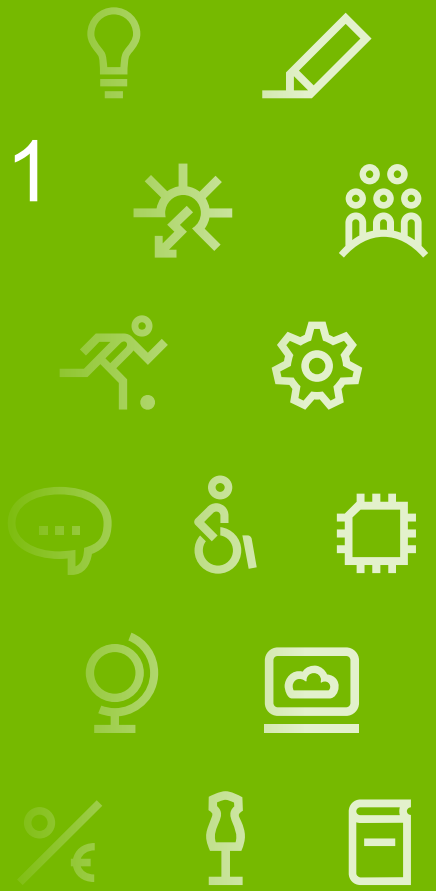
Hadoop 1 - Aufgabe 11

Häufigkeitsberechnung

Florian Symmank / 14.01.2025



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Agenda

1	Aufgabe
2	Lösungsbeschreibung
3	Parameter
4	Demo
5	Ergebnisse
6	Fazit

Häufigkeitsberechnung

- Berechnen Sie die Vorkommenshäufigkeit für alle Wörter in gegebenen Texten
- Stellen Sie die Ergebnisse als TOP-10 Liste getrennt für jede Sprache vor, mit Ausnahme der Stoppwörter (z.B. “a”, “an”, “the”, “of”, “und”, “mit”, “la”, “и”, “же” usw.)

Lösungsbeschreibung

Zählen der Wörter

Schritte:

- Angabe der Textdatei und Sprache
- Hinzufügen der Stoppwortliste
- Spezifizierung des Ausgabeorts
- Verteilung auf mehrere Map-Reducer-Prozesse

Mapper-Prozess:

- Ermittlung der Wortgrenzen
- Umwandlung der Wörter in Kleinbuchstaben
- Suche nach Wörtern in der Stoppwortliste
- Erstellung von Tupel-Einträgen, z. B. („wort“, 1), für nicht-stoppwörter
- Verwendung eines HashSet für effiziente Stoppwortsuche O(1) Lookup

Reduce-Prozess:

- Aufsummierung der Tupel (Wort und Anzahl)
- Ausgabeformat: [.., ("wort", 2030), ("wortarm", 1), ...]

Zählen der Wörter

```java
@Override
protected void map(Object key, Text value, Mapper<Object, Text, Text, IntWritable>.Context context)
        throws IOException, InterruptedException {
    Pattern pattern = Pattern.compile("[\\p{L}|\\d]+");
    Matcher matcher = pattern.matcher(value.toString());

    while (matcher.find()) {
        String word = matcher.group().toLowerCase(Locale.ROOT);
        if (stopwords.contains(word)) {
            continue;
        }
        context.write(new Text(word), new IntWritable(1));
        context.getCounter("WordCount", "TotalWords").increment(1);
    }
}
```

Sortierung der Wörter nach Häufigkeit

Schritte:

- Angabe Ausgangsdatei und Speicherort
- Verwendung eines Reducers ⇒ globale Sortierung

Mapper-Prozess:

- Eingelesener Text (Wort und Häufigkeit) wird genutzt
- Erstellung des Schlüssels (Anzahl, Wort)
- Sortierung nach:
  - Primär: Anzahl (Häufigkeit)
  - Sekundär: Wort (bei gleicher Anzahl)

Reduce-Prozess:

- Ausgabeformat:
  [(5683, "rief"), (5610, "hand"), ...]

