Web Information Retrieval (67782)
Ex3: Review Search

1 Exercise Description

In this exercise, your goal is to search for reviews and for products, based on a specific context, and then to return these results in ranked order. You will be implementing three different search methods, based on three different ranking functions. You will submit the following class, along with classes used to construct and read from your index (from the previous exercises). More details about the ranking functions appear below the code description.

```
package webdata;

public class ReviewSearch {

    /**
      * Constructor
      */
    public ReviewSearch(IndexReader iReader) {}

    /**
     * Returns a list of the id-s of the k most highly ranked reviews for the
     * given query, using the vector space ranking function lnn.ltc (using the
     *      SMART notation)
     * The list should be sorted by the ranking
     */
    public Enumeration<Integer> vectorSpaceSearch(Enumeration<String> query, int k) {}

    /**
     * Returns a list of the id-s of the k most highly ranked reviews for the
     * given query, using the language model ranking function, smoothed using a
     * mixture model with the given value of lambda
     * The list should be sorted by the ranking
     */
    public Enumeration<Integer> languageModelSearch(Enumeration<String> query,
                                                    double lambda, int k) {}

    /**
     * Returns a list of the id-s of the k most highly ranked productIds for the
```

```
     * given query using a function of your choice
     * The list should be sorted by the ranking
     */
    public Collection<String> productSearch(Enumeration<String> query, int k) {}

}
```

A query is an enumeration of tokens. You can assume that these tokens are already normalized, i.e., in lower case.

The first two functions simply implement ranking functions studied in class, in order to find reviews that are most relevant to a given query. Note that you should only return reviews that contain at least one word from the query. If there are only m reviews containing words from the query, and $m < k$, then the functions should return m results.

In the third function, you are returning products, instead of reviews. Note that:

- A single product can have multiple reviews
- Reviews also give a star rating for products
- Reviews are annotated with their helpfulness

Your goal is to develop a ranking function that takes all of these factors (and possibly others) into consideration, when returning products, given the user's keyword query. This is your opportunity to be creative, when developing a ranking function.

Note that while you should try to make your code reasonably efficient, we will only test your code on small-scale datasets (up-to 10000 reviews).

1.1 Exercise Submission

The code for your exercise should be submitted in a jar format via the course website. Only one submission should be made for each pair! Please make sure that the jar file uploaded is in the correct format, and contains all relevant files, including: all source files, a README (including any compiling or running issues, and the logins and teudat zehut numbers of the students submitting the project).

In addition to the code, you should write up a description of your productSearch function, and your intuitive justification. We will take into consideration originality and creativity in grading this part of the exercise. The write-up should be included in the jar file, within a file named ex3.pdf.

Your exercise should run correctly both on Linux and on Windows systems.