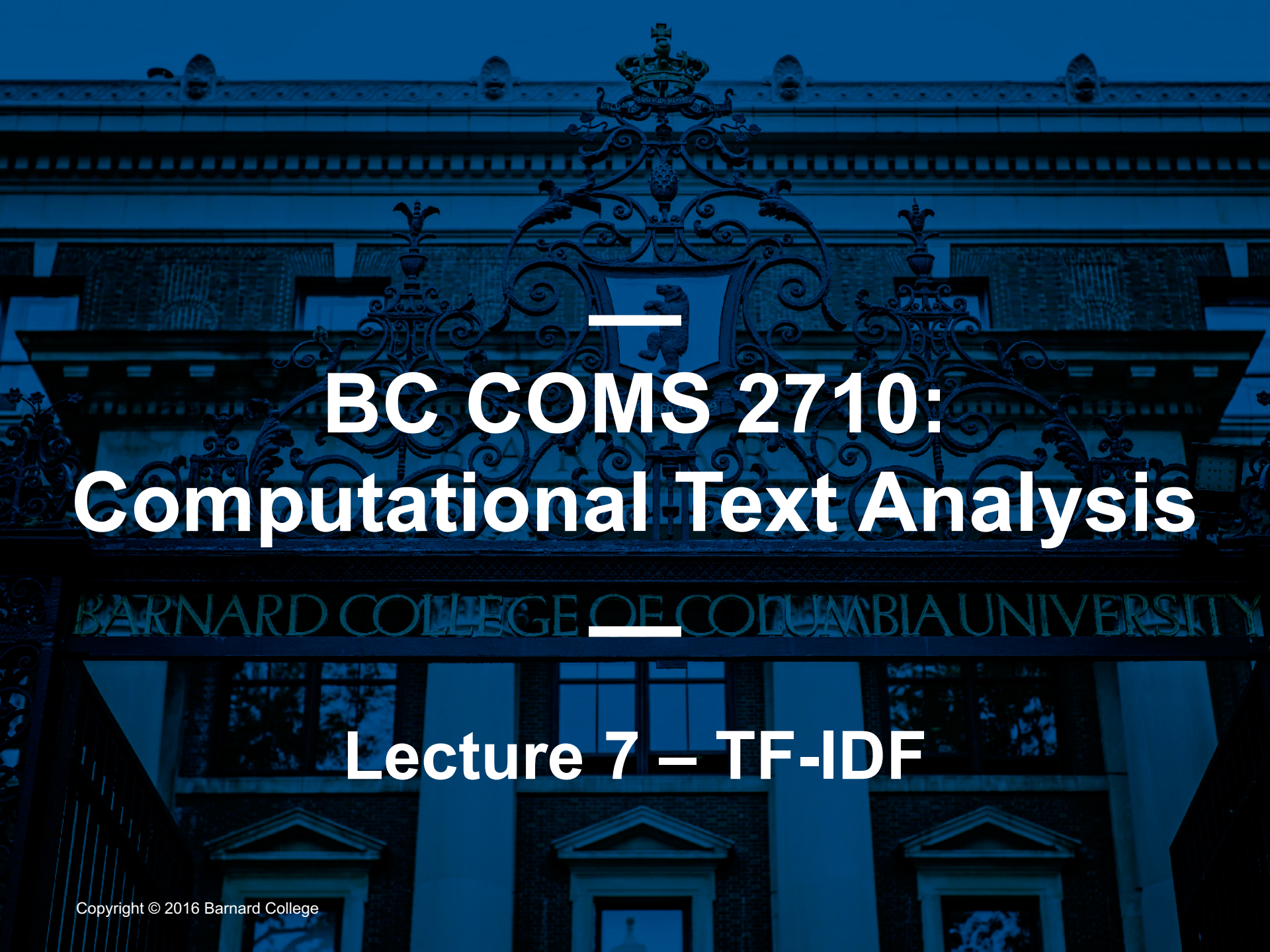


BC COMS 2710: Computational Text Analysis

Lecture 7 – TF-IDF

Announcements – Assignments

- Readings:
  - Reading 02 – link course site, due Sunday
- Tutorial 1.3:
  - Graded half, will release scores later today
- Week 2 Tutorials:
  - 2.1 – Tokenization, lemmatization, stopwords, etc
    - Based on mondays lecture
  - 2.2 – Exploring dictionary-based methods
    - Based tomorrow's lecture

