Probabilistic IR: BM25

Alfan Farizki Wicaksono

Fakultas Ilmu Komputer, Universitas Indonesia

BM25 Scoring Regime

$$RSV_{BM25} = \sum_{t \in Q \cap D} \log\left(\frac{N}{df_t}\right) \frac{(k_1 + 1).\, tf_t}{k_1\left((1-b) + b\,\frac{dl}{avdl}\right) + tf_t}$$

?

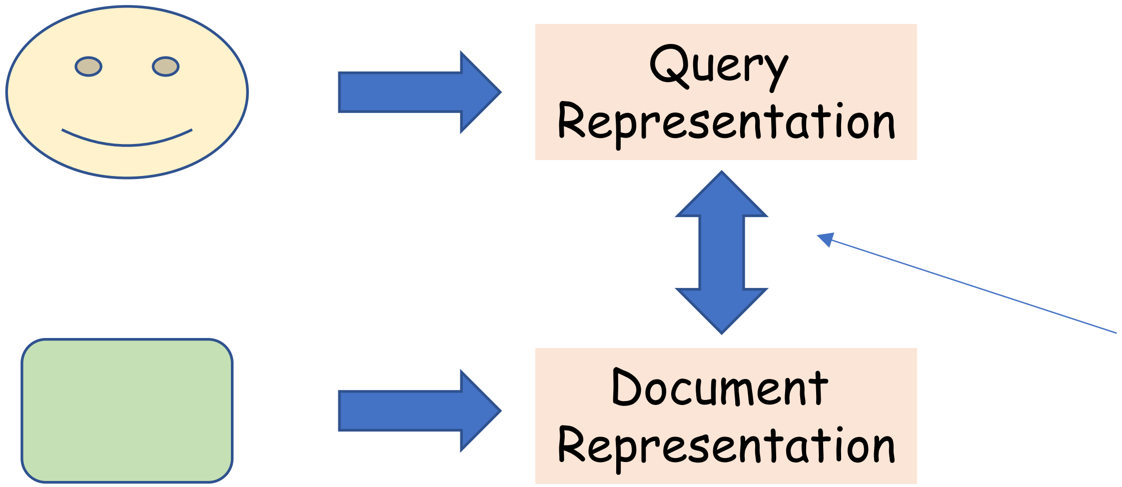
Selanjutnya kita akan belajar bagaimana BM25 bisa tercipta …

Harap sabar dan tekun, karena akan banyak notasi matematis ☺

Uncertainties in IR

Memahami information need
melibatkan ketidakpastian

User Information Need



Document

How well documents satisfy information need? Seberapa yakin dokumen tersebut relevan terhadap information need?

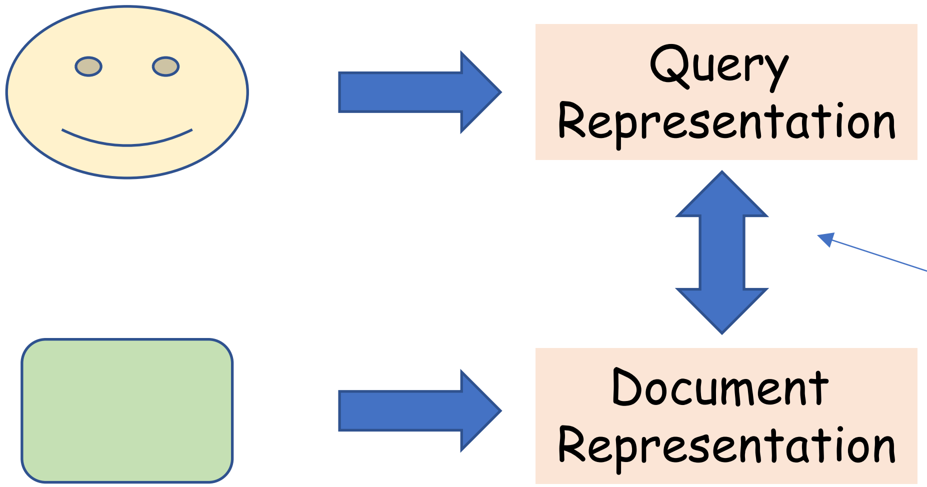
In Boolean Retrieval Models & Vector Space Models of IR, matching is done in a formally defined **but semantically imprecise calculus of index terms**.

Probabilities in IR

Teori probabilitas merupakan *principled foundation* untuk melakukan *reasoning* yang melibatkan ketidakpastian!

