Vector Space Model
Term Weighting

COMP3009J: Information Retrieval

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

UCD School of Computer Science
Beijing Dublin International College

Term Weighting

- In order to illustrate the model, we have used a simple binary weighting scheme.
- This gives the representation of the vectors in terms of 1s and 0s only.
- This is not commonly used in practice.
- Frequently used metric is **TF-IDF**:
  - **TF:** Term Frequency
  - **IDF:** Inverse Document Frequency

Term Weights

- We can also take the **number of times** a term occurs in a document into account (i.e. the *frequency* of the term), for example:
  - The term "Biden" will appear in a biography of Joe Biden, for obvious reasons.
  - The term "Biden" will also appear in a biography of Donald Trump, as he once won an election against Donald Trump. It will not be as common, however.
- The first document is more relevant to a search for "Biden" but without using weights, both would be treated equally.
- The greater number of occurrences of the term in the first document should result in a greater weight.

Term Weights: Why?

- Consider a document collection of 100,000 documents.
- A word that appears in every document is not very useful as an index term, as it tells us very little about which documents a user might be interested in.
- A word that appears in 5 documents considerably narrows the space of documents that might be of interest to the user.
- This is the basic rationale behind stopwords, but it has wider consequences also.

Term Weights

- When talking about term weights, we use the following notation:
- A weight $w_{i,j} > 0$ is associated with **every index term** k_i of **document** d_j.
- For an index term that does not appear in the document, $w_{i,j} = 0$
- This weight quantifies the importance of the index term for describing the document's semantic contents.
- Any time we examine a new IR model, we must firstly consider how its weighting scheme works.

Term Weights

- Index terms are usually assumed to be **independent**, though this is a simplification.
- Consider a document about computer networks. Here, the terms “computer” and “network” are clearly related, as the appearance of one attracts the appearance of the other.
- You may argue that their weights should be altered to take this dependency into account.

TF-IDF Weighting

- TF-IDF consists of two parts:
  - The **Term Frequency (TF)** assumes that a term that occurs many times in a document is more important in describing that document than one that occurs rarely. **Every term** has a separate term frequency for **every document** in the IR system.
  - The **Inverse Document Frequency (IDF)** argues that terms that only occur in very few documents are more important than those that are found in many documents. This is based on the same idea as stopwords. **Each term** has only **one** inverse document frequency within an entire document collection.

Term Frequency (TF)

- Term Frequency is calculated by using the formula below.
- $tf_{i,j}$ is the term frequency of term i in document j and is calculated as follows:
  - $tf_{i,j} = \frac{freq_{i,j}}{maxfreq_j}$
    - where $freq_{i,j}$ is the number of times term i occurs in document j and $maxfreq_j$ is the maximum number of times *any* term occurs in document j.
    - It is a *normalised* frequency.
- The reason we divide by the maximum frequency is to ensure that terms in **long documents** do not get an unfair advantage.

Inverse Document Frequency (IDF)

- A term's Inverse Document Frequency is designed to give a higher weight to terms that are rare.
- **Note** that each term only has one IDF score within an entire document collection.
- It is calculated using the following formula:
  - $idf_i = log_2\left(\frac{N}{n_i}\right)$
- N is the total number of documents in the collection
- n_i is the number of documents in the collection that contain term i.

TF-IDF

- Finally, the TF-IDF weight for a term i in a document j is found by multiplying its TF for that document by its IDF score, i.e.
- $w_{i,j} = tf_{i,j} \times idf_i$
- Note again that each term has a **different TF-IDF score** for **each document** in the document collection.

Example

- D1: “new york times”
- D2: “new york post”
- D3: “los angeles times”
- Q: “new new times”