IST 2018/2019

Processamento e Recuperação de Informação

Lab 05: Evaluation in Information Extraction and Retrieval

To solve the following problems, we will use the *Whoosh* search library for Python[1]. The goal is to implement and evaluate a simple search engine.

1

The file `pri_cfc.txt` is composed of several text documents from the CFC collection[2]. It contains one document per line, and the first word of each line is the document ID.

Create a Python script that indexes all the documents using the Whoosh library. Make sure that you also store the document ID in a separate field.

The following code shows an example of how to index documents.

```
from whoosh.index import create_in
from whoosh.fields import *
schema = Schema(id = NUMERIC(stored=True), content=TEXT)
ix = create_in("indexdir", schema)
writer = ix.writer()
writer.add_document(id=1,
                    content=u"This is the first document we've added!")
writer.add_document(id=2,
                    content=u"The second one is even more interesting!")
writer.commit()
```

Note: The directory `indexdir` must already exist, for the index to be created.

2

Once the documents are indexed, implement a function that takes as argument a string and performs a search over the index, returning a list of document IDs, ordered by their similarity ranking.

The following example code shows how to perform a query using Whoosh.

```
from whoosh.index import open_dir
from whoosh.qparser import *
ix = open_dir("indexdir")
```

[1] `https://bitbucket.org/mchaput/whoosh/wiki/Home`. All code excerpts shown here are adapted from the Whoosh documentation.

[2] `http://www.dcc.ufmg.br/irbook/cfc.html`

```
with ix.searcher() as searcher:
    query = QueryParser("content", ix.schema, group=OrGroup).parse(u"first document")
    results = searcher.search(query, limit=100)
    for r in results:
        print r
    print "Number of results:", results.scored_length()
```

Notes:

All strings to be processed by Whoosh must be Unicode. You can convert any string to Unicode using the `decode`[3] method.

By default, Whoosh returns the results ordered using the BM25 similarity.

3

Implement functions to measure the precision, recall and F1 of a list of search results. The functions should take the following inputs:

(a) A list of document IDs (the result of a search);

(b) A set of IDs of the relevant documents.

Each of the functions that is to be developed should return the precision, recall, and F1 values, respectively.

4

The file `pri_queries.txt` contains a set of queries and, for each query, the set of relevant documents.

Implement a script that reads the file and, for each query, executes the corresponding search and measures the precision, recall and F1 of the results. The script should print out each query and its evaluation values and a final average value for each measure.

You should limit your search results to the first 100.

5 Pen and Paper Exercises

5.1 Evaluating an Information Retrieval System

A query Q yielded as answer the ordered set $R = \{d_1, d_2 \cdots d_{20}\}$. In R, documents d_i where i is odd where judged as *relevant*, whereas documents d_i where i is even where judged *not relevant*. Find the following values:

[3] `http://docs.python.org/2/library/stdtypes.html#str.decode`

(a) Precision@5;

(b) R-precision;

(c) The interpolated precision and recall curve.

5.2 Evaluating a Classifier

Consider a binary classification problem, where each instance can be assigned to either a positive or a negative class. Consider also that you have a dataset D with 10 instances, each assigned to the corresponding class by a domain expert.

A given classification model assigned a class to each of the instances in the dataset. Considering the predicted classes C, you want to evaluate the quality of the model.

$$D =< +, +, -, -, +, +, -, -, +, - >$$
$$C =< +, +, -, +, +, -, -, -, -, - >$$

(a) Draw the confusion matrix for the aforementioned classification results.

(b) Compute the accuracy, precision, recall and F1 metrics.

5.3 Evaluating an Information Extraction System

Consider the following textual document, to be processed by an Information Extraction (IE) system in order to collect entity mentions (i.e., recognize entities such as names for persons, titles/positions, and organizations).

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer"that stifled technological innovation.

Today, Microsoft claims to "love"the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels – the coveted code behind the Windows operating system – to select customers.

"We can be open source. We love the concept of shared source,"said Bill Veghte, a company VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying...

A domain expert has manually annotated all entity mentions in each sentence:

- **organizations:** Microsoft Corporation , Microsoft , Microsoft , Free Software Foundation
- **positions:** CEO, VP, founder

- **persons:** Bill Gates , Gates, Bill Veghte , Richard Stallman

The information extraction system produced the following results:

- **organizations:** Microsoft Corporation , Free Software Foundation
- **positions:** CEO, VP, founder
- **persons:** Bill Gates , Orwellian, Windows , Bill Veghte, Microsoft VP, Richard Stallman

(a) For each class, compute the precision, recall and F1 metrics.

(b) Compute the macro-averaged and micro-averaged precision for the IE system.