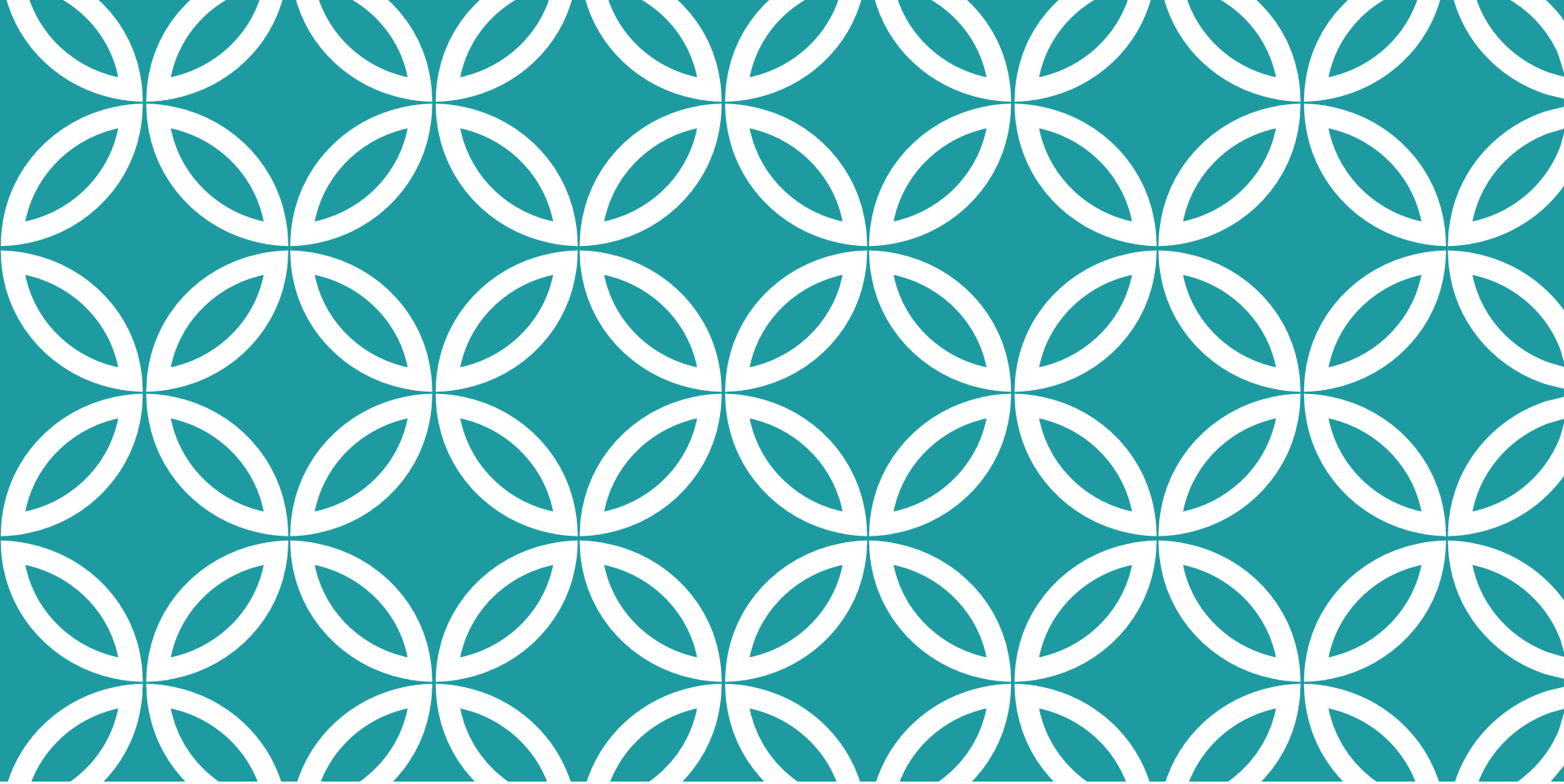


ORGANIC WEB SEARCH:
Finding Similarity
(Between web pages and a query)

