INFO/CS 4300: Language and Information
Vector Space Model Cheatsheet

General framework. In the *Vector Space Model* we represent documents, as well as queries, as vectors in the same space. The dimensions of the space correspond to all the terms $t_1, \ldots, t_N$, that are contained in a collection of documents.

A document d_j will be represented in this model by the $N-$dimensional vector:

$$\vec{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{N,j}),$$

where $w_{i,j}$ is the weight of the term t_i in document d_j meant to indicate how important that term is in the document. Note that the all document vectors have the same length, N (i.e., the number of terms), independent of how many of these terms it contains.

Similarly, a query q can be embedded in the space as another vector of length N:

$$\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{N,q})$$

.

When designing an information retrieval system, our goal is to define a similarity measure $\mathrm{sim}(\vec{q}, \vec{d_j})$ that captures well the fitness of document d_j for the information need expressed in query q.

The way in which the terms t_i, the weights $w_{i,j}$, and the similarity measure $\mathrm{sim}(\vec{q}, \vec{d_j})$ are defined is an open question, and the solution often depends on the setting in which the information retrieval system is applied. In this course we will discuss some of the possible solutions.

Defining the terms. One of the simplest ways to define the set of terms as corresponding to the vocabulary of types present in all the documents.[1]

Defining the weights.

Here are two ways to define the weights that we discussed in class so far (more to come):

- **Binary weights.** We can set $w_{i,j}$ to be 1 if term t_i occurs in document d_j, and 0 otherwise.
- **Term frequency weights.** We can set $w_{i,j} = \mathrm{tf}_{i,j}$ (the *term frequency*) which is simply the number of times the the term t_i appears in document d_j.

[1] One could also define terms as concepts; for example "Kim Kardashian" would be considered one single term.)

Defining the similarity measure.

We also discussed a couple of possible functions to calculate the match between a document and a query, when represented in the vector space:

- **Dot product.**

$$\text{sim}(\vec{q}, \vec{d_j}) = \vec{q} \cdot \vec{d_j} = \sum_{k=1}^{N} w_{k,j} w_{k,q}$$

  Note that when using binary weights, this will simply corresponds to the the number of words in common between the query and the document, since $w_{k,j} w_{k,q} = 1$ iff term t_k occurs in both d_j and q.

- **Generalized Jaccard.**

$$\text{sim}(\vec{q}, \vec{d_j}) = \text{GeneralizedJaccard}(\vec{q}, \vec{d_j}) = \frac{\sum_k \min(w_{k,j}, w_{k,q})}{\sum_k \max(w_{k,j}, w_{k,q})}$$

Using binary weights, this becomes equivalent to the Jaccard similarity we defined earlier in class: the number of common terms shared by the query **and** document divided by the number of all unique terms contained by either the query **or** the document, rewriten here using the vector space notation:

$$\text{Jaccard}(\vec{q}, \vec{d_j}) = \frac{\sum_k w_{k,j} \wedge w_{k,q}}{\sum_k w_{k,j} \vee w_{k,q}}$$