Sortierung der Wörter nach Häufigkeit

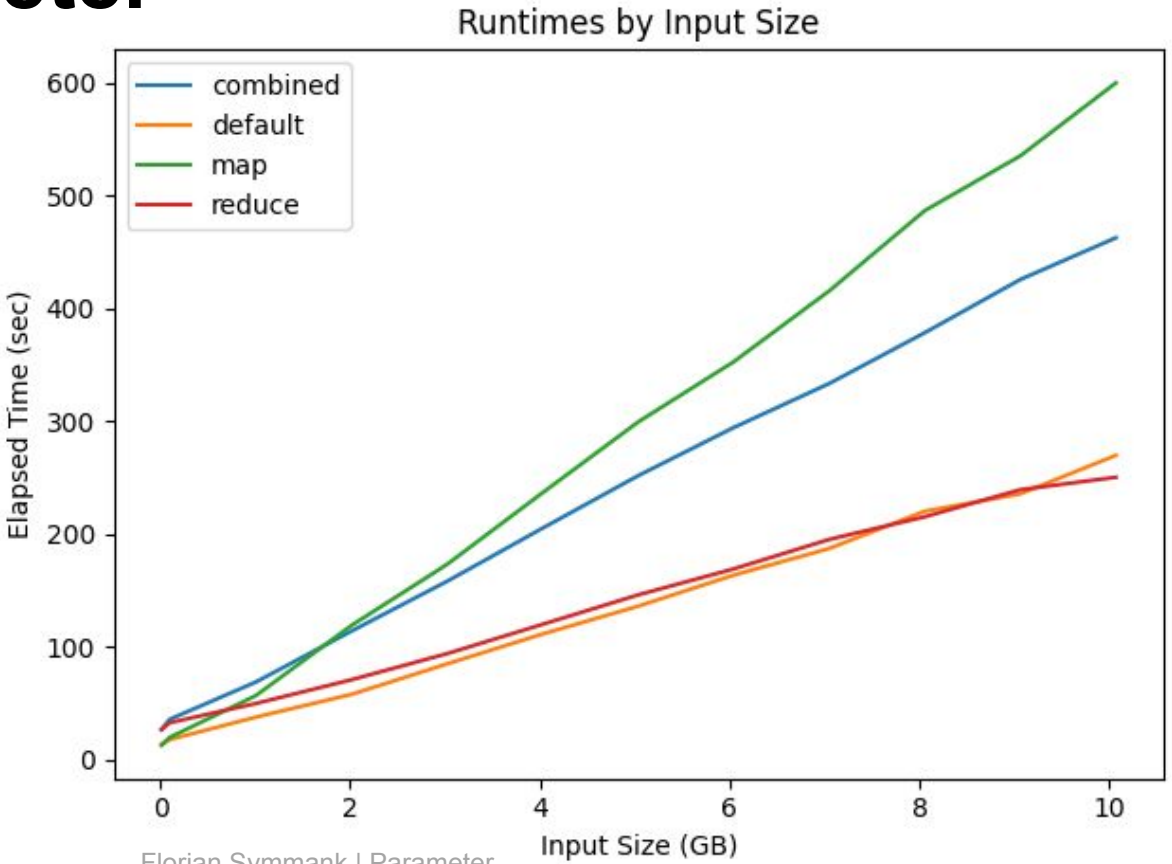
```
public void map(Object key, Text value, Context context)
        throws IOException, InterruptedException {
    // Input format: word \t count
    String[] parts = value.toString().split("\t");
    if (parts.length == 2) {
        String word = parts[0];
        int count = Integer.parseInt(parts[1]);
        compositeKey = new CompositeKey(count, word);
        context.write(compositeKey, NullWritable.get());
        context.getCounter("KeyCount", "TotalKeys").increment(1);
        return;
    }

    throw new IOException("Invalid input format: " + value.toString());
}
```

