



CSCE 670 - Information Storage and Retrieval

Lecture 4: BM25, Probabilistic Model

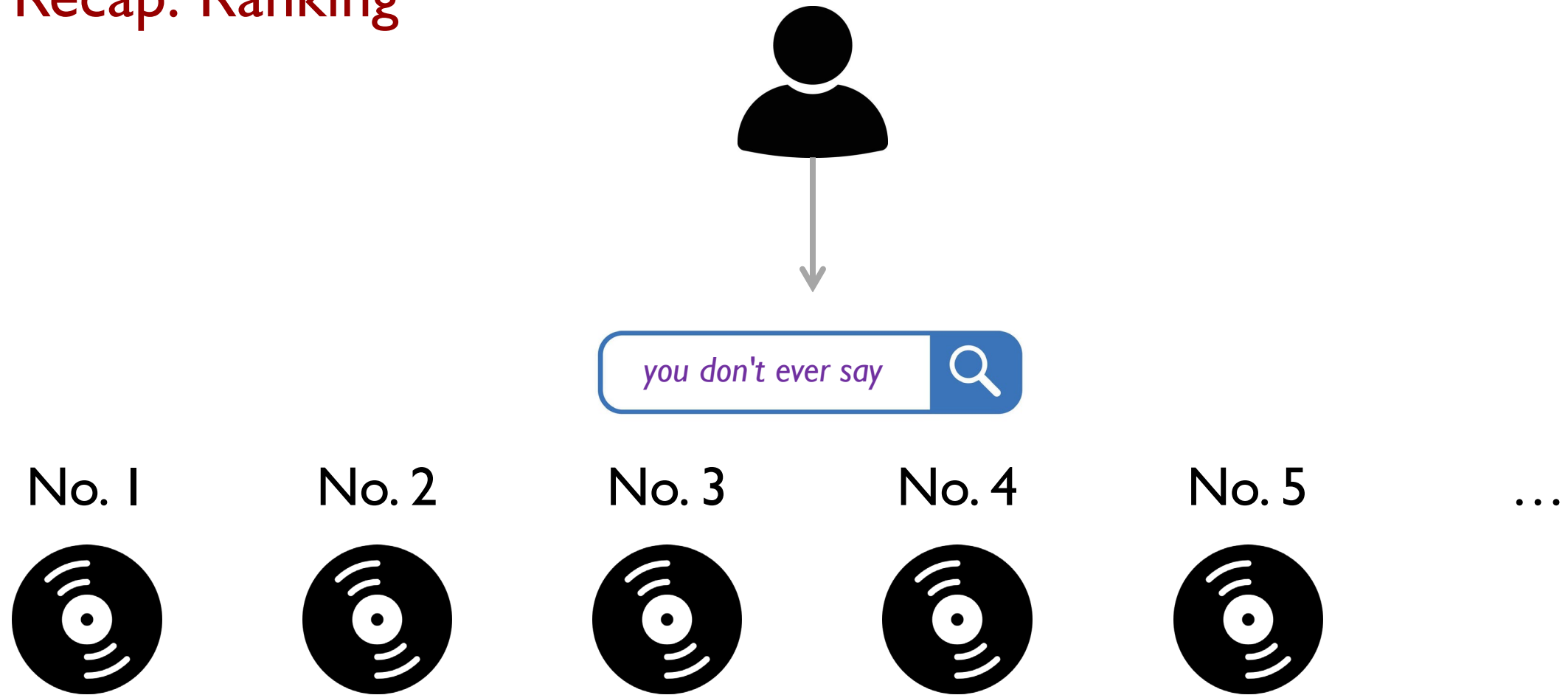
Yu Zhang

yuzhang@tamu.edu

September 4, 2025

Course Website: <https://yuzhang-teaching.github.io/CSCE670-F25.html>

Recap: Ranking



We return a **ranked** list of albums.

Recap: TF-IDF




- Given a query q and a document d , we want to calculate $\text{Score}(q, d)$.
- The query q may have one or more terms.
- For each term $t \in q$, we consider two factors:
 - **Term Frequency (TF)**: the number of times it occurs in the document
 - **Inverse Document Frequency (IDF)**: how rare it is across all documents: $\log \frac{|\mathcal{D}|}{|d \in \mathcal{D}: t \in d|}$

$$\text{tfidf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t$$

- The overall score is the sum of the TF-IDF scores of all terms.

$$\text{Score}(q, d) = \sum_{t \in q} \text{tfidf}_{t,d}$$

Our Plan: Ranking

-  Why is ranking important?
-  What factors impact ranking?
- Two foundational text-based approaches
 -  TF-IDF
 - BM25
- Two foundational link-based approaches
 - PageRank
 - HITS
- Machine-learned ranking (“learning to rank”)

BM25 (or Okapi BM25)

- **BM** = Best Match
- **25** = the 25th version of the scoring function
- Over time, BM25 has become a default scoring function used broadly across many real-world systems.
- Goal: be sensitive to term frequency and document length while not adding too many parameters

History of BM25: The 1994 Text REtrieval Conference (TREC-3)

Overview of the Third Text REtrieval Conference (TREC-3)

