Übung Big-Data-Technologien
Prof. Dr. Christoph Schmitz

Übungsblatt 3

1 MapReduce: Invertierter Index (Zählt doppelt wg. höherem Aufwand)

Schreiben Sie einen MapReduce-Job, der einen **invertierten Index** über eine Sammlung von Textdateien erstellt!

Das bedeutet: gegeben ist eine Sammlung von Dokumenten d_1, d_2, …, d_n. Diese enthalten verschiedene Terme (Wörter) t_1, …, t_m. Daraus wird ein Index folgender Art erstellt:

t_1: $d_{1,1}$, <Anzahl Vorkommen> $d_{1,2}$, <Anzahl Vorkommen> … $d_{1,n1}$, <Anzahl Vorkommen>
t_2: $d_{2,1}$, <Anzahl Vorkommen> $d_{2,2}$, <Anzahl Vorkommen> … $d_{2,n2}$, <Anzahl Vorkommen>
…
t_m: $d_{m,1}$, <Anzahl Vorkommen> $d_{m,2}$, <Anzahl Vorkommen> … $d_{m,n}$, <Anzahl Vorkommen>

In dem Index steht also zu jedem Term t_i, in welchen Dokumenten er wie oft vorkommt. Es werden nur die Dokumente für jeden Term aufgezählt, in denen der Term auch tatsächlich vorkommt.

„Invertierter Index“ heißt dies deswegen, weil aus einer Zuordnung Dokument → Terme die umgekehrte Zuordnung Term → Dokumente errechnet wird.

Weitere Anforderungen:

- Im Verzeichnis `src/main/resources` des Projektgerüsts finden Sie eine Liste `stopwords.txt` mit sogenannten Stoppwörtern, die nicht gezählt werden sollen, da sie ohnehin in fast jedem Dokument vorkommen.
- Wenn die Anzahl Vorkommen eines Terms in einem Dokument kleiner als drei ist, soll diese Kombination Term-Dokument nicht ausgegeben werden.
- Terme, für die nach dieser Filterung kein Dokument mehr übrig bleibt, sollen nicht ausgegeben werden.
- Weisen Sie die Anzahl der gezählten und nicht gezählten Wörter (also der Stoppwörter) über Counter aus!
- Weisen Sie die Anzahl der als zu klein verworfenen Term-Dokument-Kombinationen über Counter aus!

Beispielcode, Daten und Hinweise:

- Sie finden ein Projektgerüst im Stud.IP unter `bdt-uebung-3-invertedindex.zip`.
- Im Projektgerüst finden Sie den Code für das Einlesen der Stoppwörter.
- Sie finden hier ebenfalls Code dafür, wie Sie im Mapper an den Dateinamen der Eingabedaten kommen.
- Die Stoppwörter filtern Sie am besten im Mapper, die Anzahl der Vorkommen pro Dokument im Reducer.

- Machen Sie sich mit den verwendeten Klassen aus Google Guava vertraut, hier: `Resources`; `Joiner`; `Splitter`. Guava kann Ihnen viel Arbeit sparen!
- Beachten Sie die vorhandenen Unit-Tests für Mapper und Reducer. Diese können Ihnen helfen, die gewünschte Funktionalität korrekt zu implementieren.
- Sie finden Beispieldaten mit bereinigten Texten von William Shakespeare unter `/data/shakespeare` im HDFS.
- Die Ausgabe sollte ungefähr so aussehen (die genaue Einrückung usw. spielt keine Rolle):

```
...
rebel       1henryiv.txt: 4
rebellion   1henryiv.txt: 4   2henryiv.txt: 9   coriolanus.txt: 3
            john.txt: 4
recreant    john.txt: 4 richardii.txt: 4
redress     1henryvi.txt: 3   2henryiv.txt: 3   julius_caesar.txt: 6
            measure.txt: 3
refuse      coriolanus.txt: 3 merchant.txt: 5
...
```

- Die Counter sollten ungefähr so aussehen:

```
INDEX_MAP
      STOPWORD_REMOVED=539039
      WORD_COUNTED=541973
INDEX_REDUCE
      COUNTED=25639
      TOO_FEW_OCCURRENCES=91612
```

  Dabei sind `INDEX_MAP` und `INDEX_REDUCE` die Counter Groups für Mapper und Reducer, und `STOPWORD_REMOVED` usw. sind die eigentlichen Counter.