Parameter

Parameter

Parametername	Default	Map	Reduce	Combined
NumReduceTasks	1	1	8	8
mapreduce.reduce.memory.mb	1024	1024	4096	4096
mapreduce.reduce.java.opts	-Xmx768m	-Xmx768m	-Xmx3072m	-Xmx3072m
mapreduce.reduce.speculative	false	false	true	true
mapreduce.output.compress	false	false	true	true
mapreduce.output.compress.codec	Not set	Not set	org.apache.hadoop.io.compress.GzipCodec	org.apache.hadoop.io.compress.GzipCodec
mapreduce.map.memory.mb	1024	2048	1024	2048
mapreduce.map.java.opts	-Xmx768m	-Xmx1536m	-Xmx768m	-Xmx1536m
mapreduce.map.speculative	false	true	false	true
mapreduce.task.io.sort.mb	100	512	100	512
mapreduce.task.io.sort.factor	10	100	10	100

Parameter


Runtimes by Input Size
combined
default
map
reduce
Elapsed Time (sec)
0
100
200
300
400
500
600
0
2
4
6
8
10
Input Size (GB)

Demo

Vorbereitung 1

```
# Upload book-data to hadoop
docker cp <path_to_local_books> <container_name>:/usr/local/hadoop/<path_to_books>/
# Upload stopwords to hadoop
docker cp <path_to_local_stopwords> <container_name>:/usr/local/hadoop/<path_to_stopwords>/
# Upload jar
docker cp <path_to_local_jar> <container_name>:/usr/local/hadoop/<path_to_jar>

## Enter hadoop
docker exec -it -w /usr/local/hadoop <container_name> /bin/bash

# Upload book-data to hdfs
bin/hdfs dfs -put <path_to_books>/<language> /data
# Upload stopwords to hdfs
bin/hdfs dfs -put <path_to_stopwords> /data/stopwords.json

# Verify files are uploaded
bin/hdfs dfs -ls /data

# Reset outdir
bin/hdfs dfs -rm -r /output/*
```

Vorbereitung 2

```
# Count the words
bin/hadoop jar fs/hadoop_wordcount.jar de.floriansymmank.WordCountDriver de /data/de/de_all.txt
/output/de_all_wordcount /data/stopwords.json /output/stats.txt
# Sort the words
bin/hadoop jar fs/hadoop_wordcount.jar de.floriansymmank.SortByCountDriver
/output/de_all_wordcount/part-r-00000 /output/de_all_sorted

# Check the output
bin/hdfs dfs -cat /output/de_all_wordcount/*
bin/hdfs dfs -cat /output/de_all_sorted/*

# Download result from hdfs to hadoop
bin/hdfs dfs -get /output/* /tmp/output
```

