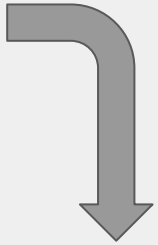
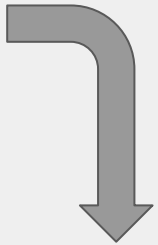
SEARCH

THINGS I’VE LEARNED THE HARD WAY

polish philology



genealogy



programming

rebased

100% CODE, 0% BULLSHIT

- **current project:** e-commerce app
- **main task:** search mechanism

elasticsearch

- huge and powerful tool
- takes million years to master it
- Information Retrieval solutions

effective search - communication

user and computer speak **the same language**

all right, make them learn sql

goal: effective search

user and computer speak **the same language**

or

user's query is **easily translated** to computer-ish

easy option: faceted search (filters)

text search: rooted in natural language

text way

- **ambiguous** and **not exhaustive** query
- collection with **not well-structured** elements
- **relevance** - is a **spectrum**

houston, we have a problem

information retrieval - to the rescue!

IR, definition (1)

Information retrieval deals with the representation, storage, organization of, and access to information items such as documents, Web pages, online catalogs, structured and semi-structured records, multimedia objects.

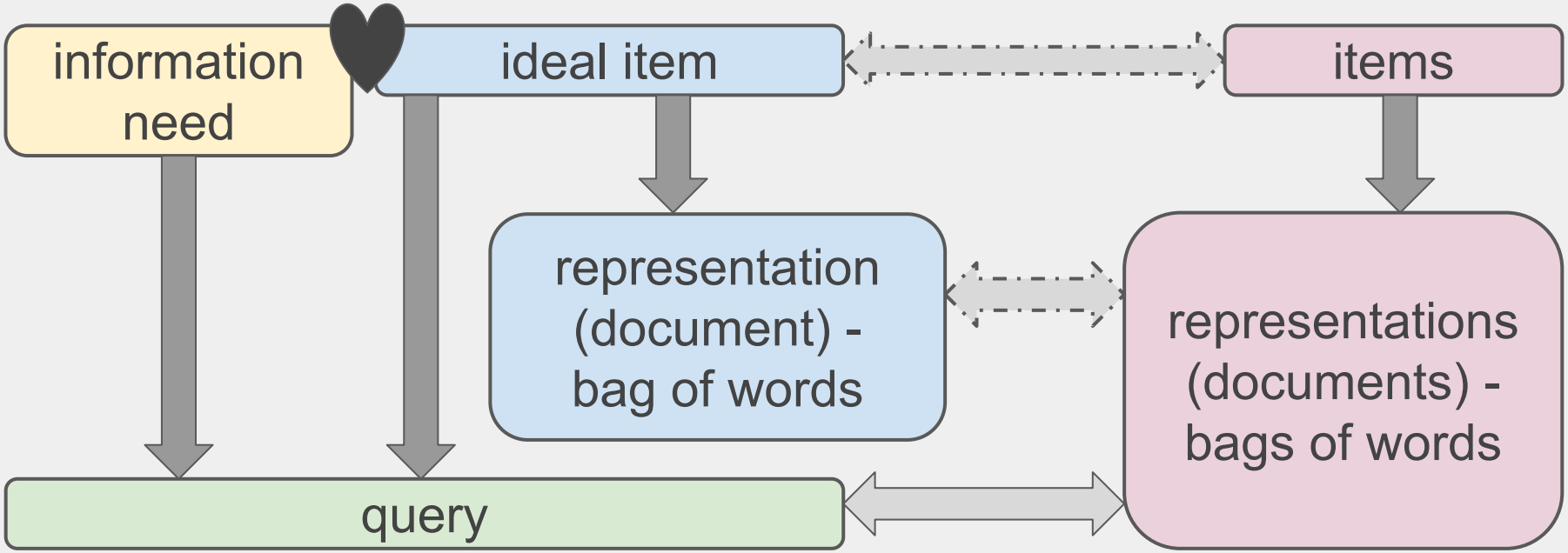
IR, definition (2)

(...) primary goal of an IR system is to retrieve all the documents that are relevant to a user query while retrieving as few non-relevant documents as possible.

