BM25 Demystified

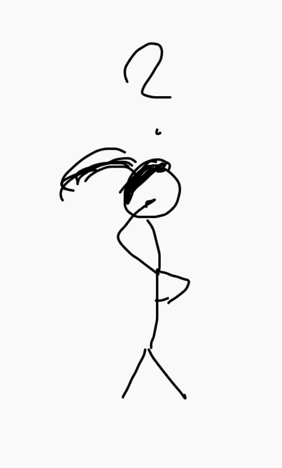
Britta Weber
6/7/2016

What is BM25?

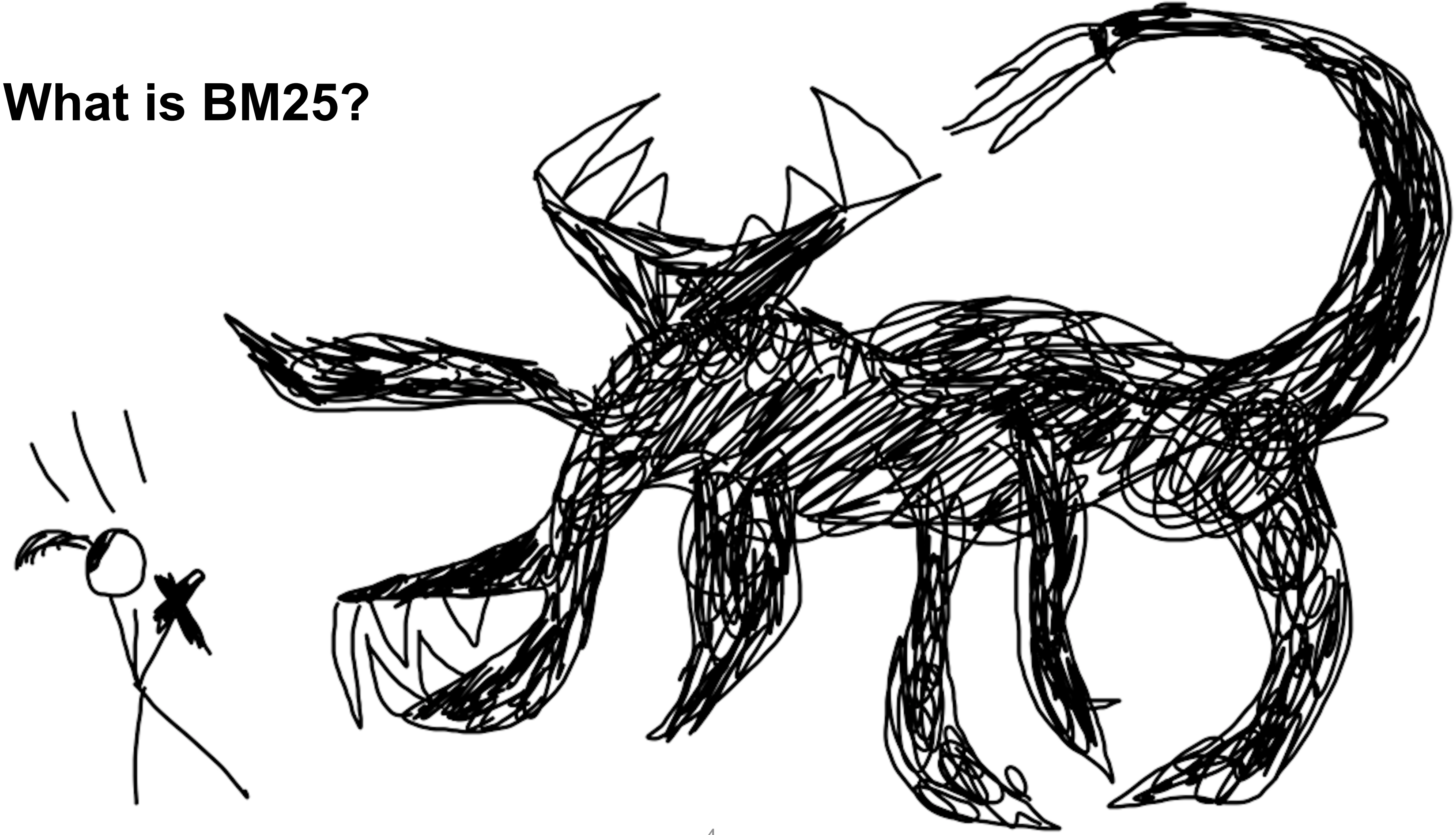
“Oh! BM25 is that probabilistic approach to scoring!”

What is BM25?

$$\mathrm{bm25}(d) = \sum_{t \in q, f_{t,d} > 0} \log\left(1 + \frac{N - df_t + 0.5}{df_t + 0.5}\right) \cdot \frac{f_{t,d}}{f_{t,d} + k \cdot (1 - b + b\frac{\mathrm{l}(d)}{avgdl})}$$

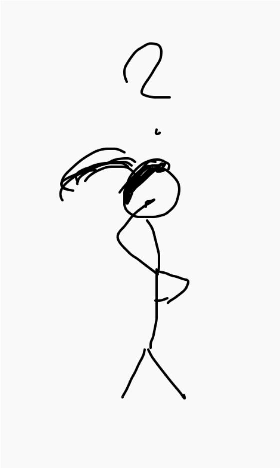


What is BM25?



What is BM25?

$$\mathrm{bm25}(d) = \sum_{t \in q, f_{t,d} > 0} \log\left(1 + \frac{N - df_t + 0.5}{df_t + 0.5}\right) \cdot \frac{f_{t,d}}{f_{t,d} + k \cdot (1 - b + b\frac{l(d)}{avgdl})}$$



Why is this so complicated?

Searching in natural language text

Often when you search you really just want to filter by…

- categories
- timestamps
- age
- ids …

```
"_source": {
    "oder-nr": 1234,
    "items": [3,5,7],
    "price": 30.85,
    "customer": "Jon Doe",
    "date": "2015-01-01"
}
```

Searching in natural language text

Tweets mails, articles,… are fuzzy

- language is ambivalent, verbose and many topics in one doc
- no clear way to formulate your query

```
"_source": {
  "titles": "guru of everything",
  "programming_languages": [
                "java",
                "python",
                "FORTRAN"
            ],
  "age": 32,
  "name": "Jon Doe",
  "date": "2015-01-01",
  "self-description": "I am a
hard-working self-motivated expert
in everything. High performance is
not just an empty word for me..."
}
```


elastic

A free text search is a very inaccurate description of our information need

What you want:

- quick learner
- works hard
- reliable
- enduring
- …

```
"_source": {
  "titles": "guru of everything",
  "programming_languages": [
                  "java",
                  "python",
                  "FORTRAN"
               ],
  "age": 32,
  "name": "Jon Doe",
  "date": "2015-01-01",
  "self-description": "I am a
hard-working self-motivated expert
in everything. High performance is
not just an empty word for me..."
}
```


elastic

A free text search is a very inaccurate description of our information need

What you want:

- quick learner
- works hard
- reliable
- enduring
- …

But you type :

“hard-working, self-motivated, masochist”

```
"_source": {
  "titles": "guru of everything",
  "programming_languages": [
                    "java",
                    "python",
                    "FORTRAN"
                 ],
  "age": 32,
  "name": "Jon Doe",
  "date": "2015-01-01",
  "self-description": "I am a
hard-working self-motivated expert
in everything. High performance is
not just an empty word for me..."
}
```


elastic

The purpose of this talk

$$\mathrm{bm25}(d) = \sum_{t \in q, f_{t,d} > 0} \log\left(1 + \frac{N - df_t + 0.5}{df_t + 0.5}\right) \cdot \frac{f_{t,d}}{f_{t,d} + k \cdot (1 - b + b\frac{l(d)}{avgdl})}$$

By the end of this talk you should

- know the monster, understand what the parameters of BM25 do

elastic

The purpose of this talk

$$\text{bm25}(d) = \sum_{t \in q, f_{t,d} > 0} \log\left(1 + \frac{N - df_t + 0.5}{df_t + 0.5}\right) \cdot \frac{f_{t,d}}{f_{t,d} + k \cdot (1 - b + b\frac{l(d)}{avgdl})}$$

By the end of this talk you should

- know the monster, understand what the parameters of BM25 do
- know why it has the label “probabilistic”

elastic

The purpose of this talk

$$\mathrm{bm25}(d) = \sum_{t \in q, f_{t,d} > 0} \log\left(1 + \frac{N - df_t + 0.5}{df_t + 0.5}\right) \cdot \frac{f_{t,d}}{f_{t,d} + k \cdot (1 - b + b\frac{1(d)}{avgdl})}$$

By the end of this talk you should

- know the monster, understand what the parameters of BM25 do
- know why it has the label “probabilistic”
- be convinced that switching to BM25 is the right thing to do

The purpose of this talk

$$\mathrm{bm25}(d) = \sum_{t \in q, f_{t,d} > 0} \log\left(1 + \frac{N - df_t + 0.5}{df_t + 0.5}\right) \cdot \frac{f_{t,d}}{f_{t,d} + k \cdot (1 - b + b\frac{l(d)}{avgdl})}$$

By the end of this talk you should

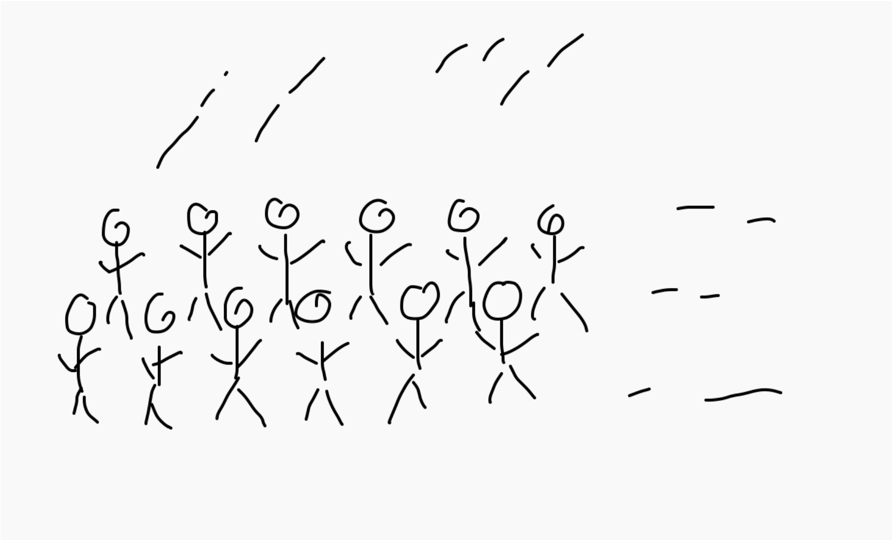
- know the monster, understand what the parameters of BM25 do
- know why it has the label “probabilistic”
- be convinced that switching to BM25 is the right thing to do
- be able to impress people with you in depth knowledge of probabilistic scoring

The current default - TF/IDF

Example: we are looking for an intern

Search in self-description of applications for these words:

- self-motivated
- hard-working
- masochist



We want to order applications by their ***relevance*** to the query.

elastic

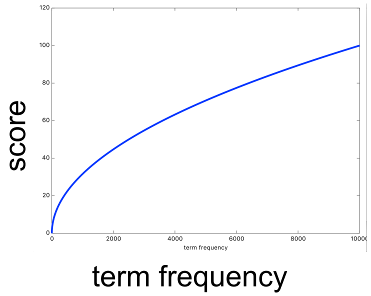
Evidence for relevance - term frequencies

Use **term frequencies** in description, title etc.