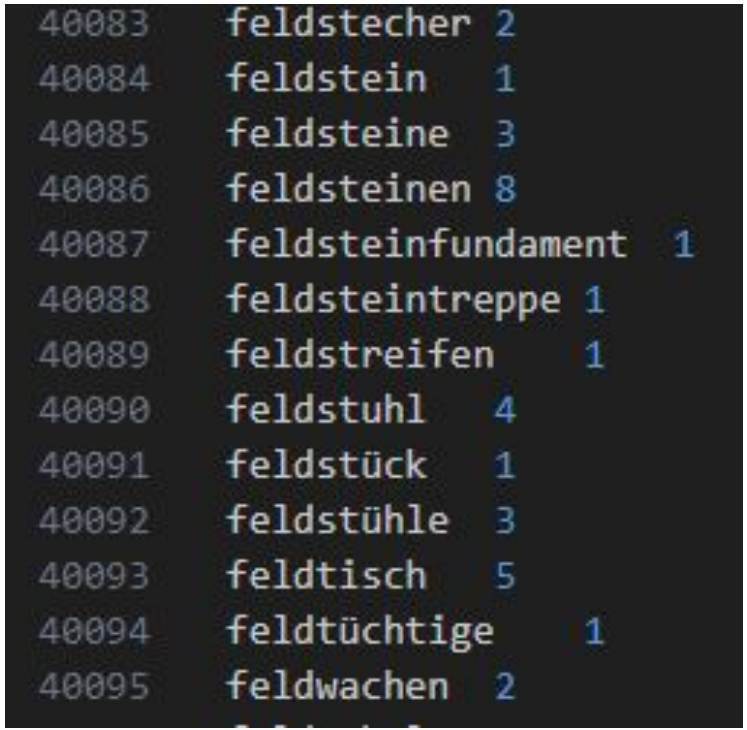
Zählen der Wörter

```
root@92574758821d:/usr/local/hadoop-2.8.1# bin/hadoop jar fs/hadoop_wordcount.jar de.floriansymmank.WordCountDriver de /data/de/de_all.txt /output/de_all_wordcount /data/stopwords.json /output/stats.txt
25/01/07 12:17:00 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
25/01/07 12:17:00 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
25/01/07 12:17:01 INFO input.FileInputFormat: Total input files to process : 1
25/01/07 12:17:02 INFO mapreduce.JobSubmitter: number of splits:1
25/01/07 12:17:02 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1736251189950_0001
25/01/07 12:17:03 INFO impl.YarnClientImpl: Submitted application application_1736251189950_0001
25/01/07 12:17:03 INFO mapreduce.Job: The url to track the job: http://92574758821d:8088/proxy/application_1736251189950_0001/
25/01/07 12:17:03 INFO mapreduce.Job: Running job: job_1736251189950_0001
25/01/07 12:17:07 INFO mapreduce.Job: Job job_1736251189950_0001 running in uber mode : false
25/01/07 12:17:07 INFO mapreduce.Job:  map 0% reduce 0%
25/01/07 12:17:13 INFO mapreduce.Job:  map 100% reduce 0%
25/01/07 12:17:17 INFO mapreduce.Job:  map 100% reduce 13%
25/01/07 12:17:19 INFO mapreduce.Job:  map 100% reduce 25%
25/01/07 12:17:21 INFO mapreduce.Job:  map 100% reduce 38%
25/01/07 12:17:23 INFO mapreduce.Job:  map 100% reduce 50%
25/01/07 12:17:25 INFO mapreduce.Job:  map 100% reduce 63%
25/01/07 12:17:27 INFO mapreduce.Job:  map 100% reduce 75%
25/01/07 12:17:29 INFO mapreduce.Job:  map 100% reduce 88%
25/01/07 12:17:31 INFO mapreduce.Job:  map 100% reduce 100%
25/01/07 12:17:32 INFO mapreduce.Job: Job job_1736251189950_0001 completed successfully
25/01/07 12:17:32 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=2768618
```

...

```
                WRONG_REDUCE=0
        WordCount
                TotalWords=2164266
        File Input Format Counters
                Bytes Read=34102705
        File Output Format Counters
                Bytes Written=2166304
Stats:
Input File: de_all.txt
Input File Size (bytes): 34102705
Language: de
Total Words: 2164266
Elapsed Time (ms): 31742
Words per Minute: 4090982
```

Zählen der Wörter



/output/de_all_wordcount/part-r-00000

Sortierung der Wörter nach Häufigkeit

```
root@92574758821d:/usr/local/hadoop-2.8.1# bin/hadoop jar fs/hadoop_wordcount.jar de.floriansymmank.SortByCountDriver /output/de_all_wordcount/part-r-00000 /output/de_all_sorted
25/01/07 12:24:47 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
25/01/07 12:24:47 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
25/01/07 12:24:47 INFO input.FileInputFormat: Total input files to process : 1
25/01/07 12:24:48 INFO mapreduce.JobSubmitter: number of splits:1
25/01/07 12:24:48 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1736251189950_0002
25/01/07 12:24:48 INFO impl.YarnClientImpl: Submitted application application_1736251189950_0002
25/01/07 12:24:48 INFO mapreduce.Job: The url to track the job: http://92574758821d:8088/proxy/application_1736251189950_0002/
25/01/07 12:24:48 INFO mapreduce.Job: Running job: job_1736251189950_0002
25/01/07 12:24:52 INFO mapreduce.Job: Job job_1736251189950_0002 running in uber mode : false
25/01/07 12:24:52 INFO mapreduce.Job:  map 0% reduce 0%
25/01/07 12:24:55 INFO mapreduce.Job:  map 100% reduce 0%
25/01/07 12:24:59 INFO mapreduce.Job:  map 100% reduce 100%
25/01/07 12:24:59 INFO mapreduce.Job: Job job_1736251189950_0002 completed successfully
25/01/07 12:24:59 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=343087
```

