Unit 8 – TF-IDF

CS 201 - Data Structures II
Spring 2022
Habib University

Syeda Saleha Raza

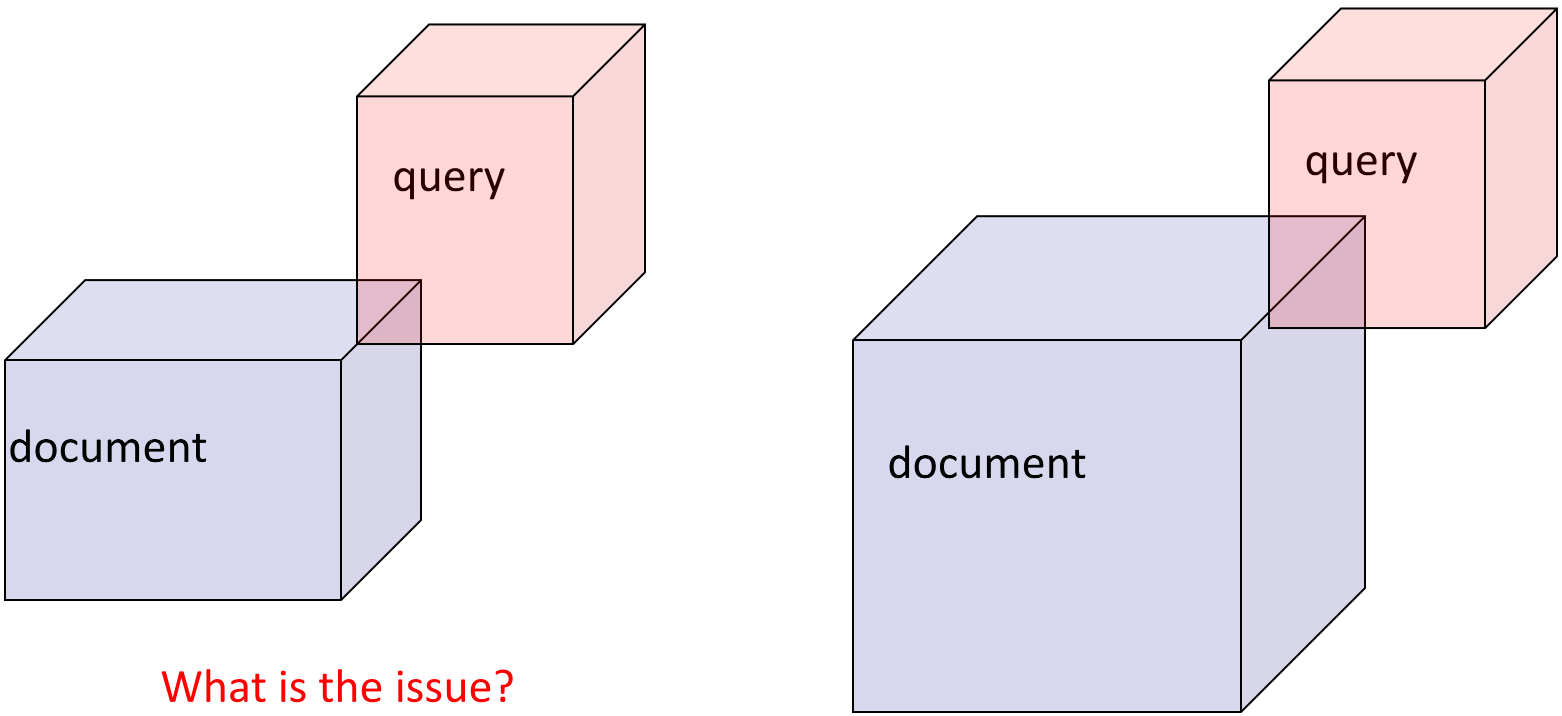
Bag of words

1. The sun is shining
2. The weather is sweet
3. The sun is shining and the weather is sweet

```
{'the': 5, 'shining': 2, 'weather': 6, 'sun': 3, 'is': 1, 'sweet': 4,
'and': 0}
```

```
[[0 1 1 1 0 1 0]
 [0 1 0 0 1 1 1]
 [1 2 1 1 1 2 1]]
```

Some things to be careful of…



What is the issue?

