Reconocimiento de Escritura

Daniel Keysers

Image Understanding and Pattern Recognition
German Research Center for Artificial Intelligence (DFKI)
Kaiserslautern, Germany

Mar-2007

UNIVERSIDAD POLITECNICA DE VALENCIA

Outline

Segmentation-based Off-line Handwriting Recognition

- Introduction
- Preprocessing
- Field Preprocessing/Normalization
- Segmentation into Characters
- Character Recognition: Neural Network
- Search for Optimal Solution
- Training

Outline

Segmentation-based Off-line Handwriting Recognition

- Introduction
- Preprocessing
- Field Preprocessing/Normalization
- Segmentation into Characters
- Character Recognition: Neural Network
- Search for Optimal Solution
- Training

Literature

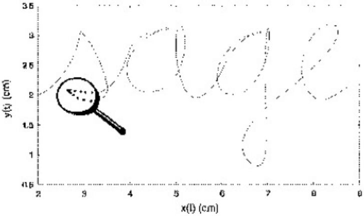
This lecture is to a large extent based on:

- T.M. Breuel:
  A system for the off-line recognition of handwritten text.
  Procs. 12th Int. Conf. Pattern Recognition, Vol. 2, pp. 129-34, 1994.
- T.M. Breuel:
  Recognition of handwritten responses on US Census forms.
  Procs IAPR Workshop Document Analysis Systems, pp. 237-64, 1995.

Previously

- isolated classification problems
  - given feature vector x, make optimal decision $D(x)$
  - usually using $P(\omega|x)$
- but that's not sufficient for a complete recognition system
- this lecture:
  overview of a complete handwriting recognition system

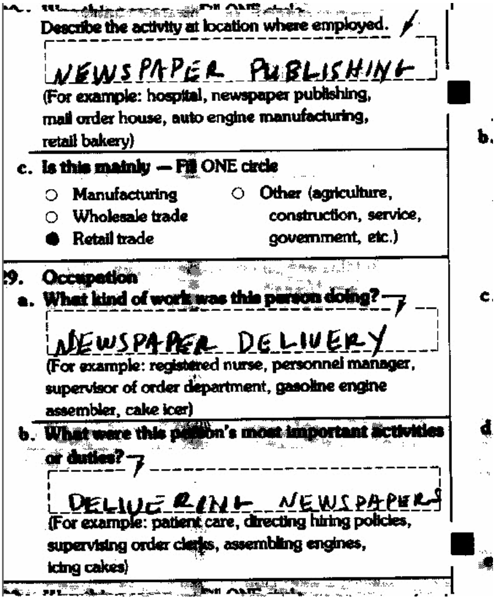
Off-Line Connected Handwriting Recognition
(also called 'static', vs. 'dynamic' on-line recognition)



Off-Line Connected Handwriting Recognition

- Scanned from Paper (no pen input)
- Applications:
  - Census
  - Post Office / Mail Delivery
  - Medical / Insurance Forms
  - Surveys
  - Paper Based Personal Assistants
  - Paper/Digital Interface (annotations, editing marks, etc.)
  - whiteboard reading

Example (US Census Task)



Describe the activity at location where employed.

NEWSPAPER PUBLISHING

(For example: hospital, newspaper publishing, mail order house, auto engine manufacturing, retail bakery)

c. Is this mainly — Fill ONE circle

- ○ Manufacturing
- ○ Wholesale trade
- ● Retail trade
- ○ Other (agriculture, construction, service, government, etc.)

29. Occupation

a. What kind of work was this person doing?

NEWSPAPER DELIVERY

(For example: registered nurse, personnel manager, supervisor of order department, gasoline engine assembler, cake icer)

b. What were this person's most important activities or duties?

