The Amazing Emoji Searcher

Richard Luong · 2015-06-10

The Amazing Emoji Searcher

Agenda

Introduction
About myself
Motivation
Related work
Emoji

Project design

Crawling
Searching
Presenting

Results and Discussion
Advantages
Drawbacks
Future work

DEMO!

Introduction

About myself

Richard Luong
Tsinghua University
Lund University

Industrial engineering
front end: web, iOS

Goal

emoji search engine

combines visual content with natural language

easy for non-technology person to understand

Movitation

emoji search engine

I will kill you.
I will kill you. 😋

Related work

emoji

Google Scholar: 768 results

Emoji

Unicode:

Emoji are pictographs (pictorial symbols) that are typically presented in a colorful form and used inline in text.

Emoji

絵 文字

e
drawing

moji
character

History

First introduced by NTT DoCoMo in 1998 in Japan.

First set included 172 12x12 pixel emoji, encoded into two-byte sequences.

Now

Unicode standard, v6.0
Version 7.0 DRAFT

Different set for different platforms.

In 2015 Apple released a new set, focusing on skin tone, family combinations and country flags.

Project Design

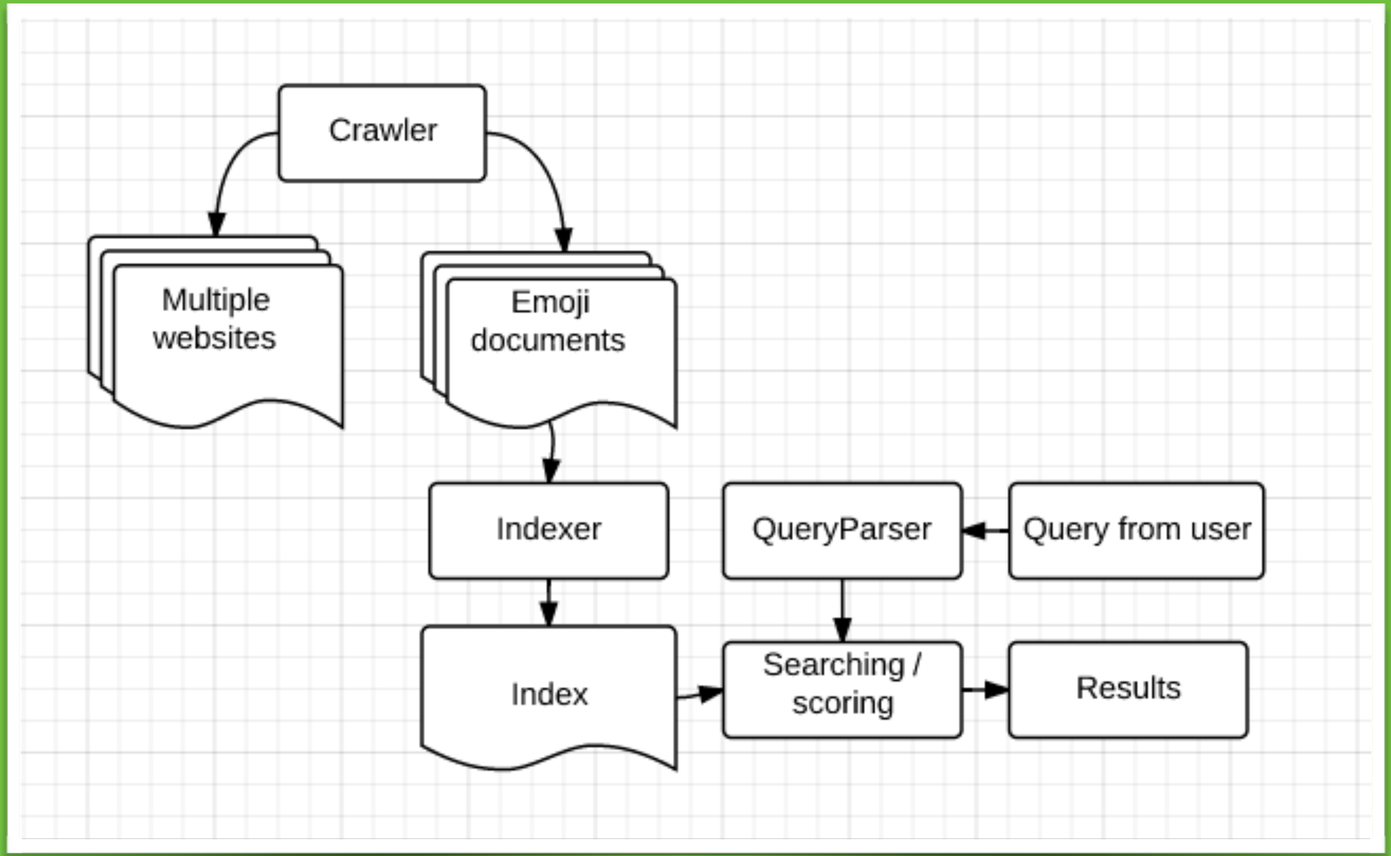
Project Design

crawling

searching
indexing

presenting

Flowchart


Crawler
Multiple websites
Emoji documents
Indexer
QueryParser
Query from user
Index
Searching / scoring
Results

web crawler

to download relevant information about emojis

crawling

Framework used:

Beautiful Soup

crawling

Sources

http://unicode.org

Full Emoji Data

www.unicode.org/emoji/charts/full-emoji-list.html

Full Emoji Data

Emoji Chart Index

This chart provides a list of the Unicode emoji characters, with images from different vendors, version and source information, default style, and annotations. The ordering of the emoji and the annotations are based on Unicode CLDR data.

For details about the format and fields, see Emoji Chart Index and *UTR #51 Unicode Emoji*. Hovering over an emoji shows the name; clicking goes to the Full Emoji List entry for that emoji.