"I got my PhD in Semiotics at the University of ….but I am still ***hard-working***! … It takes a ***masochist*** to go through a PhD…"

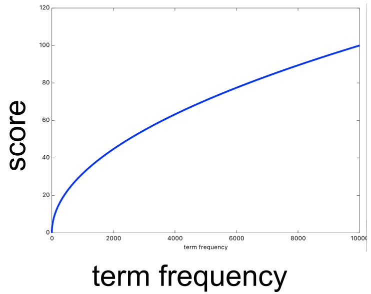
Major tweaks

- term frequency: more is better

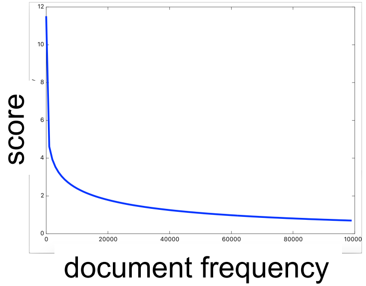


Major tweaks

- term frequency: more is better

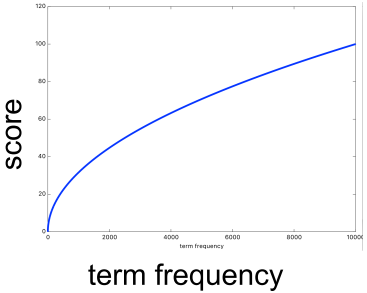


- inverse document frequency: common words are less important

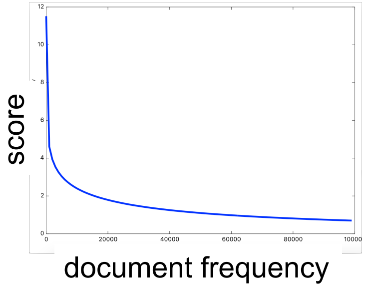


Major tweaks

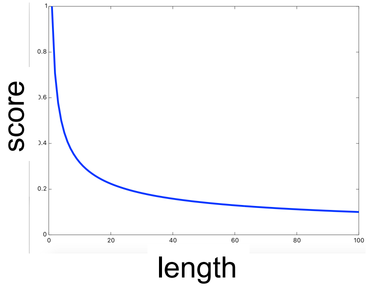
- term frequency: more is better



- inverse document frequency: common words are less important



- long documents with same tf are less important: norm



Bool query and the coord- factor

Query: holiday, china

"Blog: My holiday in Bejing"

term frequencies:

holiday: 4
china: 5

"Economic development of Sichuan from 1920-1930"

term frequencies:

holiday: 0
china: 15

Coord factor: reward document 1 because both terms matched

TF/IDF

- Successful since the beginning of Lucene
- Well studied
- Easy to understand
- One size fits most

What is wrong with TF/IDF?

It is a heuristic that makes sense intuitively but it is somewhat a guess. (Ad hoc.)

So…can we do better?

Probabilistic ranking and how it led to BM25

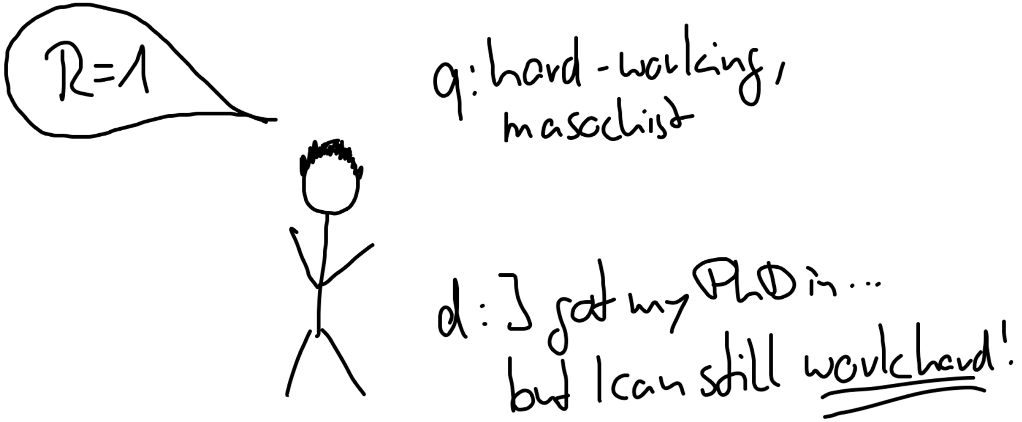
The root of BM25: Probability ranking principle (abridged)

> "If retrieved documents are ordered by decreasing probability of relevance on the data available, then the system's effectiveness is the best that can be obtained for the data."

K. Sparck Jones, S. Walker, and S. E. Robertson, "A probabilistic model of information retrieval: Development and comparative experiments. Part 1,"

Estimate relevancy

- simplification: relevance is binary!
- get a dataset queries - relevant/ irrelevant documents
- use that to estimate relevancy

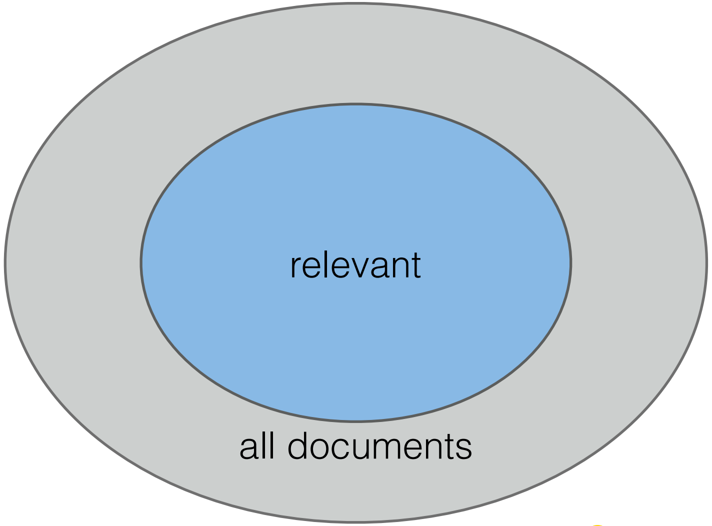


Estimate relevancy

elastic

Estimate relevancy

get a dataset queries - relevant/irrelevant documents and use that to estimate relevancy



In math

$$
\begin{aligned}
P(A|B) &= \text{probability of A given B} \\
R &= \text{relevancy (1/0)} \\
d &= \text{document} \\
q &= \text{query}
\end{aligned}
$$

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

For each document, query pair - what is the probability that the document is relevant? Order by that!

elastic

In math

$P(A\|B)$	= probability of A given B
R	= relevancy (1/0)
d	= document
q	= query

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

	R=1	***R=0***
d1	0.1	0.9
d2	0.2	0.8
d3	0.7	0.3
…	…	…

In math

$P(A\|B)$	= probability of A given B
R	= relevancy (1/0)
d	= document
q	= query

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

	R=1	***R=0***
d1	0.1	0.9
d2	0.2	0.8
d3	0.7	0.3
...	...	...

for each query *q*!

elastic

In math

$$
\begin{aligned}
P(A|B) &= \text{probability of A given B} \\
R &= \text{relevancy } (1/0) \\
d &= \text{document} \\
q &= \text{query}
\end{aligned}
$$

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

	R=1	***R=0***
d1	0.1	0.9
d2	0.2	0.8
d3	0.7	0.3
…	…	…

for each query *q*!

No way we can ever get a list of that, no matter how many interns we hire….

…here be math…

Foundations and Trends® in
Information Retrieval
Vol. 3, No. 4 (2009) 333–389

DOI: 10.1561/1500000019

The Probabilistic Relevance Framework: BM25 and Beyond

By Stephen Robertson and Hugo Zaragoza

Contents

1 Introduction **334**

2 Development of the Basic Model **336**

$P(A|B)$ = probability of A given B
R = relevancy (1/0)
d = document
q = query
t = term
$f_{t,d}$ = frequency of term in document
$F = f_{t,d}$ = term frequency is $f_{t,d}$
$F = 0$ = term not in document

…and we get to…

$$W(d) = \sum_{t \in q, f_{t,d} > 0} \log \frac{P(F = f_{t,d} | R = 1) P(F = 0 | R = 0)}{P(F = f_{t,d} | R = 0) P(F = 0 | R = 1)}$$

elastic

…and we get to…

$P(A\|B)$	= probability of A given B
R	= relevancy (1/0)
d	= document
q	= query
t	= term
$f_{t,d}$	= frequency of term in document
$F = f_{t,d}$	= term frequency is $f_{t,d}$
$F = 0$	= term not in document

P(tf of "hard-working" = 1| R=1) = 0.1
P(tf of "hard-working" = 1| R=0) = 0.12
P(tf of "hard-working" = 2| R=1) = 0.3
…

$$W(d) = \sum_{t \in q, f_{t,d} > 0} \log \frac{P(F = f_{t,d} | R = 1) P(F = 0 | R = 0)}{P(F = f_{t,d} | R = 0) P(F = 0 | R = 1)}$$