DELIVERING NEWSPAPERS

(For example: patient care, directing hiring policies, supervising order clerks, assembling engines, icing cakes)

Isolated vs. Connected

Isolated Character Recognition

- handwriting is assumed to consist of isolated characters
- characters are isolated by preprinted boxes
- extract images of isolated characters by coordinates
- compute feature vector, then classify

Connected Character Recognition

- characters touch
- cannot extract a single feature vector for each character, since feature vector depends not only on location, but also extent of character
- multiple segmentations are plausible ('m' / 'rn')

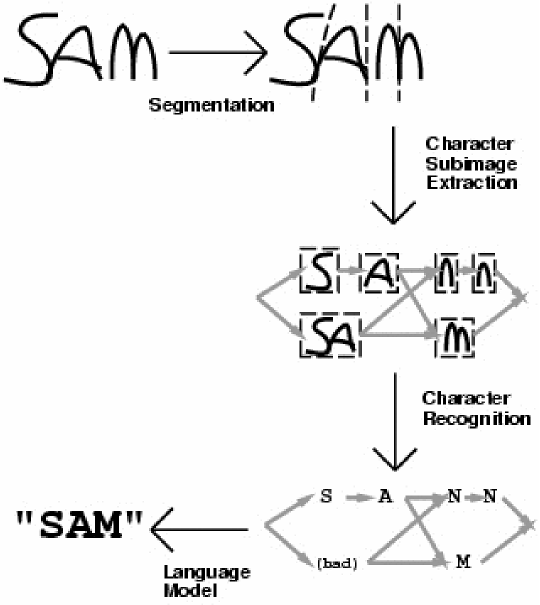
Dealing with Touching Characters

- classify whole words
  - small vocabulary e.g. bank checks
  - cannot share knowledge between words
- no separate segmentation step
  - point at a character and recognize it whether or not it is touching another character
  - LeNet (LeCun et al.)
- "dense" segmentation consider many segmentations
  - speech-like approaches/Hidden Markov Models
  - can reuse a lot of speech machinery
  - generally can't separate characters very well
  - works well for on-line methods (handhelds, tablets)
- "sparse" segmentation
  - separate segmentation step generates small number of character hypotheses
  - probably most common
  - will focus on this one

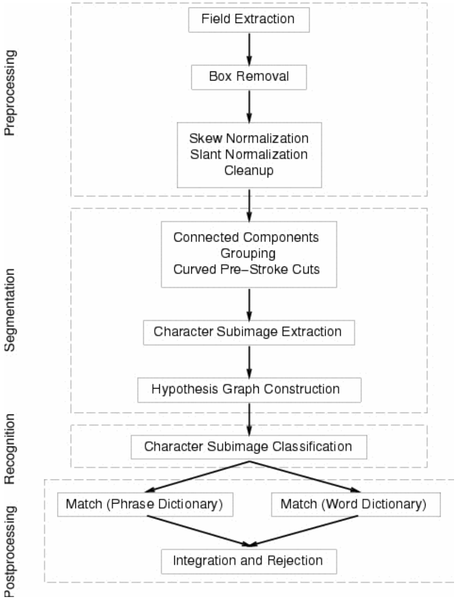
General Approach

- extract fields of handwriting from input forms
- preprocess to reduce noise, variability, background
- segment the input into multiple, overlapping segments, potentially corresponding to symbols
- compute posterior probabilities for each segment (context dependent, if necessary)
- create segmentation lattice
- integrate with language model
- compute Bayesian-optimal solution (use probabilities!)

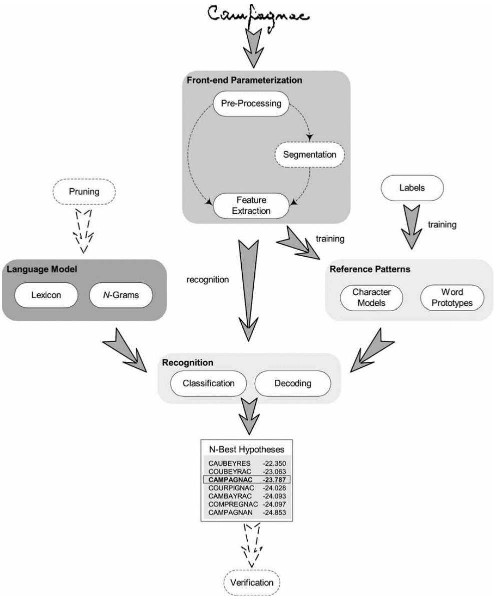
Application to Handwriting Recognition



System Overview



System Overview (Koerich' 2003)



