68_Adnan Shaikh

Boolean Retrieval Model

In [70]:

```
from nltk.book import *
import numpy as np
import pandas as pd
from nltk.tokenize import word_tokenize
from essential_generators import DocumentGenerator
```

In [119]:

```
class Boolean_Retrieval:

    def __init__(self,documents):

        self.documents = []
        self.tokenized_document = {}
        self._total_documents = 0
        self.inverted_index = {}
        self.add_document(documents)

    def add_document(self,documents):
        assert type(documents) == list or type(documents) == str, "Type must string or list of strings"
        if type(documents) == str:
            documents = [documents]

        for document in documents:
            self._total_documents += 1
            self.documents.append(document)
            self.tokenized_document[self._total_documents] = word_tokenize(document)
            self._create_inverted_index()


    def _create_inverted_index(self):

        for word in self.tokenized_document[self._total_documents]:
            if word in self.inverted_index:
                self.inverted_index[word].add(self._total_documents)
            else:
                self.inverted_index[word] = set([self._total_documents])


    def boolean_query(self,query):
        tokenized_words = word_tokenize(query.replace("^"," "))
        documents = None
        for word in tokenized_words:
            if word in self.inverted_index:
                if documents:
                    documents.intersection(set(self.inverted_index[word]))
                else:
                    documents = set(self.inverted_index[word])

        return documents,tokenized_words
```

In [120]:

```
gen = DocumentGenerator()
bl = Boolean_Retrieval([gen.paragraph(100,1000),gen.paragraph(100,1000),gen.paragraph(100,100)])
```

In [121]:

```
bl.boolean_query("european^district")
```

Out[121]:

```
({1, 2}, ['european', 'district'])
```

In [122]:

```
bl.inverted_index["european"]
```

Out[122]:

```
{1, 2}
```

In [123]:

```
bl.inverted_index["district"]
```

Out[123]:

```
{1, 2}
```