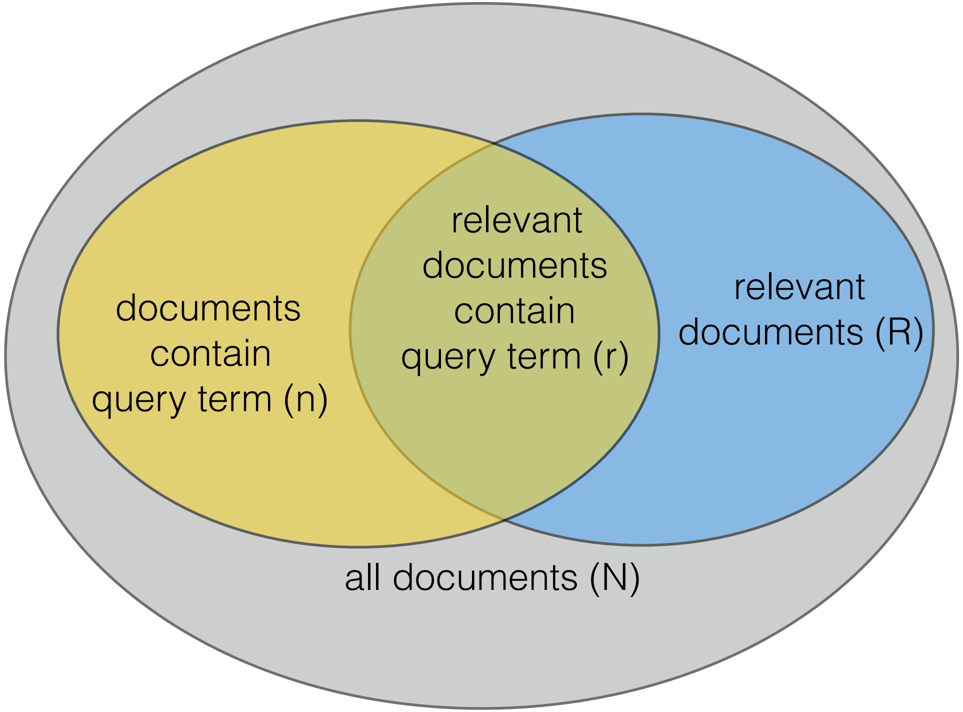
P("hard-working" does not occur in document| R=1) = 0.1
P("hard-working" does not occur in document| R=0) = 0.4

…but at least we know we only need two distributions!

How to estimate all these probabilities

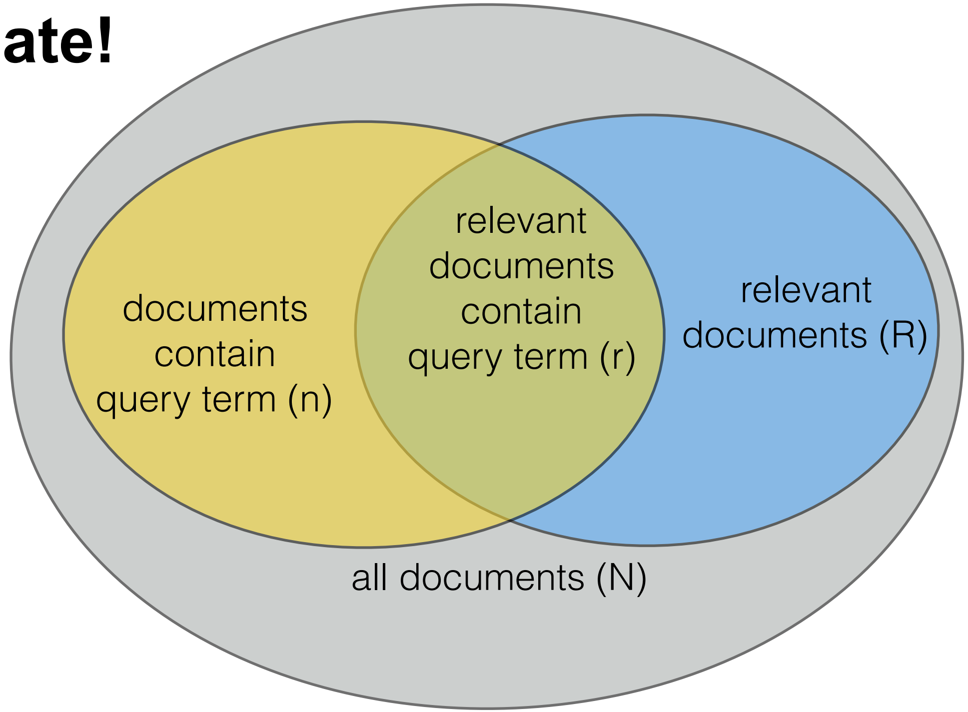
The binary independence model - a dramatic but useful simplification

query term occurs in a document or doesn’t - we don’t care how often



Use actual counts to estimate!

$$P(F=1|R=1) \approx \frac{r+0.5}{R+1}$$



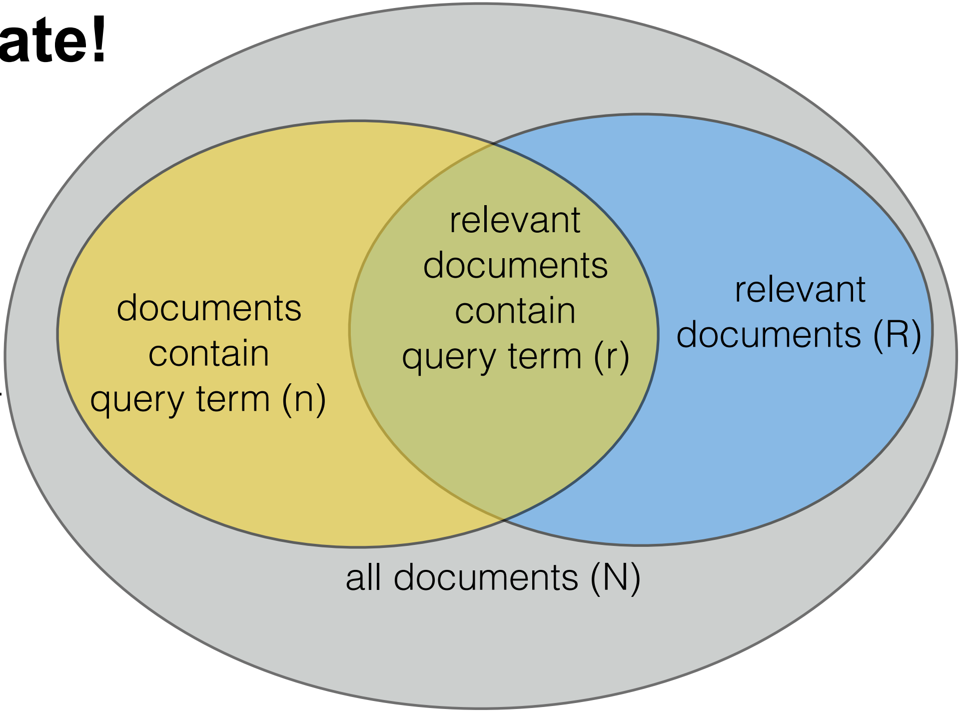
Stephen Robertson and Karen Spark Jones, Relevance Weighting of Search Terms

elastic

Use actual counts to estimate!

$$P(F=1|R=1) \approx \frac{r+0.5}{R+1}$$

$$P(F=1|R=0) \approx \frac{n-r+0.5}{N-R+1}$$



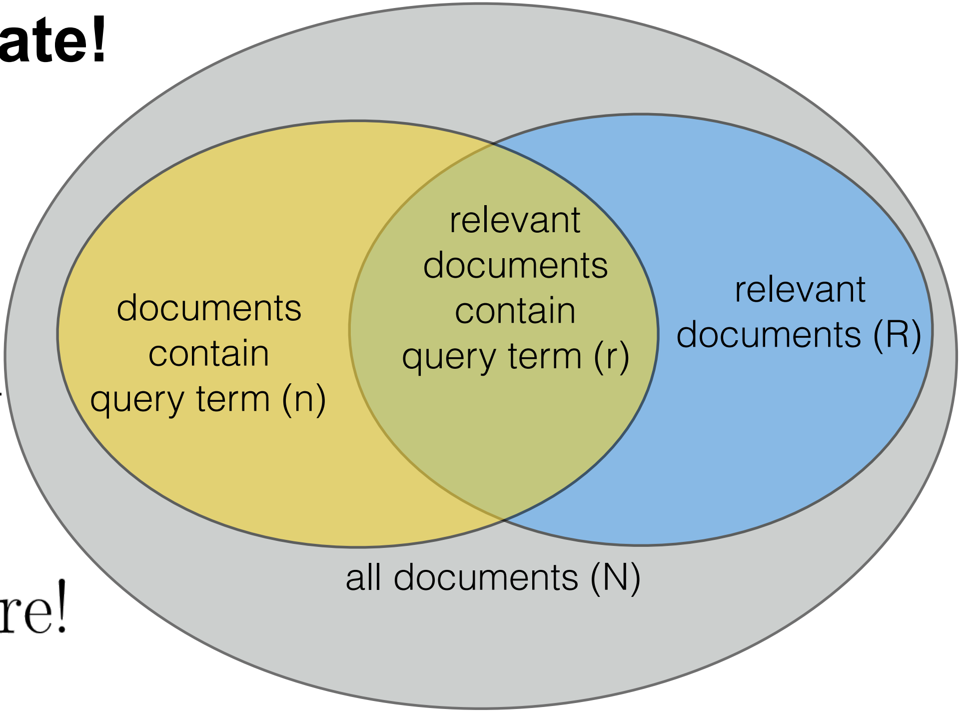
Stephen Robertson and Karen Spark Jones, Relevance Weighting of Search Terms

Use actual counts to estimate!

$$P(F=1|R=1) \approx \frac{r+0.5}{R+1}$$

$$P(F=1|R=0) \approx \frac{n-r+0.5}{N-R+1}$$

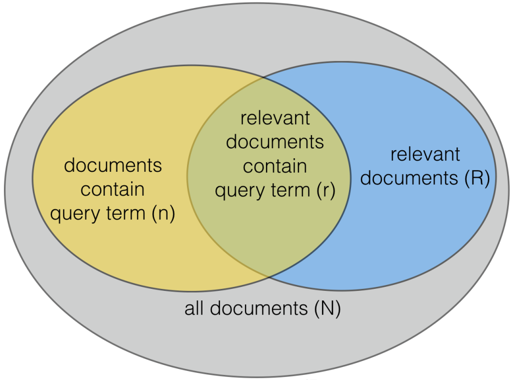
$$P(F=2|R=1) \approx \text{don't care!}$$



Stephen Robertson and Karen Spark Jones, Relevance Weighting of Search Terms

elastic

Use actual counts to estimate!