Mathematical / Bayesian Basis

Maximize over all strings:

$$P(\text{string}|\text{image}) = P(W|x)$$

Rewrite by summing over segmentations S:

$$P(W|x) = \sum_S P(W,S|x)$$

Bayes' formula:

$$P(W,S|x) = \frac{P(x|W,S)P(W,S)}{P(x)}$$

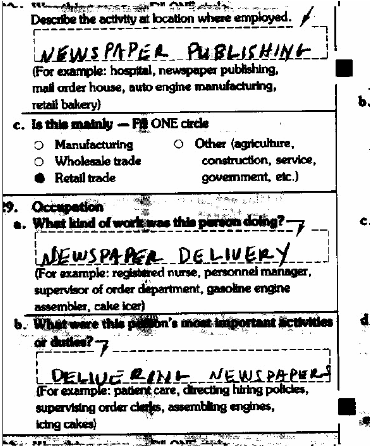
Independence Assumption:

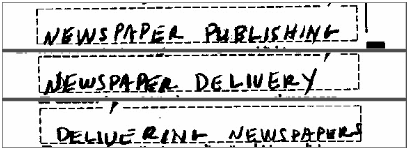
$$P(W,S|x) \approx P(W) \prod_i \frac{P(w_i|x_i)}{P(w_i)} P(S)$$

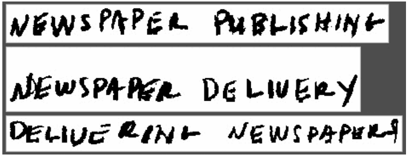
when $\text{len}(W) = \text{len}(S)$, 0 otherwise

Outline

Forms Matching / Field Extraction







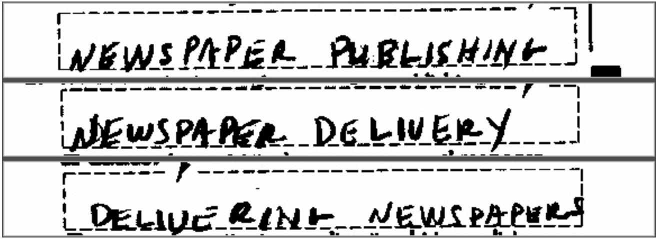
Field Extraction / Approaches

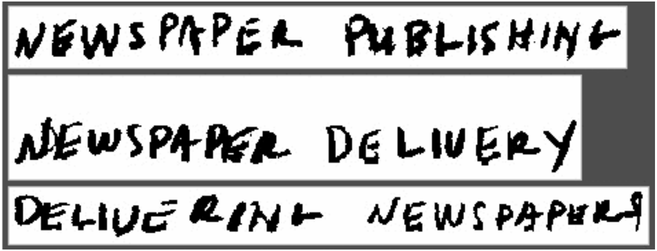
- usually model/example based approach
- model created manually
- sources of variability
  - bleed/cutting
  - non-linear scanning

- x and y projection-based matching for rough alignment
- field extraction
  - background subtraction (morphological, statistical)
    requires very precise alignment
  - generic background models
    more robust, less accurate

Background Removal

here: based on morphological model of 'horizontal dash'





Prevention: Forms Design

- give examples to user
- drop-out ink for fields
- uniform field height
- uniform text/field distances
- ‘natural’ height
- ‘proportional’ width
- per-character boxes for numbers and names

Outline

UNIVERSIDAD POLITECNICA DE VALENCIA

Field Preprocessing

- goal: remove variability prior to recognition

- skew correction
- baseline normalization
- thresholding and background removal
- noise removal
- stroke width normalization

- general approach: generate and test

Other preprocessing methods:

- slant and slope/skew correction
- smoothing
- normalization
- ...

Skew and Slant

Generate and Test

- skew: generate multiple projections, pick ‘most peaky’
- thresholding/background, noise:
  generate multiple thresholds, evaluate #connected components, stroke width, noise, etc.
- stroke width normalization:
  generate morphological erosions/dilations, evaluate #connected components, stroke width, noise, etc.

