A Hidden Markov Model for Alphabet-Soup Word Recognition

Shaolei Feng
Nicholas R. Howe
R. Manmatha

1University of Massachusetts, Amherst
2Smith College
Motivation: Inaccessible Treasures

• Historical document collections
  – Scanned images available
  – Transcription often prohibitive ($$$)
  – Unprocessed format limits use
• Many such collections
  – Washington’s letters: 140K pages
  – Isaac Newton’s manuscripts
  – Scientific field notebooks
  – Antiquities

Goal: automated search/retrieval
Challenges of Historical Documents

• Offline handwriting OCR: success in constrained domains
  – Postal addresses, bank checks, etc.
• Historical documents are much harder
  – Few constraints
  – Fading & stains
  – Hyphenation
  – Misspellings
  – Ink bleed
  – Slant
  – Ornaments

Excerpts from the GW20 collection
APPROACH
Word Recognition & Rare Words

• Most previous work with GW data employs **full-word** recognition.
• Zipf’s Law: frequency of $i^{\text{th}}$ most common word proportional to $i^{-1}$
  $\Rightarrow$ Most words appear only rarely
  57% of vocabulary: **single** example
• Hard to learn from one example
• Even harder to learn from zero examples (OOV = out-of-vocabulary)
• Rare words may be most significant!

George K. Zipf
Character-Based Recognition: How?

• Character segmentation is hard & error-prone
• Easier to locate putative letters without segmentation
• Borrow techniques from object recognition
Alphabet Soup

• Letter detection sounds good, but how do we make whole words?
• Employed new inference model (or new twist on good old HMM)
• Remainder of talk:
  I. Letter Detection
  II. Inference Model
  III. Experimental Results
LETTER DETECTION
What are Latest Detection Results?

• Object detection
  – Use many *features*
  – Statistical methods pick indicative combinations
  – Torralba, Murphy & Freeman: *joint boosting*
Histograms of Gradient Orientations (HoG)

Original Binary

9 gradient directions

Spatial sums over regions around central point at varying resolutions
Training a Letter Detector

- Human identifies ~16 samples per character
- Samples are aligned
- Additional samples found automatically
- HoG feature vector created for each
- Joint boosting trains classifier on all characters
- Classifier looks at all points on midline of unknown word
Letter Detections

- Candidates include false positive detections
- Correct detections also
- Choice of many possible sequences
- Helpful hints:
  - Detection score
  - Letter sequence
  - Spatial separation
INFERENCE
Inference Model

- State-per-slice or state-per-detection leads to complex HMM
- If number of letters in word known, can make small HMM with one state per letter
  - We don’t know, so make multiple HMMs, one for each length
  - Try all lengths
  - Observations are detections

\[
P(O, S) = \prod_{i=1}^{m} P(s_i | s_{i-1})P(o_i | s_i)
\]
Generative Probabilities $P(o_i/s_i)$

- $P(o_i/s_i)$ taken as exponential of detection score (times some very small constant)

- More complex modeling didn’t work very well
Transition Probabilities $P(s_i|s_{i-1})$

$P(s_i|s_{i-1})$ estimate has two components:

- **Character transitions**
  - Bigram or trigram
  - Estimated on training corpus using smoothing

- **Spatial separation**
  - Mean separation assumed dependent on characters at $s_i$ and $s_{i-1}$
  - Variation assumed normal around mean
Character Separation Model

• Missing data problem for mean separations
• Model: \( S_{ij} = \frac{1}{2} (w_i + w_j) \)

• Observed separations overconstrain \( w_i \)
  – Use least squares solution
• Assume normal variation; estimate variance
Dynamic Programming

• Run Viterbi for HMM of each length
  – Reuse partial results for efficiency
• Dynamic programming computes likelihood of $i^{th}$ detection in $j^{th}$ word position ($i \geq j$)
Word Decoding

- Scores in bottom row correspond to HMM solutions for each length word
- Normalize by word length & choose highest
- Backpointers allow word decoding
EXPERIMENTS
GW20 Corpus

- 20 pages of George Washington’s letters
  - Written by multiple (30) secretaries
  - Available from UMass CIIR web site
- Cross-validation format
  - Train on 19 pages, test on 1
  - Rotate through all pages
Accuracy: Base Results

Observation: Choice of word length could be improved
- Results improve ~10% when length is given
Using Lexicon Constraints

- Some bad predictions are not words: *Octoper*
- Restricted technique: constrain prediction to top-scoring word from training lexicon
  - OOV words not handled

```
Octoper ➔ October
Forsythe ➔ forest
```
Hybrid Prediction

- Idea: Use relative scores to choose between original and restricted predictions
Results: Lexicon Restriction & Hybrid

*Best prior result
Medieval Latin

• Results for Terence’s Comedies

![Bar chart showing comparison between All Words and Characters for different linguistic measures: Bigram, Trigram, and Edwards, et al.]

- Bigram
- Trigram
- Edwards, et al.
Final Remarks

• All components of inference are important
  – Detection score
  – Character bigram/trigram
  – Physical separation

• Is HoG + joint boosting the best? Maybe...
  *Any detector may be used!*

• Try some alphabet soup for yourself!
And suddenly there it was, the perfect opening for Tommy's novel, lying at the bottom of his bowl of Alphabet Soup.
Finding Baselines

Instructions
270. Letters, Orders
Letters Orders and Instructions. October 29, 1755.

Sir,

The provisions you have ordered are in course of being sent to you, and I trust you will receive them in due time. If you find any defect, please inform me immediately.

Yours,

G.W.

April 21, 1756.

To Ensign Hubbard,

Commanding at Enrock's Fort.

You are hereby directed to proceed with the party under your command and proceed to Edwards's Fort to take charge of the provisions and supplies you have been ordered to receive. Your instructions will be issued accordingly.

Yours,

G.W.

April 21, 1756.

To Captain Harrison,

Commanding at Edwards's Fort.

It is of the utmost importance that you immediately proceed to Edwards's Fort to receive the provisions and supplies ordered for you. Please ensure that everything is delivered as per the instructions.

Yours,

G.W.
Locating Letters

• Easier to locate known letters than unknown
  – Only allow correct letter transitions
  – Use all possible detections
  – Gives position data for estimating separations
Example: Boosting

• Base rule must classify at least half of examples correctly.
Example: Boosting

• Base rule must classify at least half of examples correctly.
• Reweight data before training new rule (focus on errors)
Example: Boosting

• Base rule must classify at least half of examples correctly.
• Reweight data before training new rule (focus on errors)
• Each new rule has different viewpoint
Example: Boosting

• Base rule must classify at least half of examples correctly.
• Reweight data before training new rule (focus on errors)
• Each new rule has different viewpoint
• Combined predictions are better than single classifier alone.