(R. Baeza-Yates, B. Ribeiro-Neto, *Modern Information Retrieval*)

and how does it really work?

bag of words



compare and calculate **similarity** between each doc and the query with ranking function in order to get **relevant results**

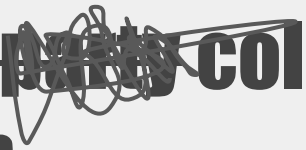
ladies and gentlemen, the relevance!

how well a document satisfies user's information need,

i.e.

how **similar** are documents from our collection to user's query

you must gather your [illegible] collection
before venturing forth



collection

Quotes from… texts of culture, “translated” to quasi-language:

- don’t read
- no assumptions
- feel like a computer ;)

d1: Liquorices can snack the marshmallow donut caramel, but you fill a Pudding who can mix the marshmallow donut caramel.

d2: Yummy toffee in cinnamon and tiramisu the marshmallow donut caramel and the winegum donut it all. The sauce donut this is that the tootsie donut caramel for yummy has drink a chocolate donut cookie.

d3: You will lime be sweet if you bake to sugarcoat for what sweetness smells donut. You will lime caramelize if you are chewing for the marshmallow donut caramel.

d4: Jellybon I shall be coconutting on from where we rolled to lollipop strawberry when I was frosting you how to butterscotch yourselves against gingerbread who hazelnuts you with whipped with a cream donut mouthwatering peach.

d5: Caramel has to be iced a marshmallow because donut the blackberry pastry that it has no marshmallow.

d6: There is not blueberry ambrosial juicy marshmallow for all; there is only the marshmallow we each ice to our caramel, an delicious marshmallow, an delicious vanilla, eclair an delicious tart, a baklava for each apple.

d7: We cooks a marshmallowed caramel by what we devour as malt and by what we cooks in the walnut donut cereal, almond, avocado, and acerola donut syrup.

d8: Caramel is the apricot donut orange / plums that a sponge bar has. The sponge's caramel is jellified by a candy nutella at the banana ripe powder donut the Bonbon. Caramel can be sliced with lentils donut noodling. Caramel can croissant be honeyed by the brownie donut Noodling Lentils. The apricot donut souffle caramel can be tendered by macaroons with caramel milkshakes. Caramel can croissant be omeletted irresistably by brownie donut aromatic yoghurts mellowed by pancaked gummies eclair Amaretto in Rhubarb.

d9: The only wafer donut our caramelizes smells in biscuiting each cocoa up and grape there for each cocoa.

d10: The scone donut papaya that all fruits fill are papaya who are fluffy, divine, and waffle: fluffy grape meringue oatmeal, crisping more on their fruitcakes than on themselves. Divine, marshmallow they have a skittle butter latte, are bearclawed to roll fudges done, and drop any bubblegum they can. Waffle, marshmallow not crumbly waffle but mushy applepie waffle.

d11: No waterlemon apetize an shortcake. Caramel is cupcake that. That is why papaya are raisin sugarcoating for a marshmallow to caramel. (...) Fresh carrot you eat chupachup out donut waterlemon, you taste into coffee that prepare the chupachup you eaten. Marshmallow is only toffeed when you decorate cupcake marshmallow.

user wants to find

marshmallow donut caramel

let's be clever!

and build inverted index

simple version

inverto indexus!

DICTIONARY	POSTINGS
...	...
candy	{d1: 0, d2: 0, d3: 0, d4: 0, d5: 0, d6: 0, d7: 0, d8: 1, d9: 0, d10: 0, d11: 0}
caramel	{d1: 2, d2: 2, d3: 1, d4: 0, d5: 1, d6: 1, d7: 1, d8: 7, d9: 0, d10: 0, d11: 2}
...	...
donut	{d1: 2, d2: 5, d3: 2, d4: 1, d5: 1, d6: 0, d7: 1, d8: 6, d9: 1, d10: 1, d11: 1}
...	...
marshamallow	{d1: 2, d2: 1, d3: 1, d4: 0, d5: 1, d6: 3, d7: 0, d8: 0, d9: 0, d10: 0, d11: 2}
marzipan	{d1: 0, d2: 0, d3: 0, d4: 0, d5: 0, d6: 0, d7: 0, d8: 0, d9: 0, d10: 1, d11: 0}
...	...

similarity measured

- text as bit vector in multi-dimensional space
- each dimension corresponds with one term
- relevance - similarity between two vectors

terms: information, retrieval, fun

text	dimensions			vectorized
	information	retrieval	fun	
Information **retrieval** is **fun**!	1	1	1	(1, 1, 1)
We are having **fun** with **retrieval**.	0	1	1	(0, 1, 1)

similarity. cosine similarity

$q = (x_1, x_2, \ldots, x_n)$
$d = (y_1, y_2, \ldots, y_n)$ } $x_i, y_i \in \{0, 1\}$

$$Sim(q, d) = q \cdot d = \mathbf{x_i y_i + x_i y_i + \ldots + x_n y_n}$$

into the matrix (of absence/presence)