Memahami information need melibatkan ketidakpastian

User Information Need

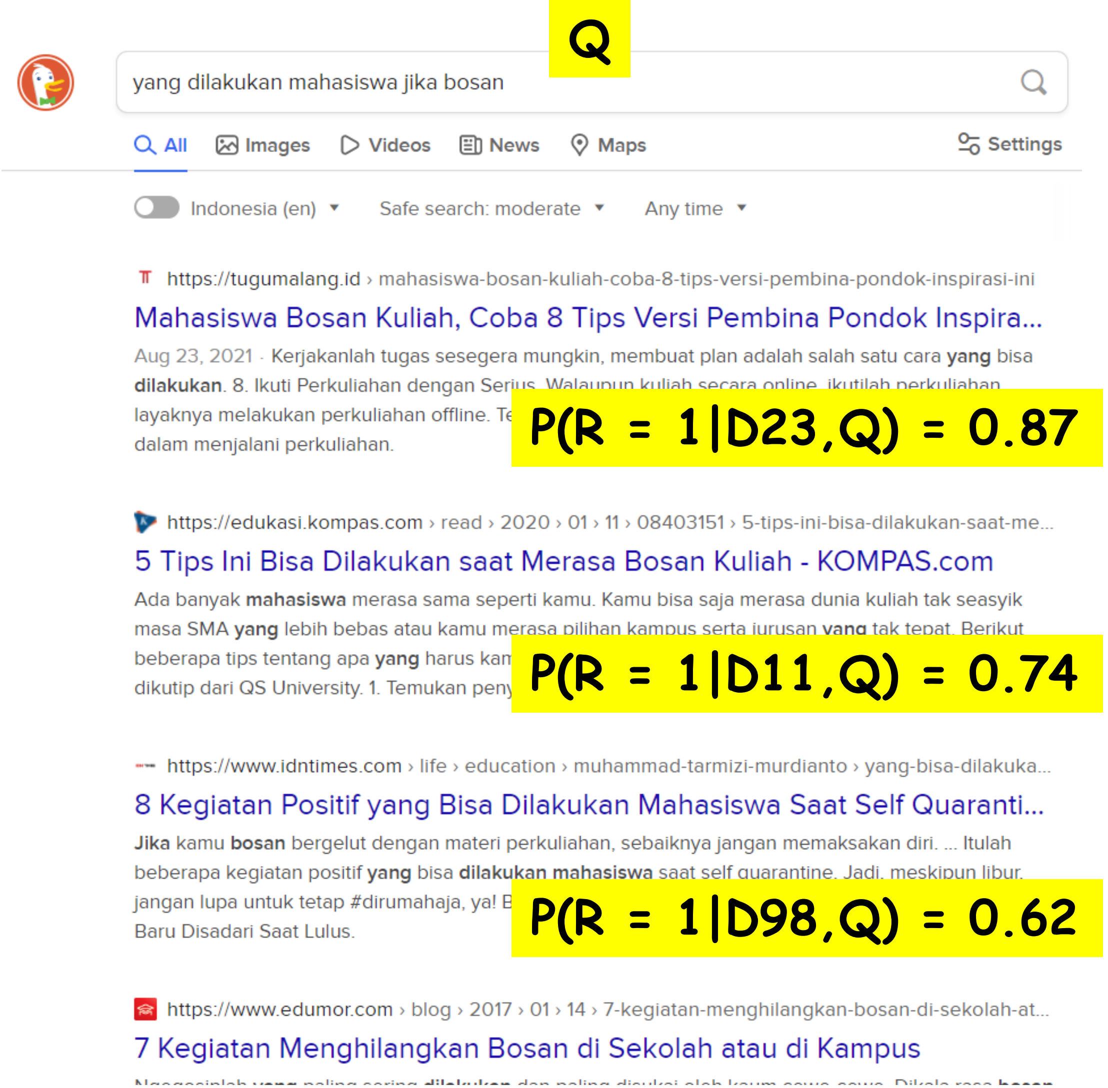


Document

How well documents satisfy information need? Seberapa yakin dokumen tersebut relevan terhadap information need?

In Boolean Retrieval Models & Vector Space Models of IR, matching is done in a formally defined **but semantically imprecise calculus of index terms**.

Probabilistic Ranking Problem



Ide:

- dokumen-dokumen di-ranking berdasarkan nilai probabilitas relevansi dokumen terhadap kebutuhan informasi; dan
- **disusun terurut mengecil** berdasarkan nilai probabilitas tersebut.

$$P(R = 1|d, q)$$

Sementara kita asumsikan relevan bersifat biner: relevan (1) atau tidak (0)

Probability Ranking Principle (PRP)

[1960s/1970s] S. Robertson, W. S. Cooper, M. E. Maroon

Van Rijsbergen 1979, 113-114

> "If a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of relevance to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose, the overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data."

Okapi BM25 [Robertson et al., 1994]

- BM25 “Best Match 25”
- Banyak sukses di kompetisi TREC (Text Retrieval Conference)

- **Goal:** be sensitive to term frequency and document length while not adding too many parameters

Jadi, Bagaimana Kita Melakukan Scoring?

Semuanya dimulai dari **Odds** di teori probabilitas.

$$score(Q,D) = O(R|Q,\overrightarrow{TF}) = \frac{P(R=1|Q,\overrightarrow{TF})}{P(R=0|Q,\overrightarrow{TF})}$$

TF vector representing D

Vocab = **{kernel, dan, model}**

D1: model, dan, kernel, kernel, dan, dan

Vektor TF D1 -> **[2, 3, 1]**

Asumsi: Kemunculan TF dari sebuah term **independent** dengan TF dari term lain!