...

```
        File Input Format Counters
                Bytes Read=268374
        File Output Format Counters
                Bytes Written=268374
Stats:
Output File: de_all_sorted
Input File: part-r-00000
Input File Size (bytes): 268374
Total Keys: 19546
Elapsed Time (ms): 11877
Keys per Minute: 98742
```

Sortierung der Wörter nach Häufigkeit

```
 1   5683   rief
 2   5610   hand
 3   5172   augen
 4   4815   herr
 5   4071   fragte
 6   3924   leben
 7   3701   sprach
 8   3680   frau
 9   3648   fort
10   3630   stand
11   3553   vater
```

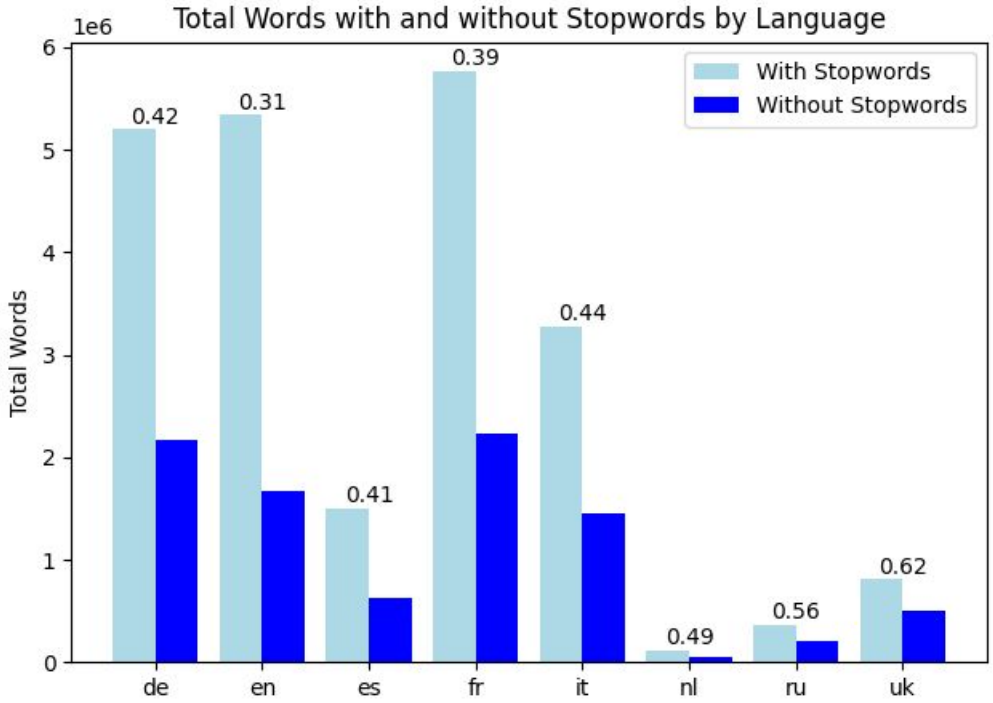
/output/de_all_sorted/part-r-00000

```
 1   171278   und
 2   145824   die
 3   138922   der
 4   74708    den
 5   71233    zu
 6   69542    in
 7   69358    er
 8   66081    ich
 9   64111    das
10   63488    sie
11   48475    sich
```