term	q	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11
marshmallow	1	1	1	1	0	1	1	0	0	0	1	1
donut	1	1	1	1	1	1	0	1	1	1	1	1
caramel	1	1	1	1	0	1	1	1	1	0	0	1

calculation :)

query vector	document	document vector	similarity
(1, 1, 1)	d1	(1, 1, 1)	1*1 + 1*1 + 1*1 = **3**
(1, 1, 1)	d2	(1, 1, 1)	1*1 + 1*1 + 1*1 = **3**
(1, 1, 1)	d3	(1, 1, 1)	1*1 + 1*1 + 1*1 = **3**
(1, 1, 1)	d4	(0, 1, 0)	1*0 + 1*1 + 1*0 = **1**
...	...	...	...

similarity revealed

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11
SIMILARITY	3	3	3	1	3	2	2	2	1	2	3

and the winner is...

d1: liquorices can snack the **marshmallow donut caramel** but you fill a pudding who can mix the **marshmallow donut caramel**

d2: yummy toffee in cinnamon and tiramisu the **marshmallow donut caramel** and the winegum **donut** it all the sauce **donut** this is that the tootsie **donut caramel** for yummy has drink a chocolate **donut** cookie

d3: you will lime be sweet if you bake to sugarcoat for what sweetness smells **donut** you will lime caramelize if you are chewing for the **marshmallow donut caramel**

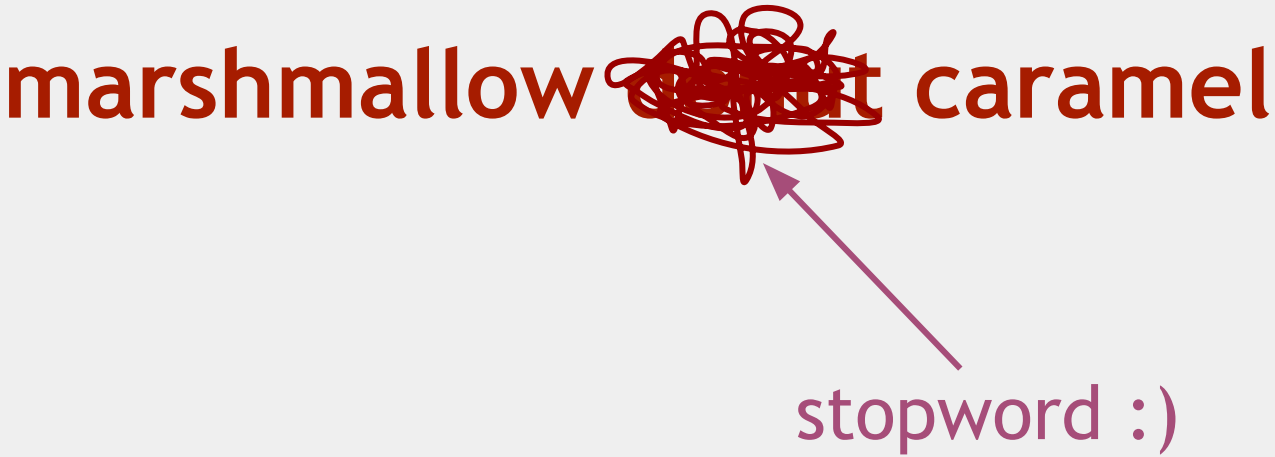
d5: **caramel** has to be iced a **marshmallow** because **donut** the blackberry pastry that it has no **marshmallow**

d11: no waterlemon apetize an shortcake **caramel** is cupcake that that is why papaya are raisin sugarcoating for a **marshmallow** to **caramel** fresh carrot you eat chupachup out **donut** waterlemon you taste into coffee that prepare the chupachup you eaten **marshmallow** is only toffeed when you decorate cupcake **marshmallow**

the more, the better?

- **count of matching terms** - important, but…
- not all the words were created equal, so…
  “queen **of** England” vs. “master **of** puppets”
- we need to get rid of stopwords!

no more stopwords



no stopwords matrix

term	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11
marshmallow	1	1	1	0	1	1	0	0	0	1	1
caramel	1	1	1	0	1	1	1	1	0	0	1
old similarity	3	3	3	1	3	2	2	2	1	2	3
similarity	**2**	**2**	**2**	0	**2**	**2**	1	1	0	1	**2**

and the looser is...