Announcements - Home

- Homework 01
  - Extended to Saturday
- Homework 02
  - Based on today's material
  - Released tomorrow or Friday
    - will have a week to complete
  - More open-end than Homework 01
  - NYTimes Obituaries:
    - Finding document specific terms
    - Finding similar obituaries

Yesterday

- Document matrix
- Started TF-IDF
  - Not so great

Today

- TF-IDF:
  - Overview
  - Computing it in Sklearn
  - Most important/interesting terms
  - Most similar documents

TF-IDF

TF-IDF use cases

1. Discover interesting terms

2. Compare documents in a corpus

Term Frequency (tf):

Frequency of word ***w*** in document ***d***

How to compute it?

$$\frac{|w|}{|\,Document\,|}$$

*number of times **w** appears in **D***

*divided by of number tokens in **D***

Why not use word counts?

TF normalizes for different document lengths

Issue with Term Frequency (TF)?

- Most frequent words are often not informative
- Why?
  - Zipf's law
  - Common across documents in a corpus

- Solution:
  - Weigh a word's TF based on how the word is spread across the corpus

Document Frequency

How common word ***w*** is across the corpus

How to compute it?

$$\mathbf{DF(w)} = \frac{|\, tf(w,d) \neq 0 \,|}{|\, D \,|}$$

Number of documents that contain ***w*** divided by number of document

Inverse Document Frequency

How common word ***w*** is across the corpus

How to compute it?

$$\mathbf{IDF(w)} = \frac{|\,\boldsymbol{D}\,|}{|\,\boldsymbol{tf(w,d)\neq 0}\,|}$$

Number of documents divided
by number of documents that contain ***w***

Inverse Document Frequency examples

1. ***the*** appears in every document

IDF***(the)*** =

Inverse Document Frequency examples

1. ***the*** appears in every document

IDF***(the)*** = 1

Inverse Document Frequency examples

1. ***the*** appears in every document

   IDF***(the)*** = 1

2. ***superfragilistic*** appears in one document

   IDF(***superfragilistic)*** **=**

Inverse Document Frequency examples

1. ***the*** appears in every document

   IDF***(the)*** = 1

2. ***superfragilistic*** appears in one document

   IDF(***superfragilistic)*** **=** number of documents

TF-IDF:
Term Frequency - Inverse Document Frequency

TF-IDF of word ***w*** in document ***D***:

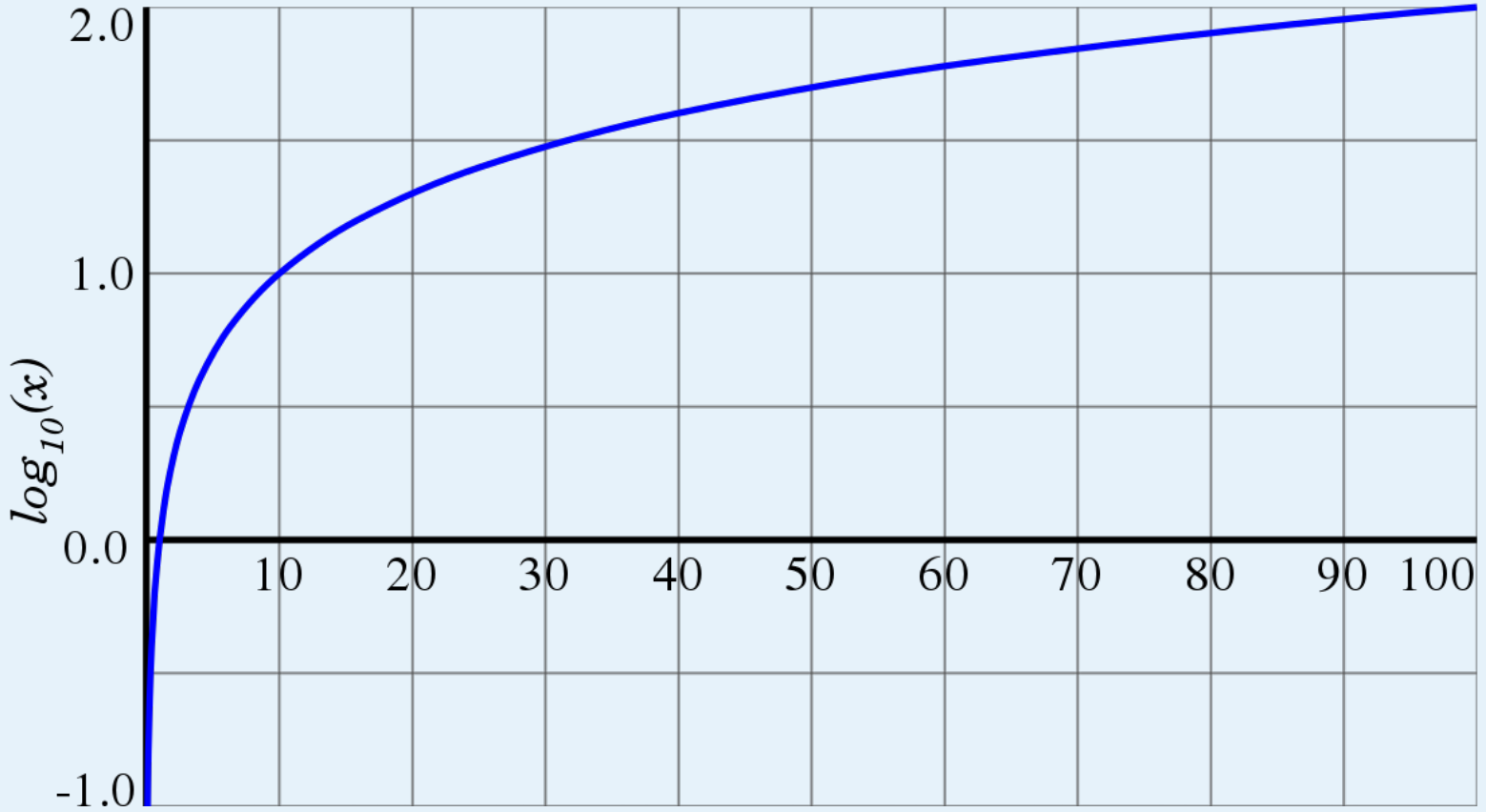
Term Frequency * Inverse Document Frequency

