TF-IDF

Princeton AI4ALL

Produced by Princeton University for AI4ALL

Today's Challenge

Based on a search query, how to find the most relevant documents?

Example: Searching a bunch of documents for "dumpling recipe"

Relevance: Problem Setting

Ex: You have 10,000 newspaper articles. Find the articles most relevant to some phrase (e.g. “the brown cow”).

“corpus” is the set of articles

“document” is any individual article

How could you approach this problem?

Discuss ideas with a partner!

Relevance: Ideas

Return any article which contains all the keywords
Return any article containing one keyword or more
Return the article with the most mentions of the keywords
Other ideas?

Relevance: Challenges

What challenges can you foresee for this problem?

Relevance: Challenges

Filter out common words like “the”
- Example: search for “the boat”
- How to do this?

Avoid favoring longer articles
- Example: What if one article is a long book, while the other one is a few paragraphs?
- How to do this?

Relevance: Term Frequency (TF)

Address “longer article” problem

The proportion of words which equal the keyword

- Instead of counting the raw number of matches

Calculate the TF of “cow” for these examples:

- “The brown cow mooed”
- “Cow is a cow”
- “Cow cow cow cow cow”

Higher is better

Relevance: Inverse Document Frequency (IDF)

We care about rarer, special words
What proportion of documents actually contain this word?
- “the” gives a high value
- “cow” gives a low value

Take the reciprocal of this
- “the” gives a low value
- “cow” gives a high value

Finally, take the log base 2

$$\log \frac{N}{n_t}$$

Relevance: TF-IDF

Combine the two scores by multiplying TF with IDF
Then, add the score for each word in the query
Example 1:

Query: “cow”
Documents: “the cow moos”, “the dog barks”, “dog dog”
TF for “cow”: ⅓, 0, 0
IDF for “cow”: log(3/1)≈1.58
TF-IDF scores: 0.52, 0, 0

Example 2:

Query: “the cow”
Documents: “the cow moos”, “the dog barks”, “dog dog”
TF for “the”: ⅓, ⅓, 0
IDF for “the”: log(3/2)≈0.58
Total TF-IDF scores: 0.19 + 0.52 = 0.71, 0 + 0.19 = 0.19, 0 + 0 = 0

Clarifications

TF is calculated *per document*

Represents how relevant the document is to a keyword

IDF is calculated *per corpus*

Represents how much we care about that keyword

Practice Time!

Practice Problems:

Calculate the TF-IDF scores for each of the following:

Phrase: “panther”
Documents:

1) “I like that panther.”
2) “Panther panther panther.”
3) “TF-IDF is awesome.”
4) “Nothing here.”
5) “Hello.”

Phrase: “the computer broke”
Documents:

1) “We fix computers, we’re the best!”
2) “If you’re broke, you can’t buy that.”
3) “The cat is cute.”
4) “The book is funny.”

Practice Coding

Time to code TF-IDF in Python!