Term Frequency

Documents are similar if they contain the same keywords (frequently)

- Therefore use the frequency *freq(i,j)* of the keyword k_i in the document d_j to determine the weight of the document vectors

(Normalized) term frequency of term k_i in Document d_j

$$tf(i,j) = \frac{freq(i,j)}{\max_{k \in T} freq(k,j)}$$

An obvious difference that can be made among terms is with respect to their frequency of occurrence in a document. Thus, a weighting scheme for documents can be defined by considering the (relative) frequency of terms within a document. The term frequency is normalized with respect to the maximal frequency of all terms occurring within the document.

Inverse Document Frequency

We have not only to consider how frequent a term occurs within a document (measure for similarity), but also how frequent a term is in the document collection of size n (measure for distinctiveness)

Inverse document frequency of term k_i

$$idf(i) = \log(\frac{n}{n_i}) \in [0, \log(n)]$$

n_i number of documents in which term k_i occurs

Inverse document frequency can be interpreted as the amount of information associated with the term k_i

Term weight (tf-idf)	$w_{ij} = tf(i,j)\ idf(i)$

Therefore, we should consider not only the frequency of a term within a document, when determining the importance of the term for characterizing the document, but also the discriminative power of the term with respect to the whole document collection. For that purpose, the inverse document frequency is computed and included into the term weight as factor.

With this weighting scheme we notice that eliminating stop words is in fact an optimization of computing similarity measures in vector space retrieval. Since stop words normally occur in every document of a collection, their term weights will normally be 0 and thus these terms will not receive any weight. Therefore, it makes sense to exclude them already on the preprocessing of documents.

Similarity Computation in Vector Space Retrieval

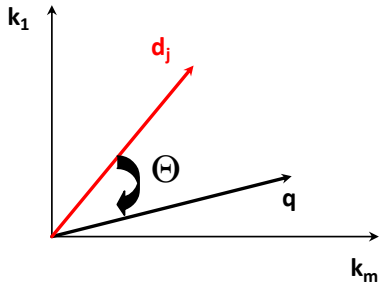
$$\vec{d}_j = \left(w_{1j}, w_{2j}, \ldots, w_{mj}\right), w_{ij} > 0 \quad if\ k_i \in d_j$$

$$\vec{q} = \left(w_{1q}, w_{2q}, \ldots, w_{mq}\right), w_{iq} \geq 0$$

$$sim(\vec{q}, \vec{d}_j) = \cos(\theta) = \frac{\vec{d}_j \bullet \vec{q}}{\left|\vec{d}_j\right| \left|\vec{q}\right|} = \frac{\sum_{i=1}^{m} w_{ij} w_{iq}}{\left|\vec{d}_j\right| \left|\vec{q}\right|}$$

$$|v| = \sqrt{\sum_{i=1}^{m} v_i^2}$$

Since $w_{ij} > 0$ and $w_{iq} \geq 0$, $0 \leq sim(q, d_j) \leq 1$



For information retrieval, the distance measure for vectors must satisfy the following properties:

• If two vectors coincide completely their similarity should be maximal, i.e., equal to 1.

• If two vectors have no keywords in common, i.e., if wherever the query vector has positive weights the document vector has weight 0, and vice versa – or in other words if the vectors are orthogonal – the similarity should be minimal, i.e., equal to 0.

• in all other cases the similarity should be between 0 and 1.

The scalar product (which is equivalent to the cosine of the angle of two vectors) has exactly these properties and is therefore (normally) used as similarity measure for vector space retrieval.

A good question is why not use Euclidean distance, instead of cosine distance. The problem with using Euclidean distance is the following: different documents with the same or similar distribution of term weights, but of different vector length (for example, because the documents have different length) would give very different results, whereas their meaning would be the same or similar. For example, documents with vectors (1,1,0) and (2,2,0) would probably have the same meaning. But their Euclidean distances to a query vector would be very different (for illustration, consider the query vector (1,1,0)).

Recall and Precision

Recall is the fraction of relevant documents retrieved from the set of total relevant documents collection-wide

Precision is the fraction of relevant documents retrieved from the total number retrieved (answer set)

	Relevant	Non-relevant
Retrieved	True positives (tp)	False positives (fp)
Not Retrieved	False negatives (fn)	True negatives (tn)

$$R = \frac{tp}{tp + fn} = P(retrieved|relevant)$$

$$P = \frac{tp}{tp + fp} = P(relevant|retrieved)$$

The results of IR systems can be compared to the expected result in two ways:

1. **Recall** measures how large a fraction of the expected results is actually found.
2. **Precision** measures how many of the results returned are actually relevant.

Important note: This measure evaluates an **unranked** result set. All elements of the result are considered as equally important.

Mean Average Precision (MAP)

Given a set of queries Q

For each $q_j \in Q$ the set of relevant documents $\{d_1, \dots d_{m_j}\}$

D_{jk} the top k relevant documents for query q_j

$P_{int}(D_{jk})$ interpolated precision of result D_{jk}

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} P_{int}(D_{jk})$$

Mean Average Precision has been shown to be a robust measure for evaluating the quality of a ranked retrieval system for a query collection Q. When a relevant document is not retrieved at all, the precision value in the MAP is 0.

Note that we have a slight (ab)use of notation. We write $P_{int}(D_{jk})$ to denote $P_{int}(k)$ for the top-k recalled documents for query q_j .