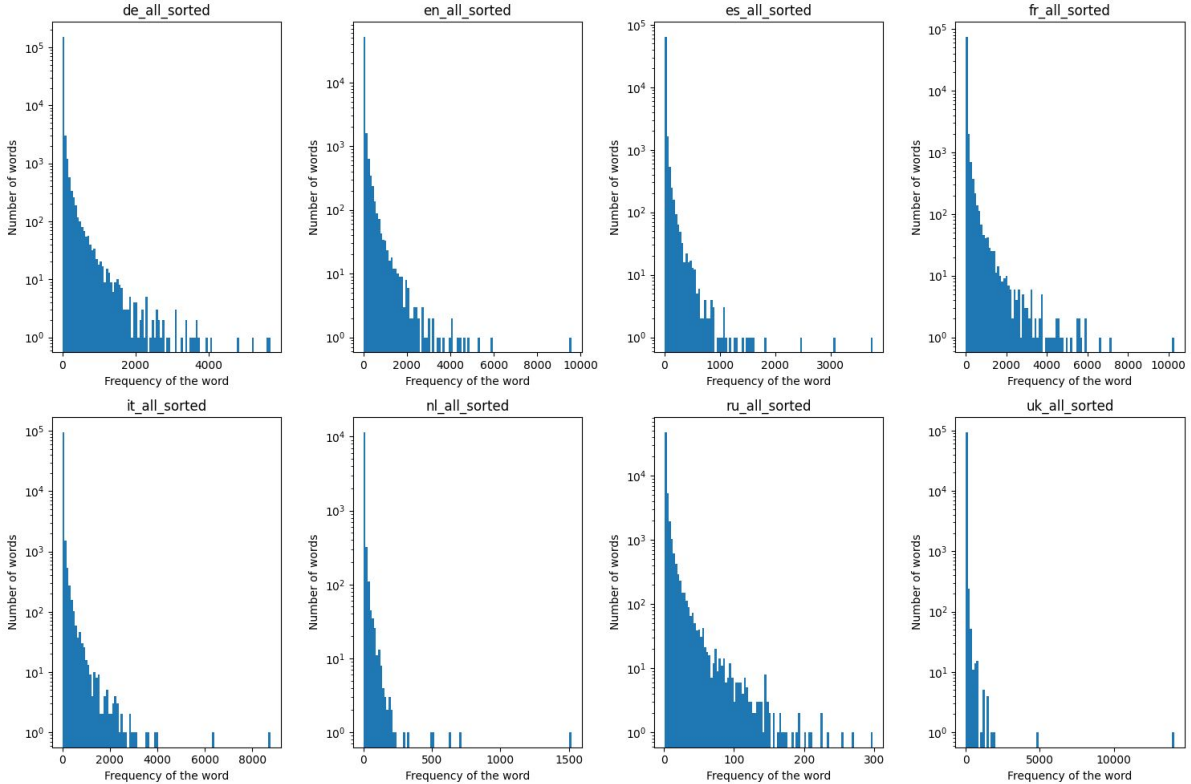
/output/de_all_xx_sorted/part-r-00000

Ergebnisse

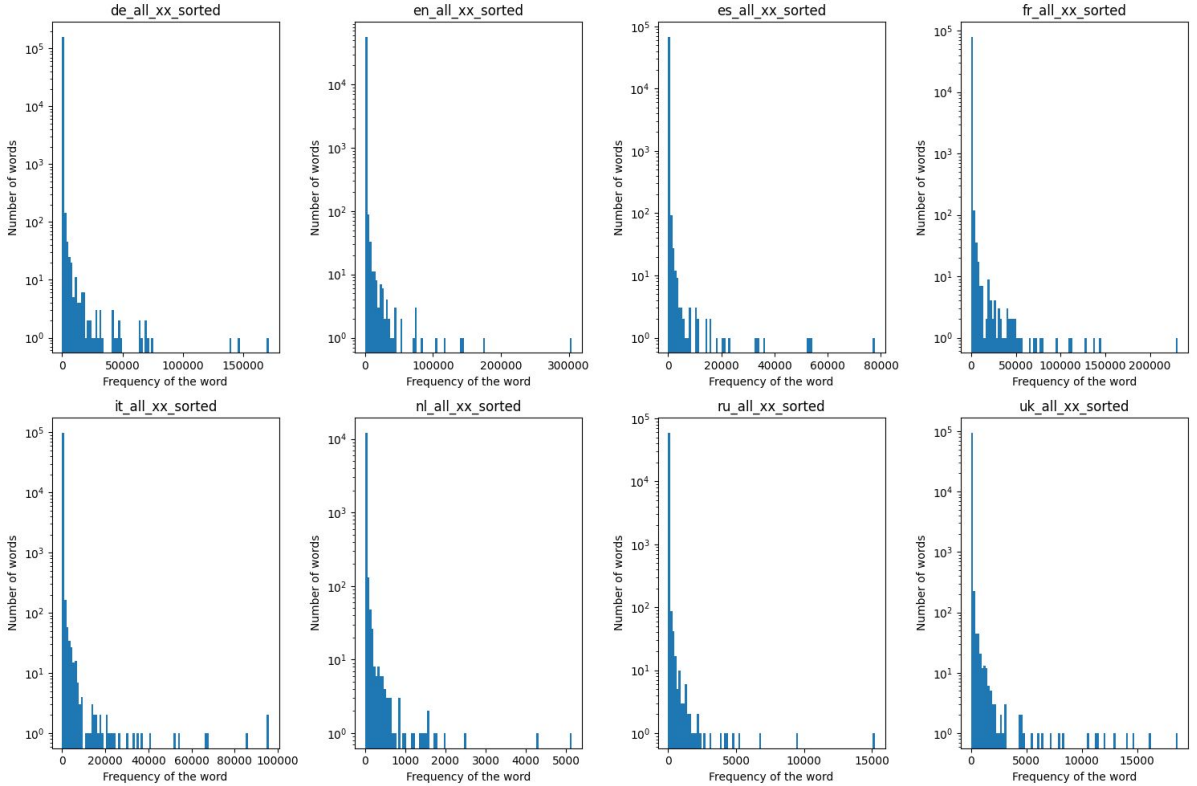
Stoppwörter zu nicht-Stoppwörter



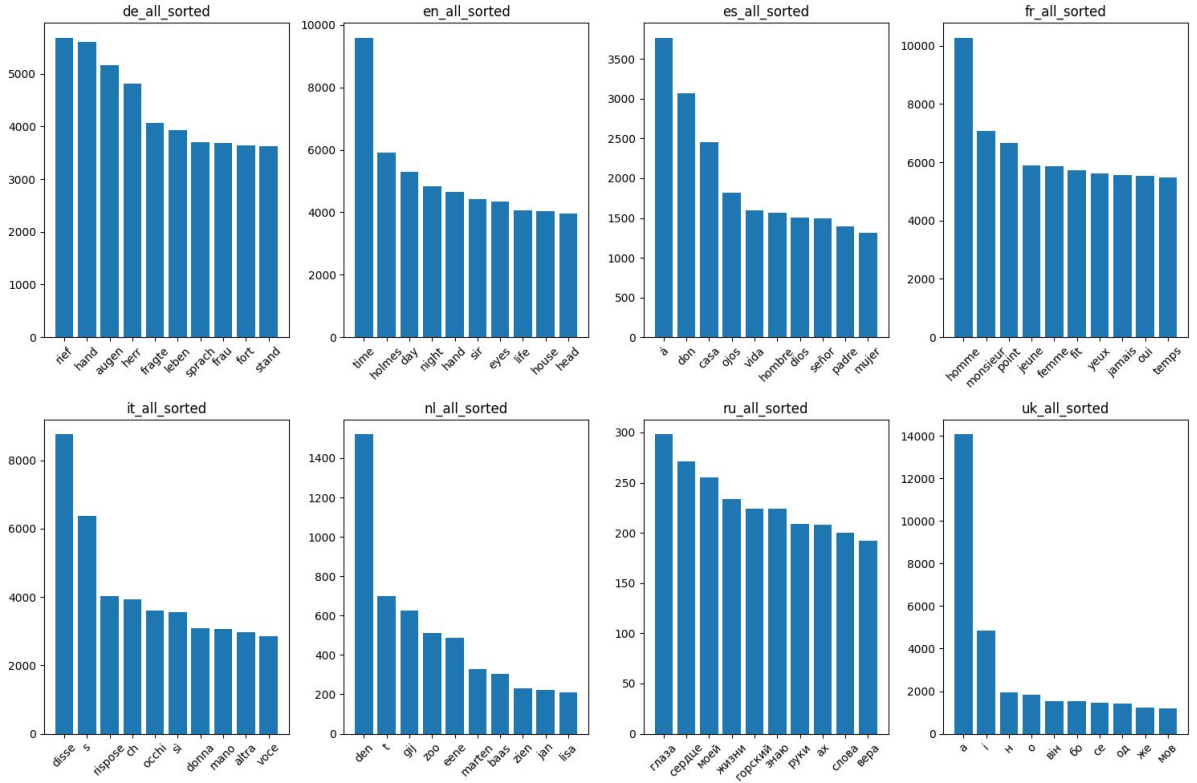
Worthäufigkeit - ohne Stoppwörter



