Department of Computing

Imperial College of Science, Technology and Medicine

example

example description

Author: Anton Zhitomirsky

Contents

1 TF-IDF

1.1 Motivating Problem

If we have a search in Google for “low carb breakfast” they get shown generic results for breakfast. Some of these terms shouldn’t have equal weighting to each-other.

The problem is that the search is using basic string/keyword matching. It treats all words as equally important. Less relevant results are ranked as highly as more relevant ones.

1.2 Solution

Rank more important words as more important.

1.3 Introduction

This is pre-neural networks.

- **Term Freqeuency (TF)**: Mesaures how often a term occurs in a document. The more often a term appears in a document, the more important it is for that document.
- **Inverse Document Frequency (IDF)**: Measures how common or rare a term is across all documents in the corpus. Terms that appear in amny different documents are less significant that those that appear in a smaller number of documents.

This means that a search for “low-carb breakfast” willl prioritize recipes where “low-carb” is a significant term, rather than returning recipes with the more common “breakfast” term.

TF-IDF

Term Frequency (TF):
Upweights words w that are more
Important to d

Frequency of w occurring together with d

$$\mathrm{TF}_{w,d} = \frac{\mathrm{count}(w,d)}{\sum_{w'} \mathrm{count}(w',d)}$$

Inverse Document Frequency (IDF):
Downweights words that appear
everywhere

Size of document collection

$$\mathrm{IDF}_{w,D} = log \frac{|D|}{|\{d \in D : w \in d\}|}$$

Number of documents in D that contain w

$$\text{TF-IDF}_{w,d,D} = \mathrm{TF}_{w,d}\mathrm{IDF}_{w,D}$$

7