Formal Languages and Computability CMPT440

Project Proposal

Michael Guarino 200 66 251

2017

A classic component of a user interface to allow the user to input text which the machine interprets or are used to communicate with other uses. Often times the user spells a word incorrectly and the machine may or may not help the user offering suggestions. In a perfect world the machine should be able to correct the users’ input the known word that the user originally intended. Several publications have proposed novel solutions to efficiently solve this problem. This document focuses on solutions proposed through the use of Deterministic Finite Automata.

A seminal work that solves the auto correction problem through the use of Deterministic Finite Automata is “Fast String Correction with Levenshtein-Automata” by Schultz and Mihov. This publication suggests using a Levenshtein-Automata, a variant of a Deterministic Finite Automata. This paper suggests measuring similarity of the user’s input word with other words by using Levenshtein distance. The Levenshtein distance between “is the minimal number of insertions, deletions or substitutions that are needed to transform one word into the other” (Schultz 2). Given a controlled search lexicon the word generated to auto correct user input would be the word where the Levenshtein distance between the words does not exceed a given boundary.

Another popular work that suggests using Deterministic Finite Automata to solve the auto correction problem is “Fast approximate string matching with finite automata” by Mans Hulden. This paper suggests similar methods for determining a single word w within a larger set of words W that most closely resembles w (Hulden 1). The method proposed also uses a distance metric that works to determine similarity between the user’s input word and the intended word. The distance metric proposed by Mans is to use minimum edit distance, similar to Levenshtein distance, is a metric derived from the minimum number of editing operations: insertion, deletion, substitution.

Using known methods of incorporating Deterministic Finite Automata I would like to explore the research within this field to find an optimal solution to zero time auto correct.

Work Cited

Hulden, Mans. "Fast approximate string matching with finite automata." *Inteligencia Artificial* 3.7 (2009): n. pag. Web.

Schultz, Klaus U., and Stoyan Mihov. “Fast String Correction with Levenshtein-Automata.” *Preceedings of 3rd International Conference on Document Analysis and Recognition (1995):* n. pag. Web.