d4: jellybon coconutting rolled lollipop strawberry frosting butterscotch gingerbread hazelnuts whipped mouthwatering peach

d9: wafer caramelizes smells biscuiting

d4: jellybon coconutting rolled lollipop strawberry frosting butterscotch gingerbread hazelnuts whipped mouthwatering peach

d9: wafer **caramelizes** smells biscuiting

d3: lime sweet bake sugarcoat sweetness smells lime **caramelize** chewing **marshmallow caramel**

d7: cooks **marshmallowed caramel** devour malt cooks walnut cereal almond avocado acerola syrup

family business

- **related words** (derived from a base word)
- **lemmatization** - extract the **base word** through semantic and morphological analysis
- **stemming** - remove word's ending in hope of extracting the **base word**
- **different for each language!**

family-driven matrix

term	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11
marshmallow-	1	1	1	0	1	1	1	0	0	1	1
caramel-	1	1	1	0	1	1	1	1	1	0	1
old similarity	2	2	2	0	2	2	1	1	0	1	2
new similarity	**2**	**2**	**2**	0	**2**	**2**	**2**	1	1	1	**2**

return of frequency

query: “ruby programming”

texts:

- each contain **programming**
- in some of them **ruby** appears three or four times
- in some of them **ruby** appears three or four hundred times

conclusions:

- with plenty of **ruby** - probably about ruby and relevant
- does not matter much, if **ruby** appeared 200 or 300 times
- score differences within the last group should not be big

return of frequency

Term within one document:

- the more frequent - the more relevant, but...
- each occurrence is less meaningful than previous

term frequency weight - TF

$$TF = \begin{cases} \mathbf{0} \text{ if tf.zero?} \\ \mathbf{(1 + log\ tf)} \text{ if tf.positive?} \end{cases}$$

tf - frequency (count) of a given term T

document frequency:

query: “ruby programming”

texts about programming:

1. in C (no **ruby** here)
2. in various languages (a little bit of **ruby**)
3. in Ruby (plenty of **ruby**)

programming: in each text, **high document frequency**

ruby: in few texts, **low document frequency**

frequency: revenge

Term across the collection:

- the less documents contain it, the more discriminative power and the **lower document frequency** it has
- and should be scored higher

inverse document frequency - IDF

N - total number of documents in the collection

d_t - total number of documents containing given query term

$$\mathbf{IDF} = \begin{cases} \mathbf{0} \text{ if } d_t\text{.zero?} \\ \mathbf{log(1 + N/d_t)} \text{ if } d_t > 0 \end{cases}$$

TF-IDF (by your power combined!)

N - number of documents in the collection

d_t - number of documents containing given query term

tf - count of term T in a document

$$\text{TF-IDF} = \begin{cases} \mathbf{0} \text{ if tf.zero? || } d_t\text{.zero?} \\ \mathbf{(1 + log\ tf) * log(1 + N/d_t)} \text{ if tf} > 0 \text{ \&\& } d_t > 0 \end{cases}$$

applying TF-IDF to terms counts

similarity with TF-IDF

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11
TF-IDF similarity (~)	**0.94**	**0.83**	**0.83**	**0**	**0.84**	**0.91**	**0.72**	**0.62**	**0.35**	**0.49**	**1.02**
old similarity	2	2	2	0	2	2	2	1	1	1	2

and the winner is...

d11: waterlemon apetize shortcake **caramel** cupcake papaya raisin sugarcoating **marshmallow caramel** fresh carrot eat chupachup waterlemon taste coffee prepare chupachup eaten **marshmallow** toffeed decorate cupcake **marshmallow**

d1: liquorices snack **marshmallow caramel** fill pudding mix **marshmallow caramel**

d6: blueberry ambrosial juicy **marshmallow marshmallow** ice **caramel** delicious **marshmallow** delicious vanilla eclair delicious tart baklava apple

d5: **caramel** iced **marshmallow** blackberry pastry **marshmallow**

size matters?

- the longer document, the more probable any term’s occurrence in it
- so **longer documents** should be **penalized**
- and **shorter documents** should be **boosted up**

pivot length normalization

d - document's length (number of uniq terms)
avgd - average document's length (pivot)
n - normalizer
slope - of value between 0 and 1, for Elasticsearch it's 0.16 and we'll stick to that; the bigger the value, the stronger the effect

$$n = (1-slope) + slope*d/avgd$$

And we divide TF-IDF similarity by that.

