Assignment-1

Q1. Why different fields use different kinds of java class? i.e., StringField and TextField are used in this example, why?

- There is a distinction in the way "StringField" and "TextField" classes index the data given it based on whether we want to index whole of data given as one or tokenize it first.

- We use "StringField" class where we want to create index and tokenize a field as whole. That means taking into consideration whole the entire field value and indexing as a single token.
  Example: For field like <DOCNO>, <BYLINE>, <DATELINE> and <HEAD> where we have very little data to index in each field, we prefer to use StringField class. Also, this is useful where we have terms like "New York" or "ABC-001", etc. to be indexed without breaking the term vector or access them using sorted file cache.

- We use "TextField" class where we want it to create index and tokenize a field, without considering term vectors.
  Example: For a field like <TEXT> tag that contains the bulk of a content this class is useful.

Q2. Table of Observation

Analyzer	Tokenization applied?	How many tokens are there for the corpus?	Stemming applied?	Stop words removed?	How many terms are there in the dictionary?	Other observations?
KeywordAnalyzer	No	84474	No	No	84061	Number of tokens is equal to number of documents. Each doc considered as a long token to index
SimpleAnalyzer	Yes	37324636	No	No	170023	Performs same as StopAnalyzer except this

						does not remove stopword
StopAnalyzer	Yes	26212591	No	Yes	169990	Took least amount of time for indexing
StandardAnalyzer	Yes	76184842	No	Yes	233439	Uses stopwords and simple tokenizing