# Assignment 04

**Digital Libraries and Foundations of Information Retrieval**

Winter semester 2022

1542011 Franka Brunen, 1365848 Andreas Schneider

**Task 1:** Boolean Retrieval 3+3+4+3+2 Points

(a)

| | $D_1$ | $D_2$ | $D_3$ | $D_4$ |
|---|---|---|---|---|
| christmas | 1 | 1 | 1 | 1 |
| snow | 1 | 0 | 1 | 0 |
| skating | 1 | 0 | 0 | 0 |
| wine | 1 | 1 | 0 | 0 |
| skiing | 0 | 1 | 0 | 0 |
| ropeway | 0 | 1 | 0 | 0 |
| punch | 0 | 1 | 0 | 0 |
| cross-country | 0 | 0 | 1 | 0 |
| ice | 0 | 0 | 1 | 1 |
| strudel | 0 | 0 | 0 | 1 |
| cinnamon | 0 | 0 | 0 | 1 |

(b)

| | | | |
|---|---|---|---|
| **christmas** | **4** | → | **1** **2** 3 4 |
| **snow** | **2** | → | **1** **3** |
| **skating** | **1** | → | **1** |
| **wine** | **2** | → | **1** **2** |
| **skiing** | **1** | → | **2** |
| **ropeway** | **1** | → | **2** |
| **punch** | **1** | → | **2** |
| **cross-country** | **1** | → | **3** |
| **ice** | **2** | → | **3** **4** |
| **strudel** | **1** | → | **4** |
| **cinnamon** | **1** | → | **4** |

(c)

1 evaluate *(ice **OR** punch)*

i. t1 = „ice“
ii. p1 = {3, 4}
iii. t2 = „punch“
iv. p2 = {2}
v. p1 ∪ p2 = {2, 3, 4}

2 evaluate *christmas **AND** (ice **OR** punch)*

i. t1 = „christmas“
ii. p1 = {1, 2, 3, 4}
iii. t2 = *(ice **OR** punch)*
iv. p2 = {2, 3, 4}
v. p1 ∩ p2 = {2,3,4}

(d) Information need: *Should I do sport on the ice or drink punch to warm up on christmas?*
Relevant documents: $D_1$ (skating on ice), $D_2$, $D_3$, $D_4$
Precision: 3/3
Recall: 3/4
Recall was not perfect, as I added $D_1$ by my understanding of the term *ice* as in *skating on ice*. The Boolean Query Algorithm does not account for such far fetched relations and thus can not find this explicit-positive document.

(e) The term christmas appears in all documents. That means it either does not have any effect on a query, or, when disjunctive, always returns all documents, or, when negated and conjunctive, the result set is always empty.

**Task 2:** Tokenisation 3+12 Points

(a) A token is an instance of a limited character string that occurs in a given document and is grouped into a semantically meaningful unit for further processing.
A token can occur more than once in a document.

A Term is a (possibly „normalized“) type that is added to the vocabulary. Normalization can be done for example with respect to upper and lower case, morphology (part of speech, flection, etc.), spelling.

(b) *Punctuation marks*
During tokenization the following punctuaction marks are normally ignored:
`. , ; : ? ! ’ ":`
O'Connor as „O“ and „Connor“

*Hyphens*
In the example sentence, it is not clear how „Peter-Paul-and-Mary“ is tokenized.
Peter-Paul-and-Mary as „Peter“, „Paul“„and“ and „Mary“ or „PeterPaulandMary“ or „Peter-Paul-and-Mary“

*Umlauts*
In this example sentence, the word „Kürenz“ is also unclear when tokenized.
Kürenz as „Kuerenz“ or „Kurenz“

*Hyphenation at line end*
In the example sentence, the word „Ceremony“ is separated by a hyphen at the end of the line, making tokenization more difficult.
Ceremony as „cere“ and „mony“

**Task 3:** Document preprocessing 15 Points

(a) **Tokens:** analyzing, online, schema, extraction, approaches, for, linked, data, knowledge, bases, elements, of, computer, science, artificial, intelligence

(b) **Stop words (to be removed):** for, of

(c) **Porter Algorithm**, where (1b-2) means *Step 1b rule 2* as per the original definition[1]:

| | |
|---|---|
| analyzing | (1b-3) analyz |
| online | (5a-1) onlin |
| schema | |
| extraction | (4-12) extract |
| approaches | (1a-4) approache (5a-1) approach |
| linked | (1b-2) link |
| data | |
| knowledge | (5a-1) knowledg |
| bases | (1a-4) base (5a-2) bas<br>(*nltk.stem.PorterStemmer returned „base“ but b-**as** is m=1, bas is/ends CVC and the second c is not W, X, or Y...*) |
| elements | (1a-4) element |
| computer | (4-4) comput |
| science | (5a-2) scienc |
| artificial | (4-1) artifici |
| intelligence | (4-3) intellig |

[1] `https://tartarus.org/martin/PorterStemmer/def.txt`