The meaning of the term *information retrieval* (IR) can be very broad. For instance, getting your ID out of your pocket so that you can type that in a document is a form of information retrieval. However, as an academic field of study, *information retrieval* might be defined thus:
Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers). While there are several definitions of IR, many agree that IR is about technology to connect people to information. This includes search engines, recommender systems, and dialogue systems etc.

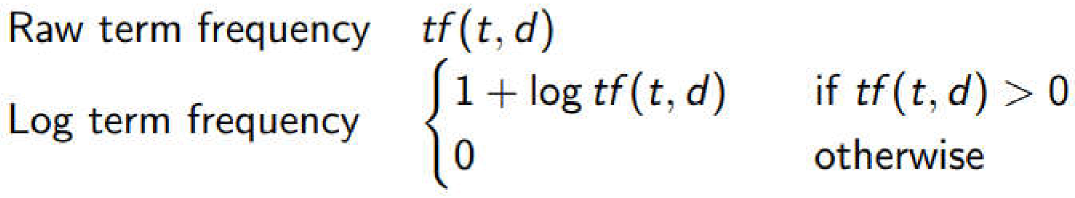
Term based retrieval methods are mathematical framework to defining query-document matching based on exact syntactic matching between a document and a query. The idea is that a pair of document and search query are represented by terms they contain. This article explains the intuition behinds most common term based retrieval methods such as DM25, TF-IDF, Query Likelihood Model.

1. TF-IDF

TF-IDF deals with information retrieval problem based on Bag of Words (BOW) model, which is probably the simplest IR model. TF-IDF contains two parts: TF(Term frequency) and IDF (Inverse Document Frequency).

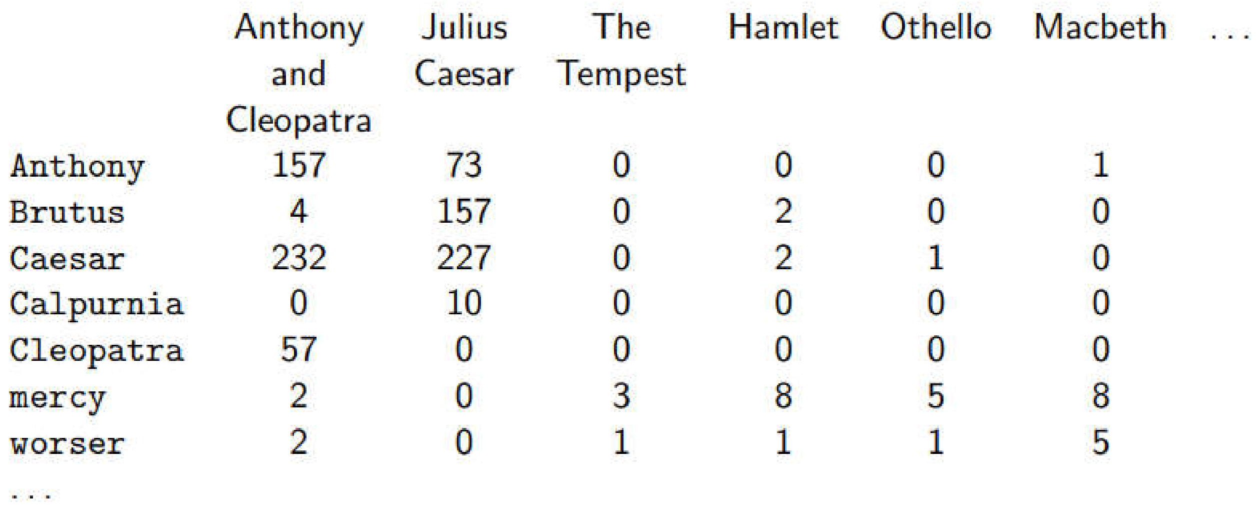
TF - Term Frequency

the idea is that we assign a weight to each term in a document depending on the number of occurrences of the term in the document. The score of the document is hence equal to the term frequency, that is intended to reflect how important a word is to a document



Raw term frequency $tf(t,d)$

Log term frequency $\begin{cases} 1 + \log tf(t,d) & \text{if } tf(t,d) > 0 \\ 0 & \text{otherwise} \end{cases}$



	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
Anthony	157	73	0	0	0	1	
Brutus	4	157	0	2	0	0	
Caesar	232	227	0	2	1	0	
Calpurnia	0	10	0	0	0	0	
Cleopatra	57	0	0	0	0	0	
mercy	2	0	3	8	5	8	
worser	2	0	1	1	1	5	
...							

