TÉCNICO LISBOA

Gestão e Tratamento de Informação

2nd Project

Deadline at 21 Nov. 2014 :: Online submission at IST/Fénix

The file at *http://feeds.dn.pt/DN-Ultimas* contains the RSS newsfeed for a well known Portuguese newspaper. The following exercises will be based on the information contained in this feed and in other similar data sources. Thus, you can test your solutions using the above newsfeed, other newsfeeds of the same newspaper, or any other feeds (e.g., from other newspapers) that have the same structure.

Exercise 1

1.1 - Write an XQuery user-defined function capable of producing a well-formed XML document encoding information about the news items in the RSS feed. The produced XML document should contain, for each news item, the title, the description, and the link to the original news item. The news items should be grouped by category and sorted in descending order by publication date. The output format of the function is shown in Fig. 1.

```
<news>
   <category name="Ciência”>
      <item date="19-10-2014" title="Man bites dog" link=”http:...”>
         A man bites a dog, after a serious discussion. Two days
later, both fell in love and decided to get married. “The Catholic
Church is opposed”, said Bishop Salid Salalasa, from the ...
      </item>
      <!-- remaining news items in the category -->
   </category>
   <!-- remaining categories in the list of news -->
</news>
```

Figure 1: Example of the format produced as output for question 1.1.

1.2 – Create an XQuery user-defined function that addresses the following two issues:

a. The function should take as input the XML document containing parliamentary transcripts used in MP1, the XML output of the function defined in Question 1.1, and a parameter *N*;
b. The function should produce, as output, a well-formed XML document containing, for each parliamentary session, all the news items related to the speeches it contains. We say that a news item is related to a set of speeches if they have at least *N*% of the individual words in common, in either the title or description. Case should be ignored.

The output format for this user-defined function is exemplified in Figure 2.

```
<related-news>
   <session date=”2014-03-01”>
      <item date=”19-10-2014” title=”Man bites dog” />
      <!-- remaining related news items -->
   </session>
   <!-- remaining sessions -->
</news>
```

Figure 2: Example of the format produced as output for question 1.2.

1.3 - Write an XQuery user-defined function similar to that of Question 1.2, but that, in this case, considers a news item to be related to the speeches if its description includes the name of any of the politicians that participated in the session. You should consider that the name matches if at least two words of the name match (e.g., the name *Pedro Passos Coelho* should match with the string *Passos Coelho*).

Exercise 2

Consider the two HTML fragments from the figures shown below, which correspond to web pages encoding information about news items.

```
<div>
    <span>
        <p>Title: Man bites dog</p>
        <p>Section: Society</p>
        <strong>Summary: Man and dog have a fight.</strong>
        <strong>By: John Slashstory</strong>
        <em>Photo by: Michel Lasucette</em>
    </span>
    <span>
        <p>Title: King of Spain injured by carnivorous plant</p>
        <p>Section: International</p>
         <strong>Summary: Unexpected accident during an elephant
safari.</strong>
        <em>Photo by: Michel Lasucette</em>
    </span>
</div>
```

Figure 3 : An example fragment from an HTML web page.

```
<div>
    <span>
        <p>Title: Masterchef won with plate of cockroaches</p>
        <p>Section: Art</p>
         <strong>Summary: The jury was overwhelmed by the contestant's
self-confidence.</strong>
        <strong>By: Livingstone Dreadlock</strong>
        <em>Photo by: Tonton Laluche</em>
    </span>
</div>
```

Figure 4: Another example fragment from an HTML page.

2.1 - Show the wrapper that a Web data extraction system, following the approach of the *RoadRunner* system, would produce, in order to extract information from the two HTML fragments given as examples. Explain how the wrapper would be obtained, through the application of the *ACME* algorithm.

Exercise 3

Figure 5 shows an example of a newsfeed, rendered using non-formated text. Each feed can contain a date, time, title, newspaper name and section. Develop an HMM to process such a list of feeds and extract the day, month, title, and name of the newspaper.

22-10-2014 , The Daily Disaster – Doctors show that eating cement helps digestion
23-10-2014 15:30 Unkown virus causes people to be less stupid – The Global Newspaper: Science
23-10-2014 16:48 Cure for the unknown virus found in German laboratory, The Global Newspaper: Science
24-10-2014 Science News Montlhy – "This is not rocket science", states rocket scientist
25-10-2014 10:22 "We are ready for an alien invasion", says Minister of Defense, The Global Newspaper: Politics

Figure 5: Example of a newsfeed.

3.1 – Present and explain the proposed HMM. You should draw a graphical representation of the model and explain clearly what the states represent and what are the observation symbols. To simplify, ignore words with less than 4 letters, and reduce all words to the lower-cased singular form (e.g. "*rockets*" will be the same as "*rocket*").

3.2 – Based on the sentences shown in Fig. 5, derive the required probabilities that define your HMM. When computing the probabilities, you should use Laplace smoothing.

3.3 – Using the HMM you just built, compute the most likely sequence of states for the following sentence:

25-10-2014 11:30 The Daily Surprise: Science – Alien hamburguer causes disaster

Exercise 4

Consider the XML document containing parliamentary transcripts used in MP1.

4.1 – Write an XQuery user-defined function that takes as input the set of speeches and returns an XML document containing the number of speeches for each political party and, for each individual word in a speech, the number of speeches where it occurs, in each party. The returned document should be in the format shown in Fig. 6.

```
<model>
<party name="PSD" size="number-of-speeches-in-party"/>
<party name="PS" size="number-of-speeches-in-party"/>
...
<word token="blah">
   <party name="PSD">number-of-speeches-with-word-in-party</party>
   <party name="PS">number-of-speeches-with-word-in-party</party>
   ...
</word>
<word token="blahblah">
   <party name="PSD">number-of-speeches-with-word-in-party</party>
   ...
</word>
</model>
```

Figure 6: XML document containing party and word statistics.

4.2 – Write an XQuery user-defined function that implements a Naïve Bayes classifier. The function should take as input the XML model generated in Exercise 4.1 and a single speech from the input file of Exercise 4.1. The function should return the party of the politician that made the speech. Your classifier should use Laplace smoothing when computing the probabilities.

Exercise 5

Assume the extractor that uses the HMM defined in the Exercise 3 outputs, as a result, an XML document in the format exemplified in Fig. 7.

```
<news>
   <item newspaper="The Global Newspaper">
      <date>
         <day>23</day>
         <month>10</month>
      </date>
      <title>Unkown virus causes people to be less stupid</title>
   </item>
   <!-- remaining news items -->
</news>
```

Figure 7: Example of the format produced as output by the HMM.

5.1 - Write a mediator function in XQuery that integrates the results from the datasets resulting from Exercise 1 and Exercise 3. The output format should be that of Exercise 1.

5.2 – Write an XQuery expression for obtaining the total number of news items published per month. Write two versions of the query:

1. An expression that uses the mediator function from Exercise 5.1;
2. An expression that uses the original XML datasets that are involved in the mediator function (i.e., unfold the previous query).

Submitting the project

The solutions to the project should be submitted through the *Fénix* system, in the form of a .zip file containing a PDF report with the solutions for the exercises, as well as individual text files with the solutions for each of the exercises (i.e., documents with the XML, XSD, XSLT or XPath/XQuery code).

In the course Webpage, you can find a Microsoft word template for the project report.

In the theoretical class following the electronic submission, a printed copy of the report, with the solutions for each exercise, should also be delivered.

We will not accept deliveries through e-mail, with reports not conforming to the supplied template, or without the text files with the solutions for the exercises.

Good luck!