Project 1: Approximate String Search for Geolocation of Tweets

Michael James, 390198

August 19, 2014

1 Introduction

Decide on which algorithms I will attempt... browse the internet for python modules.

Python modules include:

- Jellyfish - offers edit distance matching via conventional methods and phonetics
  - Documentation: `https://github.com/sunlightlabs/jellyfish`
- FuzzyWuzzy - relies on **difflib** library and clever heuristics to determine how closely strings match. Is a global edit distance library.
  - Heuristics explanation: `http://chairnerd.seatgeek.com/fuzzywuzzy-fuzzy-string-matching-in-python/`
  - Documentation: `https://github.com/seatgeek/fuzzywuzzy`
- Marisa-trie
- PyTrie
  - Documentation: `http://pythonhosted.org/PyTrie/`
- Suffix Array

2 Sample Input Sets

2.1 Sample Set A

Raw lower-case US locations.

2.2 Sample Set B

Raw lower-case US locations have had the following modifications:

- Sort - Strings based on Lexicographic Ordering
- Remove - Duplicates
- Remove - Locations that contain characters which are non-alphabetic

2.3 Sample Set C

Raw lower-case US locations have had the following modifications:

- Sort - Strings based on Lexicographic Ordering
- Remove - Duplicates
- Remove - Locations that contain characters which are non-alphabetic
- Remove - Dictionary of questionable terms, including: 'school', 'oil field', 'college', 'hospital', 'church', '(historical)', 'department', 'air field', 'center', 'clinic', etc...

3 Notes

there are a fuck ton of 'new yorks' with a bunch of other shit... wat do??