Need some notion of the length of a document

http://www.stanford.edu/class/cs276/handouts/lecture6-tfidf.ppt

Term Frequency

- In document d, the frequency represents the number of instances of a given word t.

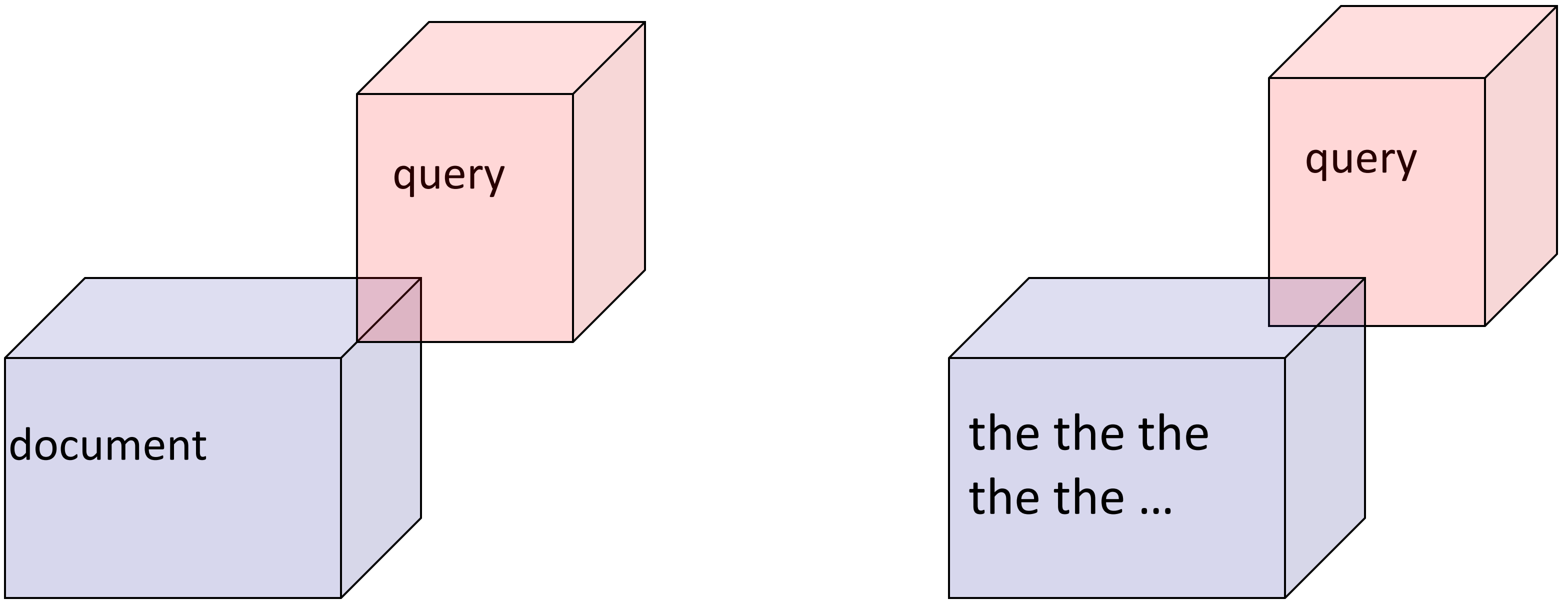
$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Pitfall with term frequency

Most frequent word on twitter in each state in the United States .

[Terrible Maps on Twitter: "The most popular word in each state https://t.co/LY7LewNohn" / Twitter]

Some things to be careful of…



Need some notion of the importance of words

http://www.stanford.edu/class/cs276/handouts/lecture6-tfidf.ppt

Term importance

- Rare terms are more informative than frequent terms
  - Recall stop words
- Consider a term in the query that is rare in the collection
- How to quantify rareness of a term?

Document frequency

- Terms that occur in many documents are weighted less, since overlapping with these terms is very likely
  - In the extreme case, take a word like the that occurs in EVERY document
- Terms that occur in only a few documents are weighted more

Inverse Document Frequency (IDF)

- Inverse Document Frequency (IDF)

$$idf_i = \log \frac{|D|}{|d : t_i \in d|}$$

  - Calculates how common a word is across documents. Most common terms are less significant.

