1. Create three simple HMMs (with no more than 5 states each) over a 26-letter alphabet (the 26 lowercase letters of English). Use the nounsverbs function to test your HMMs for how well they distinguish English nouns and verbs. (You’ll need the following files from the website: nouns.txt, verbs.txt, and nounsverbs.m). Report your results and explain/discuss. Turn in an m-file for your HMMs and separate text for your results/discussion (1pp max). [Be careful: some superficially simple HMMs take a very long time to train.]

2. **non-programming option** For ASR, we represent a speech signal as a sequence of vectors. Explain what our choices are for how many vectors and their content. Explain each one and how we might choose between them. (2pp max.)

   **programming option** Write a program that will automatically generate linear HMMs. (Each state connects only to the next state in the sequence; the last state connects only back to the first. Assume that all letters in the alphabet are equiprobable.) The program should take two arguments: a number indicating the size of the alphabet and a number indicating how many states.

3. **578 only**: Why do we need vector quantization and how does it work?

**Things to remember:**

1. This is due by the *beginning* of class on Apr. 2.

2. Remember: nothing late. Don’t wait until the last minute to do this or print things or email things. . . .

3. You may certainly talk to each other about this and other assignments, but everyone must turn in their own work. (If you do talk to others, I need to see evidence that you are doing more than listening and writing down what others say.)