$$P(F=1|R=0) \approx \ldots \quad P(F=1|R=1) \approx \ldots$$

$$P(F=0|R=0) \approx \ldots \quad P(F=0|R=1) \approx \ldots$$

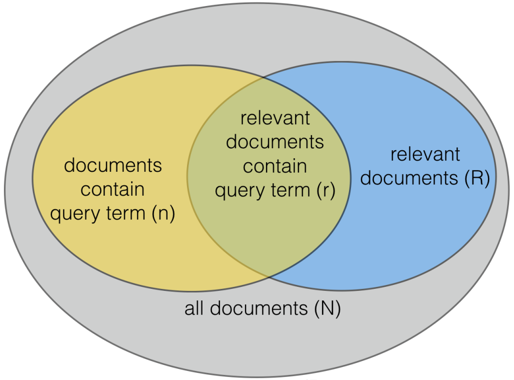
Plug this into our weight equation

$$W(d) = \sum_{t \in q, f_{t,d}>0} \log \frac{P(F=f_{t,d}|R=1)P(F=0|R=0)}{P(F=f_{t,d}|R=0)P(F=0|R=1)}$$

Stephen Robertson and Karen Spark Jones, Relevance Weighting of Search Terms

elastic

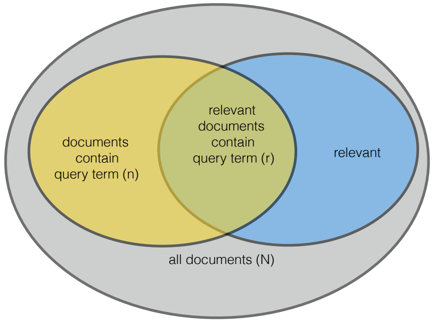
Robertson/Sparck Jones weight



$$w^{RSJ} = \log \frac{(r + 0.5)(N - R - n + r + 0.5)}{(n - r + 0.5)(R - r + 0.5)}$$

These are really just counts

So, you have an unlimited supply of interns…



$$w^{RSJ} = \log \frac{(r+0.5)(N-R-n+r+0.5)}{(n-r+0.5)(R-r+0.5)}$$

	weight
motivated	0.1
working	0.6
experienced	0.23
…	…

…but you probably don’t have that

N = number of documents

n = number of docs that contain the term

Still use Robertson/Sparck Jones weight but assume that the number of relevant documents is negligible (R=0, r=0):

$$w^{IDF} = \log \frac{(N - n + 0.5)}{n + 0.5}$$

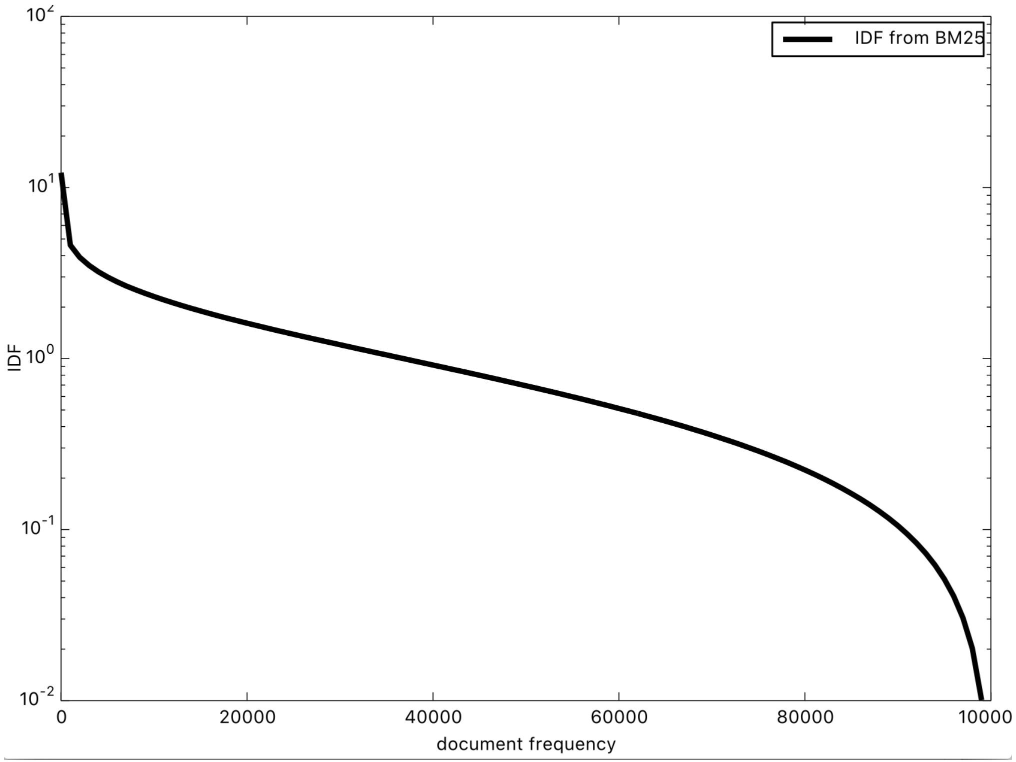
N = number of documents

df_t = number of docs that contain the term

IDF comparison

BM25

$$w^{IDF}(t) = \log\left(\frac{N - df_t + 0.5}{df_t + 0.5} + 1\right)$$



N = number of documents
df_t = number of docs that contain the term

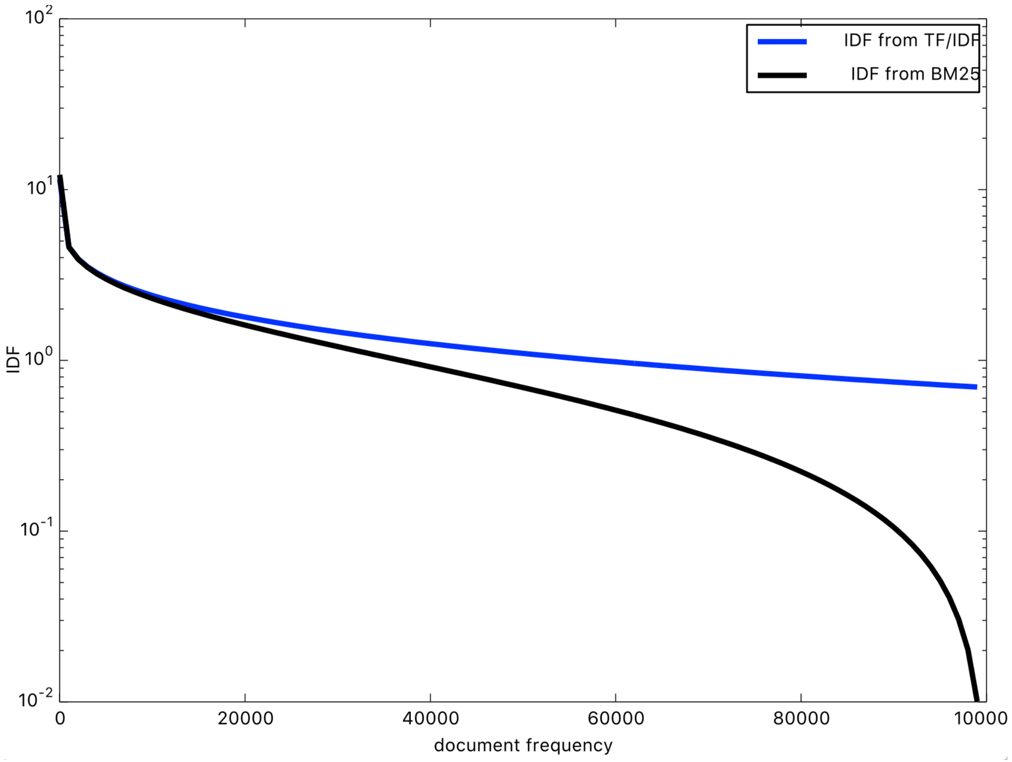
IDF comparison

BM25

$$w^{IDF}(t) = \log\left(\frac{N - df_t + 0.5}{df_t + 0.5} + 1\right)$$

TF/IDF

$$w^{IDF}(t) = \log\left(\frac{N + 1}{df_t + 1} + 1\right)$$



BM25 - We are here…

$$\mathrm{bm25}(d) = \sum_{t \in q} \log \left(1 + \frac{N - df_t + 0.5}{df_t + 0.5} \right) \cdot \frac{f_{t,d}}{f_{t,d} + k \cdot (1 - b + b \frac{\mathrm{l}(d)}{avgdl})}$$

BM25 - We are here…

idf - how popular is the term in the corpus?

$$\mathrm{bm25}(d) = \sum_{t \in q, f_{t,d} > 0} \log\left(1 + \frac{N - df_t + 0.5}{df_t + 0.5}\right) \cdot \frac{f_{t,d}}{f_{t,d} + k \cdot (1 - b + b\frac{l(d)}{avgdl})}$$

Now, consider term frequency!

What does the number of occurrence of a term tell us about relevancy?

- In TF/IDF: The more often the term occurs the better
- But…is a document about a term just because it occurs a certain number of times?
- This property is called “eliteness”

Example for “eliteness”

- “tourism”
- Look at wikipedia: Many documents are about tourism
- Many documents contain the word tourism - but are about something completely different, like for example just a country

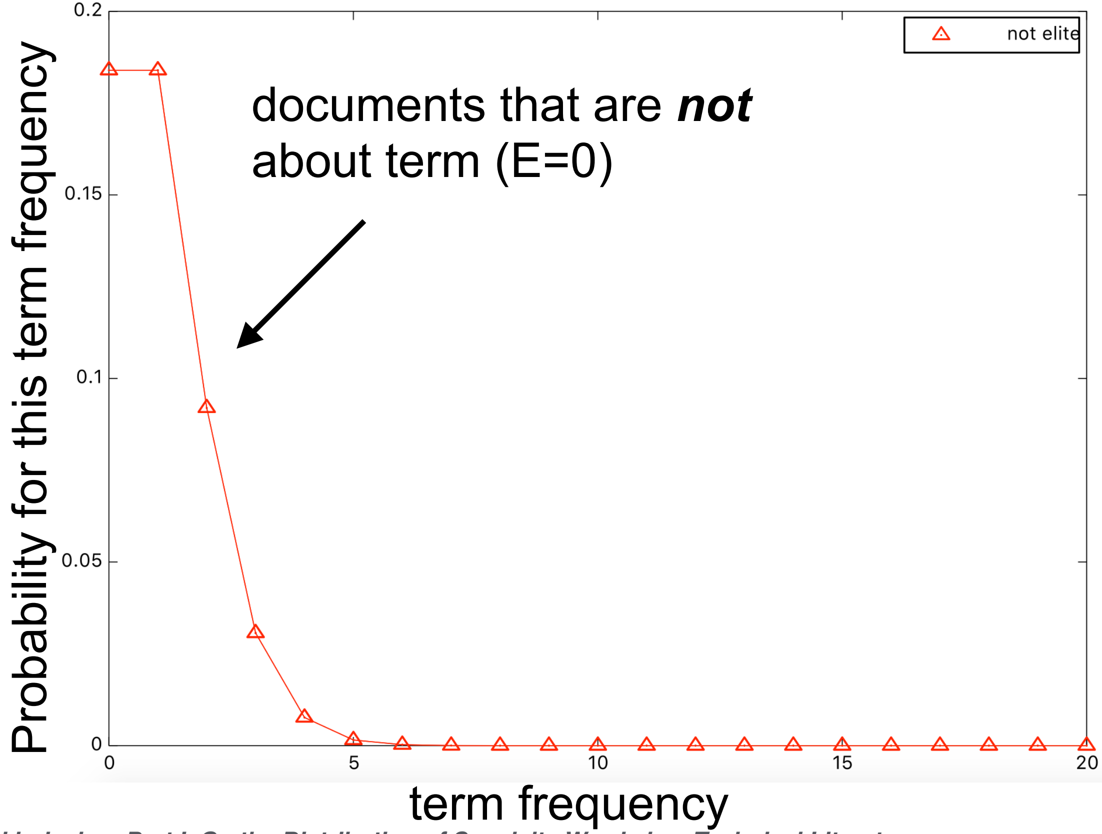
Can we use prior knowledge on the distribution of term frequency for getting a better estimate on the influence of tf?

