Project Information Retrieval

Toon Calders
toon.calders@uantwerpen.be

Deadline: December 20th (for feedback by January 6th), January 20th (final submission)

This project is to be executed in groups of up to 3 students. Groups of 3 students are expected to complete all three project parts. Groups of 2 or less students are expected to execute project parts 1 and 2. To get the data for the project, contact the lecturer to copy from an external hard drive. Send email to make an appointment to acquire the data. The preprocessed version of the complete data collection is about 9Gb. Alternatively you could download your own copy of the raw dataset (not pre-processed) from https://ia600107.us.archive.org/27/items/stackexchange/stackoverflow.com-Posts.7z.

1 Introduction

The goal of this assignment is to use an existing library, we recommend Lucene[1] (Java) or PyLucene[2] (Python), to create a document store with retrieval functionality. The document store you will be using are questions and answers from the well-known StackOverflow website[3]. As a first step, you will create a proof-of-concept application. The development of this proof-of-concept application has as purpose to study the feasibility of an information-retrieval solution. You will be required to thoroughly understand the functionality offered by the search software you use.

Given that the data in the store is intended to demonstrate your project, it is acceptable to reduce the datasets to more manageable proportions. Scaling up to the full data size, however, will be considered a plus for your project. The main core of your application is the search capability, not the interface. You can use a text-based interface.

2 The Dataset

The data to be used in the project consist of a dump of StackOverflow questions and answers. The questions have been filtered to only include those where the tags include either "python" or "c++". The dump has been preprocessed (it was one large .xml file) and split by question. Per question one file was created, containing the question and all of its answers. A prototypical example is the following file:

```
<question>
<Title>What are metaclasses in Python?</Title>
```

[1] https://lucene.apache.org/

[2] https://lucene.apache.org/pylucene/

[3] https://stackoverflow.com/

```
<Body>&lt;p&gt;What are metaclasses and what do we use them for?&lt;/p&gt;&#xA;</Body>
<Tags>python,oop,metaclass,python-datamodel</Tags>
</question>
<answer>
<Body>&lt;p&gt;&lt;em&gt;Note, this answer is for Python 2.x as it was written in 2008,
metaclasses are slightly different in 3.x.&lt;/em&gt;&lt;/p&gt;&#xA;&#xA;&lt;p&gt;
Metaclasses are the secret sauce that make 'class' work. The default metaclass for a new
style object is called 'type'.&lt;/p&gt;&#xA;&#xA;&lt;pre class=&quot;lang-none prettyprint
-override&quot;&gt;&lt;code&gt;class type(object)&#xA;  |  type(object) -&amp;gt; the
object's type&#xA;  |  type(name, bases, dict) -&amp;gt; a new type&#xA;&lt;/code&gt;&lt;
/pre&gt;&#xA;&#xA;&lt;p&gt;Metaclasses take 3 args. '&lt;strong&gt;name&lt;/strong&gt;', '&lt;
strong&gt;bases&lt;/strong&gt;' and '&lt;strong&gt;dict&lt;/strong&gt;'&lt;/p&gt;&#xA;&#xA;
&lt;p&gt;Here is where the secret starts. Look for where name, bases and the dict come from
in this example class definition.&lt;/p&gt;&#xA;&#xA;&lt;pre&gt;&lt;code&gt;class
ThisIsTheName(Bases, Are, Here):&#xA;    All_the_code_here&#xA;    def doesIs(create, a):&#xA;
... [text cut here ] ...
</answer>
<answer>
<Body>&lt;p&gt;I think the ONLamp introduction to metaclass programming is well written and gives
a really good introduction to the topic despite being several years old already.&lt;/p&gt;&#xA;&#
&lt;p&gt;&lt;a href=&quot;http://www.onlamp.com/pub/a/python/2003/04/17/metaclasses.html&quot;
rel=&quot;noreferrer&quot;&gt;http://www.onlamp.com/pub/a/python/2003/04/17/metaclasses.html
... [text cut here ] ...
</answer>
<answer>
<Body>&lt;p&gt;One use for metaclasses is adding new properties and methods to an instance
automatically.&lt;/p&gt;&#xA;&#xA;&lt;p&gt;For example, if you look at &lt;a href=&quot;
http://docs.djangoproject.com/en/dev/topics/db/models/&quot; rel=&quot;noreferrer&quot;&gt;
Django models&lt;/a&gt;, their definition looks a bit confusing. It looks as if you are only
defining class properties:&lt;/p&gt;&#xA;&#xA;&lt;pre&gt;&lt;code&gt;class Person(models.Model):&
first_name = models.CharField(max_length=30)&#xA;    last_name = models.CharField(max_length=30)&
... [text cut here ] ...
</answer>
```

