Vector Space Model: TF-IDF

Adapted from Lectures by

Prabhakar Raghavan and Christopher Manning

Bag of words model

- Vector representation doesn't consider the ordering of words in a document
  - *John is quicker than Mary* and *Mary is quicker than John* have the same vectors
- This is called the bag of words model
- Bag: multiset (multiplicity counts, abstract order)

Term frequency *tf*

- The *term frequency* $tf_{t,d}$ of term ***t*** in document ***d*** is defined as the number of times that ***t*** occurs in ***d***.
- We want to use ***tf*** when computing query-document match scores. But how?
- *Raw* term frequency is *not* what we want:
  - A document with 10 occurrences of the term may be more relevant than a document with one occurrence of the term.
  - But not 10 times more relevant.
- Relevance does not increase proportionally with term frequency.

Log-frequency weighting

- The log frequency weight of term ***t*** in ***d*** is

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{t,d}, & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

- $0 \rightarrow 0$, $1 \rightarrow 1$, $2 \rightarrow 1.3$, $10 \rightarrow 2$, $1000 \rightarrow 4$, etc.
- Score for a document-query pair: sum over terms ***t*** in both query ***q*** and document ***d***:

$$\text{score} = \sum_{t \in q \cap d} (1 + \log \mathrm{tf}_{t,d})$$

  - The score is 0 if none of the query terms is present in the document.

Document frequency

- Rare terms are more informative than frequent terms
  - Recall stop words
  - For example, consider a term in the query that is rare in the collection (e.g., *eccentric*)
  - A document containing this term is very likely to be relevant to the query *eccentric*
  - → We want a higher weight for rare terms like *eccentric*

Document frequency, continued

- Consider a query term that is frequent in the collection (e.g., high, increase, line)
  - A document containing such a term is more likely to be relevant than a document that doesn't, *but it's not a sure indicator of relevance.*
  - → For frequent terms, we want positive weights for words like *high, increase, and line*, but lower weights than for rare terms.
- We will use **document frequency** (**df**) to capture this in the score.
- df ($\leq N$) is the number of documents that contain the term

idf weight

- **df_t** is the **document_frequency** of ***t*** : the number of documents that contain ***t***
  - **df** is a measure of the informativeness of ***t***
- We define the **idf** (inverse document frequency) of ***t*** by $$\text{idf}_t = \log_{10} N/\text{df}_t$$
  - We use log N/df_t instead of N/df_t to "dampen" the effect of **idf**.

the base of the log is immaterial.

idf example, suppose N = 1 million

term	df_t	idf_t
calpurnia	1	6
animal	100	4
sunday	1,000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0

There is one **idf** value for each term ***t*** in a collection.

Collection vs. Document frequency

- The collection frequency of ***t*** is the number of occurrences of ***t*** in the collection, counting multiple occurrences.

Word	Collection frequency	Document frequency
insurance	10440	3997
try	10422	8760

- Which word is a better search term (and should get a higher weight)?

tf-idf weighting

- The **tf-idf** weight of a term is the product of its tf weight and its idf weight.

$$w_{t,d} = (1 + \log \mathrm{tf}_{t,d}) \times \log N / \mathrm{df}_t$$

- Best known weighting scheme in information retrieval
  - Note: the "-" in tf-idf is a hyphen, not a minus sign!
  - Alternative names: tf.idf, tf x idf
- Increases with
  - the number of occurrences within a document
  - the rarity of the term in the collection

TF-IDF Example Applications

- How to find similar twitter users or bloggers?
  - Build user profiles
  - Compare profiles
- How to build a user profile?
  - Twitter user profile
  - Blogger profile
- How to compare profiles?

QUESTIONS?