Outline

Segmentation-based Off-line Handwriting Recognition

- Introduction
- Preprocessing
- Field Preprocessing/Normalization
- Segmentation into Characters
- Character Recognition: Neural Network
- Search for Optimal Solution
- Training

Different Writing Styles

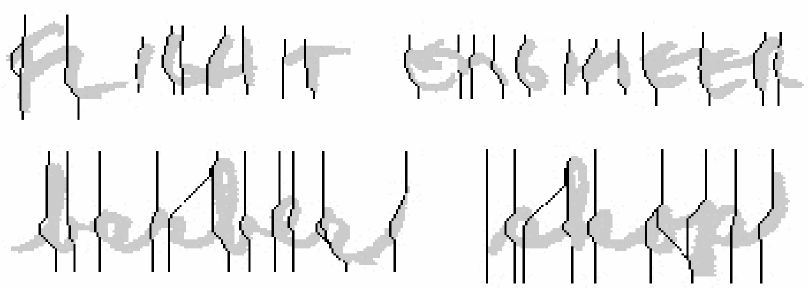
- we find four major handwriting styles
  - cursive
  - connected handprint
  - fully isolated characters
  - digits
- different styles have different segmentation needs
- cursive segmenter does not work for connected handprint
- digits benefit from custom segmenter
- connected handprint is by far the most common on forms

Segmentation is Usually a 2-Step Process

1. determine a set of potential cuts between characters or symbols
2. pick pairs of nearby potential cuts

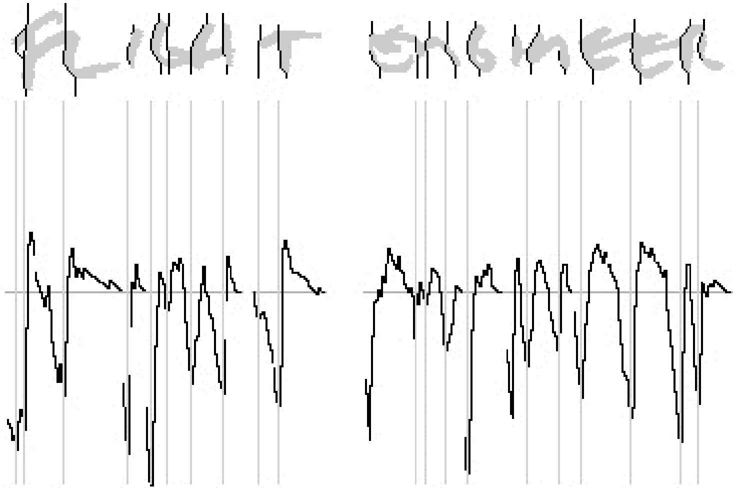
- cannot determine cuts with 100% accuracy
- failure to find a cut usually results in misrecognition
- extra cuts usually only slow down recognition
- alternative segments overlap

Cuts by Dynamic Programming

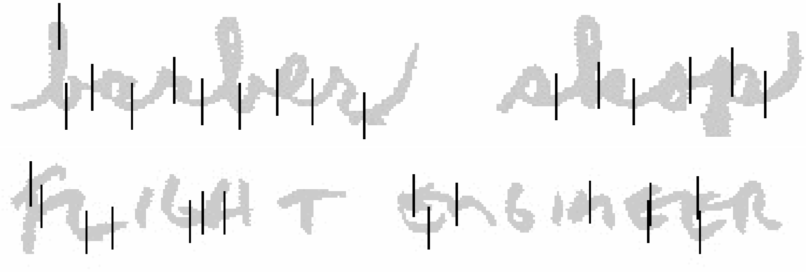


- developed for 1995 US Census
- excellent on connected handprint, acceptable on cursive
- ‘pre-stroke cuts’

Cuts by Dynamic Programming



Cuts by Valley Points



Skeletal Cuts

- compute the skeleton of the input
- find junctions
- segment the original image at the junctions
- scan segments from left to right and assemble groups into plausible characters
- works very well for digits

