COMP 336 Information Retrieval **Quiz 1, Spring 2000** **Date: March 3, 2000**

Name: _____ Student ID: _____

1. List one major advantage and one major disadvantage of the vector space model. [10]

   Advantage: one of the following: allows weighting, allows ranking, allows flexibility in matching, ...

   Disadvantage: one of the following: computation intensive, difficult to control what go into the result list, ...

2. Given an inverted file, we assume that the index file, which contains the vocabulary of the document set, is stored in main memory, and that postings lists are sorted by document Ids and on average each list occupies 2 disk pages. Now you have a new document containing 100 unique terms to be inserted into the inverted file and all of the 100 terms already exist in the index file.

   a) How many disk accesses are required for the insertion, assuming that each postings list still occupies 2 disk pages after insertion? [15]

   100 x 2 = 200 (read one page, insert new entry, write it back)
   100 x (2+1) = 300 (read entire postings list (both pages), insert new entry, write the modified page back)
   100 x (2+2) = 400 (read entire postings list (both pages), insert new entry, write both pages back in the case that insertion into the first page cause an overflow into the second page)

   b) If you have three new documents to be inserted, explain why batching the three insertions together ***might*** be less expensive than if the three documents are inserted individually. [15]

   There is a good chance that the documents share common terms. For each common term, you need to read and write the postings list once if the postings entries fall on the same disk page

3. Given a query "A AND B" what is the major difference between the Extended Boolean Model (the basic one, not the p-norm model) and a sequential combination of Boolean Model followed by ranking (as in Altavista)? [30]

   Extended Boolean model: no guarantee that high-weight documents contain both A and B, although the model strongly favors documents which contain both keywords.

   Boolean followed by ranking: every document has to contain both A and B.

4. A document contains, and only contains, the phrase "to be or not to be". Suppose every word is indexed. The document collection contains 10,000 documents and every word has equal document frequency of 1,000. What is the weight of each term according to the *tf*×*idf* weighting formula using a ***normalized*** term weight? [30]

   tf(to) = 2 idf(to) = idf(be) = idf(or) = idf(not) = log(10000/1000) = log(10)
   tf(be) = 2
   tf(or) = 1
   tf(not) = 1

   Normalize tf only: max(tf) = 2
   wt(to) = (2/2) * log(10) = log(10)
   wt(be) = (2/2) * log(10) = log(10)
   wt(or) = (1/2) * log(10) = 0.5 log(10)
   wt(or) = (1/2) * log(10) = 0.5 log(10)

If you want to normalize idf, then you need to know the maximum idf value in the entire vocabulary, which is not given in the question. Notice that you cannot normalize idf with the highest idf value ***within*** the document. You may assume that the highest idf is log(10000) but if you take the highest idf in the given document (i.e., log(10)) as the normalization factor, points will be deducted.