Count	Code	Browser	B&W*	Apple	Andr	Twit	Wind	GMail	DCM	KDDI	SB	Name	Version	Default	Annotations
1	U+1F600								missing	missing	missing	grinning face	V6.1[x]	emoji	face, grin, person
2	U+1F601											grinning face with smiling eyes	V6.0[j]	emoji	eye, face, grin, person, smile
3	U+1F602								missing			face with tears of joy	V6.0[j]	emoji	face, joy, person, tear
4	U+1F603											smiling face with open mouth	V6.0[j]	emoji	face, mouth, open, person, smile
5	U+1F604								missing	missing		smiling face with open mouth and smiling eyes	V6.0[j]	emoji	eye, face, mouth, open, person, smile
6	U+1F605											smiling face with	V6.0[j]	emoji	cold, face, open,

crawling

Sources

http://emojipedia.org

crawling

Document structure

- index
- code
- browser
- bw
- apple
- andr
- twit
- gmail
- wind
- name
- version
- default
- annotations
- emojipedia_url
- desc
- related_to
- known_as

Example document

- index: 1
- code: U+1F600
- browser: 😀
- bw: base64
- apple: base64
- andr: base64
- twit: base64
- gmail: base64
- wind: base64
- name: grinning face
- version: v6.1
- default: emoji
- annotations: [‘face’, ‘grin’, ‘person’]
- emojipedia_url: http://emojipedia.org/grinning-face/
- desc: A face with a big open (grinning) mouth, showing teeth […]
- related_to: [‘Smiling Face With Open Mouth And Smiling Eyes’]
- known_as: [‘Smiley Face Emoji’, ‘Happy Face Emoji’]

crawling

Total documents:

1281

searching
indexing

Framework used:

Whoosh

Schema

searching
indexing

```
index=ID(stored=True),
name=TEXT(analyzer=stem_ana, stored=True,
field_boost=100.0),
annotations=KEYWORD(field_boost=50.0),
related_to=TEXT(analyzer=stem_ana),
known_as=TEXT(field_boost=50.0, analyzer=stem_ana),
browser=STORED,
description=TEXT(analyzer=stem_ana, field_boost=20.0))
```

searching
indexing

Stemming

All text fields are stemmed using `StemmingAnalyzer`**.**

- tokenizer
- lower-case filter
- optional stop filter
- stem filter

Scoring

Documents are scored with the BM25F algorithm.