$f_{t,d}$ = termfrequency in doc
E = eliteness
λ = yet another parameter...

Eliteness as Poisson Distribution

Two cases:

- document is not about the term



Stephen P. Harter, A probabilistic approach to automatic keyword indexing. Part I. On the Distribution of Specialty Words in a Technical Literature

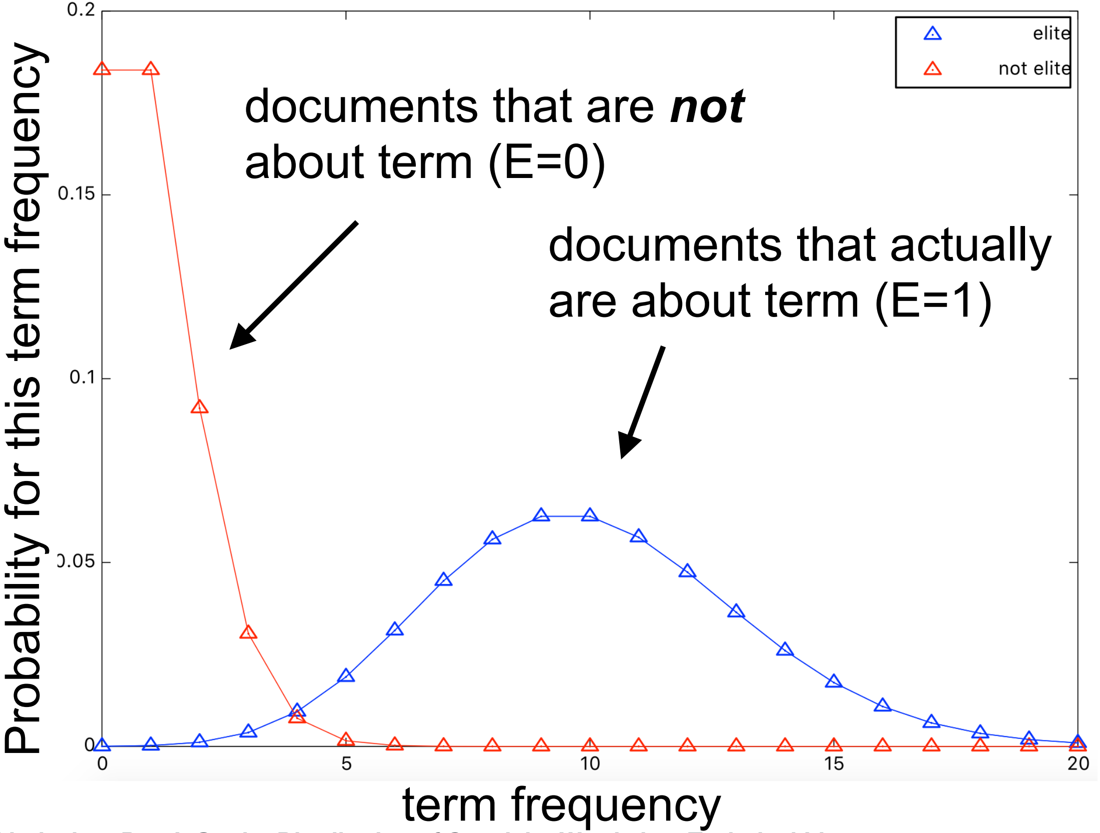
elastic

$f_{t,d}$ = termfrequency in doc
E = eliteness
λ = yet another parameter...

Eliteness as Poisson Distribution

Two cases:

- document is not about the term
- document is about the term

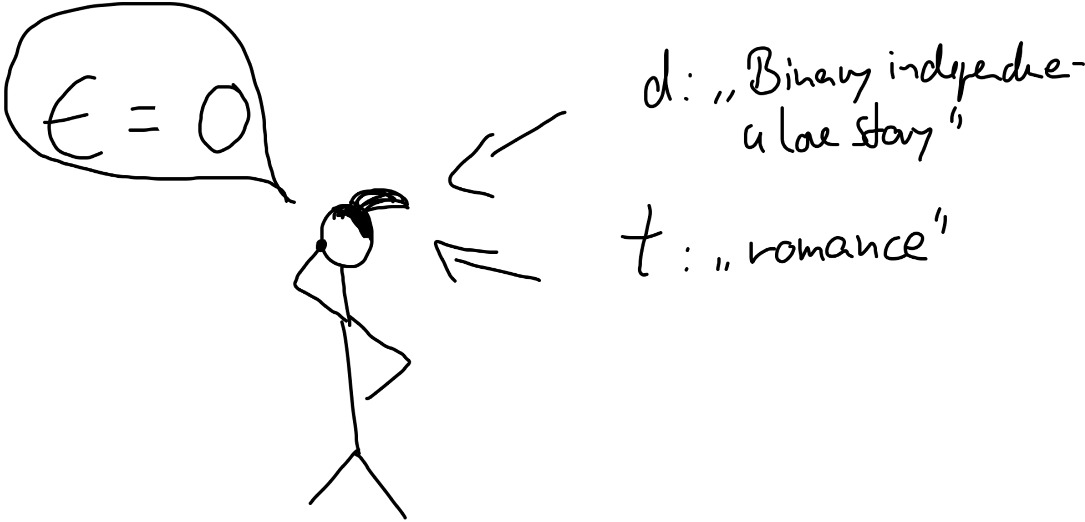


Stephen P. Harter, A probabilistic approach to automatic keyword indexing. Part I. On the Distribution of Specialty Words in a Technical Literature

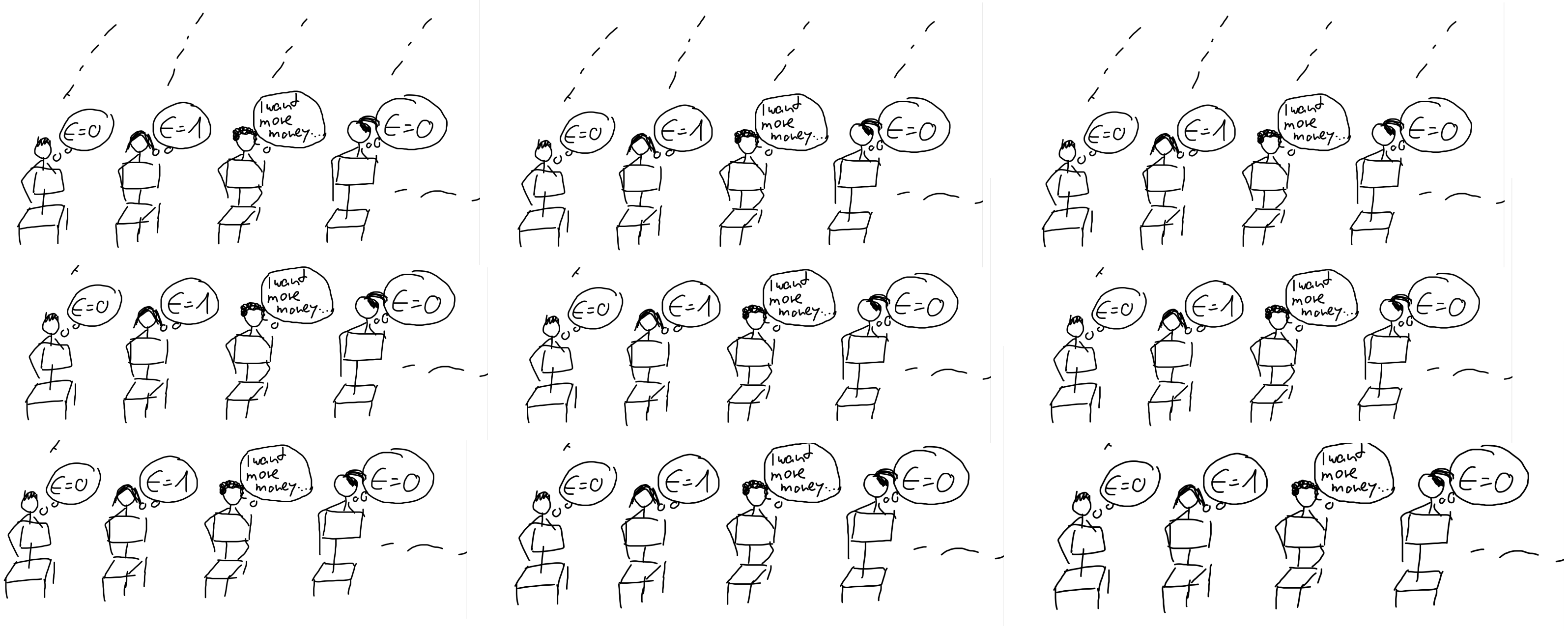
elastic

How to estimate this?

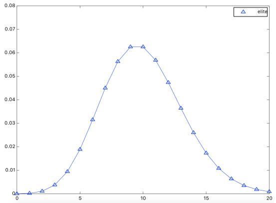
- gather data on eliteness for term
- many term frequencies -> do for many documents

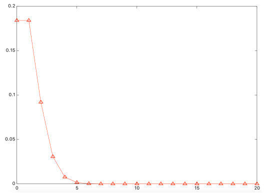


We need even more interns!



