



CALIFORNIA STATE SCIENCE FAIR 2013 PROJECT SUMMARY

Name(s) Maya R. Sankar	Project Number 33229
Project Title Misspelled! Creating an Accurate Computerized Spell Correcting Algorithm	
Objectives/Goals My goal is to create a spell-corrector using Python that corrects words with greater than 90% accuracy and that minimizes the time necessary to correct each word. Abstract Methods/Materials I will need a computer with the Python programming environment, a set of misspelled words, a corresponding list of the same words but correctly spelled, and a dictionary of correctly spelled words. I will begin by creating an algorithm modeling the idea of addition-deletion-substitution, also known as edit distance or Levenshtein distance. I will also create an algorithm to test it, which returns a list of each misspelled word and its corrections. In addition, this algorithm will return how many words have each number of suggestions, how many words had one correct correction out of the total number of words, and the time the entire algorithm took to run. Based on this data, I will decide on possible improvements to the algorithm and retest it. This continues until I find the results of the algorithm satisfactory. Results The fifth algorithm I tested was the most accurate, and thus, the best. It was a combination of transposition (switching two consecutive letters), edit distance one (the one means that only one change can be made), and sound distance two (like substitution, except sounds are substituted for each other instead of letters. eg. brayd can become braid because ay and ai are both spellings of the long a). The algorithm didn't take too long, although it returned too many possible corrections for many words. This algorithm had about 90% accuracy and corrected about 321 words per second, both of which met the criteria and constraints. Conclusions/Discussion If this algorithm was made public, it could be used by small website owners. With enough online publicity, a large number of websites might benefit. In addition, my results were quite general, and this algorithm could be used to proofread any English text. This means that this algorithm could be used by any English program dealing with text, in schools, homes, or offices. In doing this experiment, I learned to program with Python, and I began to understand a little about how we read English. For example, suffixes or silent 'e's will make a sound long, double consonants tend to make a vowel short, and w and y can change the vowels to other sounds as well. Through doing this project, I feel I learned a lot, from spell correction to programming and English.	
Summary Statement I am creating a computerized algorithm to correct spelling using Python.	
Help Received My father taught me programming and helped me debug.	