BM25F is a version of the classic Okapi BM25 ranking function that take document structure and anchor text into account.

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdl}})},$$

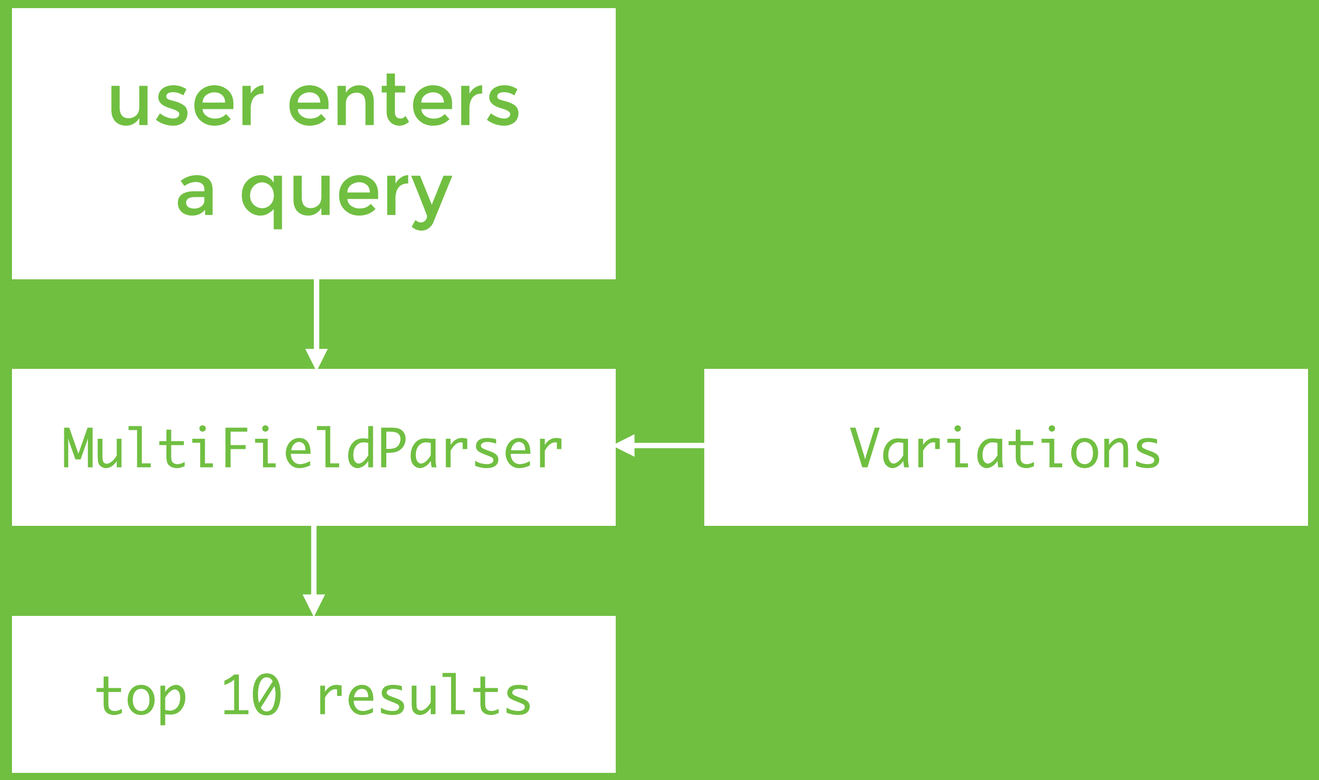
presenting

Frameworks used:

Flask

Bootstrap

presenting


user enters
a query
MultiFieldParser
Variations
top 10 results

Results and discussion

Result

the amazing emoji searcher

Result

theamazingemojisearcher.herokuapp.com

Evaluation

will be done during the demo

Advantages

- **Specialized on emojis**
  - Performs better than Google and Yahoo
- **Not only exact name match**

Drawbacks

- **No sematic search**
  - “toy” delivers 🐩 and 🔫
- **Weighting can be different between users**

Future work

- **Semantic search**
- **Keyboard extension**

One more thing …

DEMO!