UCT CSC303 2004 :: XML/IR :: Exam [25 marks]

Answer Questions 1 AND 2.

Then answer 3 questions from among Questions 3-6.

(Questions 3 and 4 are optional – Sonia will provide 2 more questions).

Question 2 : Information Retrieval [10]

1. When building an IR system to ignore all tags and index just the textual data in a collection of XML files, which parser family is more efficient: SAX or DOM? Explain why. [3]

SAX [1]

SAX does not build the entire document tree in memory and therefore is more efficient in terms of space and time. In addition, the API will allow tags to be simply ignored as the user can specify only a handler for text nodes. [2]

2. In preprocessing data, terms are often stemmed and stopped. What are stemming and stopping? How does stemming affect recall and precision? [3]

Stemming converts terms to a canonical form, to ignore prefixes/suffixes/etc.[1]

Stopping ignores common words.[1]

Stemming increases recall but decreases precision.[1]

3. What is the difference between term frequency and document frequency? [2]

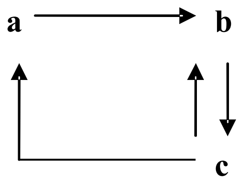
Term frequency is the number of times a term appears in a document [1] while document frequency is the number of documents a term occurs in. [1]

4. List 2 advantages of the Latent Semantic Analysis algorithm for IR. [2]

- *Smaller vectors – faster query matching.*
- *Smaller term-document space – less storage required.*
- *Automatic clustering of documents based on mathematical similarity (basis vector calculations).*
- *Elimination of "noise" in document collection.*

Question 3 : PageRank [10]

1. Apply the PageRank algorithm to the following link graph [6].



Start with equal ranks of 1/3 each. Stop after 4 iterations, or when c=5/12. Show all calculations.

(Hint: Remember that r[i](n) = Sum of r[j](n-1)/N[j] over all j in B[i])

	N	*B*	*R[0]*	*R[1]*	*R[2]*	*R[3]*	*R[4]*
A	*1*	*C*	*1/3*	*1/6*	*1/6*	*1/4*	*1/6*
B	*1*	*A* *C*	*1/3*	*1/2*	*1/3*	*5/12*	*5/12*
C	*2*	*B*	*1/3*	*1/3*	*1/2*	*1/3*	*5/12*

One mark for N, one mark for B, one mark for each correct iteration

2. Assuming that we had a collection of newspaper articles from the Cape Argus in digital form, why will the PageRank algorithm not be suitable for IR purposes? [2]

PageRank can only rank documents with links to other documents – standalone newspaper articles without links cannot be ranked. [2]

3. How does the HITS algorithm differ fundamentally from the PageRank algorithm? [2]

HITS assigns importance to, and calculates ranking on the basis of, both pages with lots of links to them (authorities) and pages with lots of links to other pages (hubs) – PageRank only considers authorities. [2]

Question 4 : XML / XSLT [10]

1. Write an XSLT template to transform

```
<university>
<class>
<lecturer>EvilTeacher</lecturer>
<student>JoeStudent</student>
</class>
</university>
```

into:

```
<dungeon>
<torture>
<torturer>EvilTeacher</torturer>
<tortured>JoeStudent</tortured>
</torture>
</dungeon>
```

Assume that the values “Hussein” and “Alapan” may differ from one document to another. Assume the source namespace prefix is “source” and the destination prefix is “dest”. Assume that the number of <class> tags is unbounded in its defining XML Schema. Use the following as a starting point. [4]

```
<xslt:template match=″source:university‶>
. . .
</xslt:template>
```

```
<xslt:template match=”source:university”>
  <dest:dungeon>
  <xslt:for-each select=”source:class”>
    <dest:torture>
      <dest:torturer>
        <xslt:value-of select=”source:lecturer”/>
      </dest:torturer>
      <dest:tortured>
        <xslt:value-of select=”source:student”/>
      </dest:tortured>
    </dest:torture>
  </xslt:for-each>
  </dest:dungeon>
</xslt:template>
```

Minus one mark for each major error.

2. Write an XML Schema type definition corresponding to the contents of the university node. [6]

```
  <complexType>
    <sequence>
      <element name="class" minOccurs="0" maxOccurs="unbounded">
        <complexType>
          <sequence>
            <element name="lecturer" type="string"/>
            <element name="student" type="string"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
```

Minus one mark for each major error.