MapReduce-алгоритм построения инвертированного индекса

Требуется построить инвертированный индекс (inverted index) для заданного корпуса текстов (текстового файла).

Входные данные map:

```
(docid, content)
```

Результирующий инвертированный индекс должен иметь следующую структуру:

```
(word, [<docid1, TF-IDF1>, <docid2, TF-IDF2>, ...])
```

- Статьи должны быть отсортированы в порядке убывания TF-IDF (Term Frequency – Inverse Document Frequency)
- Для каждого слова ограничить список статей *N* наиболее релевантными
- Определить и исключить из индекса Top20 высокочастотных слов

При вычислении TF-IDF считаем, что:

- *TF*(*t*, *d*) — это число вхождений слова *t* в документ *d* (Wiki-статью)
- *IDF*(*t*, *D*) — обратная частота, с которой слово *t* встречается во множестве документов *D* (Wiki-статьях):

$$\mathrm{IDF}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

Программы должны быть написаны на языке Java (Apache Hadoop Java API)