We'd love you to complete this coding challenge for us.

Don't spend too much time on it, ~3 hours should probably be enough - we're not looking for perfection, but rather your methodology and development process.

Let me know if you have any questions or need clarification on anything.  
  
**Overview**

Develop a predictive text mechanism similar to what you see on smartphone keyboards. Given an input sequence of characters, you should output the 3 most likely words the user is trying to type, including prefix matches. For example, with the input “di” you might suggest [“did”, “dinner”, “didn’t”]. It is fine to have fewer than 3 matches where others aren’t possible, e.g. “newe” might only return [“newer”, “newest”]

**Prior data**

Assume you have a corpus of 100k email or text messages you can train on, and this is your only source of data, not an external dictionary of words. You may assume that everything is in your target language but may of course include legitimate abbreviations like “brb” or “omg”. You can find such corpuses available free online (also feel free to use other example ones - e.g. a corpus of irc/messenger chats or whatever is available).

**Bonus**

Suggest near matches where the user has made a slight mistake, e.g. given “nebe” you might suggest [“never”], or “pelp” might return [“peep”, “people”]. *Don’t worry about doing this if you have run over time.*

**Format**

Your code can work as a command line app, a webapp or even as a hosted app with a link you share back - however you prefer. The code can be sent back by email or github, your choice.