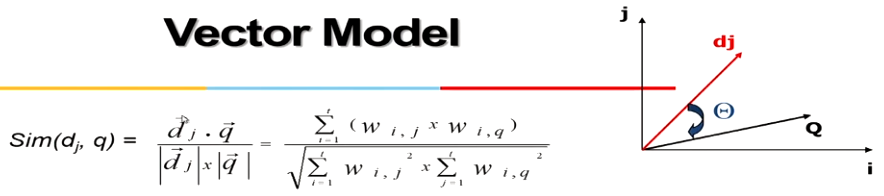
* Is based on the fact the binary weights are too limiting, since they don’t provide option for partial match or relevance.
* Non-binary weights provide consideration for partial match, where each term has a partial weight.
* These term weights are used to compute a degree of similarity between a query and each document.
* Based on the similarity weight, ranking of documents is done.
* ***Wij >= 0***is a weight associated with the pair (ki,dj)
* ***vec(dj) =* (*W1, j, W2, j, ……, Wt, j*)** is the weighted vector associated with document **dj.**
* **Wi, q >= 0** associated with the pair (ki,q).
* ***vec(q) =* (*W1, q, W2, q, ……, Wt, q*) .**
* **t** represents total no. of index terms in the collection.

Similarity between a document and a query can be denoted as,



Similarity is calculated as a cos function of angle between dj and q. If exactly same angle will be 0 (i.e. cos (0) = 1) and if totally different, angle will be 90 (i.e. cos (90) = 0).

* *tf* is defined as the term frequency or the intra document frequency*,* which is quantification for intra-documents contents (similarity)
* *Idf* is defined as the inverse document frequency, which is quantification for inter-documents separation (dis-similarity)
* ***Wij = tf \* idf***

Advantages:

* Simple model based on linear algebra.
* Terms weights are not binary.
* Allow partial match.
* Allow ranking documents based on relevance.
* Allows computing a continuous degree of similarity between queries and documents.
* Allows efficient implementation for large document collections.
* Preferred for handling large amount of records.

Limitations:

* Index terms are assumed to be mutually independent.
* Long documents are poorly represented.
* Search keywords must precisely match document terms.
* The order in which the terms appear in document is lost in the vector space representation. That is why this model is also known as ***bag of words*** model.
* Weighting is intuitive, but not very formal.