Donna Harman

**National Institute of Standards and Technology
Gaithersburg, MD. 20899**

1. Introduction

In November of 1992 the first Text REtrieval Conference (TREC-1) was held at NIST [Harman 1993]. The conference, co-sponsored by ARPA and NIST, brought together information retrieval researchers to discuss their system results on a new large test collection (the TIPSTER collection). This conference became the first in a series of ongoing conferences dedicated to encouraging research in retrieval from large-scale test collections, and to encouraging increased interaction among research groups in industry and academia. From the beginning there has been an almost equal number of universities and companies

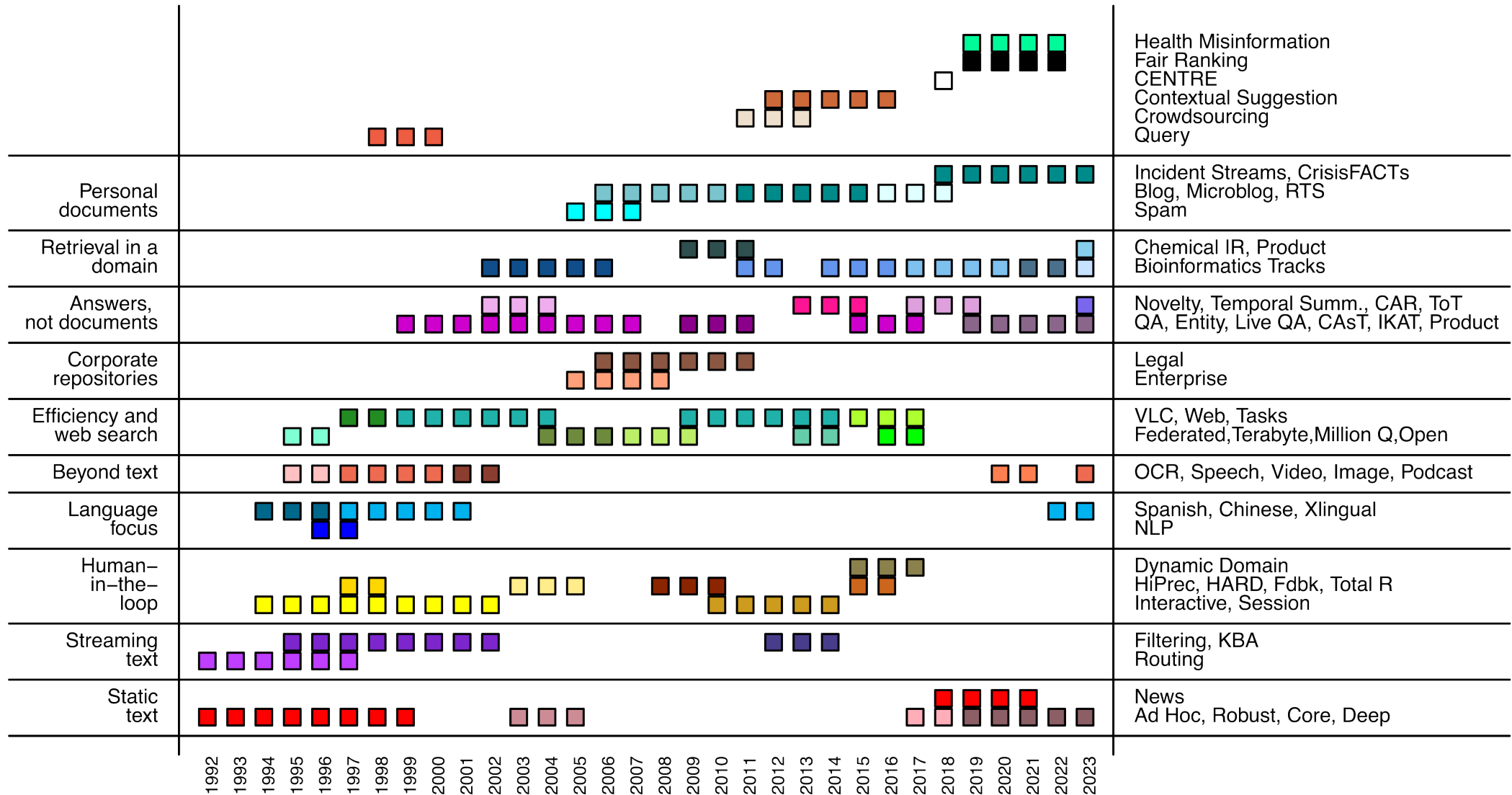
description of the test collection being used, and an overview of the results. The papers from the individual groups should be referred to for more details on specific system approaches.

2. The Task and the Participants

The three TREC conferences have all centered around two tasks based on traditional information retrieval modes: a "routing" task and an "ad hoc" task. In the routing task it is assumed that the same questions are always being asked, but that new data is being searched. This task is similar to that done by news clipping services or by li-

TREC: <https://trec.nist.gov/>

IR challenges held each year, featuring expert-annotated data (e.g., relevant query-document pairs) from various domains



TREC-3

Table 1: TREC-3 Participants (14 companies, 19 universities)

Australian National University	Bellcore
Carnegie Mellon University/CLARITECH	CITRI, Australia
City University, London	Cornell University
Dublin City University	Environment Research Institute of Michigan
Fulcrum	George Mason University
Logicon Operating Systems	Mayo Clinic/Foundation
Mead Data Central	National Security Agency
New York University	NEC Corporation
Queens College	Rutgers University (two groups)
Siemens Corporate Research Inc.	Swiss Federal Institute of Technology (ETH)
TRW/Paracel	Universitaet Dortmund, Germany
University of California - Berkeley	University of Central Florida
University of Massachusetts at Amherst	VPI&SU (Virginia Tech)
University of Minnesota	University of Toronto
Universite de Neuchatel, Switzerland	Verity Inc.
West Publishing Co.	Xerox Palo Alto Research Center

