

Building Large Scale Search Engines

Jumana Bahrainwala

(@JumzB)(jzbahrai.github.io)

who am i?

- Data Scientist at Unata
- Work mainly on the Search Engine
- Use Python, Elasticsearch and Postgresql

Things to think about:

- 1) Algorithms you are going to use
- 2) The way you structure your data
- 3) Evaluating your search

PART 1

Let's talk Algorithms and use Cases

Case1 : website Ranking

PageRank - Explained intuitively

“

”

Probability that someone will view a page

Citation model

Advantages: ONE NUMBER (a mathematical rank)

PageRank - Explained intuitively

Important

Very Important

Also Very Important

Thesauras.com

AnimalKingdom.com

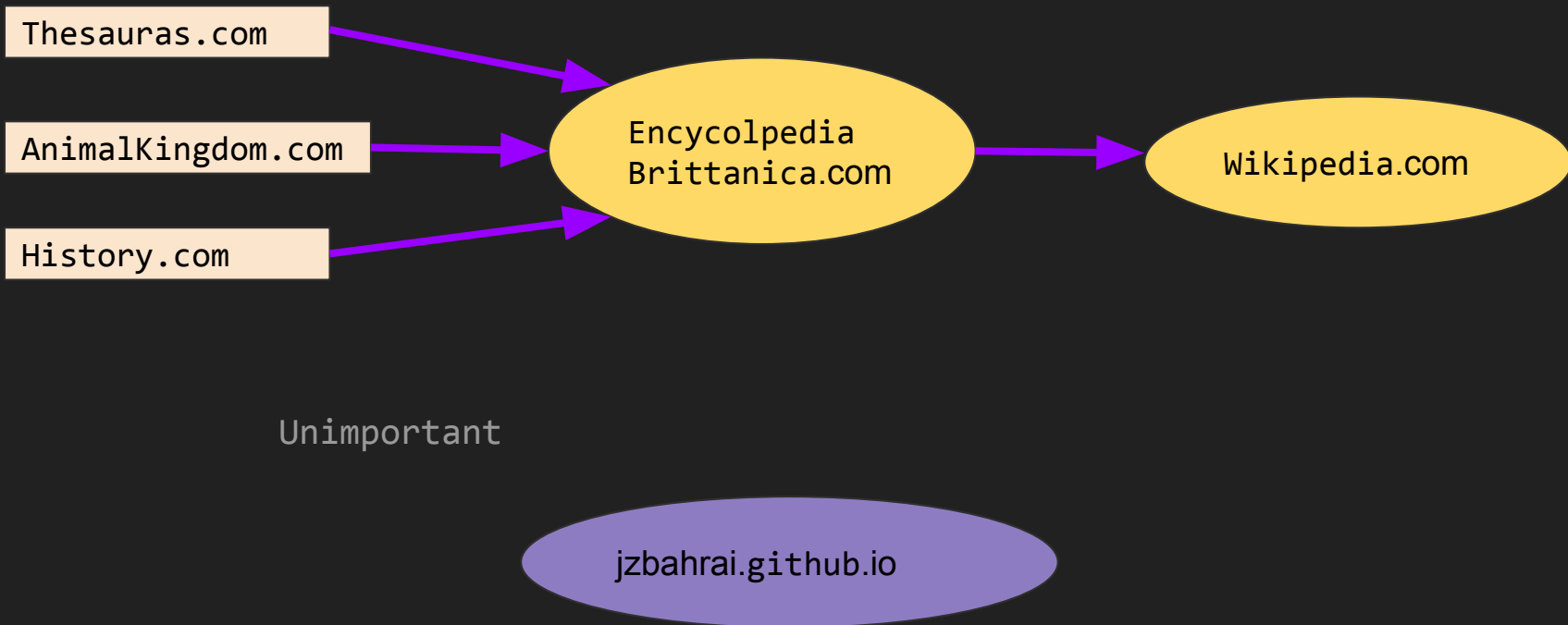
History.com

Encycolpedia
Brittanica.com

Wikipedia.com

Unimportant

jzbahrai.github.io



Case 2: what about full text search?