CT102 Information Systems

Consider the following tf*idf weights which were calculated for some terms in documents in a document collection and a query

	rent	house	crisis	cap	agreement	tenanc	evict	sim
doc1	0.2	0.3	0.1					
doc2	0.01			0.4	0.3		0.3	
doc3	0.15	0.35			0.4	0.35		
doc4	0.25	0.32	0.15		0.33	0.4		
doc5	0.1			0.43	0.3		0.5	
query	1	1			1	1		

Assuming that query terms have weight of 1

PROBLEM STATEMENT

How to find which documents are most similar to the query across all terms in the query

Consider the following tf*idf weights which were calculated for some terms in documents in a document collection and a query

	rent	house	crisis	cap	agreement	tenanc	evict	sim
doc1	0.2	0.3	0.1					
doc2	0.01			0.4	0.3		0.3	
doc3	0.15	0.35			0.4	0.35		
doc4	0.25	0.32	0.15		0.33	0.4		
doc5	0.1			0.43	0.3		0.5	
query	1	1			1	1		

Assuming that query terms have weight of 1

VECTOR SPACE MODEL

Main abstractions:

- For all documents, each document D is represented as a vector of real-valued numbers where each number corresponds to the weights of a term in the document
- Queries are also viewed as vectors
- Each position in the vector corresponds to a term from the document collection
  - Therefore, the length of the vector (number of weights/vector dimension) is the number of terms in a document collection called **the vocabulary**
  - In reality, only need to consider the positions which correspond to the terms in the query and which have non-zero weights

VECTOR SPACE COMPARISON

- A query is also represented as a vector where each term in the query can be assigned a weight of 1.0
- Comparison is done by finding the similarity between document vectors and the query vector
- e.g., in previous example:

$$\overrightarrow{query} = < 1.0\ 1.0, 0.0, 0.0, 1.0, 1.0, 0.0 >$$

	rent	house	crisis	cap	agreement	tenanc	evict
doc1	.2	.3	.1	0	0	0	0
doc2	.01			.4	.3	0	.3
doc3	.15	.35			.4	.35	
doc4	.25	.32	.15		.33	.4	
doc5	.1			.43	.3		.5
query	1	1			1	1	

$\overrightarrow{doc1} = < 0.2, 0.3, 0.1, 0.0, 0.0, 0.0, 0.0, 0.0 >$

$\overrightarrow{doc2} = < 0.1, 0.0, 0.0, 0.4, 0.3, 0.0, 0.3 >$

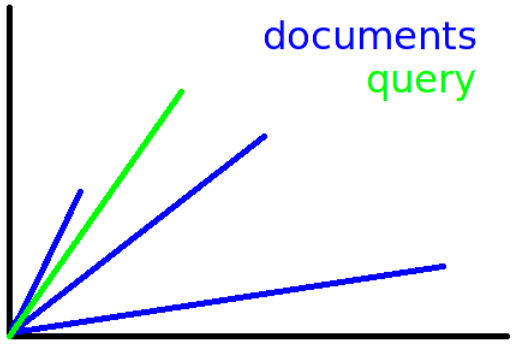
$\overrightarrow{doc3} = < 0.15, 0.35, 0.0, 0.0, 0.4, 0.35, 0.0 >$

$\overrightarrow{doc4} = < 0.25\ 0.32, 0.15, 0.0, 0.33, 0.4, 0.0 >$

$\overrightarrow{doc5} = < 0.1\ 0.0, 0.0, 0.43, 0.3, 0.0, 0.5 >$

$\overrightarrow{query} = < 1.0\ 1.0, 0.0, 0.0, 1.0, 1.0, 0.0 >$

From google images (2D vectors)



Similarity between a document vector *d* and a query vector *q*:

The idea is to "measure" the angle or distance between the vectors d and q which represent the document d and query q.

This is done using the Euclidean dot product (cosine similarity) of the two vectors

If the vectors are close (i.e. similar) the distance between them is small and the result is close to1.

