**Sanjana Sunil Agarwal**

# SEARCH ASSIGNMENT #1

For the file GenerateIndex.java,
**[To run the code, you need to pass three command line arguments.**
**args[0]: The location of the corpus on the machine you are running the code on.**
**args[1]: The location where you want to store the indexed files.**
**args[2]: The analyzer the you want to use.**
**0: KeywordAnalyzer ; 1: SimpleAnalyzer ; 2: StopwordAnalyzer ; 3: Standard Analyzer ; Any other number is default value of StandardAnalyzer ]**

1 a) The total number of docs in the corpus is 84474.
b) Different fields are treated with different java classes. There are certain fields in a document, that shouldn't be allowed to tokenize and provide results of exact match. Such fields are like docid, timestamp, urls, etc. Thus, in such cases we use the StringField. It should not have any tokenization or analysis/filters applied.

On the other hand, fields like a description field, where we need to count the occurrences of a word or analyze other statistics from the document, in that case we should use a TextField. Thus, TextFields generally have a tokenizer and a text analysis attached where we need not return the exact matches and the indexed content is broken down into smaller tokens, each token can be matched separately.
**Reference:** https://goo.gl/0BcC0S

For the file IndexComparison.java,
**[To run the code, you need to pass one command line argument.**
**args[0]: The location of the folder consisting of the indexed files.]**

2) We generate the following table after comparing the indices,

| Analyzer | Tokenization applied? | How many tokens are there for this field? | Stemming applied? | Stop words removed? | How many terms are there in the dictionary? |
|---|---|---|---|---|---|
| Keyword Analyzer | No, does not split the text. | 84474 | No | No | 84043 |
| Simple Analyzer | Yes | 37330193 | No | No | 169981 |
| Stop Analyzer | Yes | 26216524 | No | Yes | 169948 |
| Standard Analyzer | Yes | 26649729 | No | Yes | 233384 |