Information Retrieval (CS60092)
Mid-semester examination, Autumn 2013 – 2014

Time: 2 hours, **Full Marks:** 50

Attempt all questions.
Use of scientific calculator is allowed.
State any assumptions made clearly.

Q. 1>(a) Assume the word *cricket* has the following postings list:
cricket: 274, 287, 288, 300, 420
What is the variable-byte gap encoding (binary) for the above sequence?
How many bytes are required? How many bytes would have been required if the original postings list was stored (assume that one only needs to store the doc-ids)?

(b) Decode the following variable byte gap encoding and report the actual postings list:
10000101 10000010 10000100 10000011 10000001

(c) What are the largest gaps that can be encoded using one and two bytes?

(d) Can we derive Heaps' law from Zipf's law? Justify your answer. [**5 + 2 + 2 + 1 = 10]**

Q. 2>Consider the four toy documents below as your corpus:
doc-1: *cricket is a great game*
doc-2: *a game is a good good sport*
doc-3: *all sport and cricket are great great great*
doc-4: *cricket is a sport and a game*
Assume the following stop list: *a, an, the, is, are, and, of, in, any, all, every, each*
We use a vector space model. Assume that the index of a term is determined by its order of encounter in the log (*cricket* gets position 1 in the vector). Using **simple TF-IDF** (use raw TF, multiplied by IDF(t) = $\log_{10}(N/\text{DF}(t))$) for **term weighting for both queries and documents**. Do **not** apply **any** normalization on the document weights. The issued query is *the great game of cricket.* Use the same TF-IDF vectors for both parts below. You are encouraged to neatly tabulate all values used, like term frequencies and term weights.

(a) Rank the documents using the simple overlap score. Show all steps of the computation.

(b) Rank the documents using the cosine similarity. Show all steps of the computation. **[5 + 5 = 10]**

Q. 3>A retrieval system produced the following ranked list (only relevance judgments are shown) in response to a query: 1, 1, 0, 0, 1, 0, 1, 1, 0, 0. Assume the number of relevant documents in the collection to be nine.

(a) Compute the F-score for the query.

(b) Plot the precision-recall curve for this query.

(c) Plot the interpolated precision-recall curve for this query.

(d) Now assume the same system produces the list: 0, 0, 1, 1, 1, 0, 1, 0, 1, 0 for a second query. Assume the number of relevant documents for this query to be seven. Compute mean R-precision.

(e) Compute MAP for this binary relevance system. **[1 + 3 + 3 + 1.5 + 1.5 = 10]**

Q. 4>(a) Provide the tree representation of the following XML document:

```
<paper>
<frontmatter>
<author>
<firstname>Humpty</firstname>
<lastname>Dumpty</lastname>
</author>
<title>The Great Egg</title>
<abstract>This is a true story.</abstract>
<keyword>egg</keyword>
<keyword>fall</keyword>
</frontmatter>
<body>
<section number="1">
<title>Introduction</title>
<subsection>The tree was high.</subsection>
</section>
<section number="2">
<title>Method</title>
<subsection>I had a great fall.</subsection>
</section>
<section number="3">
<title>Conclusion</title>
<subsection>It was an XML tree.</subsection>
</section>
</body>
</paper>
```

(b) Expand NEXI and INEX. **[8 + 2 = 10]**

Q. 5>(a) The postings lists for *cricket, football, swimming* and *tennis* contain 35, 77, 12 and 18 documents respectively. What is the best order for processing the query that is a conjunction of all the four terms? Use parentheses to show the order clearly.

(b) If the document frequencies are assumed to be *w, x, y* and *z*, and the corpus has *N* documents in all, what is the time complexity for processing the query *(cricket OR football) AND NOT (swimming OR tennis))*?

(c) Which of stemming and lemmatization can be expected to produce a larger lexicon? Why?

(d) Name three popular stemming algorithms.

(e) Can stemming decrease precision? Justify your answer.

(f) Assume the following postings lists for *cake* and *pie*:
cake: 3, 5, 7, 10, 13, 18, 39, 53 [Skips inserted from 3 to 10 and 10 to 39]
pie: 2, 12, 15, 39, 43, 45, 49 [Skips from 2 to 39 and 39 to 49]
List the pairwise doc-id comparisons that are avoided due to the presence of skip pointers.

(g) Enumerate the permuterm vocabulary for *cake*.

(h) Without computations, state the Levenshtein distance between *calm* and *slam*.

(i) What is the Jaccard coefficient between *calm* and *clam* using character bigrams?

(j) Name one real application of *k*-gram character indexes. **[1 x 10 = 10]**