How relevance ties into that

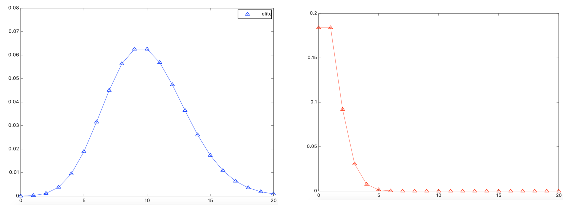




Suppose we knew the relationship of frequency and eliteness.

We need: relationship of frequency and relevancy!

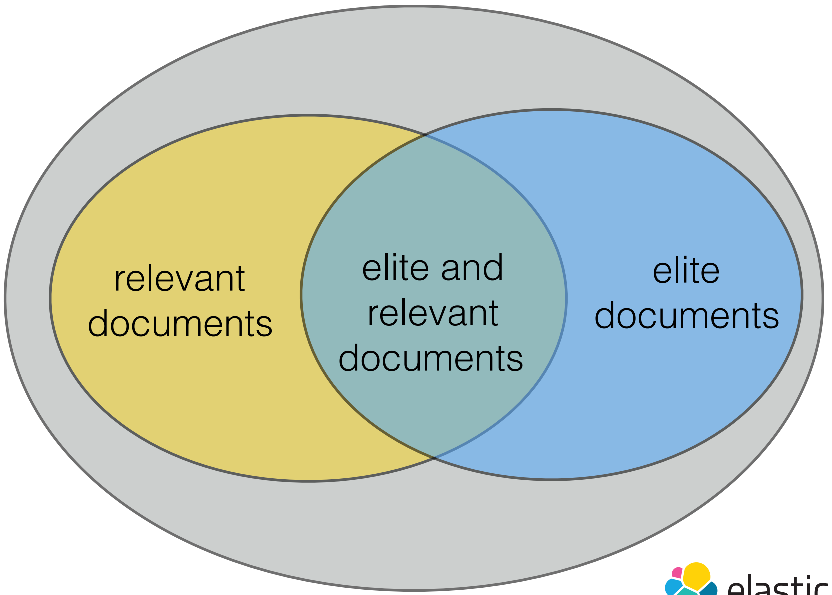
How relevance ties into that



Suppose we knew the relationship of frequency and eliteness.

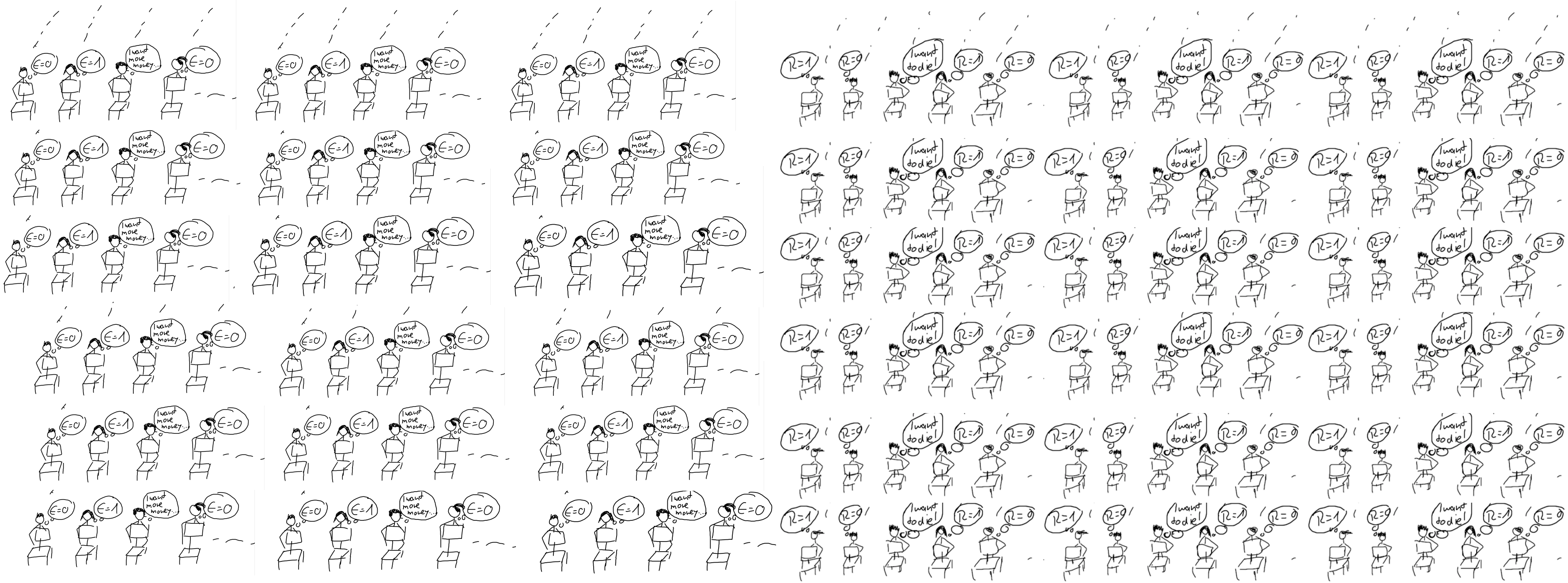
We need: relationship of frequency and relevancy!

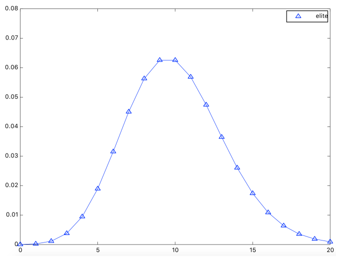
- Have yet another distribution: $P(E|R)$
- make eliteness depend on relevancy
- estimate from data

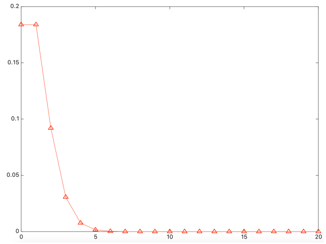


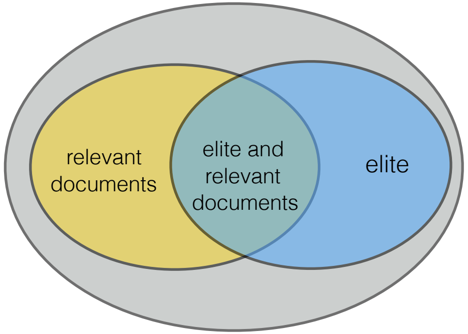
elastic

We need even more interns for the relevance too!









$$P(f_{t,d}|E)$$

$$P(E|R)$$

combine the two…

$$P(f_{t,d}|R)$$

…plug into here…

$$W(d) = \sum_{t \in q, f_{t,d} > 0} \log \frac{P(F = f_{t,d}|R = 1)P(F = 0|R = 0)}{P(F = f_{t,d}|R = 0)P(F = 0|R = 1)}$$

…here be math…

…and we get to….

…and we get to….

$$W(d,q) =$$

“This is a somewhat messy formula, and furthermore we do not in general know the values of these three parameters, or have any easy way of estimating them.”

Stephen Robertson and Hugo Zaragoza, The Probabilistic Relevance Framework: BM25 and Beyond

“…they took a leap of faith…”

Victor Lavrenko, Probabilistic model 9: BM25 and 2-poisson, youtube

What is the shape?

If we actually had all these interns and could get the exact shape then the curve…

- would start at 0
- increase monotonically
- approach a maximum asymptotically
- maximum would be the IDF we computed before!

$$W(d, q) =$$

What is the shape?

If we actually had all these interns and could get the exact shape then the curve…

- would start at 0
- increase monotonically
- approach a maximum asymptotically
- maximum would be the IDF we computed before!

$$W(d, q) =$$

Just use something similar!