TREC-3

Okapi at TREC-3

S E Robertson

S Walker

S Jones

M M Hancock-Beaulieu

M Gatford

Centre for Interactive Systems Research

Department of Information Science

City University

Northampton Square

London EC1V 0HB

UK

Advisers: E Michael Keen (University of Wales, Aberystwyth), Karen Sparck Jones (Cambridge University), Peter Willett (University of Sheffield)

1 Introduction

The sequence of TREC conferences has seen the City University Okapi IR system evolve in several ways. Be-

City at TREC-2

For TREC-2 the simple inverse collection frequency (ICF) term-weighting scheme was elaborated to embody within-document frequency and document length components, as well as within-query frequency, and a large number of weighting functions were investigated. Because of hardware failures few of the runs were ready in time, and City's official results were very poor. However, later automatic ad hoc and routing results are

Stephen Robertson (1946-)

- Professor at City University London
- Principal Researcher at Microsoft Research Cambridge
- Developed the BM25 ranking function
- A key contributor to the probabilistic model of IR
- Involved in the development of the TREC evaluation program
- Gerard Salton Award (2000)



What is BM25?
Why are we talking about a method from 1994?

BM25 in (Almost) One Slide

$$\text{BM25}(q, d) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{\text{TF}(t, d) \cdot (k_1 + 1)}{\text{TF}(t, d) + k_1(1 - b + b \cdot \frac{|d|}{\text{avgdl}})}$$

- k_1 controls term frequency scaling
 - $k_1 = 0$: binary model
 - k_1 very large: raw term frequency
- b controls document length normalization
 - $b = 0$: no document length normalization
 - $b = 1$: relative frequency (full document length normalization)
- Typically, k_1 is set between 1.2 and 2; b is set around 0.75
- $|d|$ is the length of d (in words); avgdl = average document length (in words)

The IDF Component in BM25

$$\text{IDF}(t) = \log \left(\frac{N - n(t) + 0.5}{n(t) + 0.5} + 1 \right)$$

- $N = |\mathcal{D}|$, number of documents
- $n(t) = |\{d \in \mathcal{D} : t \in d\}|$, number of documents containing the term t

Why are we talking about a method from 1994?

- Even compared to today's large language models, BM25 still delivers strong performance on text retrieval tasks. It is also **easy to implement**, requires **no machine learning training**, and does **not rely on GPU support**.

OPINION

The Neural Hype and Comparisons Against Weak Baselines

Jimmy Lin

David R. Cheriton School of Computer Science, University of Waterloo

Setup

- Two recent IR papers: Paper 1 and Paper 2
 - Several “neural” models proposed/compared in the papers
- “Traditional” (non-neural) ranking models:
 - BM25
 - QL: query likelihood with Dirichlet smoothing
 - RM3 variant of relevance models (pseudo-relevance feedback where top results returned by a QL model are treated as relevant)
- Experiments in Anserini (open-source engine built on Lucene)

Results

Condition	AP	P20
QL	0.2499	0.3556
QL + RM3	0.2865	0.3773
Neural ₁	0.2815	0.3752
Neural ₂	0.2801	0.3764
Neural ₃	0.2856	0.3766
Neural ₃ '	0.2971	0.3948
Anserini: QL	0.2496	0.3543
Anserini: BM25	0.2526	0.3604
Anserini: BM25 + RM3 (independent)	0.2954	0.3885
Anserini: BM25 + RM3 (joint)	0.2973	0.3871

Table 1: Comparison of Anserini results to Paper 1.

Condition	AP	P20
BM25	0.238	0.354
BM25 + Features	0.250	0.367
Neural _x	0.258	0.372
Neural _y	0.256	0.370
Neural _x + Neural _y	0.259	0.373
A + Neural _y	0.263	0.380
A + Neural _y + M	0.265	0.380
B + Neural _y	0.270	0.383
B + Neural _y + M	0.272	0.386
Anserini: QL	0.2481	0.3517
Anserini: BM25	0.2528	0.3598
Anserini: BM25 + RM3 (independent)	0.2991	0.3901
Anserini: BM25 + RM3 (joint)	0.2956	0.3931

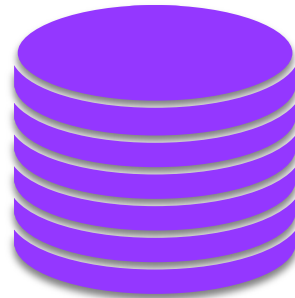
Table 2: Comparison of Anserini results to Paper 2.

How to interpret the BM25 scoring function?
Why is it so complicated?

Generative Model for Documents

- Let's consider a **probabilistic** interpretation.
- Idea: Words are drawn **independently** from the vocabulary using a multinomial distribution.

$p(\text{meet}) = 0.002$
 $p(\text{midnight}) = 0.001$
 $p(\text{the}) = 0.015$
...



... *meet me at midnight* ...

- If we use this model to generate a sentence
 - What is the probability that the first word is “*midnight*”?
 - What is the probability that the second word is “*midnight*”, given the first word is “*midnight*”?

Generative Model for Documents

- Given a specific term (e.g., “*midnight*”)
 - The distribution of its term frequency (TF) follows a **binomial** distribution.