Singkat Cerita … Kita Bermain dengan **Bayes' Rule**

$$O(R|Q,\overrightarrow{TF}) = \frac{P(R=1|Q,\overrightarrow{TF})}{P(R=0|Q,\overrightarrow{TF})}$$

$$= \frac{P(R=1|Q)P(\overrightarrow{TF}|R=1,Q)}{P(R=0|Q)P(\overrightarrow{TF}|R=0,Q)}$$

$$= O(R|Q)\prod_{i=1}^{n}\frac{P(TF_i|R=1,Q)}{P(TF_i|R=0,Q)}$$

Singkat Cerita … Kita Bermain dengan **Bayes' Rule**

$$O(R|Q,\overrightarrow{TF}) = \frac{P(R=1|Q,\overrightarrow{TF})}{P(R=0|Q,\overrightarrow{TF})}$$

$$= \boxed{\frac{P(R=1|Q)}{P(R=0|Q)}}\boxed{\frac{P(\overrightarrow{TF}|R=1,Q)}{P(\overrightarrow{TF}|R=0,Q)}}$$

$$= O(R|Q)\prod_{i=1}^{n}\frac{P(TF_i|R=1,Q)}{P(TF_i|R=0,Q)}$$

Ingat bahwa frekuensi sebuah term INDEPENDENT dengan TF dari term lain!

n = banyaknya term di vocab/dictionary

Konstan untuk sebuah query Q

Term Frequency dari Term i di dokumen

Singkat Cerita … Kita Bermain dengan **Bayes' Rule**

$$O(R|Q,\overrightarrow{TF}) = O(R|Q) \prod_{TF_i>0} \frac{P(TF_i|R=1,Q)}{P(TF_i|R=0,Q)} \prod_{TF_i=0} \frac{P(TF_i|R=1,Q)}{P(TF_i|R=0,Q)}$$

Jika kita asumsikan untuk semua term yang **tidak** muncul di Query:

$$P(TF_i = tf_i|R=1,Q) = P(TF_i = tf_i|R=0,Q)$$

$$O(R|Q,\overrightarrow{TF}) = O(R|Q) \prod_{\substack{TF_i>0 \\ TF_i^Q>0}} \frac{P(TF_i = tf_i|R=1,Q)P(TF_i = 0|R=0,Q)}{P(TF_i = tf_i|R=0,Q)P(TF_i = 0|R=1,Q)} \prod_{TF_i^Q>0} \frac{P(TF_i = 0|R=1,Q)}{P(TF_i = 0|R=0,Q)}$$

TF_i : Frekuensi term i di dokumen

TF_i^Q: Frekuensi term i di query

Singkat Cerita … Kita Bermain dengan **Bayes' Rule**

$$O(R|Q,\overrightarrow{TF}) = O(R|Q) \prod_{TF_i>0} \frac{P(TF_i|R=1,Q)}{P(TF_i|R=0,Q)} \prod_{TF_i=0} \frac{P(TF_i|R=1,Q)}{P(TF_i|R=0,Q)}$$

Jika kita asumsikan untuk semua term yang tidak muncul di Query:

$$P(TF_i = tf_i|R=1,Q) = P(TF_i = tf_i|R=0,Q)$$

$$O(R|Q,\overrightarrow{TF}) = \boxed{O(R|Q)} \boxed{\prod_{\substack{TF_i>0 \\ TF_i^Q>0}} \frac{P(TF_i = tf_i|R=1,Q)P(TF_i = 0|R=0,Q)}{P(TF_i = tf_i|R=0,Q)P(TF_i = 0|R=1,Q)}} \boxed{\prod_{TF_i^Q>0} \frac{P(TF_i = 0|R=1,Q)}{P(TF_i = 0|R=0,Q)}}$$

Konstan untuk sebuah query Q

Hanya ini yang diperlukan untuk Ranking!

Komponen yang hanya bergantung pada query -> konstan untuk sebuah query!

Scoring? -> Retrieval Status Value

$$score(Q,D) = RSV = \prod_{\substack{TF_i > 0 \\ TF_i^Q > 0}} \frac{P(TF_i = tf_i | R = 1, Q) P(TF_i = 0 | R = 0, Q)}{P(TF_i = tf_i | R = 0, Q) P(TF_i = 0 | R = 1, Q)}$$

Practical Issue!

Pakai LOG untuk menghindari underflow! Kok bisa pakai LOG?

Term yang muncul di dokumen dan query

$$= \sum_{\substack{TF_i > 0 \\ TF_i^Q > 0}} log \left(\frac{P(TF_i = tf_i | R = 1, Q) P(TF_i = 0 | R = 0, Q)}{P(TF_i = tf_i | R = 0, Q) P(TF_i = 0 | R = 1, Q)} \right)$$