elastic

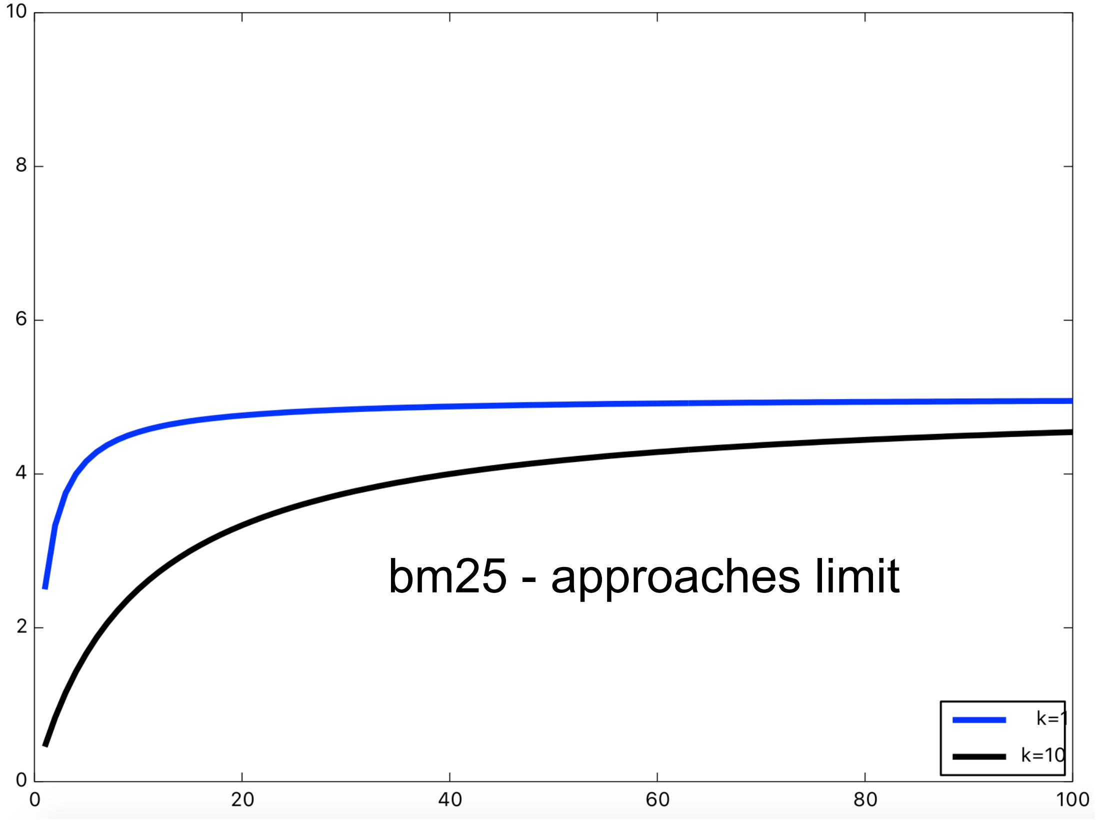
Tf saturation curve

- limits influence of tf
- allows to tune influence by tweaking k

$$w(t) = \frac{f_{t,d}}{f_{t,d} + k}$$

$f_{t,d}$ = frequency of term in document

k = saturation parameter



elastic

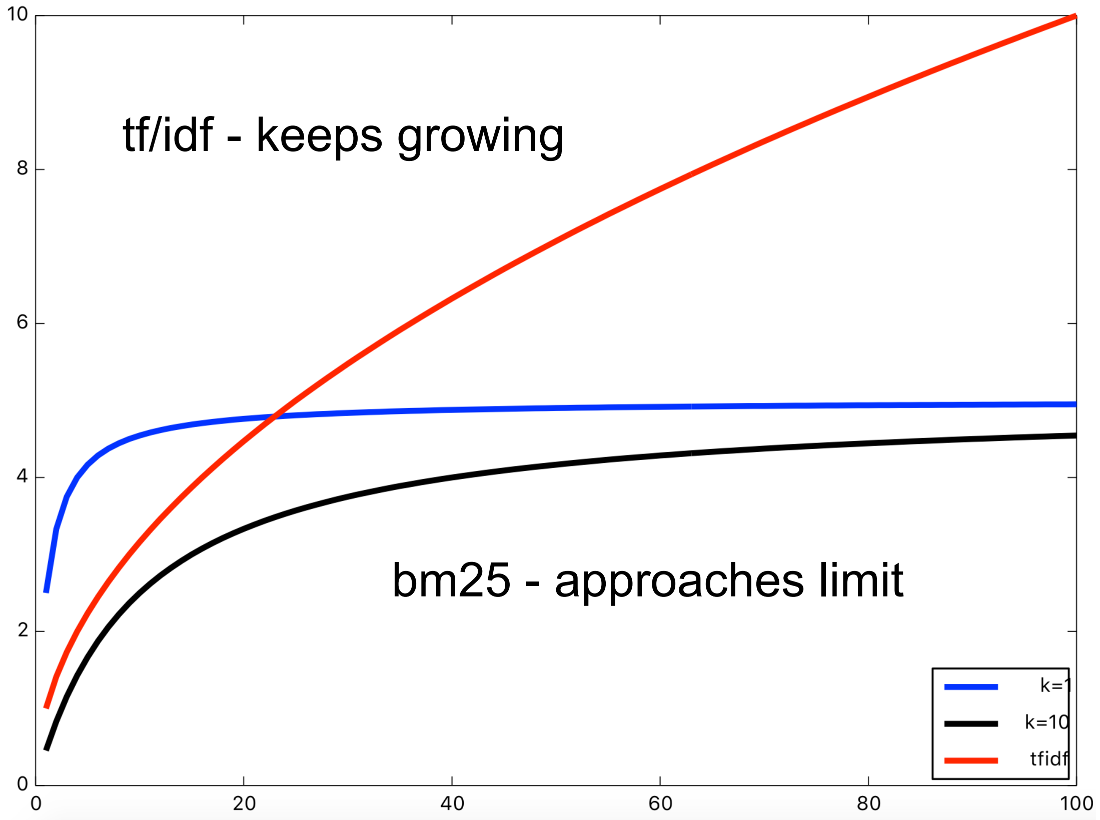
$f_{t,d}$ = frequency of term in document

k = saturation parameter

Tf saturation curve

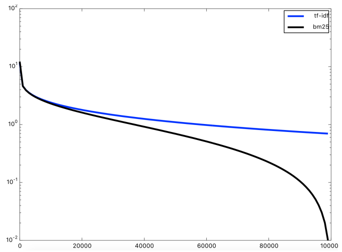
- limits influence of tf
- allows to tune influence by tweaking k

$$w(t) = \frac{f_{t,d}}{f_{t,d} + k}$$



elastic

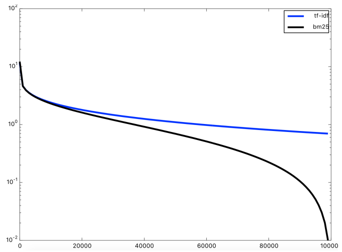
BM25 - We are here…



idf - how popular is the term in the corpus?

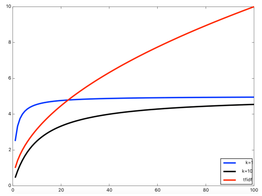
$$\mathrm{bm25}(d) = \sum_{t \in q, f_{t,d} > 0} \log\left(1 + \frac{N - df_t + 0.5}{df_t + 0.5}\right) \cdot \frac{f_{t,d}}{f_{t,d} + k \cdot (1 - b + b\frac{l(d)}{avgdl})}$$

BM25 - We are here…



idf - how popular is the term in the corpus?

$$\mathrm{bm25}(d) = \sum_{t \in q, f_{t,d} > 0} \log\left(1 + \frac{N - df_t + 0.5}{df_t + 0.5}\right) \cdot \frac{f_{t,d}}{f_{t,d} + k \cdot (1 - b + b\frac{l(d)}{avgdl})}$$



saturation curve - limit influence of tf on the score

So…we assume all documents have same length?

- Poisson distribution: Assumes a fixed length of documents
- But they don't have that (most of the time)
- We have to incorporate this too!
- scale tf by it like so:

$$\mathrm{bm25}(d) = \sum_{t \in q, f_{t,d} > 0} \mathrm{idf}(t) \cdot \frac{f_{t,d}}{f_{t,d} + k \cdot \left(1 - b + b \cdot \frac{l(d)}{avgdl} \right)}$$

Interpolation between 1 and document length/average document length

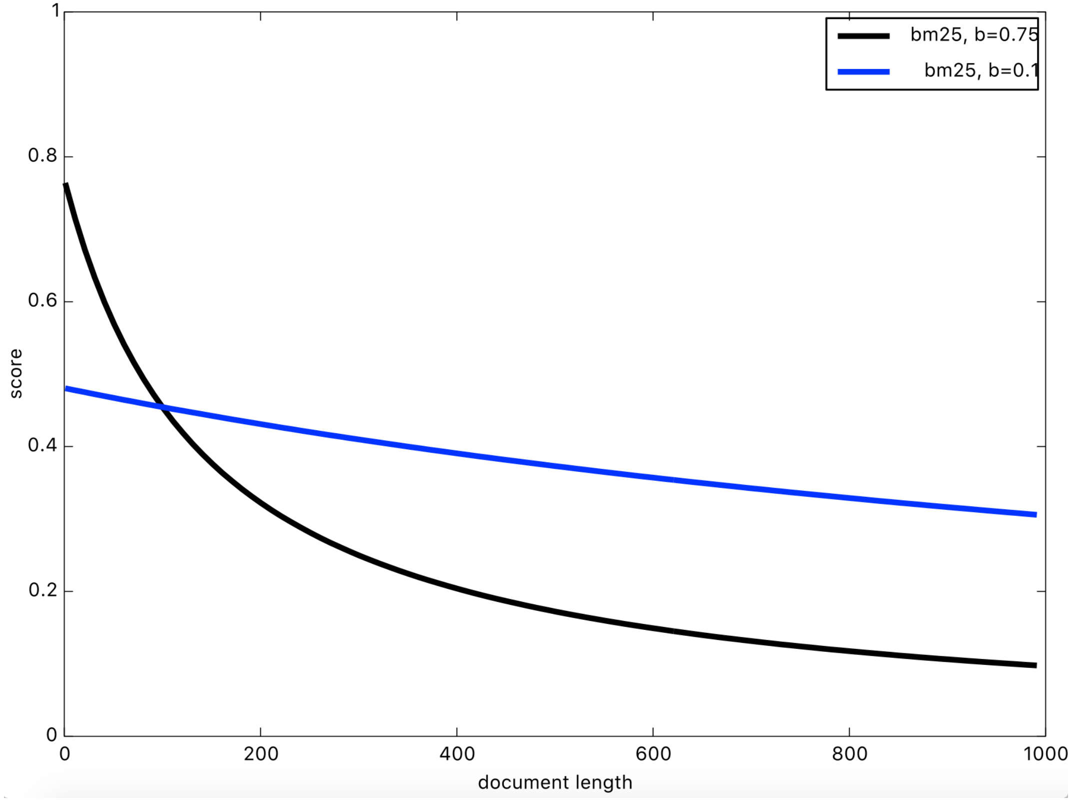
elastic

Influence of b

$$
\begin{aligned}
f_{t,d} &= \text{frequency of term in document} \\
k &= \text{saturation parameter} \\
b &= \text{length parameter} \\
l(d) &= \text{number of tokens in document} \\
avgdl &= \text{average document length in corpus}
\end{aligned}
$$

- tweak influence of document length

$$norm(t) = \frac{f_{t,d}}{f_{t,d} + k \cdot \left(1 - b + b \cdot \frac{l(d)}{avgdl}\right)}$$



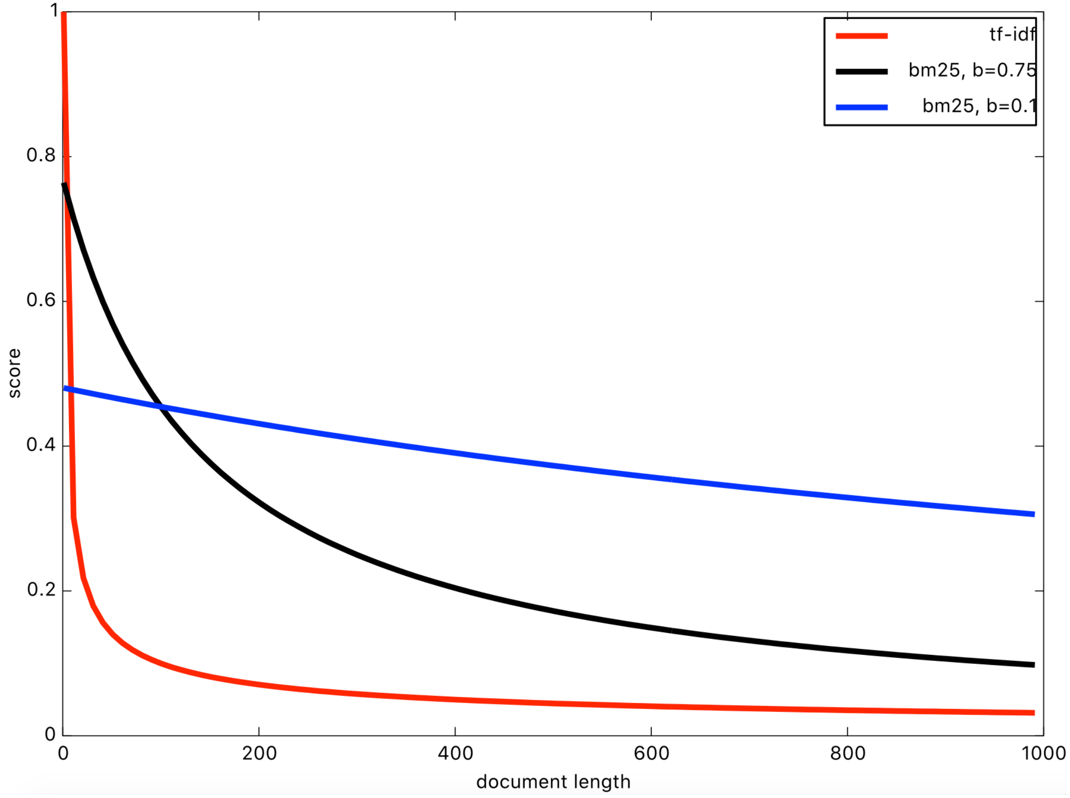
elastic

Influence of b

$f_{t,d}$ = frequency of term in document
k = saturation parameter
b = length parameter
$l(d)$ = number of tokens in document
$avgdl$ = average document length in corpus