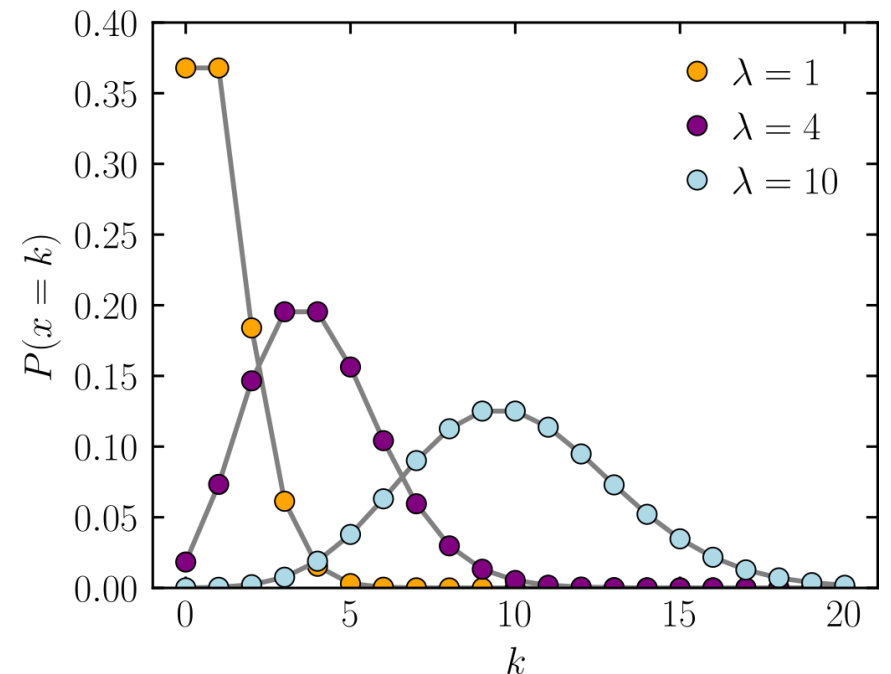
- Binomial distribution:** There is a coin for which the probability of landing heads up in a single toss is **0.001**, and tails up is **0.999**. What is the probability of getting heads exactly k times in **100** tosses? ($k = 0, 1, 2, \dots, 100$)

$$p_k = \binom{100}{k} \times 0.001^k \times 0.999^{100-k}$$

Generative Model for Documents

- When the number of tosses is large (e.g., ≥ 100) and the probability of getting heads is small, a **binomial** distribution can be approximated by a **Poisson** distribution.
- **Poisson distribution**: A call center receives incoming calls at an average rate of λ (i.e., λ calls per hour). The time interval between successive calls is independent of the time of the previous call. Under these conditions, the probability that the call center receives k calls ($k = 0, 1, 2, \dots$) in one hour is given by:

$$p_k = \frac{\lambda^k e^{-\lambda}}{k!}$$



Generative Model for Documents

- Why can we also approximate TF as following a Poisson distribution?
- Assume all documents have the same length (e.g., 100 words). *We will fix this later!*
- Given a word (e.g., “*midnight*”)
 - $p(\textit{midnight}) = 0.001$
 - “*midnight*” should “arrive” at an average rate of $0.001 \times 100 = 0.1$ per document.
 - The number of words between two occurrences of “*midnight*” is independent of the position of the previous “*midnight*”.
 - TF = the number of occurrences of “*midnight*” in a document

