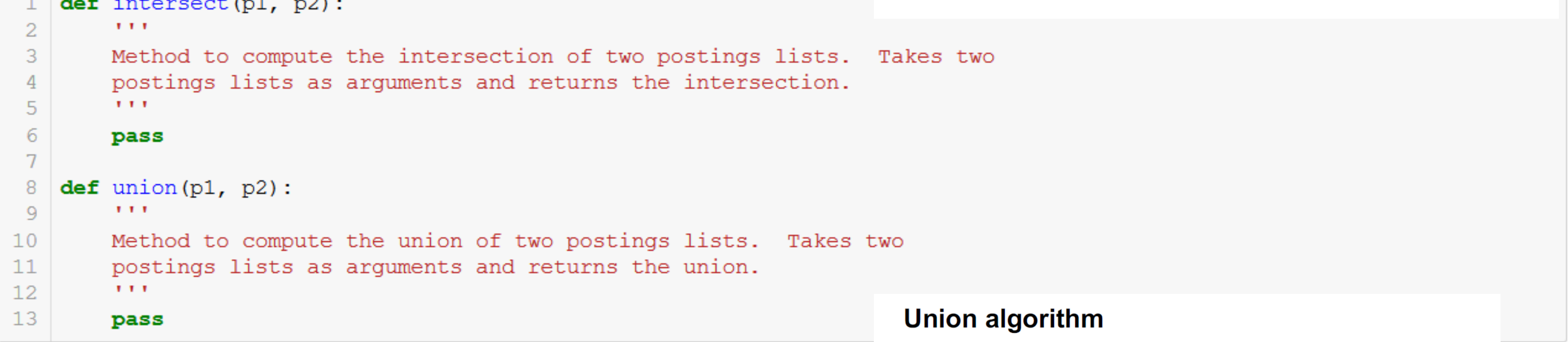
Terms	
t	1, 4, 9, 10
u	5, 6, 7
v	2, 4, 6, 8, 10
w	5
x	1, 3, 4, 7
y	5, 8, 10

Exercise 1: Boolean Retrieval

- Your part:
  - Implement the intersection and union algorithm

Intersection algorithm



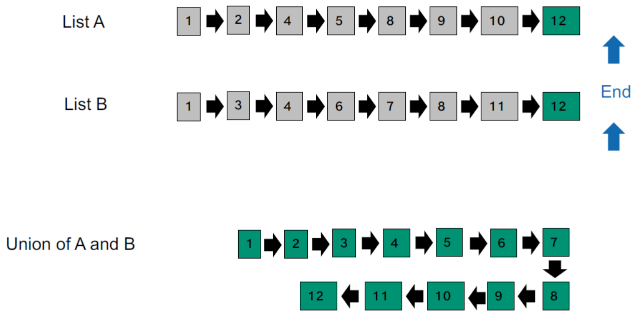


```python
def intersect(p1, p2):
    '''
    Method to compute the intersection of two postings lists.  Takes two
    postings lists as arguments and returns the intersection.
    '''
    pass

def union(p1, p2):
    '''
    Method to compute the union of two postings lists.  Takes two
    postings lists as arguments and returns the union.
    '''
    pass
```

  - Test it using provided examples

Union algorithm


List A
1
2
4
5
8
9
10
12
End
List B
1
3
4
6
7
8
11
12
Union of A and B
1
2
3
4
5
6
7
8
9
10
11
12