Chapter 7

Character and Handwriting Recognition

Optical character recognition and handwriting recognition are the task of converting images containing printed and handwritten text (respectively) into text. For cleanly printed text in high-resource languages, this is a fairly mature technology, but in other settings, this continues to be an active area of research.

A current trend is to leverage established models from automatic speech recognition. Today, we'll just highlight one approach problem, selected for its similarity to the speech model presented in the last chapter, BBN's BYBLOS system (Bazzi, Schwartz, and Makhoul, 1999).

During both training and testing, images go through several preprocessing steps.

- De-skewing: The image is rotated so that the lines are horizontal
- Line finding: The image is sliced up into horizontal lines of text. This can be done using a Hidden Markov Model, but we don't discuss this, since this is more of a vision problem than a language problem.
- Feature extraction: Each line is sliced up into vertical frames, and converted into a vector with the following features (Figure 7.1):
  - For each vertical position, how dark the frame is at that position.
  - First derivatives of the above, in both horizontal and vertical directions.
  - Local slope and correlation across a 2×2 window. (I'm not really sure how these are computed.)

  In all, there are 80 features for each frame.

Now the image has been converted into a stream of vectors, just like speech, and we can train and use a speech-recognition system on it as before. There are some differences, however:

- The language model is the same, a n-gram model over words.
- The "pronunciation" model just maps words to their spellings.
- Instead of dividing each phone into three subphones, we divide each character into 14 subcharacters. The transducer that models subcharacters can skip subcharacters, but it can't skip two in a row (Figure 7.2).

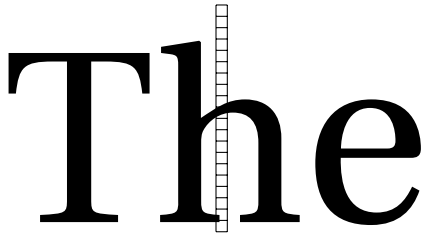

Figure 7.1: A frame and cells.

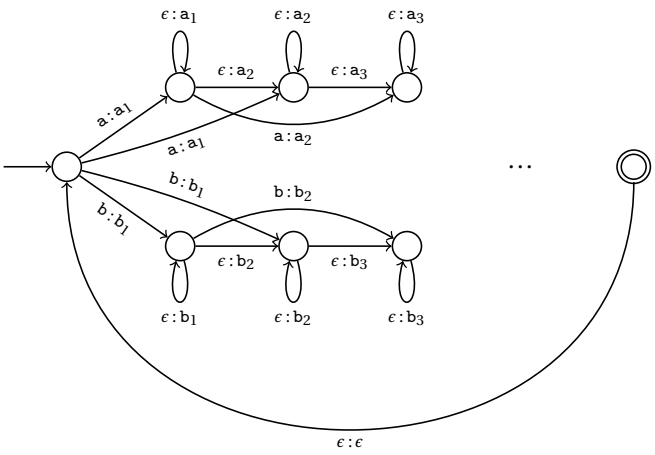

Figure 7.2: Transducer converting character sequence into frames, shown just for characters a and b. The transitionscontinue up to $\texttt{a}_{14}$ and $\texttt{b}_{14}$.

- Finally, each frame is mapped to a feature vector, just as with speech.

As before, the language model can be trained separately on as much text data as we want. The "pronunciation" model has no parameters, so it doesn't need to be trained. And the rest of the models are trained together using EM (forward-backward) just as with speech.

Bibliography

Bazzi, Issam, Richard Schwartz, and John Makhoul (1999). “An Omnifont Open-Vocabulary OCR System for English and Arabic”. In: *Trans. Pattern Analysis and Machine Intelligence* 21.6.