Captures terms that are frequent in a document and specific to the document in the corpus

However, which will be much bigger, TF or IDF?

Scaling down IDF

log function is a way to scale down idf



Understanding log

Inverse Document Frequency

How common word ***w*** is across the corpus

How to compute it?

$$\mathbf{IDF(w)} = \boldsymbol{log}\left(\frac{|\,\boldsymbol{D}\,|}{|\,\boldsymbol{tf(w,d)\neq 0}\,|}\right)$$

Number of documents divided
by number of documents that contain ***w***

Inverse Document Frequency examples

1. ***the*** appears in every document

   IDF***(the)*** = log(1) = ?

2. ***superfragilistic*** appears in one document

   IDF(***superfragilistic)*** **=** log(number of documents)

TF-IDF:
Term Frequency - Inverse Document Frequency

TF-IDF of word ***w*** in document ***D***:

Term Frequency * Inverse Document Frequency

Captures terms that are frequent in a document and specific to the document in the corpus

Remaining issues

TF-IDF of word that appears in every corpus is 0

TF-IDF of word ***w*** that never appears:

Remaining issues

TF-IDF of word that appears in every corpus is 0

Word still has some information

TF-IDF of word ***w*** that never appears:

TF(***w***) =

IDF(***w***) =

Remaining issues

TF-IDF of word that appears in every corpus is 0

We probably don't want this

TF-IDF of word ***w*** that never appears:

TF(***w***) = 0

IDF(***w***) =

Remaining issues

TF-IDF of word that appears in every corpus is 0

We probably don't want this

TF-IDF of word $\boldsymbol{w}$ that never appears:

$$\text{TF}(\boldsymbol{w}) = 0$$

$$\text{IDF}(\boldsymbol{w}) = \boldsymbol{log}\left(\frac{|\boldsymbol{D}|}{|\boldsymbol{tf}(\boldsymbol{w},\boldsymbol{d}) \neq \boldsymbol{0}|}\right)$$

Remaining issues

TF-IDF of word that appears in every corpus is 0

We probably don't want this

TF-IDF of word $\boldsymbol{w}$ that never appears:

$$\text{TF}(\boldsymbol{w}) = 0$$

$$\text{IDF}(\boldsymbol{w}) = \boldsymbol{log}\left(\frac{|\boldsymbol{D}|}{\boldsymbol{0}}\right)$$

Remaining issues

TF-IDF of word that appears in every corpus is 0

We probably don't want this

TF-IDF of word $\boldsymbol{w}$ that never appears:

$$\text{TF}(\boldsymbol{w}) = 0$$

$$\text{IDF}(\boldsymbol{w}) = \boldsymbol{log}\left(\frac{|\boldsymbol{D}|}{\boldsymbol{0}}\right)$$

Can't divide by 0

Smoothing

If you saw something happen **1 out of 3 times**, is its probability really **1/3**?

