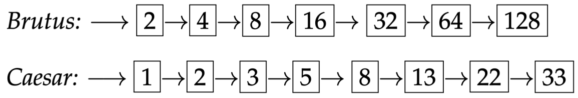
CSC 785: Information storage & retrieval

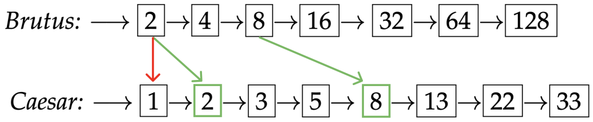
Test 05

Professor: KC Santosh, Ph.D
Jose David Cortes

1. Explain boolean query processing. Let us take an "INTERSECT" operation in a query: *Brutus AND Caeser* for the following postings lists.



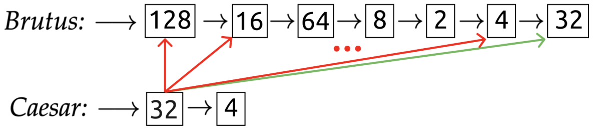
The Boolean retrieval model is being able to ask a query that is a Boolean expression:
Boolean Queries are queries using AND, OR, and NOT to join query terms.
- Views each document as a set of words.
- Is precise: document matches condition or not.



Brutus AND Caesar: 2, 8

2. Does the sorting (the postings lists) matter? Explain.

Definitely yes, It matter. Sorting the posting list is important because otherwise, we should look at the whole list for an element. For instance:

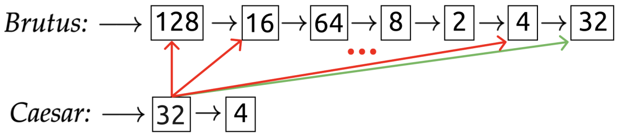


3. Can the size of the postings lists vary from one to another?

Yes, the size of the posting lists depends on the frequency of this term in the document.

How can such a query (mentioned above) be optimized?

It´s related to the size of the list, we should start to match the small one into the big one.
For example:



If you "intersect"(AND) Caesar with Brutus, you only have to match two elements again 7 otherwise you should try to match seven elements again two, and because the lists are sorted for sure you will not need to look in the whole list(average case). Also, the resulting list should be as much the size as the small one.

4. What are the issues in document parsing? Explain.

The two main issues in document parsing are related to Character sequence and choosing the document unit. There are Different encoding schemes to convert byte sequence of characters into a linear sequence. Although there are many sources to encode different types of texts, first the format of the document has to be appropriately selected and subsequently the right encoding scheme to be applied. On the other hand, languages can also cause issues especially when they don't keep a linear character order.
Depending on their nature, documents need to be indexed in different ways since they have distinct components and in this sense, choosing the document unit can become an issue.

5. Explain tokenization process in detail. Give some examples that are related to English, not other languages.

Tokenization is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols, and other elements called tokens. Tokens can be individual words, phrases, or even whole sentences. In the process of tokenization, some characters like punctuation marks are discarded. The tokens become the input for another process like parsing and text mining.

Tokenization relies mostly on simple heuristics in order to separate tokens by following a few steps:

- Tokens or words are separated by whitespace, punctuation marks, or line breaks
- White space or punctuation marks may or may not be included depending on the need
- All characters within contiguous strings are part of the token.
  Tokens can be made up of all alpha characters, alphanumeric characters, or numeric characters only.

Tokens themselves can also be separators. For example, in most programming languages, identifiers can be placed together with arithmetic operators without white spaces. Although it seems that this would appear as a single word or token, the grammar of the language actually considers the mathematical operator (a token) as a separator, so even when multiple tokens are bunched up together, they can still be separated via the mathematical operator.

Text:

In a village of La Mancha, the name of which I have no desire to call to mind, there lived not long since one of those gentlemen that keep a lance in the lance rack, an old buckler, a lean hack, and a greyhound for coursing.

Tokenized Text:

In → a → village → of → La → Mancha → the → name → of → which → I → have → no → desire → to → call → to → mind → there → lived → not → long → since → one → of → those → gentlemen → that → keep → a → lance → in → the → lance → rack → an → old → buckler → a → lean → hack → and → a → greyhound → for → coursing

CSC 785: Information storage & retrieval

6. Project: Write a code (preferably in Python) to Lemmatize any text inputs. It needs to be complete by itself (while submitting your code).

Hint: Use Wordnet Lemmatizer with NLTK as Wordnet is the standard and considered as a publicly available lexical database for English. Your input should be all least a paragraph, not just a single word.

Link to USD repository