**The University of Hong Kong**
**Faculty of Engineering**

**Department of Computer Science**

COMP7306, Web technologies

Reference solution

## 1 The Web: A Variety of Technologies (8 points)

1. (2 points)

   a) `.co.cc` is a hierarchy of the domain name system; visibly, probably because of the leniency of the owners of this domain name hierarchy, Web sites under this hierarchy hosted malicious software. Google decided to remove all Web sites with a URL in this domain hierarchy from its index.

   b) Baidu is the main search engine used in China, while Bing is the challenger at the global level of the leading search engine Google. Web search engine technologies (text preprocessing in particular) heavily depend on the language used. It seems Baidu has deemed that by using Microsoft's technologies on English language Web sites, possibly by just calling a Web service on Bing's side, the quality of their results on English language queries would improve.

2. (4 points) 0pt or 0.5pt, nothing in between

   a) Absolute units may mean very different things in terms of screen space: 1 cm is very small on a large flatscreen, quite large on a smartphone screen. Besides, current technology does not provide a reliable pixel-to-cm conversion at the level of the browser, so these units are most of th time nonsensical.

   b) EcmaScript

   c) It is a Web server software, that responds to Web client requests by providing Web pages or Web resources, possibly interacting with server-side programs that generate these.

   d) Separating HTML and CSS code in two separate files help with the maintenance of the code (they can be independently edited by different developers, for instance). The ability of reusing the same stylesheet from one page to another provides consistent styling without code duplication, and reduces the amount of downloaded content since this stylesheet is only downloaded once for all pages that use it.

   e) Google Chrome

   f) Yahoostopped developing their own search engine and uses Bing on their Web site instead. This agreement is not true for Yahoo Japan, a separate entity, that opted for Google. The counterpart is Bing using Yahooonline advertisement services.

   g) The semi-structured data model uses flexible typing (from no typing at all, to strict typing). Semi-structured data are self-describing. Semi-structured data have graph or tree shape, compared to tables for the relational model. Semi-structured data have standardized serialization.

h) Model, View, Controller. Separation of the program code into description of the business logic (model), of the interface (view), and on the interaction code between model, view, and the user (controller).

3. (2 points, flexible grading, 1pt for each of the “good” and “bad” sides, with -0.5pt for incorrect arguments) What browsers do you develop for?

   Good: My code is geared towards browsers respecting established Web standards (HTML 4.01, CSS 2). In addition, I test that Web sites I develop are still usable (and, if possible, with the same appearance) in all browsers having more than 1% of the market (for instance, IE6) by testing them thoroughly on each different browser. Finally, when I use some cutting edge feature, I pay extra attention to the behavior of all commonly used browsers.

   Bad: a list of browsers such as "IE9, IE10, Chrome".

## 2 Attractions of European cities (12 points)

1. (2 points: -.5 to -1 point for each different class of errors)

   We choose to use morphological stemming here, and a reasonable list of grammatical stop words.

| | |
|---|---|
| $d_1$ | eiffel, tower, famous, monument, paris |
| $d_2$ | paris, miss, louvre, museum |
| $d_3$ | anatomy, museum, pavia, interest |
| $d_4$ | ashmolean, museum, oxford, old, museum, europe |
| $d_5$ | paris, numerous, museum, orsay, museum |
| $d_6$ | jewel, tower, london, famous |
| $d_7$ | luxembourg, garden, paris, luxembourg |

2. (2 points: -1 point for each different class of errors)

| | |
|---|---|
| famous | $(d_6, \frac{1}{4}\log(\frac{7}{2})), (d_1, \frac{1}{5}\log(\frac{7}{2}))$ |
| museum | $(d_5, \frac{2}{5}\log(\frac{7}{4})), (d_4, \frac{1}{3}\log(\frac{7}{4})), (d_2, \frac{1}{4}\log(\frac{7}{4})), (d_3, \frac{1}{4}\log(\frac{7}{4}))$ |
| paris | $(d_2, \frac{1}{4}\log(\frac{7}{4})), (d_7, \frac{1}{4}\log(\frac{7}{4})), (d_1, \frac{1}{5}\log(\frac{7}{4})), (d_5, \frac{1}{5}\log(\frac{7}{4}))$ |
| tower | $(d_6, \frac{1}{4}\log(\frac{7}{2})), (d_1, \frac{1}{5}\log(\frac{7}{2}))$ |

3. (3 points: -1 point for each different class of errors)

```
function map(id, document)
  foreach distinct token in document
    stem := morphologicalStemming(term)
    if (stem not in stopWords)
      tf := count(term, document)/length(document)
      output (term, (id, tf))

function reduce(term, list)
  foreach (id, tf) in list
    output (term, tf * log(nbDocuments/length(list)))
```

One assumes here that nbDocuments is a globally known constant. Other approaches are of course possible.

4. (1 point) A possibility is to extract city names from Wikipedia (or some knowledge base) and to store them in a trie. This provides efficient extraction of entity from a corpus, with a linear-time runtime in the size of the corpus. An inconvenient concerns homonyms that may make us believe that a name (e.g., Luxembourg) is a city when it is used with another meaning.

5. (2 points: 1pt for the technique, 1pt for the evaluation) With the help of a part-of-speech annotation and a syntax parser, we propose to combine the following two techniques:

   - Whenever there is a pattern “in CityName” with CityName an identified city name (or just a name starting with a capital), look for all noun phrases NP of more than one word in the same sentence that precede this expression (without article), and extract the fact locatedIn(NP, CityName). When going back in the sentence, we never cross another “in”.
   - Whenever a noun phrase is of the form “N of CityName” with N a noun, extract the fact locatedIn(N of CityName, CityName)

   These two techniques combined extract the following facts:

   - locatedIn(Eiffel tower, Paris)
   - locatedIn(most famous monument, Paris)
   - locatedIn(Anatomy museum, Pavia)
   - locatedIn(Ashmolean museum, Oxford)
   - locatedIn(Tower of London, London)
   - locatedIn(Luxembourg gardens, Paris)

   The precision is $\frac{5}{6}$ and the recall $\frac{5}{8}$, beyond the required thresholds.

6. (2 points) We can have “Tower of London”, “Eiffel tower” in the class “Monument” (through rdf:type), itself a rdfs:subClassOf of “Attraction”; “Luxembourg gardens” is a rdf:type of “Attraction”; the different museums in “Museum” another subclass of “Attraction”, and all cities in an independent “City” class. “:locatedIn” relations are present between entities as indicated in the previous question.

END OF SOLUTION