Pure negation query in lucene

In many information-retrieval system, you can use queries like “term1 AND (NOT term2)” but you cannot use queries like “NOT term2” on their own (e.g. to get only documents that do not contain term2). At least the system returns no result even if some documents do not contain term2. This is because:

- Most of these system uses inverted indices, which are essentially “term => document list” mappings. There is no way to reconstruct a list of documents for “NOT term2” from such a mapping.
- The system can generate a set of all document and then subtract those that contains term2 from the set. This can be resource-intensive (e.g. needs O(doc count) storage and processing time) and can be misused by users.

Despite the difficulties, users of some system might need this kind of query. In Lucene, there are two ways to support it:

- When indexing, create a field (e.g. “exists”) with a constant value across documents (e.g. “true”). Then rewrite the query “NOT term2” into “+exists:true -term2”.
- Combine MatchAllDocsQuery, which matches all document, with BooleanQuery. The code will look like this:

```
// Create a query that matches something like “everything AND NOT term2”
MatchAllDocsQuery everyDocClause = new MatchAllDocsQuery();
TermQuery termClause = new TermQuery(new Term(“text”, “term2”));
BooleanQuery query = new BooleanQuery();
query.add(everyDocClause, BooleanClause.Occur.MUST);
query.add(termClause, BooleanClause.Occur.MUST_NOT);
…
IndexSearcher searcher = new IndexSearcher(“/path/to/index”);
Hits hits = searcher.search(query);
// hits contains only documents that do not contain term2
```

Revision #1
Created Fri, Aug 29, 2008 5:30 PM by Chan, Wing Kai
Updated Fri, Aug 29, 2008 5:30 PM by Chan, Wing Kai