Predictive Auto Complete

Tries, hashmaps, and a practical application

Introduction

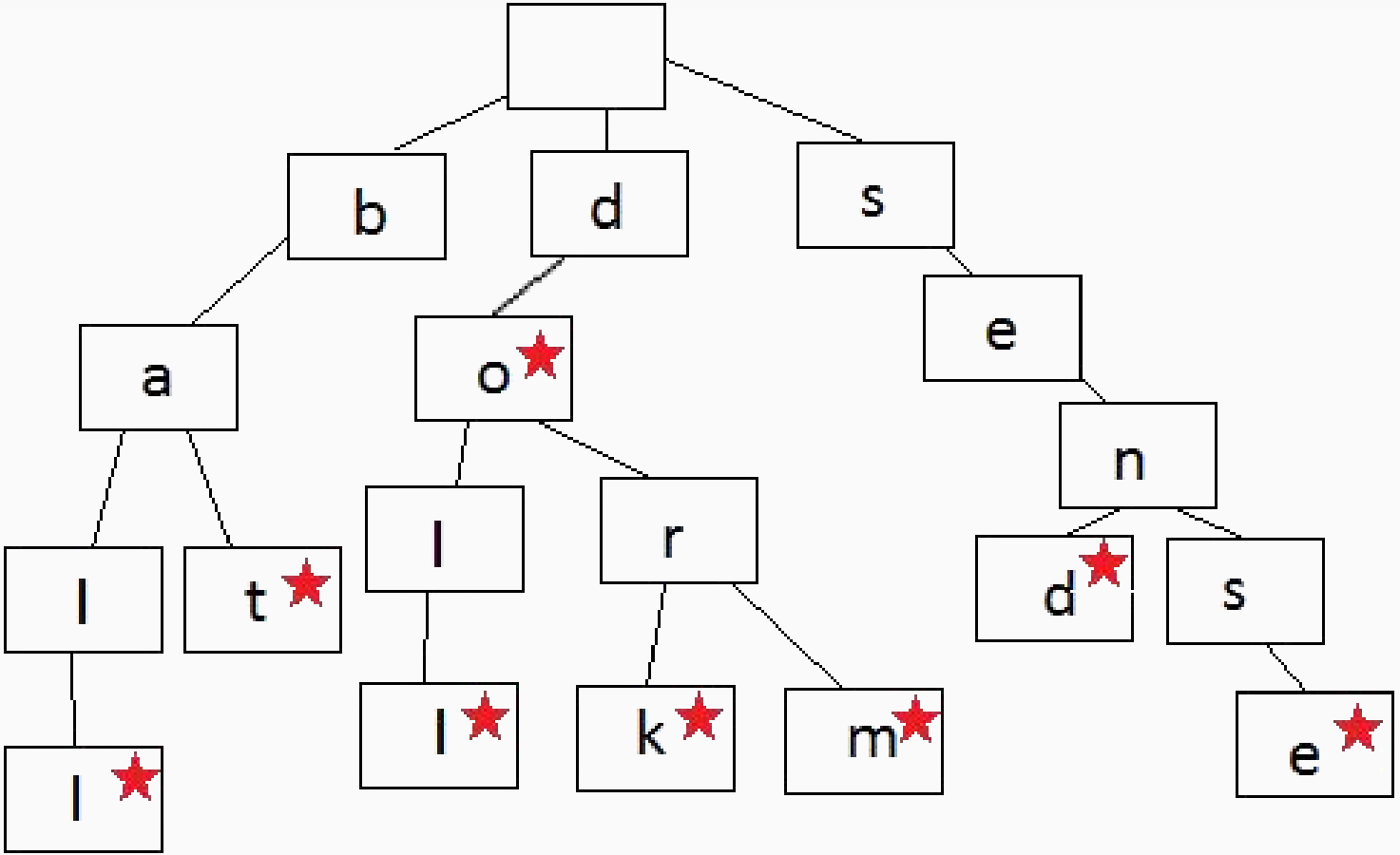
- What is a Trie and how are they used?
- Auto complete
- Moving forward

Tries

What is a Trie?