Log(a*b*c) = Log a + Log b + Log c

Jadi …

Jadi, sebenarnya kita hanya perlu menghitung:

$$P(TF_i = tf_i | R, Q)$$

StatProb: Poisson Distribution

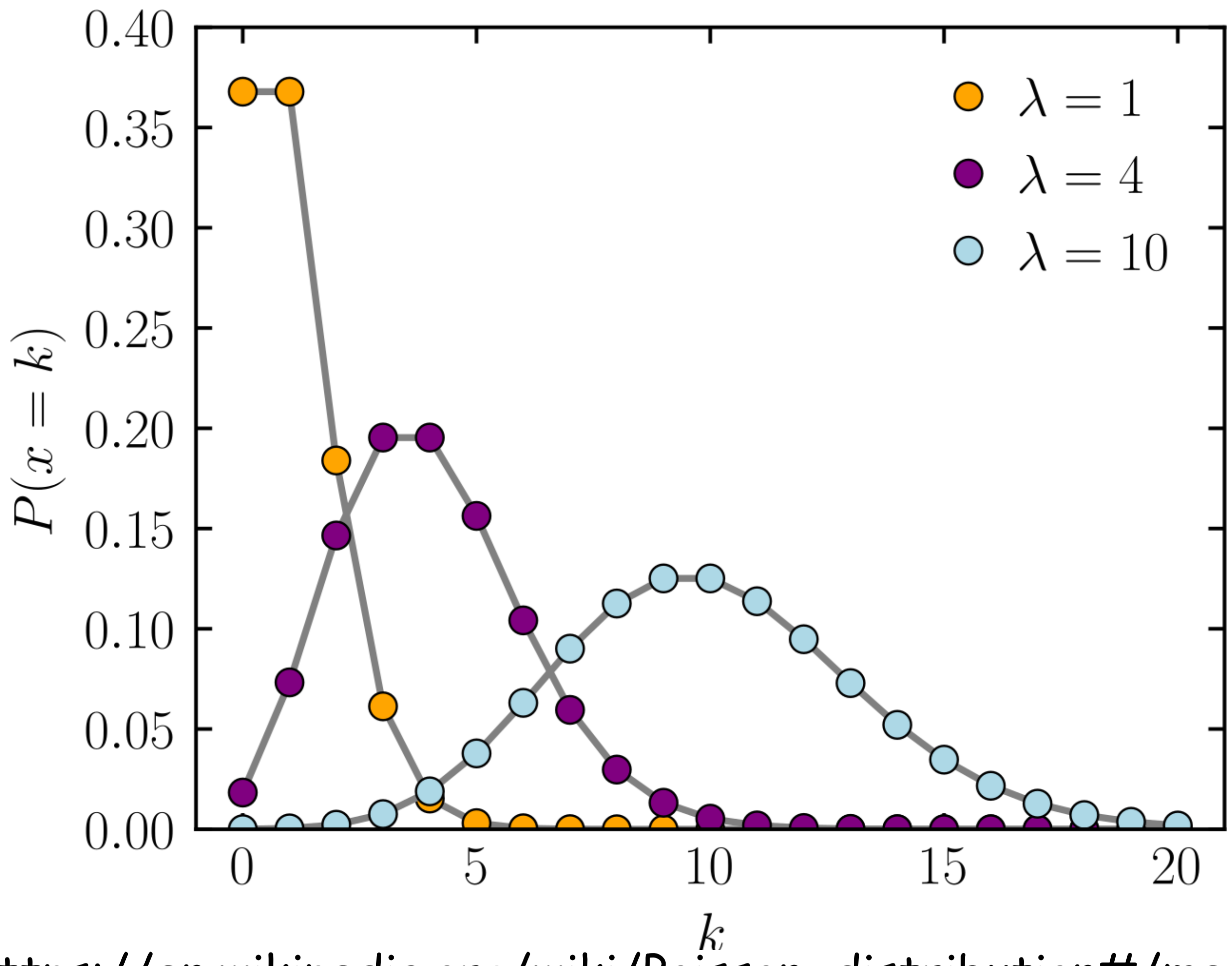
- *Poisson Distribution* models the probability of ***k***, the number of events occurring in **a fixed interval of time/space**.

$$P(k) = \frac{\lambda^k}{k!} e^{\lambda}$$

Lambda is the **average rate**

- Contoh:
  - Banyaknya mobil yang tiba di gerbal tol dalam 1 menit
  - Banyaknya typo dalam sebuah halaman
  - **Banyaknya kemunculan term (TF) di sebuah dokumen**

StatProb: Poisson Distribution



Sumber gambar: https://en.wikipedia.org/wiki/Poisson_distribution#/media/File:Poisson_pmf.svg

Poisson as a Model for "TF"

It's a reasonable fit for **general terms**; but **not** for **specific terms**.

			Documents containing *k* occurrences of word (λ = 53/650)												
	Freq	**Word**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
General Terms	**53**	**expected**	**599**	**49**	**2**										
	52	*based*	600	48	2										
	53	*conditions*	604	39	7										
Specific Terms	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

Harter, "A Probabilistic Approach to Automatic Keyword Indexing", JASIST, 1975

Elite Terms vs Non-Elite Terms

A term is elite in a document **if the document is about the concept denoted by them.**

The **National Football League Draft** is an annual event in which the **National Football League** (**NFL**) **teams select eligible college football players**. It serves as the **league's** most common source of **player recruitment**. The basic design of the **draft** is that each **team**is given a **position** in the **draft order** in **reverse order** relative to its **record**…

Eliteness bergantung dengan relevansi; dan **merupakan informasi yang hidden (secara umum tidak dapat diobservasi)**!

$$P(E_i|R,Q) = ?$$

E_i = 1 jika term i elite

E_i = 0 jika term i tidak elite

Balik lagi ke $P(TF_i = tf_i | R, Q)$

Chain Rule "mempermudah" hidup kita :)

$$P(TF_i = tf_i | R, Q) = P(E_i = 1 | R, Q).P(TF = tf_i | E_i = 1, Q) + P(E_i = 0 | R, Q).P(TF = tf_i | E_i = 0, Q)$$

$$= P(E_i = 1 | R, Q).\frac{\lambda^k}{k!}e^{-\lambda} + (1 - P(E_i = 1 | R, Q)).\frac{\mu^k}{k!}e^{-\mu}$$

Distribusi Poisson untuk yang Elite Terms

Distribusi Poisson untuk yang Non Elite Terms

Sebelumnya, kita sudah amati bahwa frekuensi Elite Terms dan Non Elite Terms berbeda!