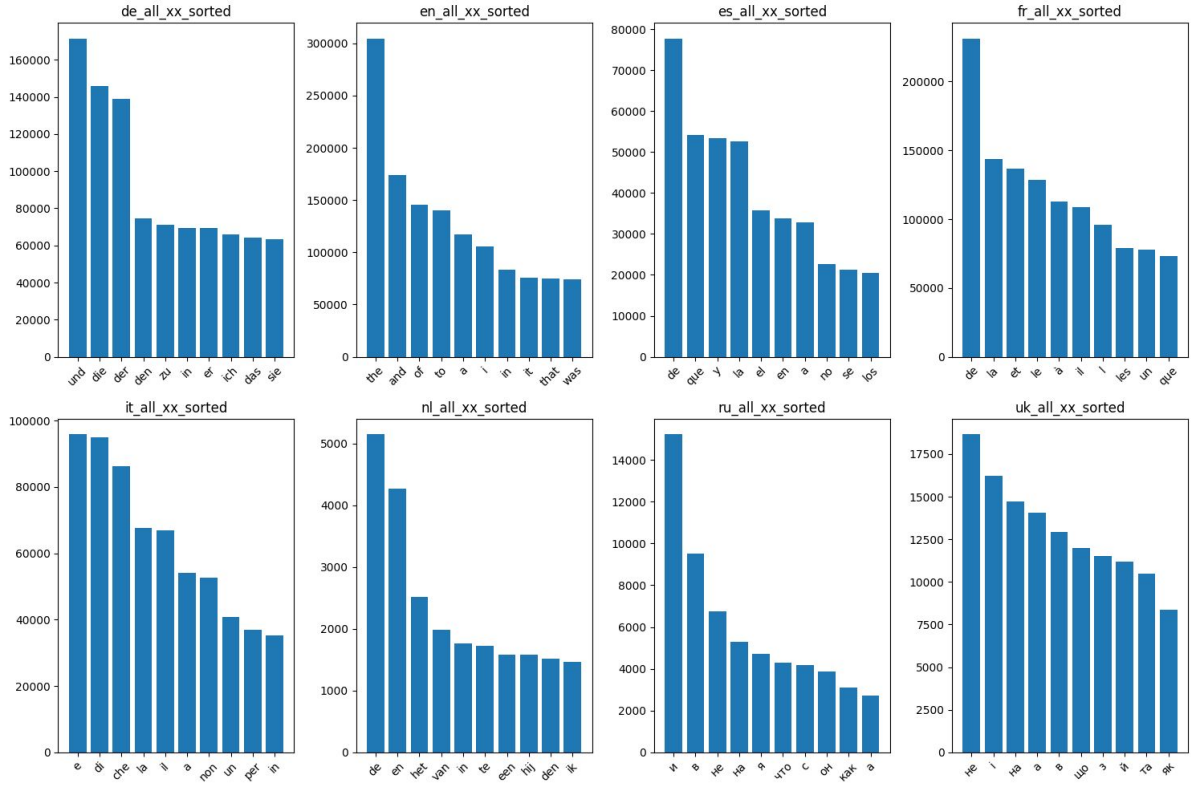
Worthäufigkeit - mit Stoppwörter



TOP-10 Liste - ohne Stoppwörter



TOP-10 Liste - mit Stoppwörter



Fazit

Fazit

- Übernimmt Boilerplate-Code bei MapReduce
- Lernkurve vorhanden ⇒ Einarbeitungszeit
- Komponenten klar strukturiert
- Logischer Aufbau
- Geringer Aufwand möglich
- Overkill für manche Aufgaben

Vielen Dank.

www.htw-berlin.de

Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

www.htw-berlin.de