

CS512 FUN Project - Spell Checking System

Shengjie Li
Rutgers University
Piscataway, NJ, USA
Email: shengjie.li@rutgers.edu

Weikang Li
Rutgers University
Piscataway, NJ, USA
Email: wl494@scarletmail.rutgers.edu

Junlin Lu
Rutgers University
Piscataway, NJ, USA
Email: jl2364@scarletmail.rutgers.edu

Abstract— This article mainly introduces the achievement of a word spell checking system. This system is a handy tool for every user who wants to check whether there are errors in their text. In the system, we use a less memory-consuming version of trie — ternary search tree for storing the dictionary. We use several methods including edit distance, similarity keys, and n-grams to improve the accuracy of word correction suggestion.

I. PROJECT DESCRIPTION

There are a lot of ways to detect spelling mistakes and correct the word. This system uses dictionary searching to detect possible correct spelling. Spelling checking plays an important part in our daily life, especially for international students whose mother language is not English, therefore developing a spell-checking system helps a lot and is extremely useful. We use a Trie to store the whole dictionary. More specifically, we use a ternary search tree, which is one kind of implementation of Trie. The correction of a word is a process of finding the most similar word. We achieve it using a combination of several methods like edit distance, similarity keys, and n-grams. Each one in our group is proficient in data structures and algorithms, so it is feasible for us to complete it within a semester. By avoiding some number of calculations and combining several different ways to calculate the difference between two words, we can somehow improve the efficiency and accuracy of this system. This is the novel part of this system. We have not encountered any stumbling block yet. But we think the improvement of the accuracy of the spelling correction would be very hard after a certain point.

The project has four stages: Gathering, Design, Infrastructure Implementation, and User Interface.

A. Stage1 - The Requirement Gathering Stage.

- The general description of this project's deliverables: This project is a spell checking system which is able to detect spelling errors in a text inputted by users.
- The three types of users (grouped by their data access/update rights):
 - 1) **Administrator**: Administrators are able to manage database, manage user privileges and roles, and check system logs.
 - 2) **Normal users**: The basic users of this system. They use the system to check whether their words are correct.
 - 3) **System maintenance personnel**: System maintenance personnel is responsible for the maintenance of the dictionary.
- The user's interaction modes: A user typically uses a keyboard to input text, and uses mouse clicks to interact with this GUI spell checking system. The user just needs to type their text in the text area, click the detect button, then several correction suggestions will be revealed on the right. The user is also able to report their mistakes corresponding to the correct word.
- The real world scenarios: Please insert the real world scenarios in here, as follows.
 - Scenario1 description: A general user, like a student who is learning English, and he wants to know if the words that he writes is correct.
 - System Data Input for Scenario1: Words or sentences
 - Input Data Types for Scenario1: String
 - System Data Output for Scenario1: Recommendations for modification of wrong words
 - Output Data Types for Scenario1: String
 - Scenario2 description: An author who wants to check if all words he used are correct
 - System Data Input for Scenario2: Sentences
 - Input Data Types for Scenario2: String
 - System Data Output for Scenario2: Sentence with highlights of wrong words
 - Output Data Types for Scenario2: String
 - Scenario3 description: A system maintenance person who wants to add some new words
 - System Data Input for Scenario3: Words
 - Input Data Types for Scenario3: String
 - System Data Output for Scenario3: A message of whether the operation was successful or not
 - Output Data Types for Scenario3: String
 - Scenario4 description: A system maintenance person who wants to delete some words or correct some wrong words
 - System Data Input for Scenario4: Words
 - Input Data Types for Scenario4: String
 - System Data Output for Scenario4: A message of whether the operation was successful or not
 - Output Data Types for Scenario4: String
 - Scenario5 description: An administrator wants to

- add a new system maintenance person
 - System Data Input for Scenario5: Username and password
 - Input Data Types for Scenario5: String
 - System Data Output for Scenario5: A message of whether the operation was successful or not
 - Output Data Types for Scenario5: String
 - Scenario6 description: An administrator wants to check the log of the system
 - System Data Input for Scenario6: Username and password
 - Input Data Types for Scenario6: String
 - System Data Output for Scenario6: The log of the system
 - Output Data Types for Scenario6: String
- Project Time line and Division of Labor.
 - Stage 1: Before Oct. 26
 - * Tasks of Shengjie Li:
 - Format designing using \LaTeX
 - Writing an abstract of the project
 - Writing the timeline and division of the project
 - * Tasks of Weikang Li:
 - Writing six real-world scenarios of this system
 - * Tasks of Junlin Lu:
 - Writing general description of this project
 - Writing 3 types of users of this system
 - Writing the user's interaction modes of this system
 - Stage 2: Before Nov. 9
 - * Tasks of Shengjie Li:
 - Writing a brief description of algorithms and data structures
 - * Tasks of Weikang Li:
 - Drawing flow diagram
 - * Tasks of Junlin Lu:
 - Writing high-level pseudo code
 - Stage 3: Before Nov. 23
 - * Tasks of Shengjie Li:
 - Implementing the system
 - * Tasks of Weikang Li:
 - Testing and evaluating the system
 - * Tasks of Junlin Lu:
 - Writing documentation
 - Stage 4: Before Dec. 7
 - * Tasks of Shengjie Li:
 - Designing a GUI
 - * Tasks of Weikang Li:
 - Writing a project report
 - * Tasks of Junlin Lu:
 - Preparing for a power point presentation