Bali

Chain

$P(TF_i$

Sudah dijelaskan …
Masalahnya Informasi Eliteness dari suatu term tidak terobservasi!

$$= P(E_i = 1|R,Q).\frac{\lambda^k}{k!}e^{-\lambda}$$

$$+(1 - P(E_i = 1|R,Q)).\frac{\mu^k}{k!}e^{-\mu}$$

Distribusi Poisson untuk yang Elite Terms

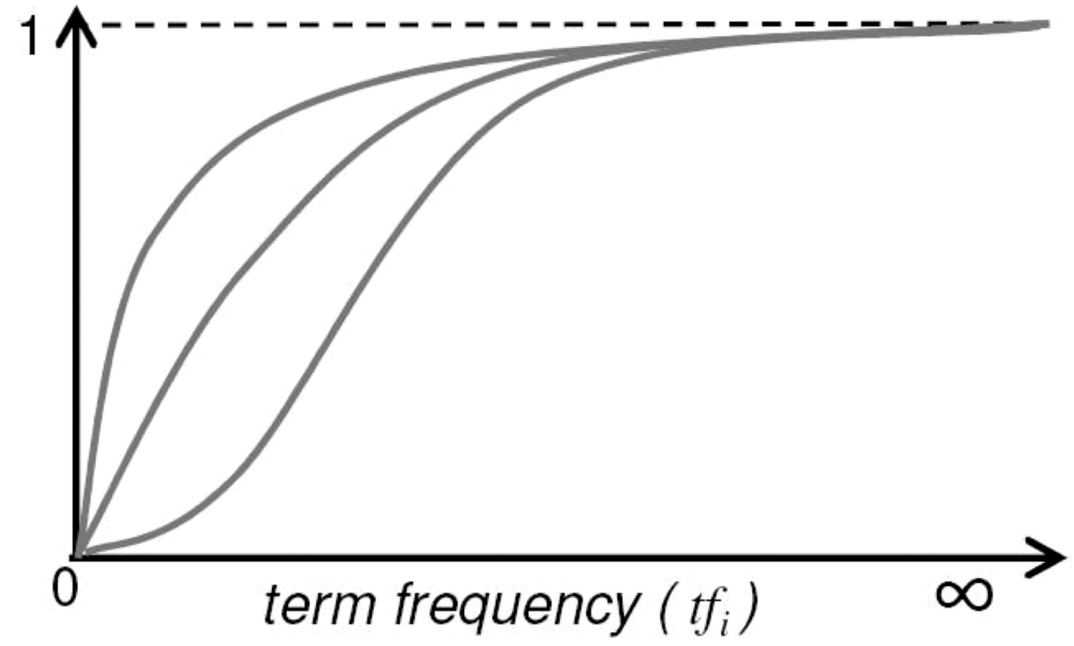
Distribusi Poisson untuk yang Non Elite Terms

Sebelumnya, kita sudah amati bahwa frekuensi Elite Terms dan Non Elite Terms berbeda!

