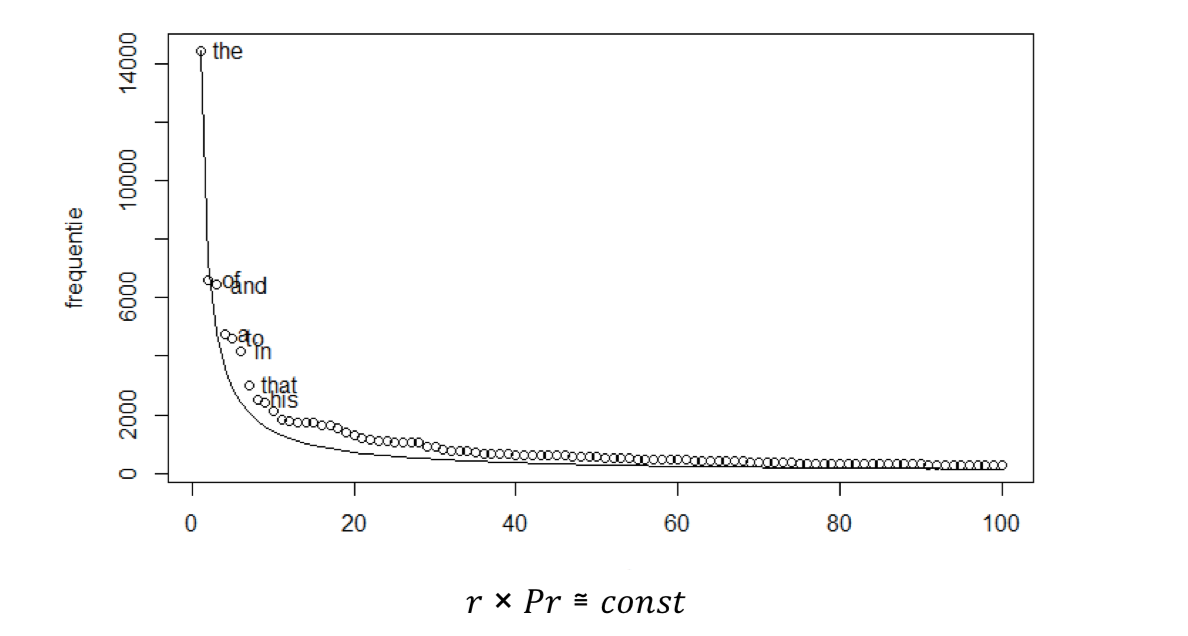
In the Boolean model the results are not ranked. Documents either contain the word or do not.

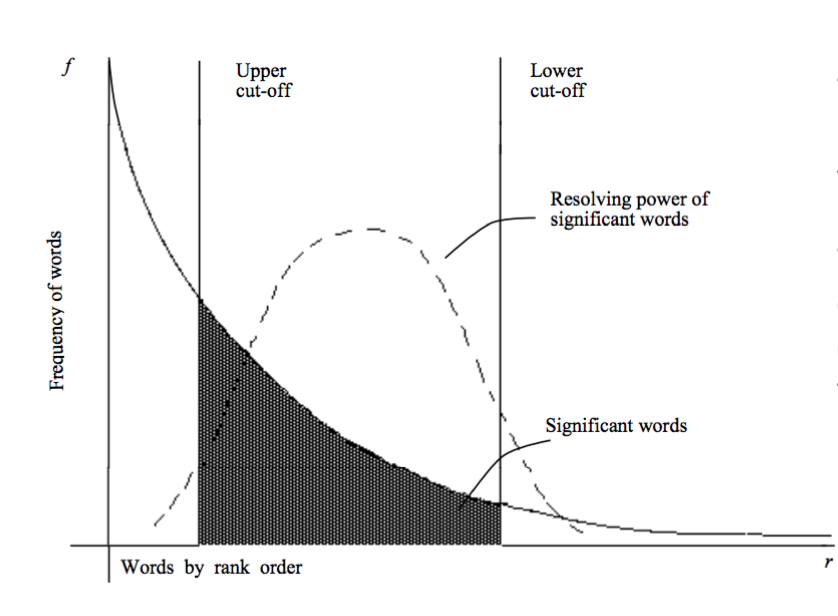
Ranked retrieval

* It is important to rank search results in terms of relevance. Users only look at the 10 top results at most.
* Can rank the documents using a score.
* How can you measure the relevance?
* An idea: Rank based on how many times query terms appear in the document.
* Problems: Document lengths vary, so in larger documents query terms may occur more frequently than in smaller documents.
* And: Rare terms are more informative than frequent terms. E.g Biology appearing once in a document may have more relevance than the term ‘the’ appearing 10 times.

Zipf’s law

* Frequency of any word in a given collection is inversely proportional to its rank in the frequency table.
* r \* frequency ≈ constant
* For example the 2nd ranked will appear about half as often as the first most often term. The 3rd ranked will appear about as 3rd as often as the most frequent term and so on. Term occurrence is proportional to (occurrence of most frequently used word) \* **1/rank**.
* Question: But what is this trying to say?
* [Zipf's Law YT vid](https://www.youtube.com/watch?v=fCn8zs912OE)

Luhn’s hypothesis

* The words exceeding the upper cut off are considered to be common
* The words below the lower cut-off are considered rare, and therefore do not contribute significantly to the content of the article.

Term Frequency (tf)

* tf*t,d,* The number of times the term occurs in the document.
* Do not want **raw term frequency?**

Document frequency (df)

* In how many documents term appears
* Consider how rare the term is.
* Rare terms more informative than frequent terms