Lab 3 Evaluation

Information Retrieval, 2020

Introduction

Goal The objective of this lab is for you to get acquainted with varying parameters and learn several evaluation measures of retrieval systems.

Instructions

In this lab we will work with the Swedish MedEval test collection containing different types of articles within the medical domain. The idea is to see how the information retrieval from Swedish medical data is affected by noun decompounding.

`/local/course/ir/data/MedEvalTK`

Collection has 42K documents in trectext format. There are 57 unique topics(queries) that have corresponding qrel files in the category that we will work with - "None". See the mapping file:

`/local/course/ir/lab3/mapping_qrels_topics.txt`

Qrel files have relevance judgements [0,3].

The experimental setup

We will build two indices by specifying our own index parameters and by using IndriBuildIndex.

The indices are built on two corpora. The first is the original corpus - korpus-utf-8.xml and second one in the decomposed compounds - korpus-utf-8.xmlposlemsgm. If needed, look up the instructions of the Lab2. For building indices call:

```
IndriBuildIndex <parameter_file>
```

The corpora are in:

`/local/course/ir/data/MedEvalTK/qpa1003/`

Store your two indices separately, we will need them later in the lab.

Write a function that converts 50 topics which have corresponding qrel files into a query parameter file. The most unassuming approach - query should simply contain all words in TITLE. A better approach is to exclude function words, and keep content words only. You can use combine belief operator. The output will serve as a query parameter file for the next step and should look like this:

```
<parameters>
<query>
<number>1</number>
<text>#combine(fettsnål kosts inverkan på LDL och HDL)</text>
</query>
<query>
<number>2</number>
<text>#combine(försiktighet vid behandling med erytromycin under graviditet)</text>
</query>

...
</parameters>
```

The topics are in:

```
/local/course/ir/data/MedEvalTK/Topics/MedTopics.txt
```

Run your query parameter file of 50 queries on both indices. For this lab we will limit the number of returned results to 100, by setting -count=100. Store both resulting files for evaluation. You can call IndriRunQuery:

```
IndriRunQuery <parameter_file> > <result_file>
```

Evaluation measures

Now we will evaluate the effect of de-compounding on Swedish medical language test collection MedEval. The goal of an evaluation metric is to measure the quality of a particular ranking of known relevant/non-relevant documents.

P@k As the first measure that measures how many relevant documents are at the top k retrieved documents, we will use Precision @ rank K. Note that P@k uses binary relevance judgments - relevant/non-relevant. For that we will regard relevance judgements [1,3] as 1 and [0] as 0.

Implement a function that performs the evaluation for 50 queries @10. Present results for original and decompounded corpus in a table.

Then choose one query, and try out a few values of k (e.g., 1 to 10) and provide a table of results. What values of k seem to perform best?

MAP As the second measure that provides a single figure over the entire batch, we will use MAP. Note that MAP uses binary relevance judgments - relevant/non-relevant. For that we will regard relevance judgements [1,3] as 1 and [0] as 0.

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk})$$

Implement a function that performs the evaluation for mean average precision (for a set of queries is the mean of the average precision scores for each query) according to the procedure described in the book and the lectures. Present results for original and decompounded corpus in a table. Discuss the difference.

Lab report

Report and discuss your results for all evaluation metrics. Hand-in your code with appropriate comments and with short instructions how to run your implementation. Discuss what setting gives better results. Why? Summarise your results in tables.

VG part

Experiment with varying other parameters, like query length and number of returned documents. Visualize (performance vs. parameter) and analyse observed differences. Discuss your results for all evaluation metrics.