If you saw something happen **0 out of 3 times**, is its probability really **0**?

If you saw something happen **3 out of 3 times**, is its probability really **1**?

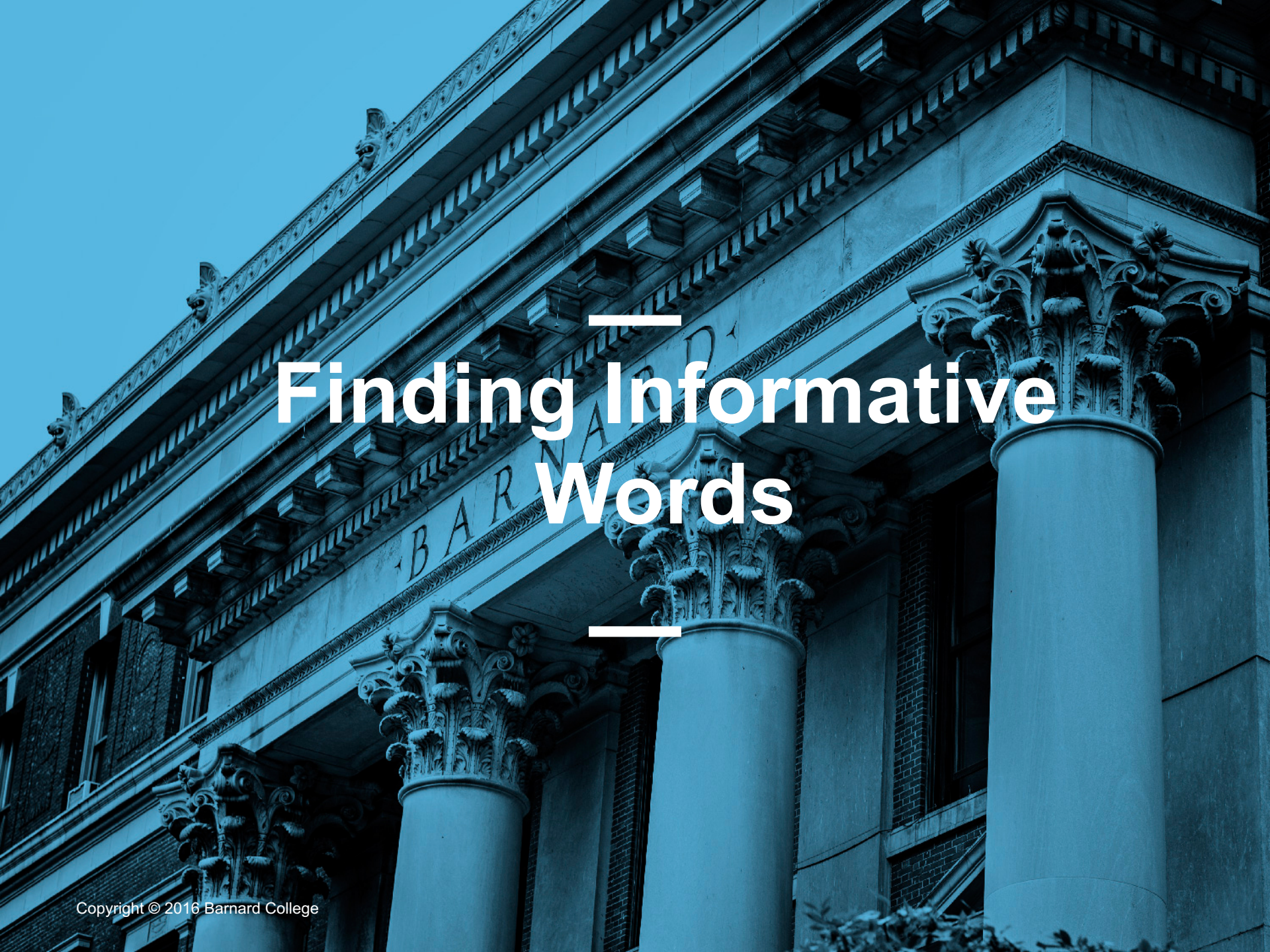
Slide from Jason Eisner

Smoothing - IDF

Let's add one document that contains each word

Smoothing IDF:

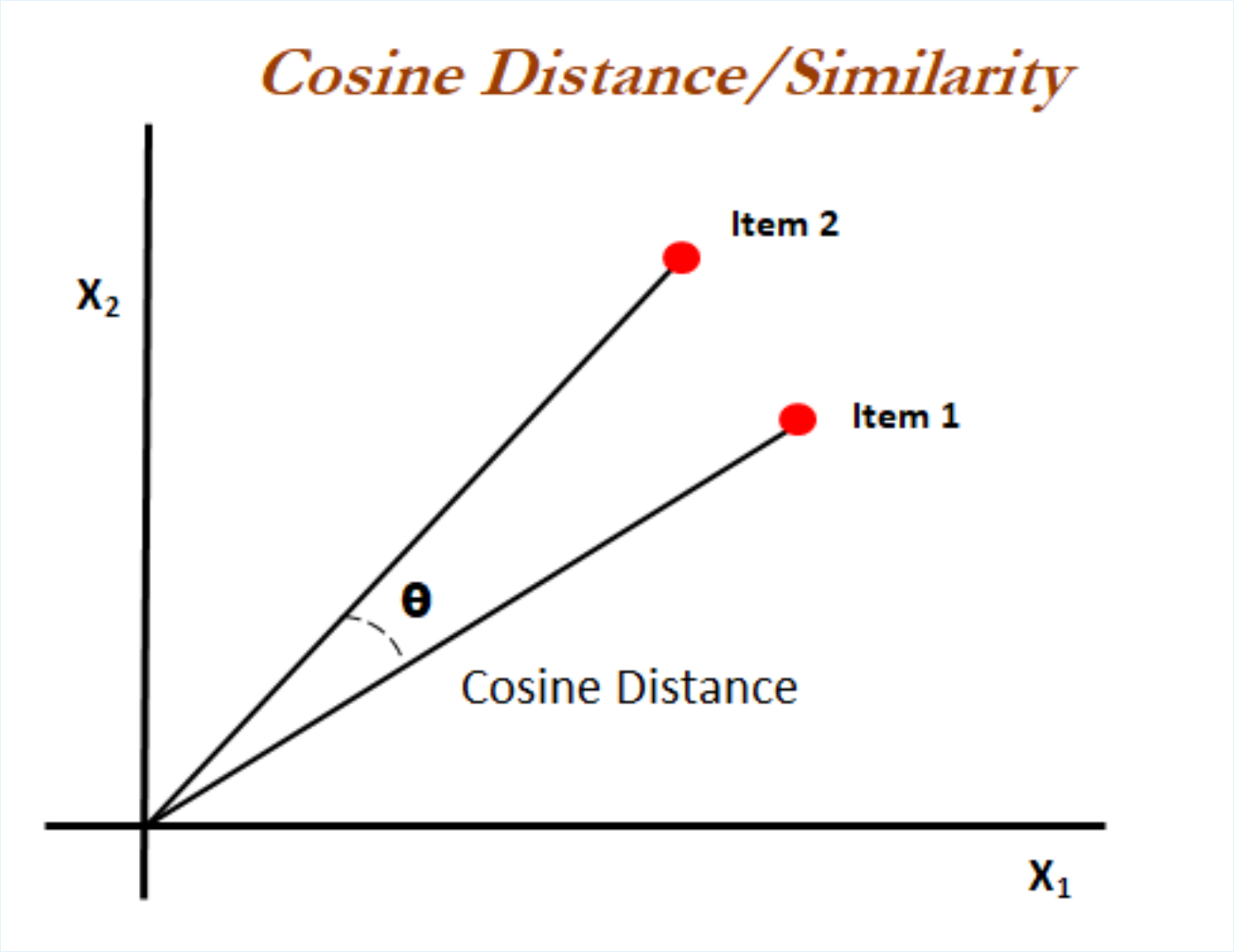
$$\mathbf{IDF(w)} = \boldsymbol{log}\left(\frac{|\,\boldsymbol{D}\,|}{|\,\boldsymbol{tf(w,d)\neq 0}\,|+\mathbf{1}}\right)+\mathbf{1}$$



Finding Informative Words

Comparing documents

Cosine Simalrity



Cosine Similarity

Similarity defined as:

cosine of the angle between the vectors

Compute $\cos(\theta)$:

the normalized dot product of vectors A and B

Dot product of A and B:

$$A * B = \sum_{i}^{n} a_i b_i$$