Every file starts with one question (included within `<question>` and `</question>`) and zero or more answers, each included in `<answer>` and `</answer>`. Questions have sub-element title, body, and tags, of the answers only the body has been preserved. In the body text special symbols such as < and > have been encoded using `&[code];`. For instance, `<p>` is encoded as `&lt;p&gt;`. Furthermore, there is one special tag `<code>` that the users of StackOverflow can use to typeset code fragments. Special symbols and codes can be ignored, yet it is conceivable that better performance will be achieved if they are taken into account.

3 Assignment

The assignment consists of three parts. The first two parts will be graded together, the third part separately. The parts can be submitted one by one for feedback that can be incorporated into the next parts. All page limits are guidelines only, but avoid lengthy discussions of standard components; copying or repeating the content of lecture slides or course books does not show your mastery of the subjects.

1. Study the feasibility of using Lucene to add retrieval capabilities to the StackOverflow repository provided to you. Describe the functionality offered by Lucene, such as the types of indices that are included, the different score models, etc. How does Lucene store the index, does it have spell correction? Include a benchmarking study of the performance of the retrieval performance. Manually label some documents. One acceptable approach is to use the titles of th questions as the query, and only index the remaining parts of the documents. In this way a ground truth can be generated rather easily. This first part is expected to be about 10 pages.

2. Study the impact on retrieval performance when implementing one or more of the following techniques: (a) Rocchio algorithm for relevance feedback, (b) query expansion based on either pseudo-relevance feedback, or on manually or automatically generated thesaurus-based query expansion, or (c) Latent Semantic Indexing. This part is expected to be around 5 pages and should discuss the impact on precision and recall of the applied techniques. The second part should not be stand-alone, it is an addendum to the first part.

3. Last but not least: implement locality sensitive hashing for finding near-duplicate questions (e.g., reposts). This part is separate from the repository indexed by Lucene. You can use existing libraries for implementing LSH, but it is critical to show in your report that you have a very good understanding of LSH and that you apply it appropriately (use a suitable similarity measure, a fitting family of hash functions) and provide a correct analysis of the guarantees provided by the theoretical analysis as seen during the lectures. This report is expected to be about 5 pages.

4 Deliverables

1. The code of your project. Do not include bulky software libraries or large datasets in emails. The preferred way to share code is via a link to a publicly available GitHub repository. Include the link in your report.

2. The report in pdf format, to be submitted via BlackBoard. Do not submit zip-files, word documents etc., only the submission of a single pdf file will be accepted.

A Note on Plagiarism

There is absolutely nothing wrong with using existing materials, you will even be commended for not reinventing the wheel, as long as you are not violating the copyright of other authors. Nevertheless, it is expected from you to clearly indicate whenever you used material that was not created by yourself. Clearly indicate in your submissions which parts constitute original work, which parts are taken from other works, and which parts were adapted from external sources. These sources have to be properly acknowledged in all your submissions. Concretely, this means at least the following guidelines are observed:

- Papers, books, webpages, blogs, etc. that were inspected while making the assignment will be referenced in a separate section "References". Citations to these materials are included in the text where appropriate.

- Text fragments exceeding one sentence that are copied from other sources are clearly marked as such. You could for instance include quoted text, definitions, etc. in italics, followed by a reference. An example of how to do this: Bela Gip (2014) defines plagiarism as "*The use* *of ideas, concepts, words, or structures without appropriately acknowledging the source to benefit in a setting where originality is expected*"

References: (at the end of the document) Gipp, Bela. "Citation-based plagiarism detection." *Citation-based plagiarism detection.* Springer Vieweg, Wiesbaden, 2014. 57-88.

- When using code from other sources, indicate so in the report, and in the source code. This could for instance be done by adding a comment with a reference to the source of the function for each function that was copied from another source. It is recommended to include a separate folder "sources" in your GitHub repository with the original files from other authors that you used. Include source in the message of your commits.