AN OCR MODEL FOR NORTHERN HAIDA

TRAINING & QUALITY ASSESSMENT
THE OVERALL IDEA

John R. Swanton’s pre-phonemic transcriptions
> 110,000 words
> 1,000,000 characters
• scanning
• image pre-processing

first NA indigenous language model
• model training
• comparing different approaches
• quality assessment

first large-scale electronic corpus of Haida
• post-processing of OCR output
• quality assessment
• formatting
WHY SWANTON?

• good quality prints of large amounts of natural text are essential for an OCR model
  • large amounts of text: rare in any NA indigenous language
  • good quality prints of large amounts of text: even rarer
• Swanton texts are both!
  • additionally: not a translation of a Western text or trade documentation
  • preserve culture & identity through traditional stories
Wa'gïen wâ'daalekû na-i èAi À'nâ ña l'isâ'wane. Lâ'we wë À'nâ ña l'îsdaî'ane. Gînaga'-ì lâ'ña qoanyû'anane Aî gam hawî'dan wë l'îsîtcla'idâ'ânane. Wa'gïen si'ñias lû wë lâ'ña îsîtclaiañi. Wa'gïen L! qLa'slas gièn wa'dalekû î'sîn l'la'gân l'îs'ai'yaiyan. Wa'gïen na-i èAi L! wa'luwan lâ'ña idja'ne. Lnagà'-ì è'skî'han la'gân À'a'ngulgañan. Wa'gïen L! îsîtclas lû gînaga'-ì qLâ'tguì À'nâ ña l'îsdaî'an. Wa'gïen qà'gâlân èsa l'gusâ'lan. 'Dì qà'gâlân nana'ñ àt di dala'ñ dâ'îngan. Aldji'âlû digâ ga qLa'ngâñ. Dì dala'ñ qoñado'iyas lû gam dì ì'l'ade'selâ'ñgân. Dala'ñ-o qû'nañ dì ì'l'ade'seldan. Aldji'âlû gam gînaga'-ì t'ëdj qOgâ'ñ l istâ'sâ'ñasàñ. Dì xë'nañas klìa'l hin dala'ñ ñan î wa'asàñ, 'hin qà'gâlân l' sudai'an. 'Ada'lu dala'nggû l gië'slasga. Së'îguà gï'daga-i À'nâ ña l k!wai'indaasañ. Dala'ñ lî ìsî'ñ êkû l'k!wai'indaasañ. Dala'ñ lî ëtsî lû èdëf dala'nggû l gië'slaasañ.
**OCR SYSTEMS**

**pattern recognition:** matching a physical shape on a page to an internally stored prototype

**OCR A & OCR B**

**Farrington 7b**
“[T]he graphic coding system we use to represent language in visual form is complex but well-suited to human abilities. Because it is woven so deeply into the fabric of our culture, it is unlikely to undergo rapid change. We shall not learn to read bar codes. We must therefore depend on OCR to eradicate computer illiteracy.”

- Nagy et al. 2000, p.9
WE NEED AN ENGINE!

- should be accessible for other researchers in the future
- (small or) no licensing costs & copyrights
- must have ability to be trained for various languages
- Unicode!
THE TESSERACT ENGINE

- GNU open source – Apache License
  - no licensing costs, no copyright
  - engine source code can be adapted & customized (if one is so inclined)
- exhaustive training system
- handles Unicode characters well
- the most accurate open source engine available Boschetti et al. 2009

TRAINING THE OCR MODELS

THE IMAGE GENERATION APPROACH

• recommended in official training handbook *Training Tesseract 2*
• requires hand-typing of *ground truth* to generate training images

THE SOURCE TEXT APPROACH

• used in Gutenberg project alongside image generation approach
• requires existing OCR model with similar character set
• requires hand-validating files

*to this day, there is no agreement as to which approach works better* Beusekom et al. 2008
THE IMAGE GENERATION APPROACH

• hand-type at least one page of text 100% correctly
  • aim at 20 examples for frequent characters, 5 for rare ones
  • in our project: 12 pages (2/person, because minions)
• use tesseract (or a graphical frontend like jTess) to generate tif image files from ground truth
• tesseract will then “learn” the form-to-character mappings from the image/text file pairings
## Unicode Character Encoding