TERM FREQUENCY:

“The Number of times a term appears in **a** document”

“It was, he thought, the difference between being dragged into the arena to face a battle to the death and walking into the arena with your head held high. Some people, perhaps, would say that there was little to choose between the two ways, but Dumbledore knew - and so do I, thought **Harry**, with a rush of fierce pride, and so did my parents - that there was all the difference in the world.” - *Harry Potter and the Half Blood Prince*

Total # words = 72

of times **Harry** shows up - 1

Inverse Document Frequency)

Inverse Document Frequency

“The Number of times a term appears in **all** documents”

Why does this matter?

- Words such as “the”, “of”
- Give weight to uncommon words

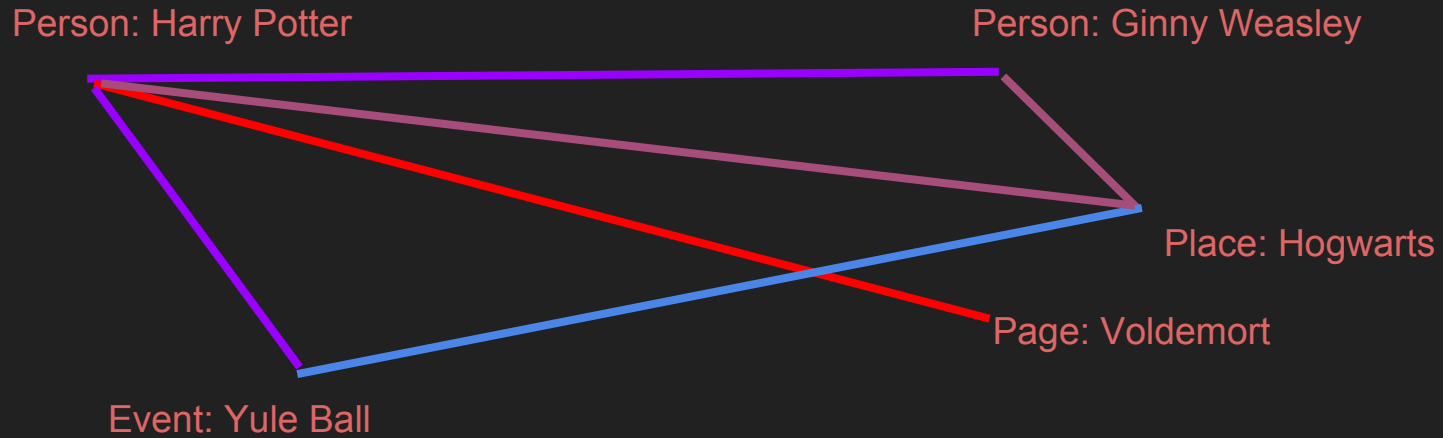
Inverse Document Frequency

“I have stolen princesses back from sleeping barrow kings. I burned down the town of Trebon. I have spent the night with Felurian and left with both my sanity and my life. I was expelled from the University at a younger age than most people are allowed in. I tread paths by moonlight that others fear to speak of during day. I have talked to gods, loved women, and written songs that make the minstrels weep. You may have heard of me.” - *The Name of the Wind, Partrick Rothfuss*

“Eliza, I don’t have a dollar to my name. An acre of land, a troop to command, a dollop of fame. All I have’s my honor, a tolerance for pain. A couple of college credits and my top-notch brain. Insane, your family brings out a different side of me. Peggy confides in me, Angelica tried to take a bite of me. No stress, my love for you is never in doubt. We’ll get a little place in Harlem and we’ll figure it out. I’ve been livin’ without a family since I was a child. My father left, my mother died, I grew up buckwild. But I’ll never forget my mother’s face, that was real. And long as I’m alive, Eliza, swear to God. You’ll never feel so...” - *Hamilton the Musical*

Case3: Entity Relationship Matching

Graph Search



PART 2

How are you preprocessing your data

Natural Language Processing

Parsing:

1. hello@world.com
2. MY name is "Jumana Bahrainwala"
3. Meet me at 10 STOP We are going for tacos STOP

Break the word up by keywords, whitespace, commas, ngrams

This is called "tokenization"

Natural Language Processing

Stemming tokens:

1. Natural Language Processing to tokens (nltk)
2. Word forms, tenses, plurals
3. Jumping-> Jump, Brownies->Brownie, hope->HOPE->Hope, mine->my

Natural Language Processing

Fuzziness/ Spell Checker:

- Levenshtein Distance
- Basic computer science problem:
 - How many changes are needed to make “Mourning” to “Morning”?

Natural Language Processing

Things to think about:

Should you match all words typed in or a proportion?

Should you match on fuzzy words before typed words, how do you make that distinction?

Should you match on description, name and tags equally?

PART 3

Evaluating Search

Retrieved Documents -

Your 10,000 google results for the word "magic"

Relevant Documents -

The 50,000 google results that could be returned for "magic"

“Search is figuring out a balance between the retrieved and relevant results at any given time.”

Goal - Calculate Precision and Recall

Precision:

Your first page of google results ($\text{Relevant Results} / \text{Retrieved Doc}$)

Recall:

Think of it as the result set you would get if you get all
RELEVANT RESULTS. ($\text{Relevant Results} / \text{Relevant Doc}$)

Buzzfeed - Build a search engine in 60 seconds

1. What's your use case?
 - a. Product Search, Grep, List Search
2. What algorithm are you leaning towards?
 - a. Graph Search, TF-IDF, Page Rank
3. Is there open source tech for this?
 - a. Elastic Search, Apache Solr
4. Preprocess your data
 - a. Tokenization, Lexemes
5. Evaluate your results
 - a. Precision and Recall

Resources

References:

Page Rank: <http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf>

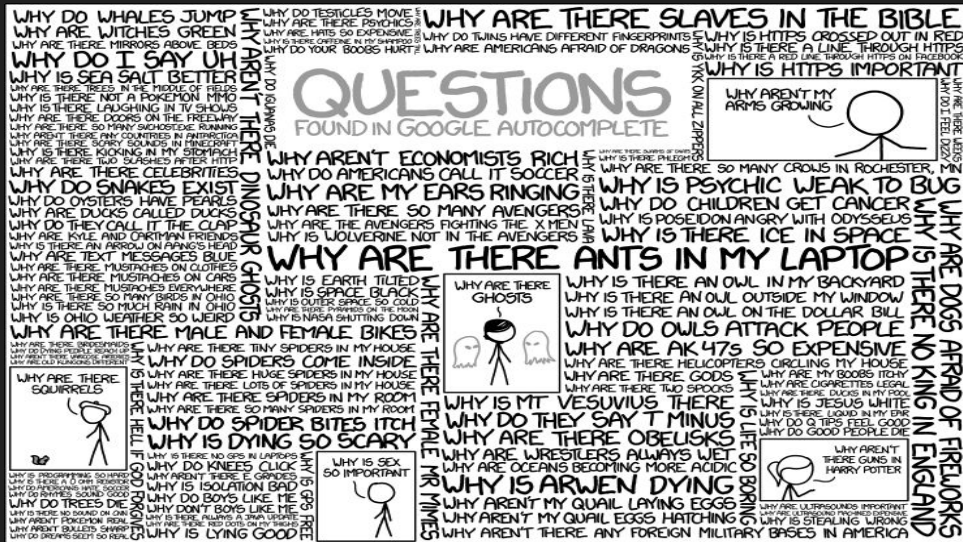
TF-IDF :

<https://pdfs.semanticscholar.org/4f09/e6ec1b7d4390d23881852fd7240994abeb58.pdf>

Building the Social Graph :

<https://research.facebook.com/publications/unicorn-a-system-for-searching-the-social-graph>

Contact Info: jzbahrai@uwaterloo.ca @JumzB



[illegible]