B. Stage2 - The Design Stage.

Transform the project requirements into a system flow diagram, specifying the different algorithms, data types and structures required for processing and their associated operations. The deliverables for this stage include the system flow diagram containing a graphical representation and textual descriptions of the corresponding data transformations, high level pseudo code of the overall system operation, and overall system time and space complexity.

Please insert your deliverables for Stage2 as follows:

- Short Textual Project Description. Please insert here the flow diagram textual description here together with its overall time and space complexity.
- Flow Diagram. Please insert your system Flow Diagram here.
- High Level Pseudo Code System Description. Please insert high level pseudo-code describing the major system modules as per your flow diagram.
- Algorithms and Data Structures. Please insert a brief description of each major Algorithm and its associated data structures here.
- Flow Diagram Major Constraints. Please insert here the integrity constraints:
 - Integrity Constraint. Please insert the first integrity constraint in here together with its description and justification.

Please repeat the pattern for each integrity constraint.

C. Stage3 - The Implementation Stage.

Specify the language and programming environment you used for your implementation. The deliverables for this stage include the following items:

- Sample small data snippet.
- Sample small output
- Working code
- Demo and sample findings
 - Data size: In terms of RAM size; Disk Resident?; Streaming ?;
 - List the most interesting findings in the data if it is a Data Exploration Project. For other project types consult with your project supervisor what the corresponding outcomes shall be. Concentrate on demonstrating the Usefulness and Novelty of your application.

D. Stage4 - User Interface.

Describe a User Interface (UI) to your application along with the related information that will be shown on each interface view (How users will query or navigate the data and view the query or navigation results). The emphasis should be placed on the process a user needs to follow in order to meet a particular information need in a user-friendly manner. The deliverables for this stage include the following items :

- The modes of user interaction with the data (text queries, mouse hovering, and/or mouse clicks ?).

- The error messages that will pop-up when users access and/or updates are denied
- The information messages or results that will pop-up in response to user interface events.
- The error messages in response to data range constraints violations.
- The interface mechanisms that activate different views in order to facilitate data accesses, according to users' needs.
- Each view created must be justified. Any triggers built upon those views should be explained and justified as well. At least one project view should be created with a justification for its use.

Please insert your deliverables for Stage4 as follows:

- The initial statement to activate your application with the corresponding initial UI screenshot
- Two different sample navigation user paths through the data exemplifying the different modes of interaction and the corresponding screenshots.
- The error messages popping-up when users access and/or updates are denied (along with explanations and examples):
 - The error message:
 - The error message explanation (upon which violation it takes place): Please insert the error message explanation in here.
 - The error message example according to user(s) scenario(s): Please insert the error message example in here.
- The information messages or results that pop-up in response to user interface events.
 - The information message: Please insert the error message in here.
 - The information message explanation and the corresponding event trigger
 - The error message example in response to data range constraints and the corresponding user's scenario Please insert the error message example in here.
- The interface mechanisms that activate different views.
 - The interface mechanism: Please insert the interface mechanism here.

II. PROJECT HIGHLIGHTS.

- Only working applications will be acceptable at project completion. A running demo should be presented to your project advisor at a date to be specified after the second midterm. A version of your application shall be installed in a machine to be specified later during the semester. Your final submission package will also include a final LaTeX report modeled after this document, as well as a Power Point Presentation.
- The presentation (7 to 8 minutes) should include at least the following items (The order of the slides is important):
 - 1) Title: Project Names (authors and affiliations)

- 2) Project Goal
- 3) Outline of the presentation
- 4) Description
- 5) Pictures are essential. Please include Interface snapshots exemplifying the different modes of users' interaction.
- 6) Project Stumbling Blocks
- 7) Data collection, Flow Diagram, Integrity Constraints
- 8) Sample Findings
- 9) Future Extensions
- 10) Acknowledgements
- 11) References and Resources used (libraries, languages, web resources)
- 12) Demo (3 minutes)

Please follow the sample presentation mock up that is posted on Sakai.

- By Dec 1 your group should have completed the final submission. This includes a presentation (7 to 8 minutes) to your project advisor as well as a convincing demo of your project functionalities (3 minutes): every group member should attend the demo (and presentation) indicating clearly and specifically his/her contribution to the project. This will allow us to evaluate all students in a consistent and fair manner.
- Thank you, and best of luck!