Learning of Segmentation

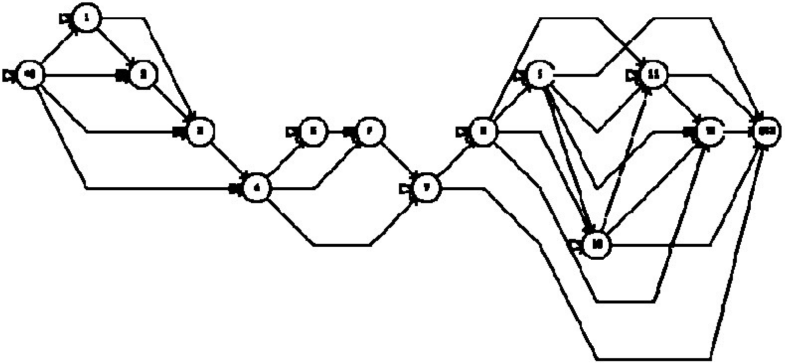
Idea: use learning here as well:

- propose a lot of cutting points by using techniques like those discussed before
- with a set of correctly segmented words, train a pattern recognition system for the decision good/bad cutting point
- only use those points for which the decision is 'good'

Segmentation/Subimage Extraction

Segmentation Graph

compactly encodes all possible segmentations of the input



Outline

Character Recognition

- goal: estimate $P\left(\text{character}\middle|\text{image, context}\right)$
- need good probability estimates
- neural networks/nonlinear regression provides good estimates
  - least-square training of regression function
  - c.f. logistic regression
  - studied in detail for backpropagation networks

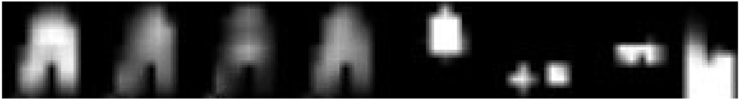
Backpropagation Networks

- model: $y_i = \sigma(\sum \beta_{ij} \sigma(\sum \alpha_{jk} x_k))$
- x: input feature vector
- y: indicator function for class membership
- trained using least square error, gradient descent
- cross-validation to avoid overtraining

- model is a universal approximator
- extensively studied (theoretically, applied)
- known to approximate posterior probabilities

- SVM may be better classifier, but $P(.|.)$?

Features



eight normalized feature maps:

- seven 10×10 unit topographic representations of feature maps corresponding to the character subimage (four gradients, holes, end-points, junctions)
- one encodes the ascent, descent, width, height, and center relative to the baseline of the character subimage using a unary code

Other Possible Features

- low-level
  - smoothed traces of the word contour
  - stroke direction distributions
  - pieces of strokes between anchor points
  - local shape templates
  - ...
- medium-level
  - edges
  - end-points
  - concavities
  - diagonal strokes
  - ...
- high-level
  - ascenders
  - descenders
  - loops, holes
  - dots
  - t-bars
  - ...

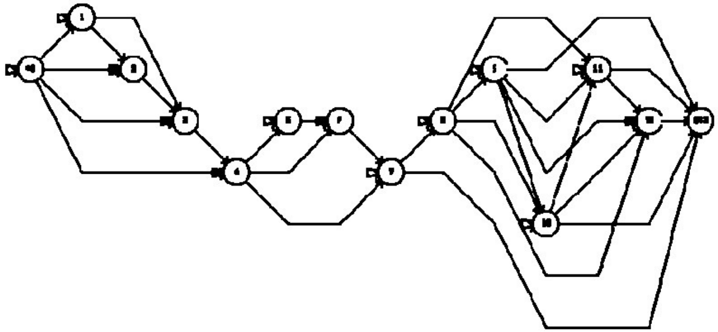
Neural Network Notes

- good isolated character performance (but not best)
- best classifier often not best overall
- good probability estimates more important than improving classification

- coarticulation can be formulated as prediction problem predict the previous character given the image of the current character
- surprisingly: few or no coarticulatory/context effects

Outline

Best Match, No Language Model



$$\max_{W,S} P(W) \prod_i \frac{P(w_i|x_i)}{P(w_i)} P(S) \stackrel{?}{=} \max_{W,S} c \prod_i \frac{P(w_i|x_i)}{P(w_i)} P(S)$$