“One” Poisson Model

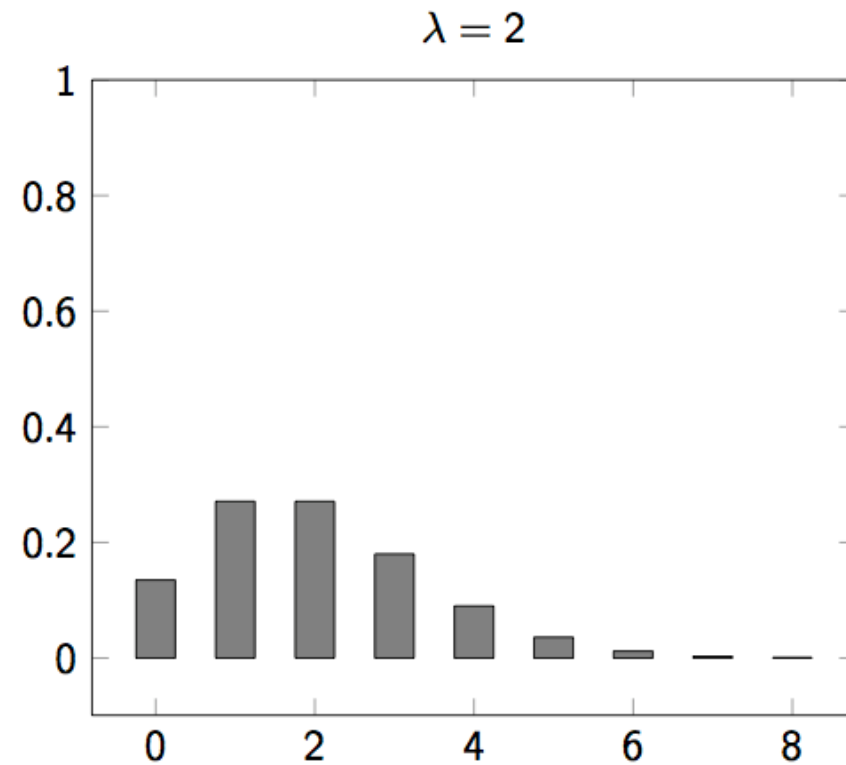
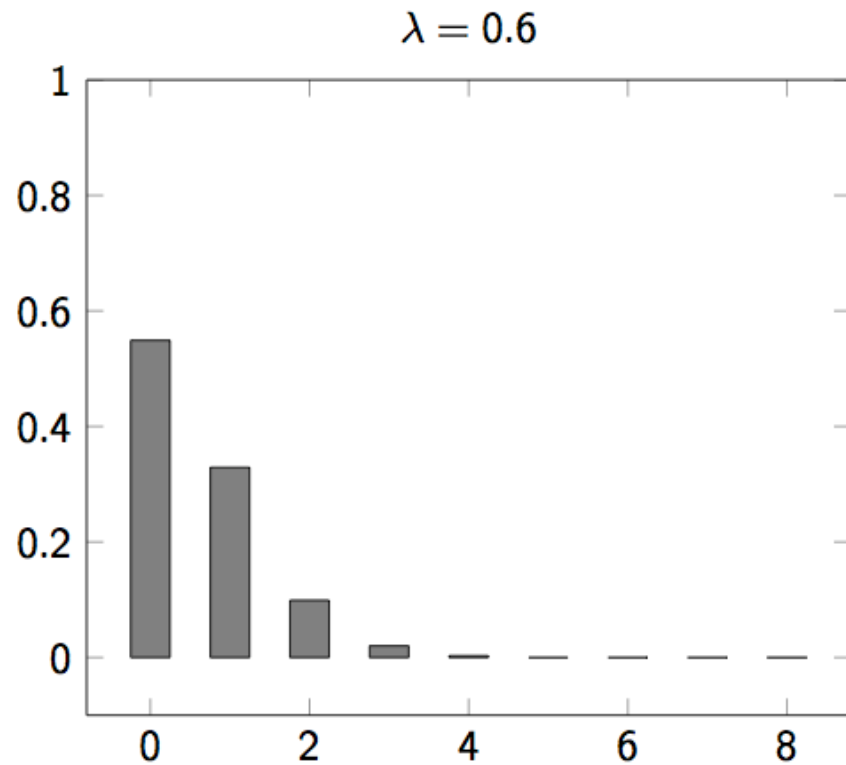
- Given 10,000 documents and a word “*information*”
 - For each document, “*information*” has a term frequency.
 - $tf_1, tf_2, \dots, tf_{10000}$
 - How should these number be distributed?
- Assume “*information*” appears M times in total in these 10,000 documents, then

$$\lambda = \frac{M}{10000}$$

(Definition of λ : the average number of occurrences in a document)

“One” Poisson Model

- $tf_1, tf_2, \dots, tf_{10000}$ should follow a Poisson distribution with $\lambda = \frac{M}{10000}$.



Theory vs. Practice

Table 1. Frequency Distributions for 19 Word Types and Expected Frequencies Assuming a Poisson Distribution with $\lambda = 53/650$

Frequency	Word Type	Number of Documents Containing k Tokens													
		k	0	1	2	3	4	5	6	7	8	9	10	11	12
51	act	608	35	5	2										
51	actions	617	27	2	0	2	0	2							
54	attitude	610	30	7	2	1									
52	based	600	48	2											
53	body	605	39	4	2										
52	castration	617	22	6	3	1	1								
55	cathexis	619	22	3	2	1	2	0	1						
51	comic	642	3	0	1	0	0	0	0	0	0	1	1	2	
53	concerned	601	45	4											
53	conditions	604	39	7											
55	consists	602	41	7											
53	factor	609	32	7	1	1									
52	factors	611	27	11	1										
55	feeling	613	26	7	3	0	0	1							
52	find	602	45	2	1										
54	following	604	39	6	1										
51	force	603	43	4											
51	forces	609	33	6	2										
52	forgetting	629	11	3	2	2	1	1	0	0	0	1			
53	expected, assuming Poisson distribution	599	49	2											

Limitations of the “One” Poisson Model

- In the table on the previous slide
 - According to the theory, it is almost impossible for a word to appear ≥ 5 times in a document.
 - In practice, a word can appear 10, 11, or 12 times in a document.
 - Why?
- Each word can be a background word or a topic-specific word in a document.
 - “information” may be a topic-specific word in information retrieval papers. (*Discussions revolve around the term “information retrieval”.*)
 - “information” may be a background word in sports news. (*Discussions typically do not revolve around this word; it is only used when generally needed.*)
- “One” Poisson model gives a reasonable fit for background words but a poor fit for topic-specific words.