Manning et al., "Introduction to Information Retrieval"

IDF - Inverse document frequency

However, term frequency suffers from a critical problem: All terms are considered equally important when it comes to assessing the document relevance on a query, although this is not always the case. In fact, certain terms have little or no discriminating power in determining relevance (e.g. ”the” may appear a whole lot in one document but contribute nothing to the relevance). To this end, we introduce a mechanism to reduce the effect of terms that occur too often in the collection to be meaningful for determining the relevance. To identify only the important terms, we can report document frequency (DF) of that term, which is the number of document in which terms occurs. It says something about the uniqueness of the terms in the collection. The smaller DF, the more uniqueness of the given terms

$$\mathrm{df}(t) := \#\{d : \mathrm{tf}(t; d) > 0\}$$

So, we are doing better. But what is the problem with DF? DF alone unfortunately tells us nothing. For instance, if DF of the term "computer" is 100, is that a rare or common term? We simply don’t know. That's why DF needs to be put in a context, which is the size of the corpus/collection. If the corpus contains 100 documents, then the term is very common, if it contains 1M documents, the term is rare. So, let’s add the size of corpus, called N and divide it to document frequency of the term.

$$idf(t) = \log \frac{N}{df(t)}$$

- $df(t)$ – document frequency of term t
- N – total number of documents in a collection

Great, we are doing fine. But let’s say the corpus size is 1000 and the number of documents that contain the term is 1, then N/DF will be 1000. If the number of documents that contain the term is 2, then N/DF will be 500. Obviously, a small change in DF can have a very big impact on N/DF and IDF Score. It is important to keep in mind that we mainly care when DF is in a low range compared to corpus size. This is because if DF is very big, the term is common in all documents and probably not very relevant to a specific topic. In this case, we want to smooth out the change of N/DF and one simple way to do this is to take the log of N/DF. Plus, applying a log can be helpful to balance the impact of both TF and N/DF on final score. As such, if the term appears in all documents of the collection, DF will be equal to N. Log(N/DF) = log1 = 0, meaning that the term does not have any power in determining the relevance.

Together, we can define TF-IDF score as follows:

$$\text{TF-IDF}(t, d) = tf(t, d) \cdot idf(t)$$

BM25

BM25 is a probabilistic retrieval framework that extends the idea of tf-idf. TF-IDF is good but BM25 is a better ranking system which improves some drawbacks of TF-IDF which involve term saturation and document length. The full BM25 formula looks a bit scary but you might have noticed that IDF is a part of BM25 formula. Let's break down the remaining part into smaller components to see why it makes sense.

$$BM25 = \sum_{t \in q} \log \left[\frac{N}{df(t)} \right] \cdot \frac{(k_1 + 1) \cdot tf(t, d)}{k_1 \cdot \left[(1 - b) + b \cdot \frac{dl(d)}{dl_{avg}} \right] + tf(t, d)}$$

- $k_1,\ b$ – parameters
- $dl(d)$ – length of document d
- dl_{avg} – average document length

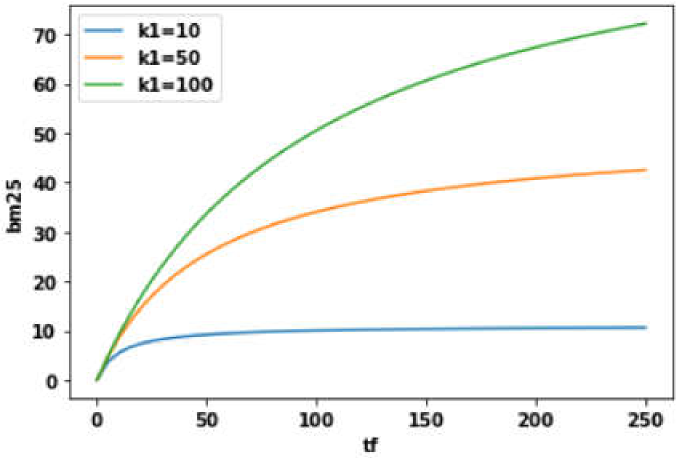
Term Saturation and diminishing return

If a document contains 100 occurrences of the term "computer,” is it really *twice* as relevant as a document that contains 50 occurrences? We could argue that if the term “computer” occurs a large enough number of times, the document is almost certainly relevant, and any more occurrence doesn’t increase the likelihood of relevance. So we want to control the contribution of TF when a term is likely to be saturated. BM25 solves this issue by introducing a parameter k that controls the shape of this saturation curve. This allows us to experiment with different values of k and see which value works best.