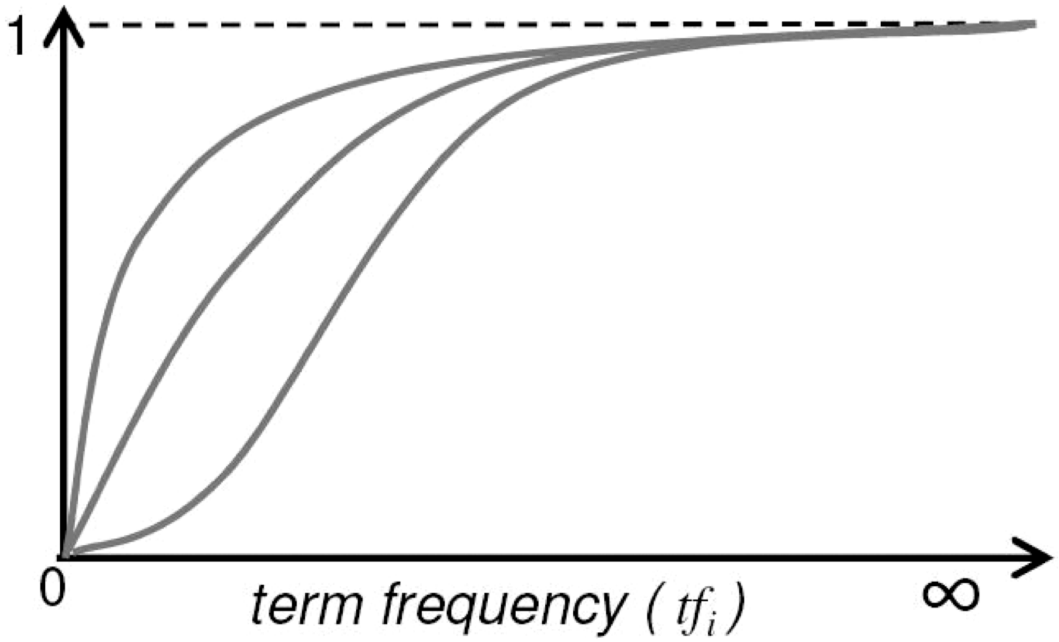
Cari cara lain -> fitting the line

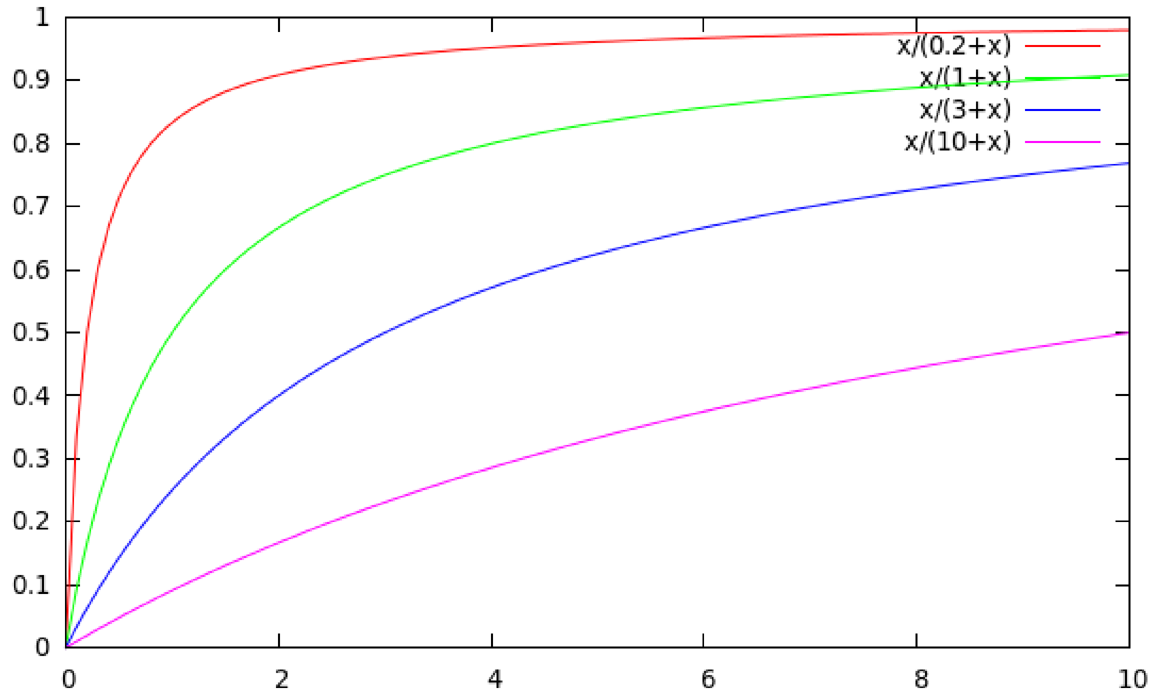
$$RSV = \sum_{\substack{TF_i > 0 \\ TF_i^Q > 0}} \boxed{\log\left(\frac{P(TF_i = tf_i | R = 1, Q) P(TF_i = 0 | R = 0, Q)}{P(TF_i = tf_i | R = 0, Q) P(TF_i = 0 | R = 1, Q)}\right)}$$