“Two” Poisson Model

- $\pi \in [0,1]$: how “relevant” a word t is to a document d
 - $\pi = 0$: totally irrelevant $\rightarrow t$ is a **background** word in d
 - $\pi = 1$: totally relevant $\rightarrow t$ is a **topic-specific** word in d
 - $\pi \in (0,1)$: somewhere in between
- When t is a **background** word, the distribution of its TF follows a Poisson distribution:

$$P(\text{tf} = k | \pi = 0) = \frac{\mu^k e^{-\mu}}{k!}$$

- When t is a **topic-specific** word, the distribution of its TF follows another Poisson distribution:

$$P(\text{tf} = k | \pi = 1) = \frac{\lambda^k e^{-\lambda}}{k!}$$

“Two” Poisson Model

- $\pi \in (0,1)$: somewhere in between

$$P(\text{tf} = k|\pi) = \pi \frac{\lambda^k e^{-\lambda}}{k!} + (1 - \pi) \frac{\mu^k e^{-\mu}}{k!}$$

- Given 10,000 documents and a word “*information*”
 - How can we know the distribution?
 - Hard! Because π is a **hidden** variable for each document.
 - Unlike **observed** variables (e.g, TF) that can be directly calculated

“Two” Poisson Model

- Learning latent variable models from large-scale text data used to be a central problem in IR, NLP, and even machine learning in the broader sense.

[\[PDF\] Probabilistic latent semantic indexing](#)

[T Hofmann](#) - Proceedings of the 22nd annual international ACM ..., 1999 - [dl.acm.org](#)

... success in different domains including automatic **indexing** (**Latent Semantic Indexing**, LSI)

[1... approach to LSA and factor analysis { called **Probabilistic Latent** Semantic Analysis (PLSA) { ...

☆ Save [Cite](#) Cited by 7913 Related articles All 29 versions [↗](#)

[\[PDF\] acm.org](#)

[Full View](#)

[\[PDF\] Latent dirichlet allocation](#)

[DM Blei](#), [AY Ng](#), [MI Jordan](#) - Journal of machine Learning research, 2003 - [jmlr.org](#)

We describe **latent Dirichlet allocation** (LDA), a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in ...

☆ Save [Cite](#) Cited by 57586 Related articles All 92 versions Web of Science: 23713 [↗](#)

[\[PDF\] jmlr.org](#)

[Discover Full](#)

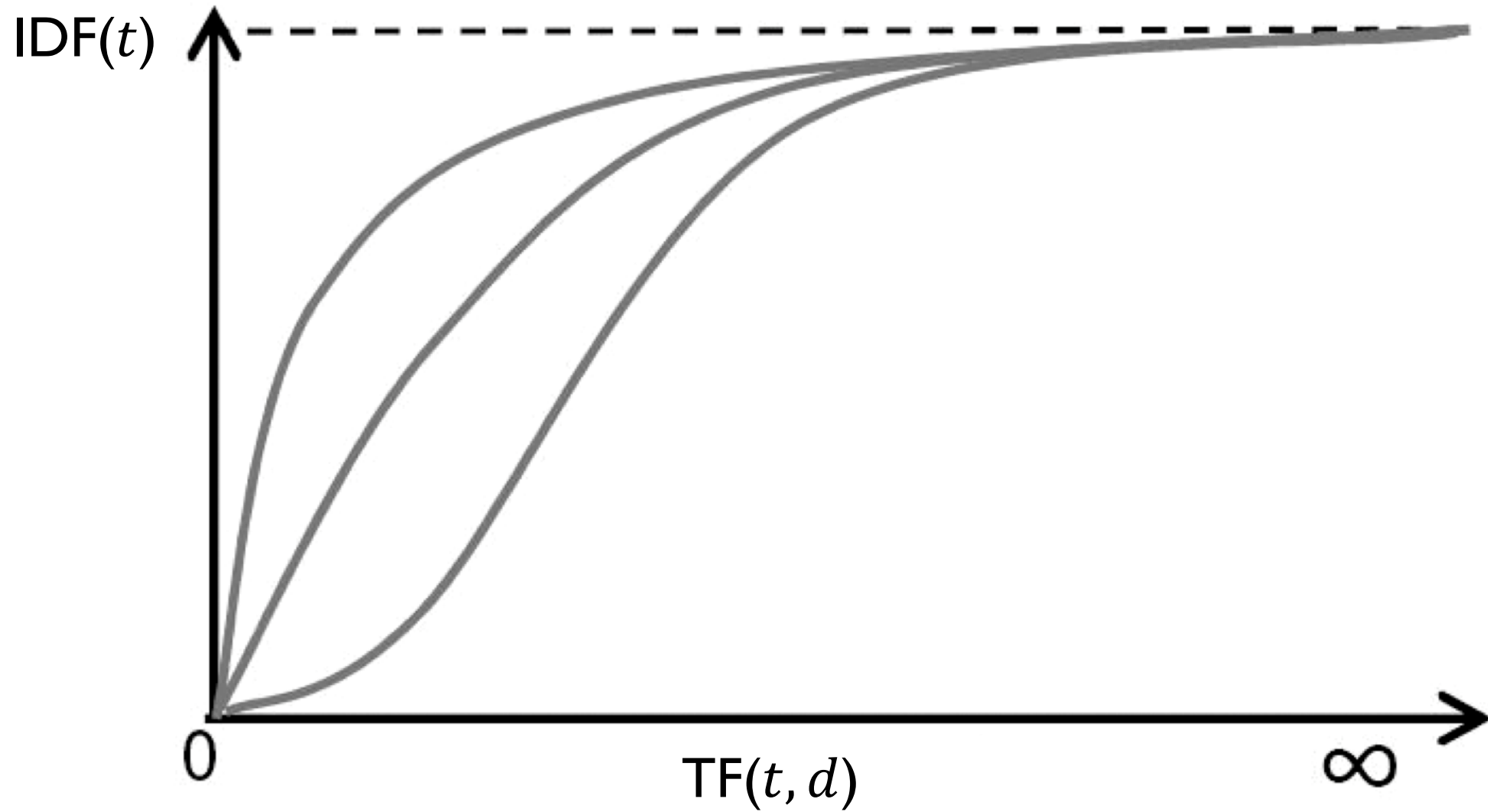
Let's make this simpler (?)