Instead of using raw TF, we will be using the following: (k1+1)* TF / (TF + k1)

$$w_t = \frac{(k_1+1)\cdot \mathrm{tf}(d;t)}{k_1 + \mathrm{tf}(d;t)} \cdot \mathrm{idf}(t)$$

So what does this do for us? It says that if k = 0, then (k+1)TF/TF+ k = 1. In this case, BM25 now turns out to be to IDF. If k goes to infinity, BM25 will be the same as TF-IDF. Parameter k can be tuned in a way that if the TF increases, at some point, BM25 score will be saturated as can be seen in the figure below, meaning that the increase in TF no longer contributes much to the score.



Document Length Normalization

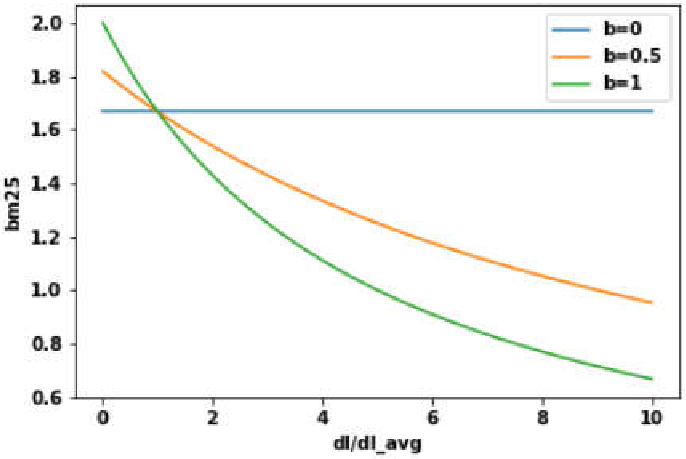
Another problem that is skipped in TF-IDF is document length. If a document happens to be very short and it contains “computer” once, that might already be a good indicator of relevance. But if the document is really, really long and the term "computer" only appears once, it is likely that the document is not about computer. We want to give reward term matches to short documents and penalize the long one. However, you do not to over-penalize though because sometimes a document is long because it contains lots of contains rather than just having lots of words. So, how can we achieve this? We will introduce another parameter b, which is used to construct the normalizer: (1-b)+b*dl/dlavg to BM25 formula. The value of parameter b must be between 0 and 1 to make it work. Now BM25 score will be the following:

$$w_t = \frac{(k_1+1)\cdot \mathrm{tf}(d;t)}{k_1\cdot((1-b)+b\cdot(l_d/l_{avg})) + \mathrm{tf}(d;t)} \cdot \mathrm{idf}(t)$$

First, let's us understand what it means for a document to be short or long. Again a document is considered long or short depends on the context of the corpus. One way to decide is to use the average length of the corpus as a reference point. A long document is simply one that is *longer than average length* of the corpus and a short one is shorter than average length of the corpus.

What does this normalizer do for us? As you can see from the formula, when b is 1, the normalizer will turn to (1 – 1 + 1*dl/dlavg). On the other hand, if b is 0, the whole thing becomes 1 and the effect of document length isn't considered at all. If b is bigger, the effects of the document length compared to the average length are more amplified.

When a document length (dl) is shorter than the corpus average's length (dlavg), the ratio dl/dlavg is smaller than 1 hence b*dl/dlavg is smaller than b, hence increase the BM25 score. On the contrary, when a document is longer than average corpus's length, the ratio dl/dlavg will be larger than 1 and the value of b* dl/dlavg will be bigger than b -…=> not sure if I am explaining it right???



In summary, **TF-IDF rewards term frequency and penalizes document frequency. BM25 goes beyond this to account for document length and term frequency saturation**.

Traditional term based retrieval methods like BM25 deal with the problem based on Bag-of-Words (BoW) representation, thus they only focus on exact matching (i.e., syntactic) and lack the consideration for semantically related words.