If the vectors are far apart (i.e. dis-simimlar), the distance between them is large and the results is close to 0.

Similarity between vectors d and q: Euclidean dot product definition (Cosine Similarity)

$$\text{sim}(\vec{d}, \vec{q}) = \frac{\vec{d} * \vec{q}}{\left|\vec{d}\right| * \left|\vec{q}\right|} = \frac{\sum_1^n (x_i * q_i)}{\sqrt{\sum_1^n {x_i}^2} * \sqrt{\sum_1^n {q_i}^2}}$$

On the top line: the weights of the corresponding terms multiplied by each other and added up: $x_i * q_i$

x_i is weight for i^{th} term in d

q_i is weight for i^{th} term in q

On the bottom line (the denominator): Square root of the sum of each weight (per document and query) squared and multiplied by each other

The denominator normalises (by the vector norm or vector magnitude) so that the number of terms in a document is considered

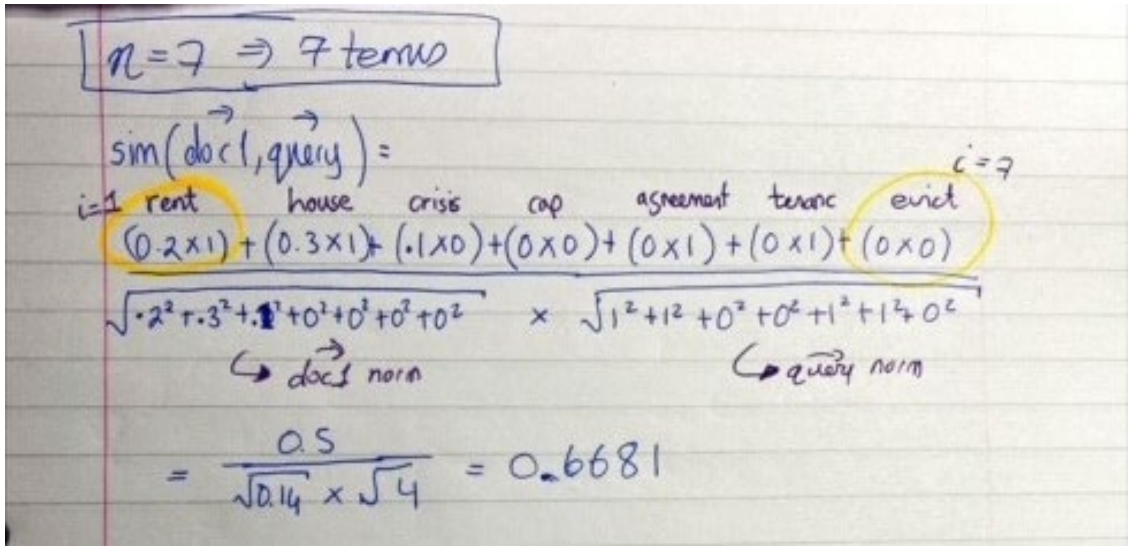
FINDING SIMILARITIES . . .

	rent	house	crisis	cap	agreement	tenanc	evict	sim
doc1	.2	.3	.1					
doc2	.01			.4	.3		.3	
doc3	.15	.35			.4	.35		
doc4	.25	.32	.15		.33	.4		
doc5	.1			.43	.3		.5	
query	1	1			1	1		

Let's start with calculating the similarity between doc1 and the query

$$\text{sim}(\vec{d}, \vec{q}) = \frac{\vec{d} * \vec{q}}{|\vec{d}| * |\vec{q}|} = \frac{\sum_1^n (x_i * q_i)}{\sqrt{\sum_1^n {x_i}^2} * \sqrt{\sum_1^n {q_i}^2}}$$

