HOMEWORK 2 – Matteo Maraziti

https://github.com/matteomaraziti/HW2IngegneriaDeiDati

ANALYZER SCELTI

CONTENUTO

```
Analyzer analyzer1 = CustomAnalyzer.builder()
                .withTokenizer(WhitespaceTokenizerFactory.class)
                .addTokenFilter(HyphenatedWordsFilterFactory.class)
                .addTokenFilter(WordDelimiterGraphFilterFactory.class)
                .addTokenFilter(LowerCaseFilterFactory.class)
                .addTokenFilter(SuggestStopFilterFactory.class )
                .build();
```

Figura 1 Analyzer per il contenuto

- WhitespaceTokenizer: divide e scarta solo i caratteri di spazio bianco. quindi crea dei token ogni volta che ci sono degli spazi vuoti fra le parole. Include nei token anche la punteggiatura.
  In: "To be, or what?"
  Out: "To", "be,", "or", "what?"
- HyphenatedWordsFilter: Se due parole sono separate da un "-" allora il token restituito e' la parola formata dalla concatenazione delle due parole scartando il "-".
  In: "hyphen- ated"
  Out: "hyphenated"
- WordDelimiterGraphFilter: Questo filtro divide i token in corrispondenza dei delimitatori di parola.
  Le regole per determinare i delimitatori sono determinate come segue:
  "CamelCase" --> "Camel", "Case".
  "Gonzo5000" --> "Gonzo", "5000".
  "hot-spot" -->"hot", "spot".
  "O'Reilly's" --> "O", "Reilly"
  "--hot-spot--" --> "hot", "spot"
- LowerCaseFilter: rende caratteri maiuscoli e minuscoli equivalenti
  In: "Down With CamelCase"
  Out: "down", "with", "camelcase"
- 5-SuggestStopFilter: si tratta di uno stop filter che tokenizza le stopwords non seguite da un separatore
  In: "The The"
  Out: "the"(2)

TITOLO

```
Analyzer analyzer2 = CustomAnalyzer.builder()
                .addCharFilter(PatternReplaceCharFilterFactory.class, map)
                .withTokenizer(WhitespaceTokenizerFactory.class)
                .addTokenFilter(WordDelimiterGraphFilterFactory.class)
                .addTokenFilter(LowerCaseFilterFactory.class)
                .build();
```

Figura 2 Analyzer Titolo

- PatternReplaceCharFilter: utilizza le espressioni regolari per sostituire determinati pattern di caratteri. in questo caso e' stato adottato per sostituire nel nome del file il ".txt" con la stringa vuota.

FILE INDICIZZATI

Per i test degli indici sono stati utilizzati tre file di testo contenenti informazioni su personaggi politici.
i tempi di indicizzazione dei tre documenti sono circa di 0.43 secondi.

QUERY UTILIZZATI

TITOLO

- Parametro di ricerca: Churchill
  Risultato: doc0 (WordDelimiter ha rimosso la "," dal token)
- Parametro di ricerca: president
  Risultato: doc0 e doc1 (pattern replace ha consentito di trovare i documenti cercando l'ultima parola del titolo)
- Paramentro di ricerca: usa president
  Risultato: doc0 doc1 (Lowercase ha reso equivalenti USA ed usa)

CONTENUTO

- Parametro di ricerca: previously
  Risultato: doc1
- Parametro di ricerca: an
  Risultato: non ha restituito alcun documento poichè si tratta di una stopword
- Parametro di ricerca: africanamerican
  Risultato: doc1 (Hyphenated ha tokenizzato "African- American" come africanamerican)
- Parametro di ricerca: states
  Risultato: doc0 doc1 doc2 (nel doc2 wordDelimiter ha individuato i token "states" e "man" a partire da "statesMan")
- Parametro di ricerca: 1874
  Risultato: doc2 (wordDelimiter ha creato due token a partire dall'anno di nascita e di morte scritti nel testo separati dal "-")
- Parametro di ricerca: lawyer
  Risultato: doc0