TF-IDF

- **TF-IDF (term frequency-inverse document frequency)** is a statistical measure that evaluates how relevant a word is to a document in a collection of documents.

$$\text{tf-idf}_{i,j} = \text{tf}_{i,j} \times \text{idf}_i$$

Bag of Words

1. The sun is shining
2. The weather is sweet
3. The sun is shining and the weather is sweet

```
{'the': 5, 'shining': 2, 'weather': 6, 'sun': 3, 'is': 1, 'sweet': 4,
'and': 0}
```

```
[[0 1 1 1 0 1 0]
 [0 1 0 0 1 1 1]
 [1 2 1 1 1 2 1]]
```

TF-IDF

1. `The sun is shining`
2. `The weather is sweet`
3. `The sun is shining and the weather is sweet`

```
{'the': 5, 'shining': 2, 'weather': 6, 'sun': 3, 'is': 1, 'sweet': 4,
'and': 0}
```

```
[[ 0.    0.43  0.56  0.56  0.    0.43  0.  ]
 [ 0.    0.43  0.    0.    0.56  0.43  0.56]
 [ 0.4   0.48  0.31  0.31  0.31  0.48  0.31]]
```

Interpreting TF-IDF

In other words, $\text{tf-idf}_{t,d}$ assigns to t a weight in document d that is:

- highest when t occurs many times within a small number of documents (thus lending high discriminating power to those documents);
- lower when the term occurs fewer times in a document, or occurs in many documents (thus offering a less pronounced relevance signal);
- lowest when the term occurs in virtually all documents.

Exercise

	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

- Compute TF-IDF score of each term in this corpus.

Cosine Similarity

$$\cos(\mathbf{a},\mathbf{b}) = \frac{\mathbf{a}\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} = \frac{\sum_{i=1}^{n}\mathbf{a}_i\mathbf{b}_i}{\sqrt{\sum_{i=1}^{n}(\mathbf{a}_i)^2}\sqrt{\sum_{i=1}^{n}(\mathbf{b}_i)^2}}$$

Example

An example is measuring the similarity between documents based on word counts:

Document	Advertising	Auto	Car	Detroit	Engine	Germany	Sales
a	5	88	123	43	35	0	36
b	71	125	42	76	0	27	88

$$a \cdot b = (5 \times 71) + (88 \times 125) + \cdots + (36 \times 88) = 22957$$