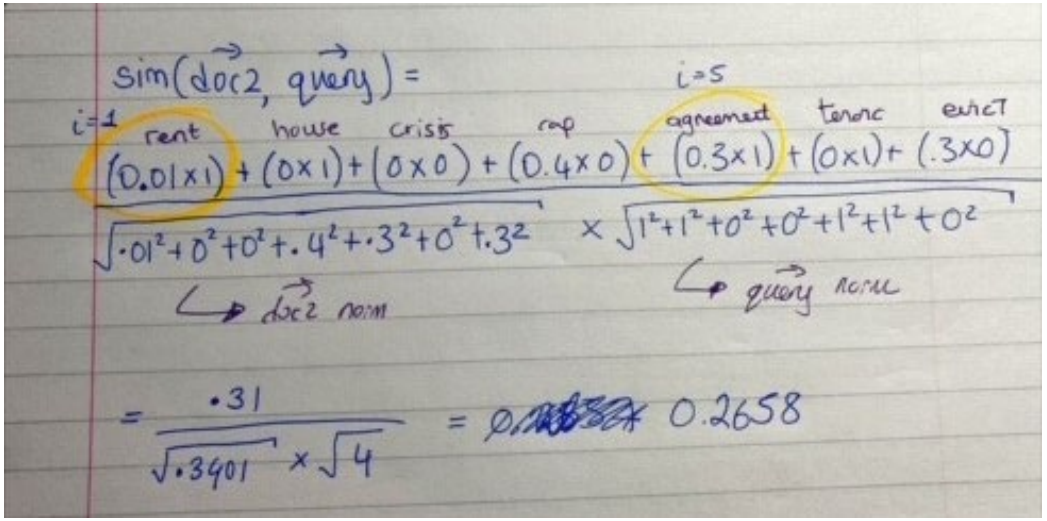
	rent	house	crisis	cap	agreement	tenanc	evict	sim
doc1	.2	.3	.1					
query	1	1			1	1		



Similarity between doc2 and the query

$$\text{sim}(\vec{d}, \vec{q}) = \frac{\vec{d} * \vec{q}}{|\vec{d}| * |\vec{q}|} = \frac{\sum_1^n (x_i * q_i)}{\sqrt{\sum_1^n x_i^2} * \sqrt{\sum_1^n q_i^2}}$$

	rent	house	crisis	cap	agreement	tenanc	evict	sim
doc2	.01	0	0	.4	.3	0	.3	
query	1	1	0	0	1	1	0	



Similarity between doc3 and the query

$$\text{sim}(\vec{d}, \vec{q}) = \frac{\vec{d} * \vec{q}}{|\vec{d}| * |\vec{q}|} = \frac{\sum_1^n (x_i * q_i)}{\sqrt{\sum_1^n {x_i}^2} * \sqrt{\sum_1^n {q_i}^2}}$$

	news	republican	candid	organ	campaign	presidenti	food	sim
doc3	.15	.35			.4	.35		
query	1	1	0	0	1	1	0	

HOW ABOUT THE OTHER DOCS?

$\text{sim}(\overrightarrow{d1}, \overrightarrow{q}) =$

$\text{sim}(\overrightarrow{d2}, \overrightarrow{q}) =$

$\text{sim}(\overrightarrow{d3}, \overrightarrow{q}) =$

$\text{sim}(\overrightarrow{d4}, \overrightarrow{q}) =$

$\text{sim}(\overrightarrow{d5}, \overrightarrow{q}) =$

SUMMARY OF SIMILARITIES:

	rent	house	crisis	cap	agreement	tenanc	evict	sim
doc1	.2	.3	.1					0.668
doc2	.01			.4	.3		.3	0.2658
doc3	.15	.35			.4	.35		0.9559
doc4	.25	.32	.15		.33	.4		0.9622
doc5	.1			.43	.3		.5	0.2735
query	1	1			1	1		

NOW COMPARING THE SIMILARITIES:

	sim
doc1	0.668
doc2	0.2658
doc3	.9559
doc4	.9622
doc5	.2735

Returned in this order:
doc4
doc3
doc1
doc5
doc2

VECTOR SPACE COMPARISON ADVANTAGES:

- Documents can be found which are most similar to the query without the need for a 100% match
- Returned documents can be sorted in decreasing order of similarity to query (so we have some ranking)
- Most commonly used approach across search engines and applied widely elsewhere also in natural language processing tasks and in machine learning approaches using natural language.