- Basically a special type of tree
- Has as many children as you want
- Typically represented as a hashmap
- Very “tall” when compared to other types of tree

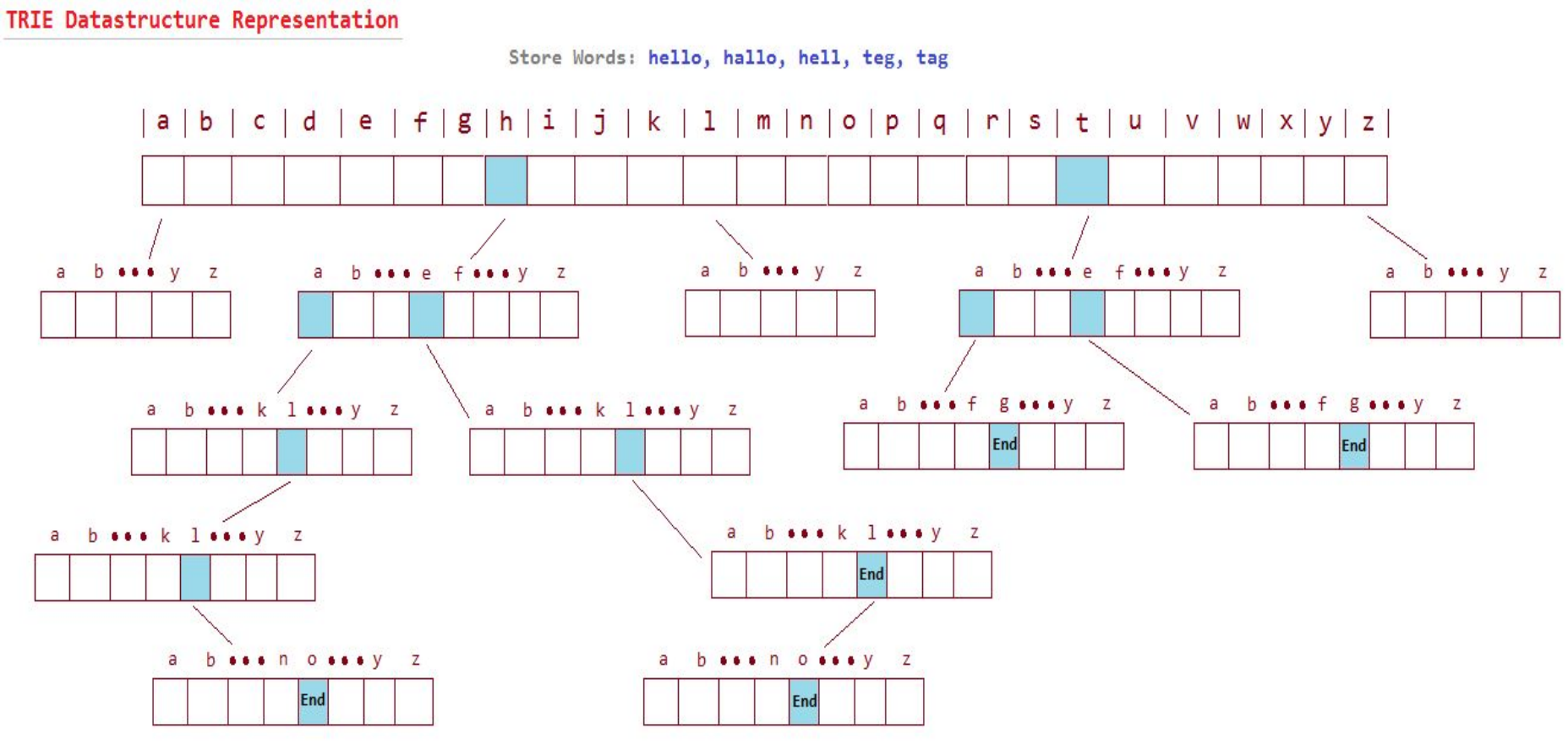
Here is what it would look like



How can they be used?

- Think about what a path represents
- A sequence of numbers, letters, symbols of any character set
- Items stored within can be seen as depth first searches

A more complete version



Application: Autocomplete

Quickly Searching Words

- Given a prefix, we can quickly see what in the tree corresponds to it
- Check for existance of children
- Generate list of children

Adding Some Heuristics

- Read in a body of text, here English novels
- Gather word frequency when inserting into trie
- Log previous word and make association table based on frequency
- Score multiplyer created when common

Getting a Suggestion

1. Run search given a prefix in real time (as keys are typed)
2. Create comprehensive list of all descendants in tree
3. Search found words for words linked with previous
4. Scale pairs based on suggestion multiplier score
5. Sort values based on final score
6. Return small subset of highest value suggestions to user

Moving Forward

Future Projects

- Interface with command line
- Interface with vim
- Multi threading support
- Capitalization
- Multiple trie support (different “languages”)

Demo

Thank You

Questions and Answers