4. Character Segmentation, Language Models and Beam Search

The heart of Tesseract

Ray Smith, Google Inc.
Approaches to Segmentation

● Segment first using only geometry.

● Maximally chop, then combine with a beam search. (Over-segmentation.)

● Sliding window to "avoid" segmentation altogether.

● Tesseract: Chop only as needed, then combine as needed.
Over-Segmentation

- Aim is to maximize recall of chops with the compromise of reduction in precision.
Segmentation Graph

- Segmentation possibilities and classifier results form a directed graph
Searching the Segmentation Graph

higher
Searching the Segmentation Graph

higher

}uglier
Integration of Language Models (General Methods)

- Implement Language Model as Finite State Machine.
- Search Language Model and Segmentation Graph in parallel.
- Combine “probabilities” in some sensible way.
- Hidden Markov Model methods are good example.
Segmentation Free = Extreme Over-Segmentation

- Slide over the word/textline with a classifier/HMM.
- Beam search + shape model probs + language model probs solves the segmentation internally.
- Really just an extreme form of over-segmentation.
Tesseract Segmentation Approach based on observations:

- Initial segmentation is often correct or close.
- Classifier generally doesn’t like incorrectly segmented text.
- Over-segmentation often leads to poor results., eg m->iii
Tesseract Segmentation Approach

Classify Initial Segmentation
Search Word: OK? Yes => Done
while any Bad Blob has any Chops available
   Chop and classify pieces of Worst Choppable Blob
      Search Word: OK? Yes => Done
while any fixable “Pain point”
   Associate adjacent blobs and classify
      Search Word: OK? Yes => Done
Types of Pain Point

- Initial: Join each adjacent pair
- Ambiguity: Eg m/rn
- Path: Neighbors of blobs in the current best path
Ratings MATRIX = Segmentation Graph

Each entry holds a BLOB_CHOICE_LIST providing classifier choices with rating and certainty.
Evaluation of a WORD_CHOICE (no params-model)

Word Rating = word_factor \sum \text{blob.choice->rating()}
word_factor = \begin{array}{c|c|c}
\hline 
\text{Condition} & \text{base word_factor} & \text{Add-ons} \\
\hline 
\text{Frequent dawg word} & 1.0 & \text{Inconsistent case +0.1} \\
\text{Other dawg word} & 1.1 & \text{Inconsistent case +0.1} \\
\text{Non-dawg word} & 1.25 +0.01 \text{ for each char over 3.} & \text{Inconsistent case +0.1} \\
& & \text{Inconsistent punc +0.2} \\
& & \text{Inconsistent chartype +0.3} \\
& & \text{Inconsistent script +0.5} \\
& & \text{Inconsistent char spacing +0.01} \\
& & \text{All except script +0.01 for each additional occurrence.} \\
\hline 
\end{array}
Evaluation of a WORD_CHOICE (with params-model)

Word Rating = word_factor \sum \text{outline length}

word_factor = weighted sum of word features:

- mean blob rating
- num inconsistent spaces
- num inconsistent char type
- num x-height inconsistencies
- num case inconsistencies
- word length (in type categories)
Tesseract Word Recognizer

Character Chopper

Done?

Segmentation Search

Done?

Try Again On Pass 2

Adapt to Word

Character Classifier
- Static
- Adaptive

Language Model
- Dictionary
- Char n-grams
- Number/Punctuation Parser
Example of Chopping (unlv/mag.3B/2/8022_028.3B.tif Col 2, line 6, word 1)

<table>
<thead>
<tr>
<th>Word</th>
<th>Distance</th>
<th>Worst blob</th>
</tr>
</thead>
<tbody>
<tr>
<td>Momm</td>
<td>212.2</td>
<td>7.7</td>
</tr>
<tr>
<td>Mommn</td>
<td>186.3</td>
<td>8.3</td>
</tr>
<tr>
<td>Momtfln</td>
<td>178.0</td>
<td>9.2</td>
</tr>
<tr>
<td>Momtain</td>
<td>124.9</td>
<td>5.3</td>
</tr>
<tr>
<td>Mounm</td>
<td>184.0</td>
<td>7.7</td>
</tr>
<tr>
<td>Mountain</td>
<td>80.6</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACCEPT!</td>
</tr>
</tbody>
</table>
Example of Combining (unlv/doe3.3B/4/2214_007.3B.tif, col 2, line 8, word 2)

<table>
<thead>
<tr>
<th>Word</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>lilllit-</td>
<td>77.58</td>
</tr>
<tr>
<td>limit</td>
<td>57.1</td>
</tr>
<tr>
<td>Emit</td>
<td>89.7</td>
</tr>
<tr>
<td>Unfit</td>
<td>95.4</td>
</tr>
<tr>
<td>Hulk</td>
<td>122.7</td>
</tr>
<tr>
<td>Bulk</td>
<td>136.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Word</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>HUM</td>
<td>120.6</td>
</tr>
<tr>
<td>MUM</td>
<td>127.0</td>
</tr>
<tr>
<td>BUM</td>
<td>134.7</td>
</tr>
<tr>
<td>1mm</td>
<td>112.8</td>
</tr>
<tr>
<td>Milk</td>
<td>147.2</td>
</tr>
<tr>
<td>huh</td>
<td>137.1</td>
</tr>
<tr>
<td>fink</td>
<td>140.7</td>
</tr>
<tr>
<td>Emu</td>
<td>129.7</td>
</tr>
<tr>
<td>BMW</td>
<td>140.3</td>
</tr>
</tbody>
</table>
Thanks for Listening!

Questions?