$$||a|| = \sqrt{5^2 + 88^2 + 123^2 + 43^2 + 35^2 + 0^2 + 36^2} = 165.13$$

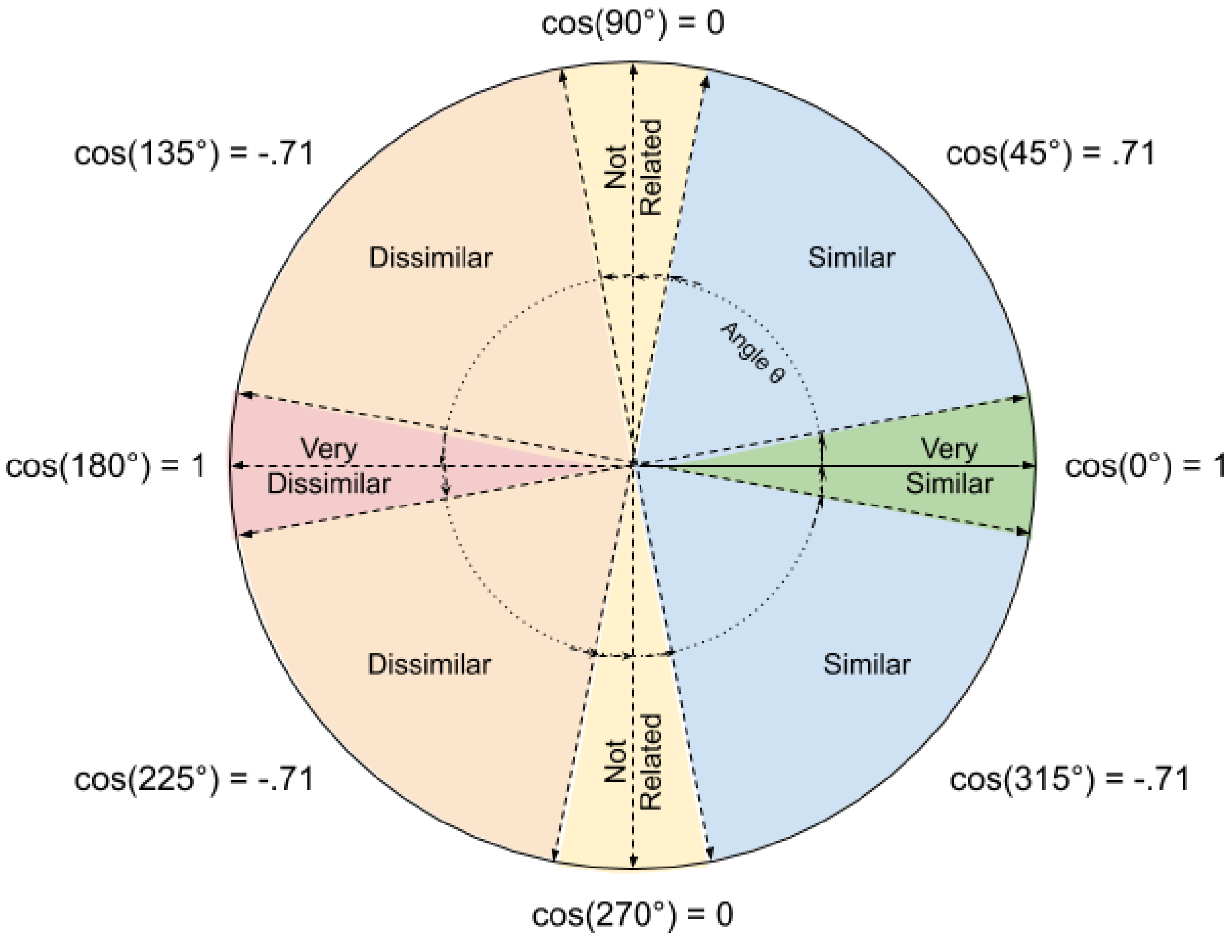
$$||b|| = \sqrt{71^2 + 125^2 + 42^2 + 76^2 + 0^2 + 27^2 + 88^2} = 191.52$$

$$||a||\;||b|| = 165.13 \times 191.52 = 31626$$

$$cosine\,similarity = cos\,\theta = \frac{a \cdot b}{||a||\;||b||} = \frac{22957}{31626} = .73$$

[Cosine Similarity — The Science of Machine Learning (ml-science.com)]

Interpreting Results



[Cosine Similarity — The Science of Machine Learning (ml-science.com)](ml-science.com)

The Three Documents and Similarity Metrics

Considering only the 3 words from the above documents: ‘sachin’, ‘dhoni’, ‘cricket’

Doc Sachin: Wiki page on Sachin Tendulkar
Dhoni - 10
Cricket - 50
Sachin - 200

Doc Dhoni: Wiki page on Dhoni
Dhoni - 400
Cricket - 100
Sachin - 20

Doc Dhoni_Small: Subsection of wiki on Dhoni
Dhoni - 10
Cricket - 5
Sachin - 1

Document - Term Matrix (Word Counts)

