YOUR NAME: REGISTRATION #:

(10 points)

(K) F u cn rd ths (1/4)

Abbreviations are hard. We are used to thinking of standard abbreviations like lb, CA, Mr or Blvd. But in fact people make up new abbreviations all the time, if they are under time pressure (e.g. instant messaging) or if they have severe space limitations (e.g. classified ads in a printed newspaper).

One place where you find lots of abbreviations is the notes taken by the overworked people who staff call centers. They have to record what was discussed, but they don't have the time to type everything out. So you often get things that look like this, from the logs of a call center run by a major telecommunications company:

cust rcvd lttr cncrng local srvc

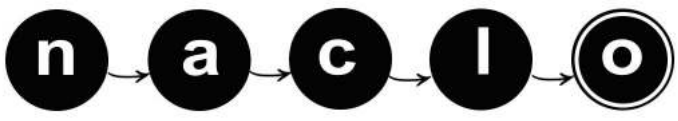
which of course is supposed to mean

customer received letter concerning local service

Let's say you are designing a computer program to try to do this kind of `normalization' automatically. You can't just have a fixed list of abbreviations: the set is pretty open ended. But what you can do is try to look at the whole corpus of data, and hope that someone somewhere has spelled out the complete words. So if for example I am looking at *rcvd lttr*, and somewhere else in the database someone has done us the favor of reporting on a different call, and used fully spelled phrase *received letter*, then we have a chance of guessing the expansion of *rcvd lttr*. That is, *rcvd* is a plausible abbreviation of *received*, *lttr* is a plausible abbreviation of *letter*, and the two occur together in the right order.

Of course, you know English, so you could have figured this out anyway. But the computer really doesn't. To the computer the problem looks as follows:

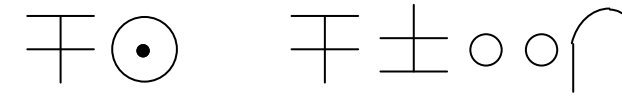
You have a bunch of abbreviated phrases (some of the words are not abbreviated, in fact), written in a bunch of symbols (remember the computer doesn't know English and to it, the strings are ultimately just a bunch of numbers anyway):

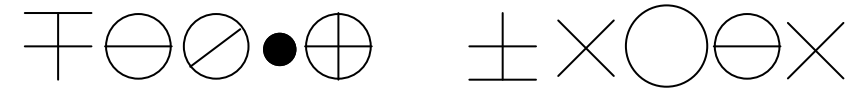


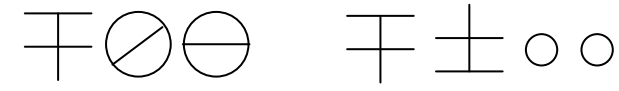
YOUR NAME: REGISTRATION #:

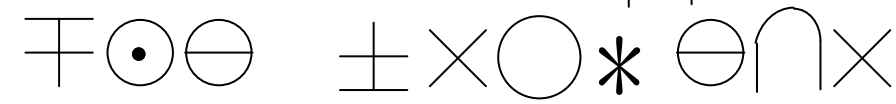
(K) F u cn rd ths (2/4)

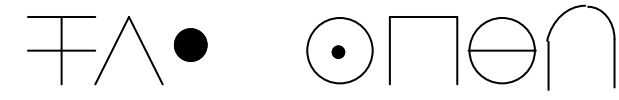
A. 

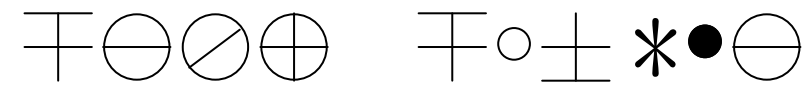
B. 

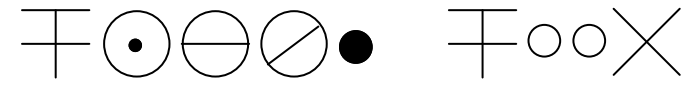
C. 

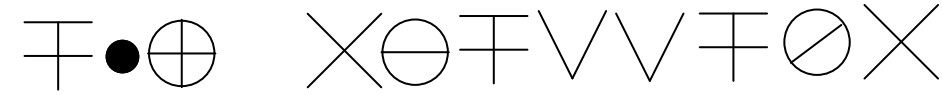
D. 

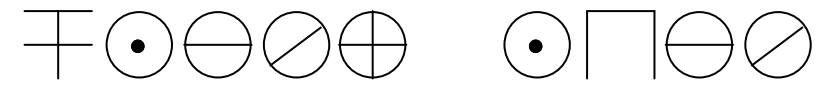
E. 

F. 

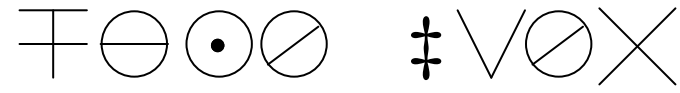
G. 

H. 

I. 

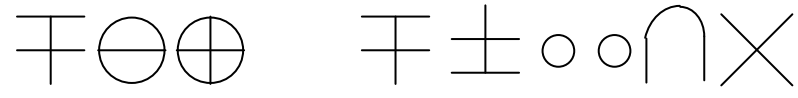
J. 

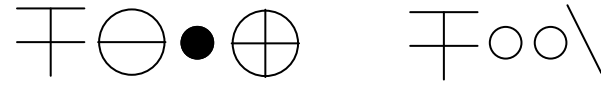
K. 

L. 

M. 

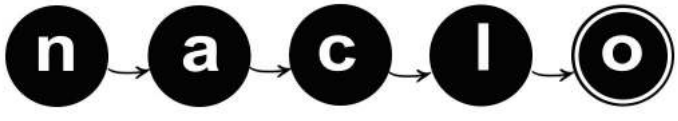
N. 

O. 

P. 

Q. 

R. 



YOUR NAME: REGISTRATION #:

(K) F u cn rd ths (3/4)

And you want to match with full phrases from elsewhere in the corpus:

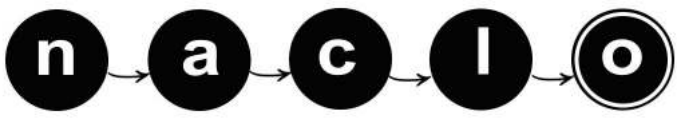
I	customer advised
II.	customer advised
III.	customer call
IV.	customer called
V.	customer called
VI.	customer called
VII.	customer called
VIII.	customer calling
IX.	customer calling
X.	customer care
XI.	customer claims
XII.	customer disconnected
XIII.	customer likes
XIV.	customer needs
XV.	customer request
XVI.	customer says
XVII.	customer understood
XVIII.	customer upset
XIX.	customer upset
XX.	customer wanted
XXI.	customer wants

There are two caveats:

1. When you are under time pressure, you make mistakes. There are actually three typos in the abbreviations—typos in that all the letters are there, but they are out of the expected order, and therefore are not strictly speaking reasonable abbreviations for the words.
2. There are three phrases that are not found in the abbreviations.

K1 (7 points). Match the encoded abbreviations from the previous page to the phrases above.

I.	IV.	VII.	X.	XIII.	XVI.	XIX.
II.	V.	VIII.	XI.	XIV.	XVII.	XX.
III.	VI.	IX.	XII.	XV.	XVIII.	XXI.



YOUR NAME: REGISTRATION #:

(K) F u cn rd ths (4/4)

K2 (3 points). Now, what phrase is abbreviated in the symbols below? Place your answer in the box at the bottom of the page.