# Vector Space Model Introduction

COMP3009J: Information Retrieval

Dr. David Lillis (david.lillis@ucd.ie)

UCD School of Computer Science Beijing Dublin International College

### Introduction

- So far, we have seen Information Retrieval (IR) as a process where a user submits a **query** consisting of some **terms** connected with **boolean operators**.
- □ This is known as the "Boolean Model" of IR, and has some serious disadvantages:
  - The model predicts that each document is either relevant or non-relevant to the query.
  - There is no notion of a "partial" match to the query conditions and so this can lead to too few documents being retrieved.
  - Every document that is considered to be relevant is treated the same, so no ranking occurs.

## Introduction

- One of the most common IR models in use is the **Vector Space Model**.
- ☐ This models both **documents** and **queries** in an **N-dimensional space**.
- Basic vector algebra is used in order to calculate the most similar documents to a query.
- The basic vector space model has been adapted for many other purposes.
- The most common are:
  - Machine Learning (e.g. K-NN algorithm)
  - Clustering
- See page 77 of Modern Information Retrieval (2<sup>nd</sup> Ed.).

#### Basis

- □ A collection contains a set of documents each of which is composed of a "bag of words" (also known as terms).
  - Each unique **term in the collection** is represented by a dimension in the vector space.
  - A document is then represented by giving a non-zero value to the dimension representing a term it contains.
    - This number is known as a term weight.
  - Other dimensions get a value of zero.
    - i.e. the weight of a term is 0 in documents that do not contain it.
- □ A similarity score is calculated based on the proximity of two vectors in the N-dimensional space.

# Partial Matching

- Unlike the Boolean Model, the Vector Space Model facilitates partial matching by using non-binary term weights.
- In models like this, each document must have a **similarity score** calculated for it, which measures how similar it is to the given query.
- This is usually shown as sim(q, d) (i.e. the similarity between a query q and a document d).
- □ These models return a **ranked list** of documents, where the documents with the highest similarity scores are at the top of the list.
- In this way, it is hoped that the most relevant documents are at the top of the result set, so that the user can find relevant information easier.

# Vector Space Model

Document and Query Representation

## Document Representation

- Consider the following documents:
  - 1. Information Retrieval is an exciting subject
  - 2. Mathematics is important in Information Retrieval
- Ignoring stopwords the collection contains 6 distinct terms: information; retrieval; exciting; subject; mathematics; important.
- The vector space will have six dimensions (one for each term in the document collection).
- ☐ This means that **EVERY DOCUMENT** is mathematically represented by six dimensions, regardless of how many terms it contains.

# Document Representation

- □ For the introduction to this topic, we first assume a simple weighting scheme:
  - if term *i* is contained in document *j*, it will have a weight of 1 (i.e.  $w_{i,j} = 1$ ).
  - if term *i* is not contained in document *j*,  $w_{i,j} = 0$ .

- In practice, the weight will be a real number where:
  - $0 \le w_{i,j} \le 1$
- We will see how this is calculated later.

Term	Doc 1	Doc 2
information	1	1
retrieval	1	1
exciting	1	0
subject	1	0
mathematics	0	1
important	0	1

# Query Representation

- The query is represented in an identical way.
- Consider the query: important information.
- This is represented in the vector model as:
  - **(**1, 0, 0, 0, 0, 1)
- □ It is important that the **order of the terms** is the same as for the document collection (i.e. the first dimension relates to the term "information", the second relates to "retrieval", etc.).
  - If the order changes, then we are no longer comparing like with like.

# Vector Space Model

- The rest of this topic is organised as follows:
  - Cosine Similarity: how to compare queries with documents to measure how similar they are.
  - **Term Weights:** Some words are more important in describing a document than others.
  - Conclusions and Summary