- roughly: pick locally best choice everywhere
- does not work well

Language Model

- $P(W)$ =const: all character sequences are equally likely
- bad assumption: most strings are nonsense ('born' vs.'bom')
- must incorporate priors $P(W)$
- $P(W)$ can take many forms: dictionary (trie), n-graph, n-gram, decision tree, ...
- given $P(W)$, must optimize

$$max_{W,S} P(W) \prod_i \frac{P(w_i|x_i)}{P(w_i)} P(S)$$

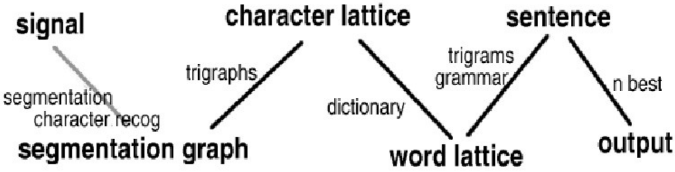
- hand-code Viterbi algorithm (dynamic programming) for many $P(W)$
- but there is a better way...

Probabilistic Finite State Transducers (Riley et al.)

- probabilistic, finite state string mapping
- operations: composition, minimization, union, difference, closure, pruning, *n*-best, alignment
- lazy or eager operations
- inner loops can be highly optimized and generic
- enables easy segmentation
- bridges multiple levels of representation (input/output alphabets can be different, e.g. char to word)
- build and precompile language models from components

- segmentation graph can be represented as PFST
- *n*-graphs, *n*-grams, dictionaries (tries), approximations of grammars
- extensively used in natural langugage processing

Finite State Tranducers can be Composed



$$\text{recognizer} = \text{minimize}(\text{grammar} \circ \text{dictionary} \circ \text{trigraphs})$$
$$\text{solution} = \text{nbest}(\text{recognizer} \circ \text{segmentation graph})$$

Back-Off

- language models: more constraints, less coverage
- need to ‘back off’ from more constrained language model to less constrained
- dictionary of phrases concatenation of words n-graphs
- backoff can itself be represented in PFST framework
- alternatively: stacked recognizers final recognizer chooses among different language models

Example: US Census Language Models

- *n*-graphs simple, insufficient
- unconstrained sequences of words reasonable performance
- dictionary of phrases best performance
- finite state approximation to arithmetic contraints (age/d.o.b.) essential for high performance numerical recognition
- other sources of probabilistic information
  - age vs. date of birth
  - first name vs. male/female
  - last name vs. race
  - ...

Outline

Segmentation-based Off-line Handwriting Recognition

- Introduction
- Preprocessing
- Field Preprocessing/Normalization
- Segmentation into Characters
- Character Recognition: Neural Network
- Search for Optimal Solution
- Training

Training Problem

- problem: training data is transcribed, not segmented/aligned

- hidden variables: segmentation/alignment information

- Expectation Maximization Algorithm
- Similar to HMM Training
- EM recovers 'hidden variables'

Training Algorithm

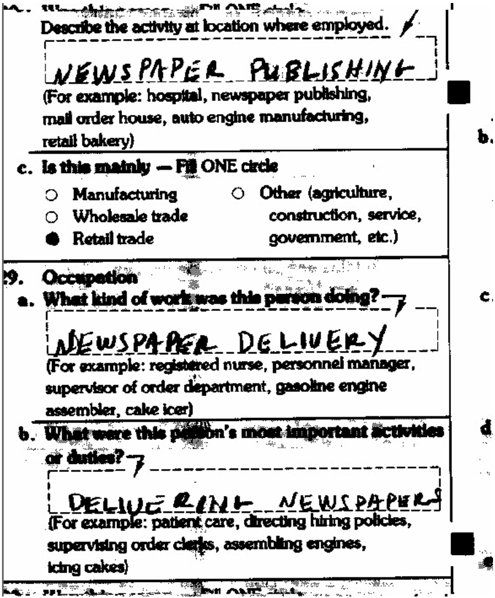
1. start with the neural network weights from isolated character recognizer
2. pick a training example with transcription
3. run the recognizer with language model consisting only of transcription
4. establishes segmentation, correspondence between transcription and input
5. retrain character recognizer with segments
6. repeat from step 2

Example: The US Census Task

- unconstrained, unsegmented handwriting recognition
- large, unmotivated writer population
- open language model
- goal: reduce cost for manual transcriptions (50% machine)

- US Census created database, ran performance evaluations
- this system placed first and second, chosen for 1995 Census

Example (US Census Task)



Describe the activity at location where employed.