- Robertson-Spärck-Jones (**RSJ**) model:

$$\text{score}(q, d) = \sum_{t \in q, \text{TF}(t, d) > 0} \log \frac{P(\text{tf} = \text{TF}(t, d) | \pi = 1) \times P(\text{tf} = 0)}{P(\text{tf} = \text{TF}(t, d)) \times P(\text{tf} = 0 | \pi = 1)}$$

- If we plug the “**One**” **Poisson model** into RSJ and add some “okay” assumptions, we get TF-IDF.
- If we plug the “**Two**” **Poisson model** into RSJ, the formula will become complicated and ugly.
 - BUT it is monotonically increasing to $\text{IDF}(t)$ when $\text{TF}(t, d) \rightarrow +\infty$!

RSJ + “Two” Poisson Model



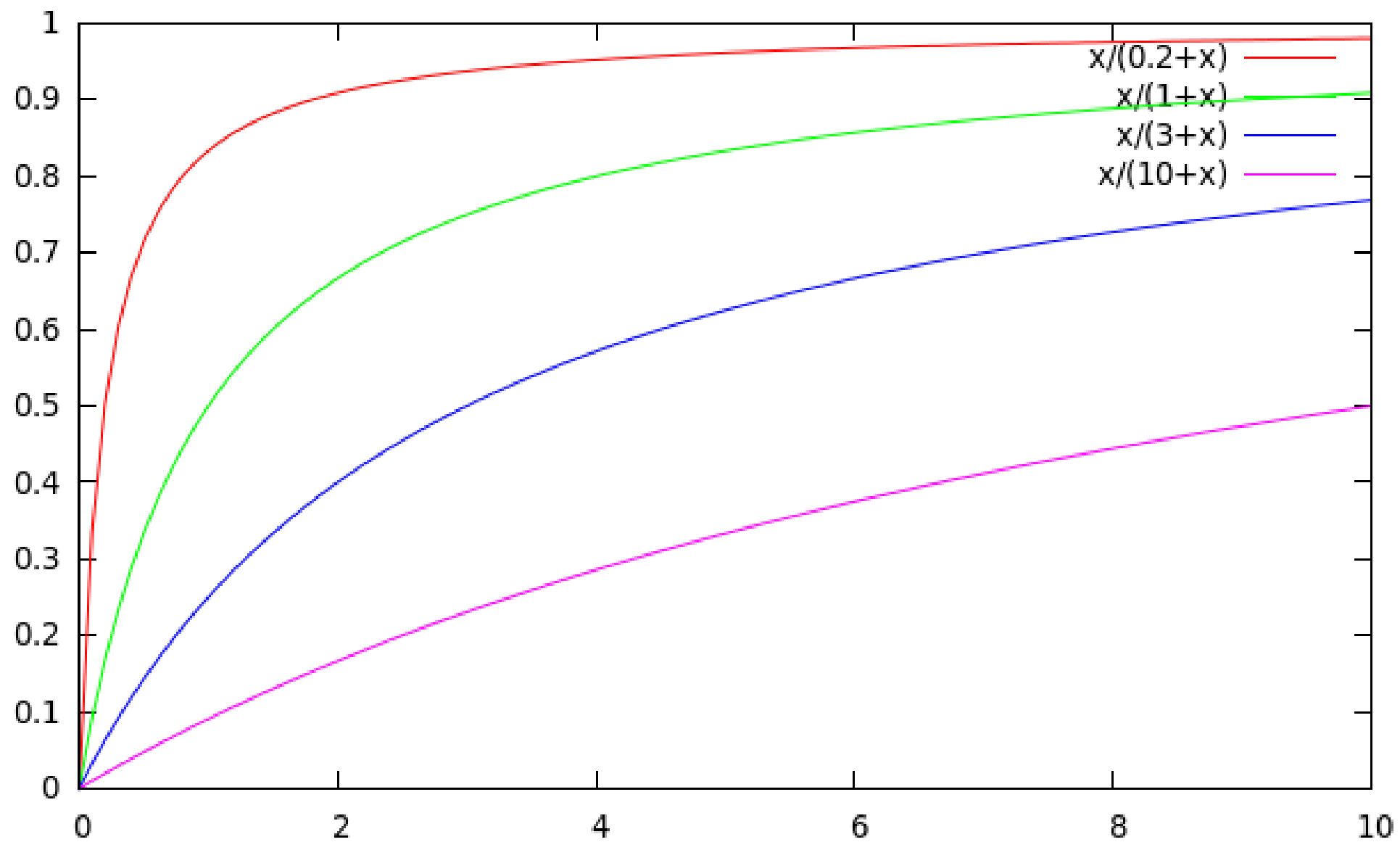
Let's make this simpler!

$$\text{score}(q, d) = g(\text{TF}(t, d)) \times \text{IDF}(t)$$

- g saturates toward a maximum value of 1, which is not true for simple TF-IDF scoring.
 - Think of raw TF or even log-based TF
- Let's approximate g with a simple parametric curve that has the same qualitative properties.