Kalau kita plot dengan berbagai kemungkinan nilai λ dan μ di slide sebelumnya



Cari cara lain -> fitting the line

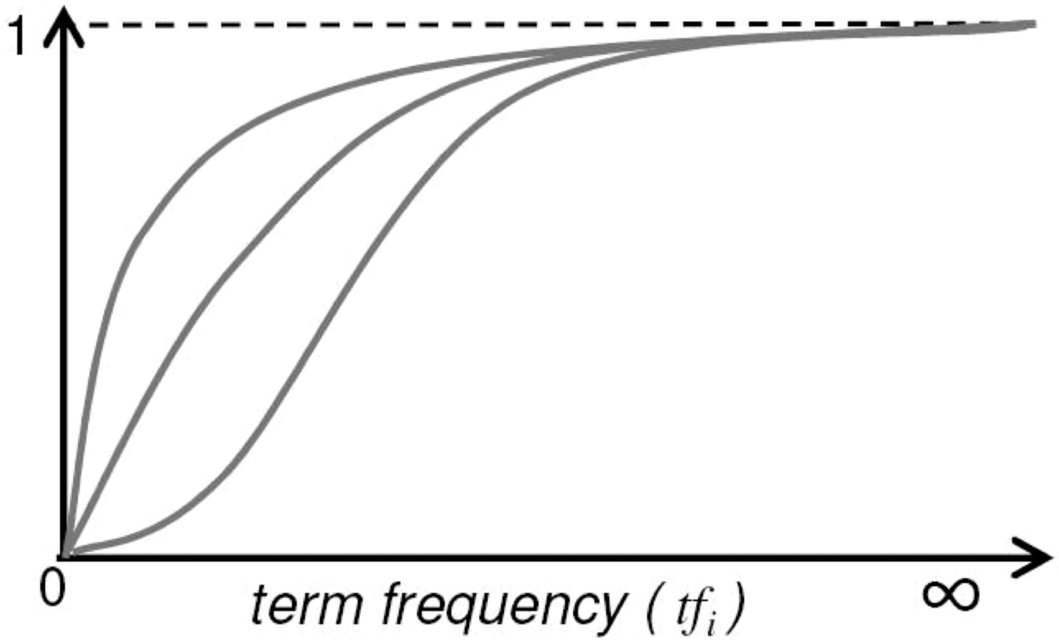


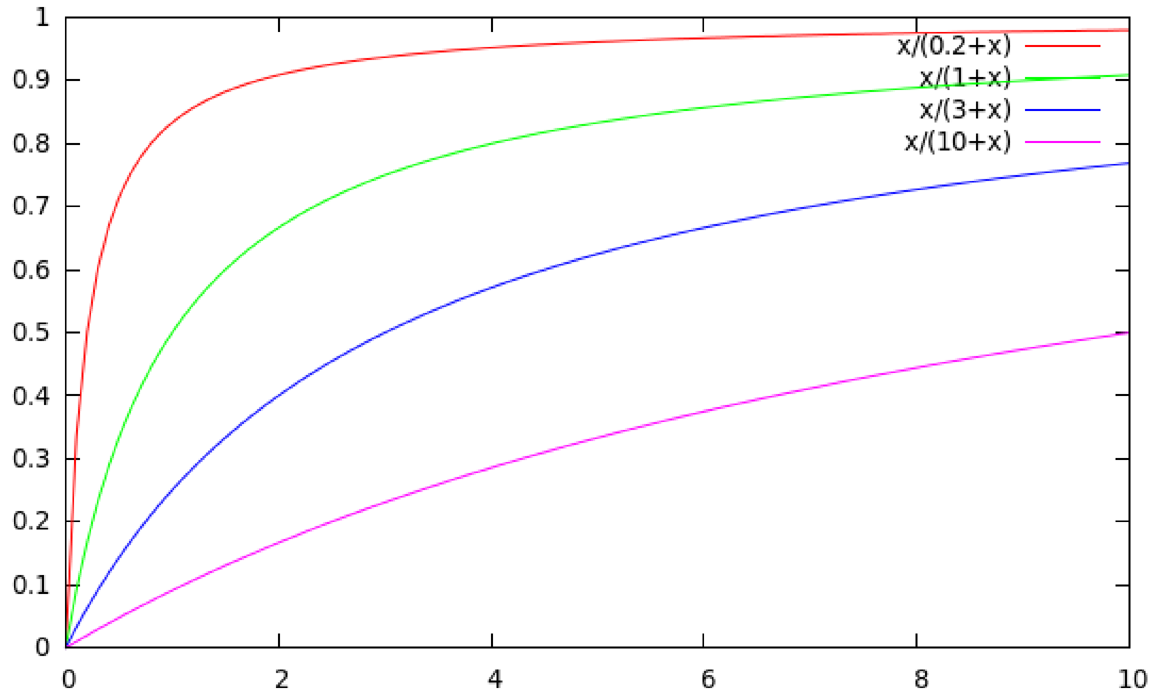


Kalau diperhatikan, kok plot-nya mirip dengan plot saturation function ya ☺

$$f(x) = \frac{x}{x+k}$$

Cari cara lain -> fitting the line





$$RSV = \sum_{\substack{TF_i > 0 \\ TF_i^Q > 0}} \log\left(\frac{P(TF_i = tf_i | R = 1, Q) P(TF_i = 0 | R = 0, Q)}{P(TF_i = tf_i | R = 0, Q) P(TF_i = 0 | R = 1, Q)}\right) \Rightarrow RSV = \sum_{\substack{TF_i > 0 \\ TF_i^Q > 0}} \frac{tf_i}{k_1 + tf_i}$$

$$RSV = \sum_{\substack{TF_i > 0 \\ TF_i^Q > 0}} \frac{tf_i}{k_1 + tf_i}$$

Our Simple Scoring Function yang mempertimbangkan **Eliteness**

$$RSV = \sum_{\substack{TF_i > 0 \\ TF_i^Q > 0}} \frac{tf_i}{k_1 + tf_i}$$

Our Simple Scoring Function yang mempertimbangkan **Eliteness**

$$RSV = \sum_{\substack{TF_i > 0 \\ TF_i^Q > 0}} \frac{(k_1 + 1).\, tf_i}{k_1 + tf_i}$$

Normalisasi: dikali dengan factor (k1 + 1) agar jika tf = 1, score untuk sebuah term adalah 1.

