Aufgabe 5:

Gegeben sind folgende Dokumente
Doc1: Neues aus der Literaturwissenschaft: Durchbruch in der Shakespeareforschung.
Doc2: ein neues buch von shakespeare entdeckt.
Doc3: Freude für Shakespeareliebhaber.
Doc4: Waren das schon alle neuen Bücher Shakespeares?

1) Konstruieren Sie einen Index durch folgende Zwischenschritte: Tokenisierung, Normalisierung(Lemmatisierung), Sortieren und Gruppieren.
2) Was liefern folgende booleschen Anfragen für die vier Dokumente in der vorherigen Aufgabe zurück:
- neu AND Shakespeare
- Buch AND neu
3) Wieweit hängen die Ergebnisse von der Normalisierung der Token (linguistische, Bearbeitung wie Lemmatisierung etc.) ab?
4) Wieweit hängen die Ergebnisse von der Tokenisierung ab? Was wäre das Ergebnis wenn ein Kompositasplitter eingebaut ist

1)
Term
Doc.freq
Postings lists

buch:2,4
durchbruch:1
entdecken:1
freude:3
literaturwissenschaft:1
neu:1,2,4
shakespeare:2,4
shakespeareforschung:1
shakespeareliebhaber:1

2)
{Doc2, Doc4}
{Doc2, Doc4}

3)
Die Normalisierung:
- Linguistische Normalisierung: Verschiedene Formvarianten eines Terms werden auf eine einheitliche Grundform reduziert, z. B. die Vollform des Terms wird zum entsprechenden Lemma ersetzt und einer Äquivalenzklasse geordnet, beispielsweise die Wortformen

„Neues“ und „neuen“ → neu „Buch“ und „Bücher“ → Buch
- Normalisierung von Wortvarianten, Flexion, Derivation: waren → sein
- Groß- und Kleinschreibung wandeln sich in Kleinschreibung: „neues“ und „Neues“
- Stopwörter werden entfernt, für, das etc.

Die Tokenisierung:
- Der Text wird vor allem durch Leerzeichen in Token segmentiert. Aber für Mehrwortlexeme ist es problematisch: New York
- Die Interpunktion ist üblich gelöscht. Aber für Abkürzung wie C.A.T darf die Interpunktion nicht gelöscht werden.
- Stoppwortliste: entfernen oder behalten
- Kompositasplitter, wie z.B. ohne Kompositasplitter findet die Anfrage „Shakespeare“ keine Dokumente mit „Shakespeareforschung“, „Shakespearesliebhaber“. Problematisch sind bei Komposita wie Staubecken → Stau Becken or Staub Ecken.

Aufgabe6

(a) Was sind Vor- und Nachteile des Booleschen Retrievals?
(b) Angenommen, die folgenden Dokumente seien von einem Booleschen Retrievalsystem indexiert worden. Dabei fand die übliche Stoppworteliminierung, sowie Stemming statt.
1. Evaluating Strategic Support for Information Access in the Daffodil System.
2. Daffodil: A User-Oriented Approach for Accessing Federated Digital Libraries.
3. Daffodil: Distributed Agents for User-Friendly Access of Digital Libraries.
4. Daffodil - Strategic Support for User-Oriented Access to Heterogeneous Digital Libraries.
5. Active Support for Query Formulation in Virtual Digital Libraries: A case study with Daffodil.
6. Daffodil: An integrated desktop for supporting high-level search activities in federated
digtal libraries.
7. User-Oriented Query Modi cation in Metaclass Systems.
8. Daffodil: Strategic Support Evaluated
Formulieren Sie möglichst knappe Boolesche Anfragen (mit AND, OR und NOT), die genau die folgenden Dokumente finden:

(i) 2 und 4
(ii) 7
(iii) 8

Vorteile
- Einfache Implementierung
- Keine Benutzung von Wahrscheinlichkeiten/Heuristiken

- Viele experimentelle Systeme
- keine hohen Anforderungen an Rechner
- Ausdrucksstarke Syntax, logische Klarheit

Nachteile

- Die Größe der Antwortmenge ist schwierig zu kontrollieren
- Keine Ordnung der Antwortmenge
- Keine Möglichkeit zur Gewichtung von Fragetermen oder gewichteter Indexierung
- Strenge Trennung zwischen gefunden / nicht gefunden
- Erstellung der Frageformulierung sehr umständlich
- schlechte Retrievalqualität
-

b)
Dokumente 2 und 4: Orient NOT Query
Dokumente 7: Modification / Metaclass;
Dokumente 8: Evaluate NOT Information

Aufgabe 7

Schauen Sie sich im Buch Manning et al. Introduction to Information Retrieval den Algorithmus Postitional Intersect an. Versuchen Sie zu verstehen was der Algorithmus macht. Geben Sie eine Variante des Algorithmus an, die Phrasenqueries aus zwei Anfragetermen verarbeitet, also z.B. ,''Stanford University''

```
POSITIONAL INTERSECT(p1, p2)
answer ← < >
while p1=/= NIL and p2 =/=NIL                zwei Pointer aus dem Dictionary für die
zwei Terme Stanford AND University
do if docID(p1) = docID(p2)                  Beide müssen im selben Dokument
vorkommen
   pp1 ← position(p1)                        Positionspointer auf den Anfang der jeweiligen
Positionsliste im Dokument gesetzt
   pp2 ← position(p2)
          while (pp1 =/= null and pp2 =/= null
            do if (pos(pp2) - pos(pp1) == 1)          Bedingung: Abstand ist 1,
Terme also nacheinander. zweiter Term muss auf den ersten folgen
                 add(answer, (docID(p1), pos(pp1), pos(pp2)))   Antwort schreiben
(möglicherweise break? boolesche Anfrage???)
                 pp1 ←  next(pp1)
                 pp2 ←  next(pp2)
               else if (pos(pp2) > pos(pp1))
weiter in den Postionslisten
                 pp1 ←  next(pp1)
               else
                 pp2 ← next(pp2)
   p1 ← next(p1)
```

```
    p2 ← next(p2)                                   weiter in den zwei postings listen
    else if docID(p1) < docID(p2)
    then p1 ← next(p1)                              weiter in der einen oder anderen
postingsliste
    else  p2 ← next(p2)
```

Aufgabe 8

Implementieren Sie für das Information Retrieval System eine Klasse, die einen invertierten Index erstellt wie er im Information Retrieval Buch Manning et. al. als Basisalgorithmus dargestellt wurde also für jeden Term gibt es eine Postingsliste mit den Dokumenten IDs. Verwenden Sie zum Test wieder die Shakespearesammlung.