$$\frac{\text{TF}(t, d)}{k_1 + \text{TF}(t, d)}$$

- $k_1 > 0$ is a hyperparameter
- $\frac{\text{TF}(t, d)}{k_1 + \text{TF}(t, d)} = 0$ when $\text{TF}(t, d)$ is 0
- $\frac{\text{TF}(t, d)}{k_1 + \text{TF}(t, d)} = 1$ when $\text{TF}(t, d)$ is ∞



An Early Version of BM25

$$\text{score}(q, d) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{\text{TF}(t, d) \cdot (k_1 + 1)}{\text{TF}(t, d) + k_1}$$

- The $(k_1 + 1)$ factor does not change ranking, but makes $\frac{\text{TF}(t, d) \cdot (k_1 + 1)}{\text{TF}(t, d) + k_1} = 1$ when $\text{TF}(t, d) = 1$.
- The model is similar to TF-IDF, but the term frequency part is bounded.

Wait! There is also the factor of document length.

- Longer documents are likely to have larger TF values
- Why might documents be longer?
 - **Verbosity**: suggests observed TF too high
 - **Larger Scope**: suggests observed TF may be accurate
- A real document collection probably has both effects, so we should apply some kind of “partial” normalization.
- **Length normalization factor**: $B = 1 - b + b \cdot \frac{|d|}{\text{avgdl}}$
 - $|d|$ is the length of d (in words); avgdl = average document length (in words)
 - $b \in [0,1]$ is a hyperparameter

Document Length Normalization

- $B = 1 - b + b \cdot \frac{|d|}{\text{avgdl}}$
- What if the length of d is exactly the average document length?
 - $B = 1 - b + b = 1$, no need to normalize
- What if d is longer than average (e.g., $|d| = 2 \times \text{avgdl}$)?
 - $B = 1 - b + 2b = 1 + b > 1$
- What if d is shorter than average (e.g., $|d| = 0.5 \times \text{avgdl}$)?
 - $B = 1 - b + 0.5b = 1 - 0.5b < 1$
- $b = 1$: full document length normalization
- $b = 0$: no document length normalization

Putting it all together ...

$$\text{BM25}(q, d) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{\text{TF}(t, d) \cdot (k_1 + 1)}{\text{TF}(t, d) + k_1 (1 - b + b \cdot \frac{|d|}{\text{avgdl}})}$$

$= B$

- k_1 controls term frequency scaling
 - $k_1 = 0$: binary model
 - k_1 very large: raw term frequency
- b controls document length normalization
 - $b = 0$: no document length normalization
 - $b = 1$: relative frequency (full document length normalization)
- Typically, k_1 is set between 1.2 and 2; b is set around 0.75
- $|d|$ is the length of d (in words); avgdl = average document length (in words)

Example

- Query q : “any zebra”
- Document d : “zebra any love any”
- 10,000 documents; “any” appears in 1,000 of them; “zebra” appears in 10 of them
- $\text{avgdl} = 10$
- $k_1 = 1.2$
- $b = 0.75$

$$\begin{aligned}\text{BM25}(q, d) &= \sum_{t \in q} \text{IDF}(t) \cdot \frac{\text{TF}(t, d) \cdot (k_1 + 1)}{\text{TF}(t, d) + k_1 (1 - b + b \cdot \frac{|d|}{\text{avgdl}})} \\ &= \sum_{t \in q} \text{IDF}(t) \cdot \frac{\text{TF}(t, d) \cdot (1.2 + 1)}{\text{TF}(t, d) + 1.2 \times (1 - 0.75 + 0.75 \cdot \frac{4}{10})} = \sum_{t \in q} \text{IDF}(t) \cdot \frac{2.2 \times \text{TF}(t, d)}{\text{TF}(t, d) + 0.55}\end{aligned}$$

Example

- Query q : “any zebra”
- Document d : “zebra any love any”
- 10,000 documents; “any” appears in 1,000 of them; “zebra” appears in 10 of them

$$\text{IDF}(t) = \log \left(\frac{N - n(t) + 0.5}{n(t) + 0.5} + 1 \right)$$

- $\text{IDF}(\text{“any”}) = \log \left(\frac{10000 - 1000 + 0.5}{1000 + 0.5} + 1 \right) = 0.9998$
- $\text{IDF}(\text{“zebra”}) = \log \left(\frac{10000 - 10 + 0.5}{10 + 0.5} + 1 \right) = 2.9789$

Example

- Query q : “any zebra”
- Document d : “zebra any love any”

$$\begin{aligned}\text{BM25}(q, d) &= \sum_{t \in q} \text{IDF}(t) \cdot \frac{2.2 \times \text{TF}(t, d)}{\text{TF}(t, d) + 0.55} \\ &= \text{IDF}(\text{any}) \cdot \frac{2.2 \times 2}{2 + 0.55} + \text{IDF}(\text{zebra}) \cdot \frac{2.2 \times 1}{1 + 0.55} = 5.9533\end{aligned}$$

Questions?

BM25 in Homework 1, Quiz 1, and Final

- Homework I
 - You need to implement the BM25 scoring function.
- Quiz I and Final Exam
 - You should be familiar with the binomial generative model and the “One” Poisson model.
 - You should know document length normalization in the BM25 scoring function.
 - You do NOT need to master the derivation of the “Two” Poisson model and the Robertson-Spärck-Jones model.
 - You will NOT be required to manually calculate the BM25 score.



Thank You!

Course Website: <https://yuzhang-teaching.github.io/CSCE670-F25.html>