NEWSPAPER PUBLISHING

(For example: hospital, newspaper publishing, mail order house, auto engine manufacturing, retail bakery)

b.

c. Is this mainly — Fill ONE circle

- ○ Manufacturing
- ○ Wholesale trade
- ● Retail trade
- ○ Other (agriculture, construction, service, government, etc.)

29. Occupation

a. What kind of work was this person doing?

NEWSPAPER DELIVERY

(For example: registered nurse, personnel manager, supervisor of order department, gasoline engine assembler, cake icer)

c.

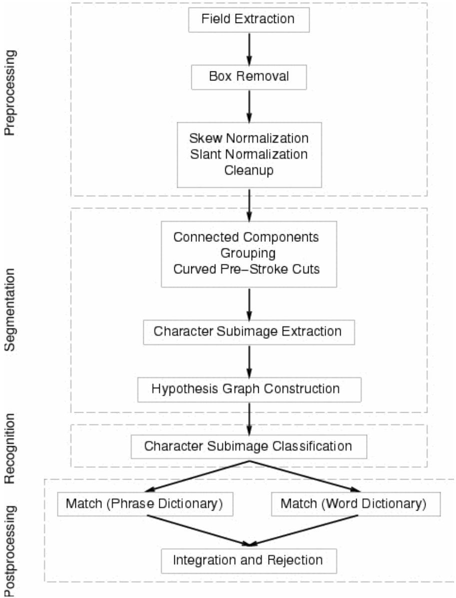
b. What were this person's most important activities or duties?

DELIVERING NEWSPAPERS

(For example: patient care, directing hiring policies, supervising order clerks, assembling engines, icing cakes)

d.

System Overview



Results (1995)

	N	err[%]
area code	1336	0.00
first name/female	4490	3.39
first name/male	3880	2.84
last name	9811	8.26
four digits (phone)	1339	0.45
month	4209	0.52

- processing times (Linux, 600MHz PIII): under 500ms/field
- more recent applications: postal addresses, ZIP codes

Results (1995)

Error rates evaluated by NIST at 50% rejection:

Participant	Field Error Rate (%)
ERIM	6.3
IDIAP	8.6
CGK	18.9
CEDAR	25.2
IBM	40.5
AT&T	43.8
NIST	52.5
Hughes	55.6
U. Bologna	57.9
MCC	74.1

This system = IDIAP

IDIAP performed better than ERIM at higher rejection rates.

Error Analysis

Error Rate	Cause
2.0%	The cost assigned to an incorrect interpretation of a character subimage is too low or the cost assigned to a correct interpretation of a character subimage is too high.
1.6%	The correct response was not in the dictionary; the system returned a close approximation from the dictionary.
0.8%	The handwritten string went past the right end of the input box and was truncated during pre-processing; the system returned a close approximation to the truncated string.
0.5%	The handwritten input consisted of two lines, one of which was (nearly) eliminated during pre-processing; the system returned a close approximation to the remaining line of text.
0.3%	The system returned a correct interpretation of the input that differed from the actual transcription.
0.3%	Spurious markings in the image were transcribed as a character (e.g., a trailing smudge as a plural S).
0.5%	Other.

Conclusions

- speech, handwriting are very similar problems
- rigorous probabilistic approach
- well-understood approximations
- standard toolbox of algorithms/approaches:
  - Mathematical Morphology
  - Generate-and-Test
  - Nonlinear Regression/Neural Networks for Probabilities
  - Probabilistic Finite State Transducers
  - probabilistic finite state models are powerful

Future Work, Open Problems

- better document segmentation: annotations, forms, etc.
- better automatic forms interpretation
- adaptive/learning segmentation methods
- more principled forms removal, figure/ground
- incorporation of world knowledge, commonsense reasoning
- more realistic assessments of human performance
- better integration with word shape approaches
- improvements on non-standard styles, highly degraded input
- robustness in small training set problems
- automatic adaptation, training