Predictive Typing

Ethan Passmore, Lane Phillips, Layne Bernardo,
Roya Rashidi, Sarah Paracha

UNIVERSITY OF ARKANSAS

Problem

- Big data can cause difficulties in searching for relevant results instantaneously

Objective

1. Develop an autocomplete program for Sorcero
2. Extend the existing functionality from the limited FAQ list to the entire corpus of documents
3. Create a user interaction that is as easy, convenient, and user-friendly as possible

Key Concepts

- Natural Language Processing aka NLP is a field of AI that extracts meaning from human language and makes decisions based on that data
- predictive typing - technology that is mainly used in search engines ex. Google Search
  - user types and technology auto-completes word or phrase

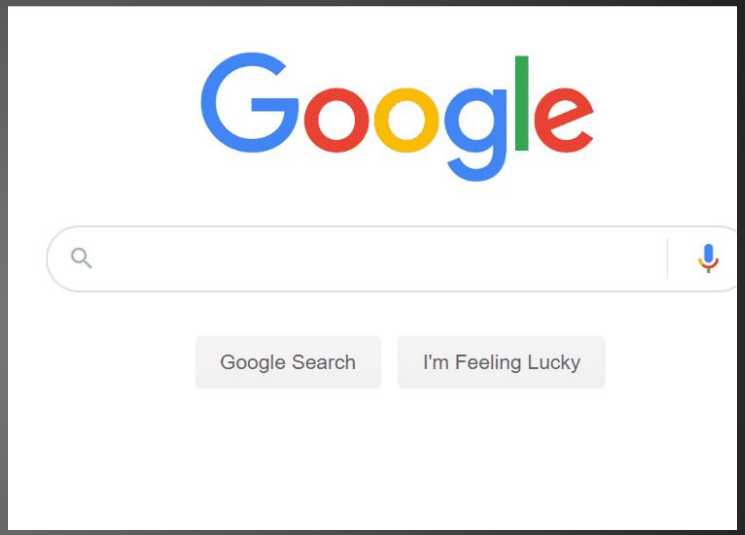
METHODS:

- “dictionary backend” - technology that is used to generate possible phrases
- adjacency matrices - used to calculate word “proximity”, or frequency of use compared to similar input in the past
- simple record - a recording of every phrase that has been searched for and suggested phrases based on frequency

Related Work

GOOGLE SEARCH ENGINE

- Google most notably has successfully developed predictive typing algorithms
- originally relied on use of adjacency matrices
- thought to have switched to machine learning algorithms



LIBRARIES

- The Embeddable Predictive Text Library Open Source Project
- Presage

Design Detail

Overview:

- N- grams: consecutive series of *n* words from a given document/corpus

  e.g The cat in the hat

  bigrams = the cat, cat in, in the, the hat
- Used in phrase completion/predictive typing application

N-gram Generation Process:

- Using python library nltk to generate n-grams
  - stored as a list data structure

Design Detail Cont'd

Buckets

- def: data type that groups objects together
  ex. when hashing, objects with same key will be placed in same "bucket"
- application: bucket for each first word of n-gram

"Alice in the" "We're all mad" "Down the rabbit"

Design Detail Cont'd

Phrase Prediction Algorithm:

1. User types one word "w" followed by a space
2. Algorithm detects this and searches for all phrases that begin with word "w"
3. As user continually types, the process is repeated until user chooses the desired suggestion produced by algorithm

Demo

Predictive Typing Demo

Associated Risks

Code Injection:

- Introducing a worm or virus onto a vulnerable system
  - Problematic in SQL, OS Commands, SMTP headers

Ensuring Loss Prevention:

- No information is modified on the corpus of documents
  - Must be flexible enough format for document corpus

Ensure Correct Phrase Completion:

- Suggestions based on metrics such as frequency, most recent
  - Higher level n grams for more accurate search results

Moving Forward

Scalability

From here we would have liked to implement more and more documents to better represent the Sorcero corpus.

Database Design:

- Nested alphabetical buckets
- Seperate database for seperate N-gram lists

Acknowledgements

This project was an industry project proposed by Sorcero, a start-up based in Washington, DC.

Sorcero is a company specializing in NLP (Natural Language Processing) Solutions.

UNIVERSITY OF ARKANSAS

References

- Sullivan, Danny. “How Google Autocomplete Works in Search.” Google, 20 Apr. 2018, https://www.blog.google/products/search/how-google-autocomplete-works-search/
- Openhub.net, Massi. “Embeddable Predictive Text Library.” Open Hub, Black Duck Software, Inc., https://www.openhub.net/p/lib378.
- Presage, https://presage.sourceforge.io/
- Sorcero, https://www.sorcero.com/about-us/

UNIVERSITY OF ARKANSAS