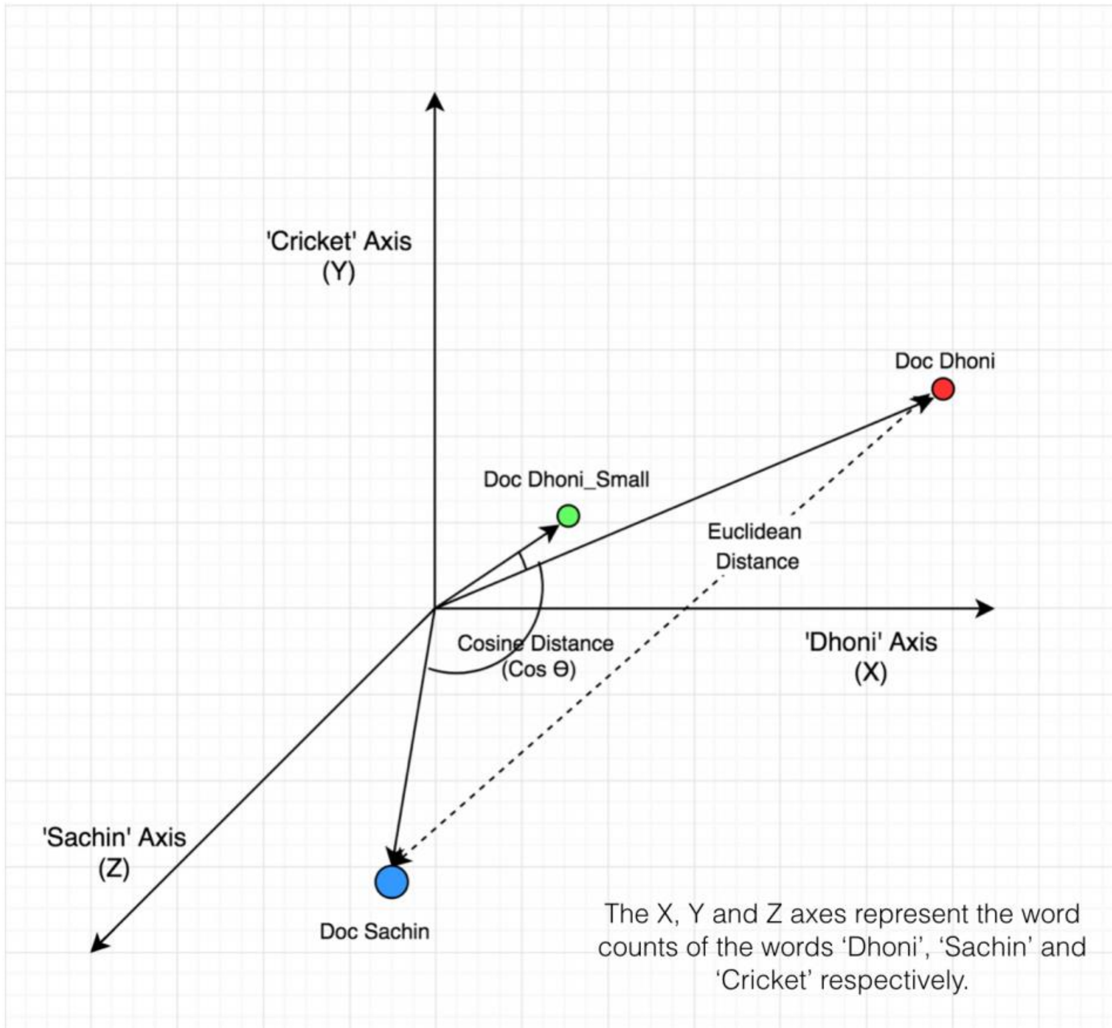
Word Counts	*“Dhoni”*	*“Cricket”*	*“Sachin”*
Doc Sachin	10	50	200
Doc Dhoni	400	100	20
Doc Dhoni_Small	10	5	1

Similarity Metrics

Similarity or Distance Metrics	*Total Common Words*	*Euclidean distance*	*Cosine Similarity*
Doc Sachin & *Doc Dhoni*	10 + 50 + 10 = **70**	432.4	0.15
Doc Dhoni & *Doc Dhoni_Small*	20 + 10 + 7 = **37**	204.0	0.23
Doc Sachin & *Doc Dhoni_Small*	10 + 10 + 7 = **27**	401.85	0.77