QUESTION:
Are query vector weights always 1?

No …. Usually query terms are expanded, some are removed (stop words) and terms are weighted according to:

- Whether term is an original part of query or whether it was added by the system (e.g., as part of thesaurus look up for example or as part of personalisation information)
- Whether term has been used previously by that person (personalisation information) and whether the term is currently being used by other people (popularity + personalisation).

EXAMPLE 2

Given the vector tf*idf representation calculated of 3 documents with terms `sql`, `database`, `program`, `comput`

Find the most relevant document to the query "database programming with sql" which is represented by the following query '`database program sql`' and the vector query:

`<1, 1, 1, 0>`

* Use precision to 4 decimal places for final answer for similarities

RECALL: tf*idf WEIGHTS (to precision of 3 decimal places)

	d1	d2	d3
sql	0.081	0.024	0.086
database	0.023	0.088	0
program	0	0.047	0.014
comput	0.021	0	0.019

Now adding in query vector (assume weights of all terms are 1, given that we are not told otherwise)

	d1	d2	d3	query
sql	0.081	0.024	0.086	1
database	0.023	0.088	0	1
program	0	0.047	0.014	1
comput	0.021	0	0.019	0

Can now start to calculate similarities:

$$sim(\vec{d}, \vec{q}) = \frac{\vec{d} * \vec{q}}{\left|\vec{d}\right| * \left|\vec{q}\right|} = \frac{\sum_1^n x_i * q_i}{\sqrt{\sum_1^n x^2} * \sqrt{\sum_1^n q^2}}$$

What is similarity of d1 to query?

	d1	d2	d3	query
sql	0.081	0.024	0.086	1
database	0.023	0.088	0	1
program	0	0.047	0.014	1
comput	0.021	0	0.019	0

$$\frac{(0.081 \times 1 + 0.023 \times 1)}{\left(\sqrt{0.081^2 + 0.023^2 + 0.021^2} \times \sqrt{3}\right)}$$

Answer = 0.692

What is similarity of d2 to query?

	d1	d2	d3	query
sql	0.081	0.024	0.086	1
database	0.023	0.088	0	1
program	0	0.047	0.014	1
comput	0.021	0	0.019	0

$$\frac{(0.024 \times 1 + 0.088 \times 1 + 0.047 \times 1)}{\left(\sqrt{0.024^2 + 0.088^2 + 0.047^2} \times \sqrt{3}\right)}$$

Answer = 0.8946

What is similarity of d3 to query?

	d1	d2	d3	query
sql	0.081	0.024	0.086	1
database	0.023	0.088	0	1
program	0	0.047	0.014	1
comput	0.021	0	0.019	0

$$\frac{(0.086 \times 1 + 0.014 \times 1)}{\left(\sqrt{0.086^2 + 0.014^2 + 0.019^2} \times \sqrt{3}\right)}$$

Answer = 0.6474

NOTE:

The same approach can be used to determine how similar documents are to each other

Where might this be useful?

	d1	d2
sql	0.081	0.024
database	0.023	0.088
program	0	0.047
comput	0.021	0

For example, the similarity between d1 and d2?

$$\frac{((0.081 \times 0.024) + (0.023 \times 0.088))}{\left(\sqrt{0.081^2 + 0.023^2 + 0.021^2} \times \sqrt{0.024^2 + 0.088^2 + 0.047^2}\right)}$$

Answer = 0.4456

SUMMARY:

$$sim(\vec{d}, \vec{q}) = \frac{\vec{d} * \vec{q}}{\left|\vec{d}\right| * \left|\vec{q}\right|} = \frac{\sum_1^n x_i * q_i}{\sqrt{\sum_1^n x^2} * \sqrt{\sum_1^n q^2}}$$

- Matching to find similar documents is usually performed using the dot product of the vector representations of documents and queries – however only those terms which are present in the query need to be considered
- The vector norm (denominator) can be pre-calculated for all documents