Hal ini tidak masalah, tidak mengubah ranking

$$RSV = \sum_{\substack{TF_i > 0 \\ TF_i^Q > 0}} \frac{tf_i}{k_1 + tf_i}$$

Our Simple Scoring Function yang mempertimbangkan **Eliteness**

$$RSV = \sum_{\substack{TF_i > 0 \\ TF_i^Q > 0}} \frac{(k_1 + 1).tf_i}{k_1 + tf_i}$$

Normalisasi: dikali dengan factor (k1 + 1) agar jika tf = 1, score untuk sebuah term adalah 1.

Hal ini tidak masalah, tidak mengubah ranking

$$RSV = \sum_{\substack{TF_i > 0 \\ TF_i^Q > 0}} \log\left(\frac{N}{df_i}\right) \frac{(k_1 + 1).tf_i}{k_1 + tf_i}$$

Tambahkan IDF untuk handle **informativeness** dari sebuah query

Informativeness + Eliteness

Earlier Version of BM25 !

$$RSV_{BM25} = \sum_{\substack{TF_i > 0 \\ TF_i^Q > 0}} \log\left(\frac{N}{df_i}\right) \frac{(k_1 + 1).\, tf_i}{k_1 + tf_i}$$

atau

$$RSV_{BM25} = \sum_{t \in Q \cap D} \log\left(\frac{N}{df_t}\right) \frac{(k_1 + 1).\, tf_t}{k_1 + tf_t}$$

Apa yang Masih Kurang?

Document Length Normalization

Ketika kita mempelajari TF-IDF dan cosine similary, **disampaikan betapa pentingnya length-normalization**.

Hal ini juga berlaku pada BM25 Scoring Function.

Document Length Normalization

Mengapa dokumen bisa menjadi Panjang?

- **Verbosity**: informasi **tf** bisa jadi terlalu tinggi
- **Larger scope**: informasi **tf** mungkin benar

Harus ada “keseimbangan” antara kedua hal tersebut! Artinya, harus bisa mempertimbangkan kedua hal tersebut dengan bobot tertentu.

Document Length Normalization

Panjang dokumen $dl = \sum_{t \in V} tf_t$

$$B = \left((1 - b) + b \frac{dl}{avdl} \right)$$

Rata-Rata Panjang dokumen di koleksi

$0 \leq b \leq 1$

Jika **b** = **1**: full document length normalization
Jika **b** = **0**: tidak menggunakan length normalization

Document Length Normalization Factor

$$B = \left((1 - b) + b \frac{dl}{avdl} \right)$$

Earlier Version of BM25

$$RSV_{BM25} = \sum_{t \in Q \cap D} \log\left(\frac{N}{df_t}\right) \frac{(k_1 + 1).\, tf_t}{k_1 + tf_t}$$

Document Length Normalization Factor

$$B = \left((1-b) + b\frac{dl}{avdl}\right)$$

Version of BM25

$$RSV_{BM25} = \sum_{t\in Q\cap D} \log\left(\frac{N}{df_t}\right)\frac{(k_1+1).tf_t}{k_1+tf_t}$$

Digabung!

Okapi BM25

$$RSV_{BM25} = \sum_{t\in Q\cap D} \log\left(\frac{N}{df_t}\right)\frac{(k_1+1).tf_t}{k_1\left((1-b) + b\frac{dl}{avdl}\right)+tf_t}$$

Okapi BM25

$$RSV_{BM25} = \sum_{t \in Q \cap D} \log\left(\frac{N}{df_t}\right) \frac{(k_1 + 1).\, tf_t}{k_1\left((1-b) + b\,\frac{dl}{avdl}\right) + tf_t}$$

- **k_1** mengatur term frequency scaling
  - **k_1** = 0 artinya binary model; **k_1** sangat besar berarti raw TF.

- **b** mengatur document length normalization

- Biasanya **1.2 <= k_1 <= 2** dan **b = 0.75**

BM25 is Better than "unnormalized" TF-IDF

- Query: machine learning
- Documents:
  - D1: learning 1024x; machine 1x
  - D2: learning 16x; machine 8x

- TF-IDF: (1 + log2 tf) * log2 (N/df)
  - D1: 11 * 7 + 1 * 10 = 87
  - D2: 5 * 7 + 4 * 10 = 75

- BM25, k_1 = 2, b = 0.75
  - D1: 7 * 2.6 + 10 * 0.02 = 18.4
  - D2: 7 * 2.8 + 10 * 2.7 = 46.6

Misal,
IDF(learning) = 7
IDF(machine) = 10
Avgdl = 100

Alternative 2

Menggunakan bentuk alternative IDF

$$RSV_{BM25} = \sum_{t \in Q \cap D} \log\left(\frac{N - df_t + 0.5}{df_t + 0.5}\right) \frac{(k_1 + 1).\, tf_t}{k_1\left((1 - b) + b\,\frac{dl}{avdl}\right) + tf_t}$$

Alternative 3

Jika **query panjang**, scaling terhadap TF pada query juga perlu dilakukan.

$$RSV_{BM25} = \sum_{t \in Q \cap D} \log\left(\frac{N}{df_t}\right) \frac{(k_1 + 1).tf(t,d)}{k_1\left((1-b) + b\frac{dl}{avdl}\right) + tf(t,d)} \frac{(k_2+1).tf(t,Q)}{k_2 + tf(t,Q)}$$

Ada bagian yang merupakan fungsi dari frekuensi term pada query