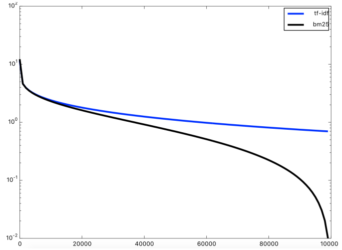
- tweak influence of document length

$$norm(t) = \frac{f_{t,d}}{f_{t,d} + k \cdot \left(1 - b + b \cdot \frac{l(d)}{avgdl}\right)}$$



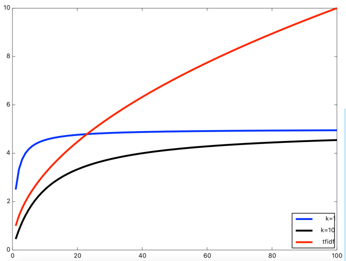
elastic

BM25 - We are here…



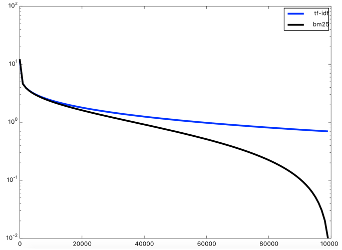
idf - how popular is the term in the corpus?

$$\mathrm{bm25}(d) = \sum_{t \in q, f_{t,d} > 0} \log\left(1 + \frac{N - df_t + 0.5}{df_t + 0.5}\right) \cdot \frac{f_{t,d}}{f_{t,d} + k \cdot (1 - b + b\frac{l(d)}{avgdl})}$$



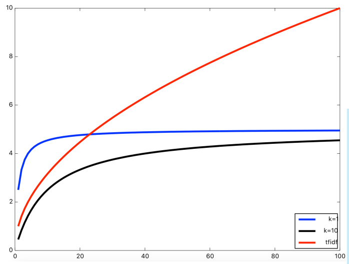
saturation curve - limit influence of tf on the score

BM25 - We are done!



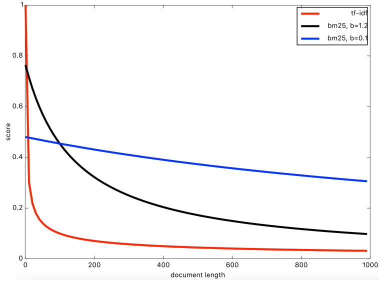
idf - how popular is the term in the corpus?

$$\mathrm{bm25}(d) = \sum_{t \in q, f_{t,d} > 0} \log\left(1 + \frac{N - df_t + 0.5}{df_t + 0.5}\right) \cdot \frac{f_{t,d}}{f_{t,d} + k \cdot (1 - b + b\frac{l(d)}{avgdl})}$$



saturation curve - limit influence of tf on the score

length weighing - tweak influence of document length

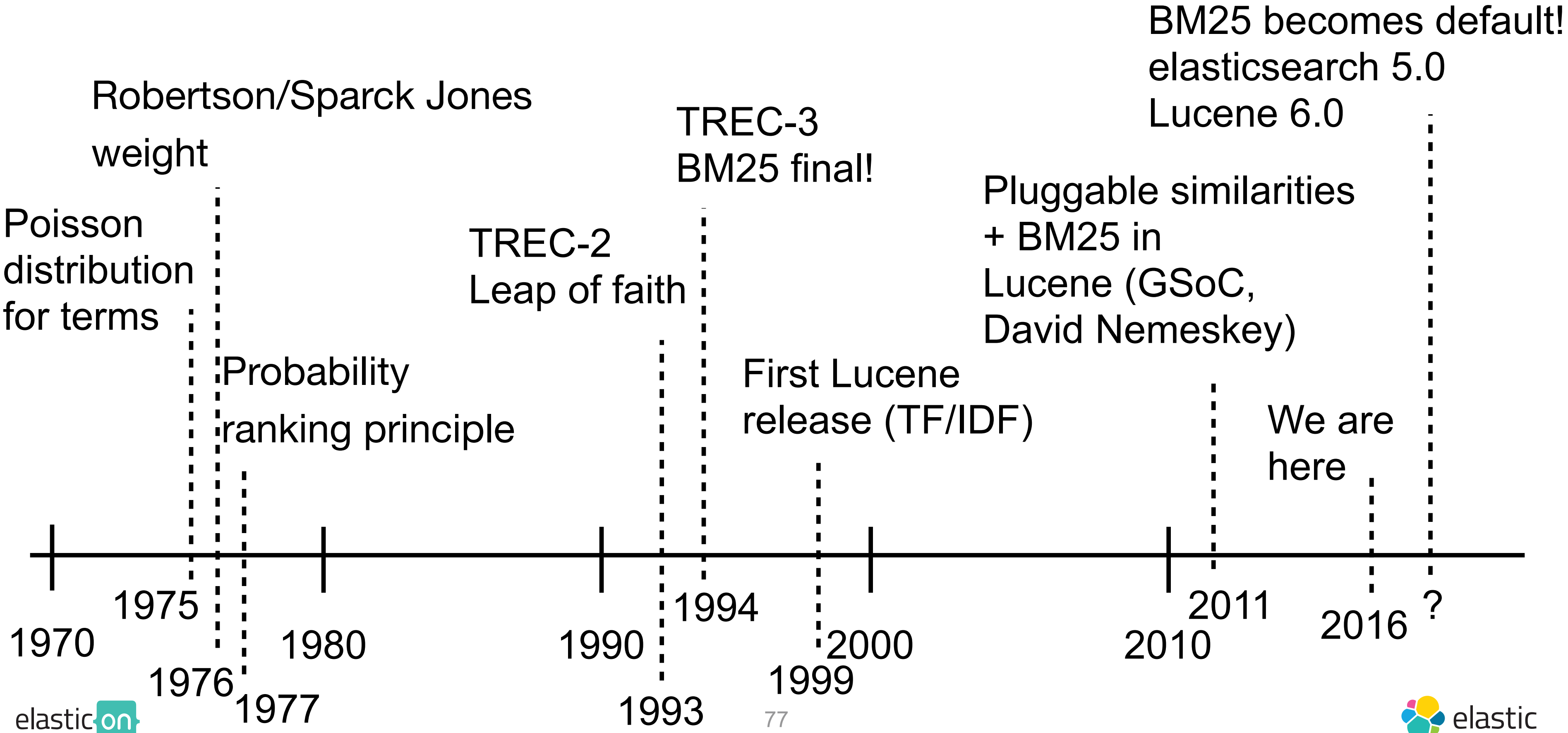


Is BM25 probabilistic?

- many approximations
- really hard to get the probabilities right even with unlimited data

BM25 is “inspired” by probabilistic ranking.

A short history of BM25



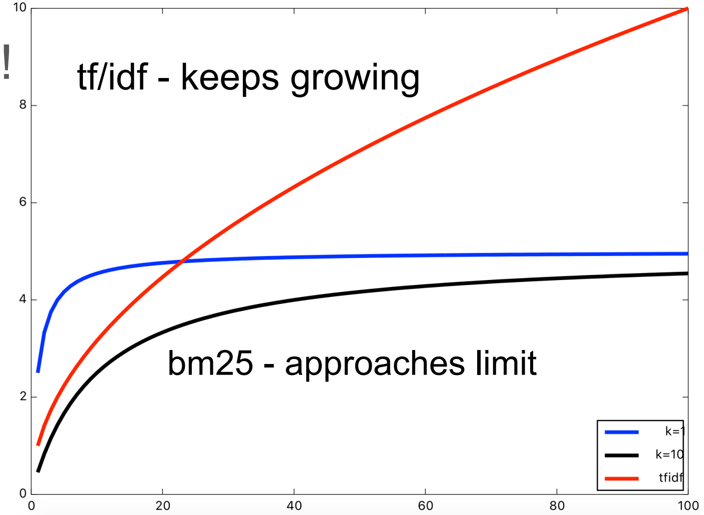
So…will I get a better scoring with BM25?

Pros with the frequency cutoff

TF/IDF: common words can still influence the score!

BM25: limits influence of term frequency

- less influence of common words
- no more coord factor!
- check if you should disable coord for bool queries?
  `index.similarity.default.type: BM25`



Other benefits

parameters can be tweaked. To update:

- close index
- update mapping (or settings)
- re-open index

Mathematical framework to include non-textual features

A warning: Lower automatic boost for short fields

With TF/IDF: short fields (title,…) are automatically scored higher

BM25: Scales field length with average

- field length treatment does not automatically boost short fields (you have to explicitly boost)
- might need to adjust boost

Is BM25 better?

- Literature suggests so
- Challenges suggest so (TREC,…)
- Users say so
- Lucene developers say so
- Konrad Beiske says so: Blog “BM25 vs Lucene Default Similarity”

But: It depends on the features of your corpus.

Finally: You can try it out now! Lucene stores everything necessary already.

Useful literature

- Manning et al., Introduction to Information retrieval
- Robertson and Zaragoza, The Probabilistic Relevance Framework: BM25 and Beyond
- Robertson et al., Okapi at TREC-3
- https://github.com/apache/lucene-solr/blob/master/lucene/core/src/java/org/apache/lucene/search/similarities/BM25Similarity.java

Thank you!