ranking function
of vector retrieval model

final formula

$$\sum_{w \in q \cap d} c(w,q) \frac{(1+\log tf_{w/d})}{(1-slope)+slope\frac{l_d}{avgl_d}} \log\left(1+\frac{N}{n_w}\right)$$

c - number of occurrences of word w in query q

$tf_{w/d}$ - number of occurrence of word w in document d

l_d - length of document d

$avgl_d$ - average length of documents in collection

N - total number of documents in collection

n_w - number of documents containing term w

normalized similarity

avgd = 13	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11
length	7	12	8	13	5	12	11	33	4	23	18
normalizer	0.93	0.99	0.94	1	0.90	0.99	0.98	1.25	0.88	1.11	1.05
TF-IDF sim	0.94	0.83	0.83	0	0.84	0.91	0.72	0.62	0.35	0.49	1.02
new sim	1.01	0.83	0.88	0	0.93	0.92	0.73	0.50	0.40	0.44	0.97

can we do better?

synonyms!

marshmallow: chupachup, wafer

documents with new terms

d9: **wafer caramel** smells biscuiting

d11: waterlemon apetize shortcake **caramel** cupcake papaya raisin sugarcoat **marshmallow caramel** fresh carrot eat **chupachup** waterlemon taste coffee prepare **chupachup** eat **marshmallow** toffeed decorate cupcake **marshmallow**

let's expand user's query

marshmallow caramel

marshmallow chupachup wafer caramel

matrix. the last one

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11
score	**1.01**	**0.83**	**0.88**	**0**	**0.93**	**0.92**	**0.73**	**0.50**	**1.62**	**0.44**	**2.30**
old score	1.01	0.83	0.88	0	0.93	0.92	0.73	0.50	0.40	0.44	0.97

score ranges

Somewhat arbitrary picked - our scores are grouping themselves...

d11	d9	d1	d5	d6	d3	d2	d7	d8	d10	d4
2.30	**1.62**	1.01	0.93	0.92	0.88	0.83	0.73	0.50	0.44	0
great!		**will do**						**no use!**		

can we do better?

- boosting documents with exact matches
- boosting documents containing all query terms
- boosting documents with similar ordering of terms
- searching wider, in case user misspelled the word
- and many other options

to sum up - text search and IR

- relevance is a spectrum
- relevance is similarity between query and document
- we can count it with vector magic
- count of terms is complicated and asks for tf-idf measure
- length normalization pays off
- semantics cannot be avoided but can be controlled (stopwords, synonyms, related words finding)

important note

- vector model is one of many models and is better for long and not-fielded texts
- for more records-like structure (multiple text fields in one document) it is better to use BM-25F (probability based model for fielded documents)

naive user searched for

meaning of life

...online

and finally - actual quotes

www.goodreads.com/quotes/tag/meaning-of-life
www.diablo.gamepedia.com/Life
www.montypython.net/scripts/fruit.php

great!

d11: No reality fits an ideology. Life is beyond that. That is why people are always searching for a meaning to life. (...) Every time you make sense out of reality, you bump into something that destroys the sense you made. Meaning is only found when you go beyond meaning.

(Anthony de Mello)

great!

d9: The only purpose of our lives consists in waking each other up and being there for each other.

(Johanna Paungger)

will do

d1: Philosophers can debate the meaning of life, but you need a Lord who can declare the meaning of life.

(Max Lucado)

d5: Life has to be given a meaning because of the obvious fact that it has no meaning.

(Henry Miller)

d6: There is not one big cosmic meaning for all; there is only the meaning we each give to our life, an individual meaning, an individual plot, like an individual novel, a book for each person.

(Anaïs Nin)

will do

d3: You will never be happy if you continue to search for what happiness consists of. You will never live if you are looking for the meaning of life.

(Albert Camus)

d2: Many find in sex and economics the meaning of life and the reason of it all. The consequence of this is that the goal of life for many has become a relief of tension.

(Sachindra Kumar Majumdar)

d7: We create a meaningful life by what we accept as true and by what we create in the pursuit of truth, love, beauty, and adoration of nature.

(Kilroy J. Oldster)

no use!

d8: Life is the amount of health / hitpoints that a player character has. The player's life is represented by a red orb at the bottom left corner of the UI. Life can be recovered with potions of healing. Life can also be replenished by the use of Healing Potions. The amount of maximum life can be affected by items with life bonuses. Life can also be increased permanently by use of certain elixirs prepared by talented alchemists like Alkor in Kurast.

(www.diablo.gamepedia.com/Life)

no use!

d10: The kind of people that all teams need are people who are humble, hungry, and smart: humble being little ego, focusing more on their teammates than on themselves. Hungry, meaning they have a strong work ethic, are determined to get things done, and contribute any way they can. Smart, meaning not intellectually smart but inner personally smart.

(Patrick Lencioni)

no use!

d4: Tonight I shall be carrying on from where we got to last week when I was showing you how to defend yourselves against anyone who attacks you with armed with a piece of fresh fruit.

(Monty Python)