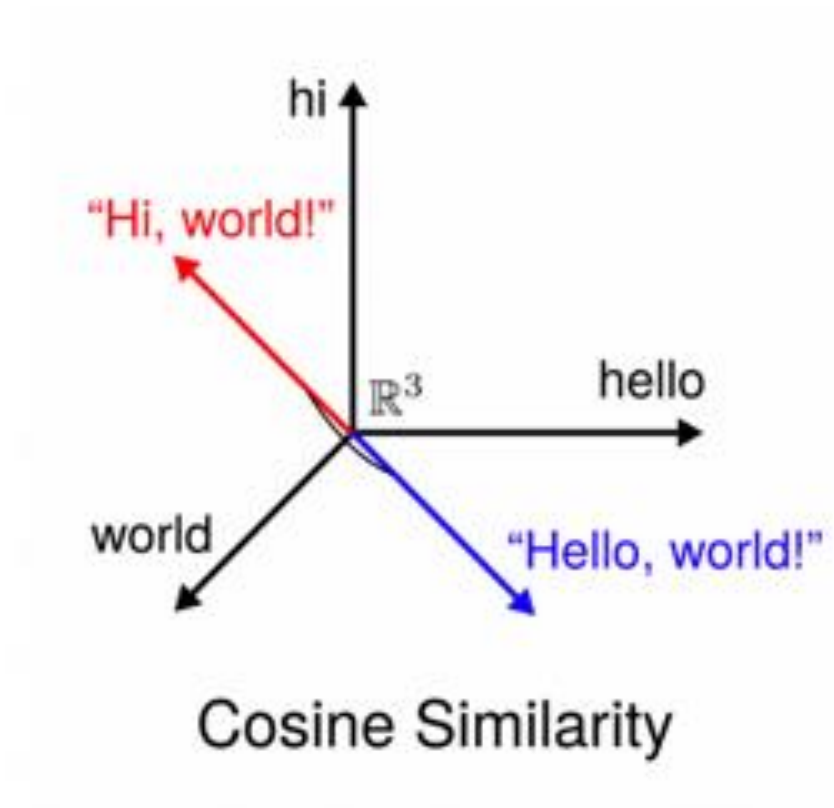
https://www.machinelearningplus.com/nlp/cosine-similarity/

Projection of Documents in 3D Space

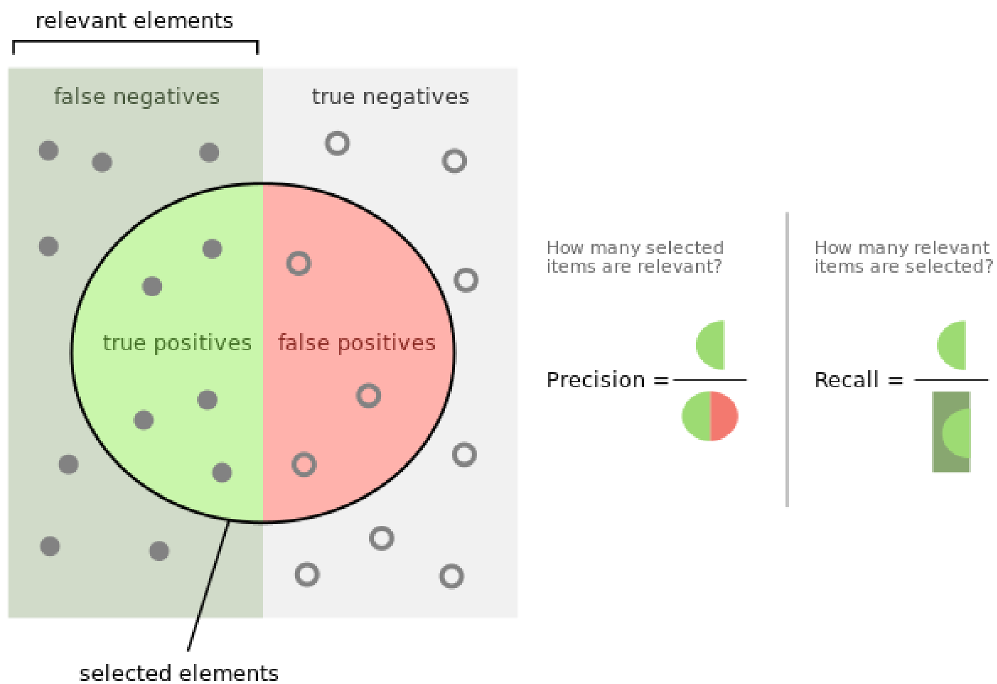


https://www.machinelearningplus.com/nlp/cosine-similarity/

Projection of documents in 3D space



Precision vs Recall



https://en.wikipedia.org/wiki/Precision_and_recall

Thanks