<table>
<thead>
<tr>
<th>Print Character</th>
<th>Unicode Specification</th>
<th>Unicode Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>’</td>
<td>Modifier Letter Acute Accent</td>
<td>U+00C2</td>
</tr>
<tr>
<td>“</td>
<td>Left Double Quotation Mark</td>
<td>U+201C</td>
</tr>
<tr>
<td>”</td>
<td>Right Double Quotation Mark</td>
<td>U+201D</td>
</tr>
<tr>
<td>‘</td>
<td>Left Single Quotation Mark</td>
<td>U+2018</td>
</tr>
<tr>
<td>’</td>
<td>Right Single Quotation Mark</td>
<td>U+2019</td>
</tr>
<tr>
<td>Print Character</td>
<td>Unicode Specification</td>
<td>Unicode Number</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Â</td>
<td>Latin Capital Letter A with Circumflex</td>
<td>U+00C2</td>
</tr>
<tr>
<td>â</td>
<td>Latin Small Letter a with Circumflex</td>
<td>U+00E2</td>
</tr>
<tr>
<td>Ā</td>
<td>Latin Capital Letter A with Diaeresis</td>
<td>U+00C4</td>
</tr>
<tr>
<td>ä</td>
<td>Latin Small Letter a with Diaeresis</td>
<td>U+00E4</td>
</tr>
<tr>
<td>Å</td>
<td>Latin Capital Letter A with Macron</td>
<td>U+0100</td>
</tr>
<tr>
<td>ä</td>
<td>Latin Small Letter a with Macron</td>
<td>U+0101</td>
</tr>
<tr>
<td>A</td>
<td>Latin Letter Small Capital A</td>
<td>U+1D00</td>
</tr>
<tr>
<td>É</td>
<td>Latin Capital Letter E with Circumflex</td>
<td>U+00CA</td>
</tr>
<tr>
<td>ê</td>
<td>Latin Small Letter e with Circumflex</td>
<td>U+00EA</td>
</tr>
<tr>
<td>É</td>
<td>Latin Capital Letter E with Macron</td>
<td>U+0112</td>
</tr>
<tr>
<td>ë</td>
<td>Latin Small Letter e with Macron</td>
<td>U+0113</td>
</tr>
<tr>
<td>Í</td>
<td>Latin Capital Letter I with Macron</td>
<td>U+012A</td>
</tr>
<tr>
<td>í</td>
<td>Latin Small Letter i with Macron</td>
<td>U+012B</td>
</tr>
<tr>
<td>Í</td>
<td>Latin Capital Letter I with Circumflex</td>
<td>U+00CE</td>
</tr>
<tr>
<td>ï</td>
<td>Latin Small Letter i with Circumflex</td>
<td>U+00EF</td>
</tr>
<tr>
<td>Ø</td>
<td>Latin Capital Letter O with Macron</td>
<td>U+016B</td>
</tr>
<tr>
<td>ŏ</td>
<td>Latin Small Letter o with Macron</td>
<td>U+00D1</td>
</tr>
<tr>
<td>Ù</td>
<td>Latin Capital Letter U with Macron</td>
<td>U+00EF1</td>
</tr>
<tr>
<td>ū</td>
<td>Latin Small Letter u with Macron</td>
<td></td>
</tr>
</tbody>
</table>
## Unicode Character Encoding

<table>
<thead>
<tr>
<th>Print Character</th>
<th>Latin Base Letter</th>
<th>Decomposed Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>Capital Letter G</td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>Small Letter g</td>
<td></td>
</tr>
<tr>
<td>l</td>
<td>Letter Small Capital L</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>Capital Letter X</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>Small Letter x</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Specification</th>
<th>Number</th>
<th>Specification</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capital Letter G</td>
<td>U+0047</td>
<td>Combining Dot Below</td>
<td>U+0323</td>
</tr>
<tr>
<td>Small Letter g</td>
<td>U+0067</td>
<td>Combining Dot Below</td>
<td>U+0323</td>
</tr>
<tr>
<td>Letter Small Capital L</td>
<td>U+029F</td>
<td>Combining Dot Below</td>
<td>U+0323</td>
</tr>
<tr>
<td>Capital Letter X</td>
<td>U+0058</td>
<td>Combining Dot Below</td>
<td>U+0323</td>
</tr>
<tr>
<td>Small Letter x</td>
<td>U+0078</td>
<td>Combining Dot Below</td>
<td>U+0323</td>
</tr>
</tbody>
</table>
Font issues

- requirement: same font as in textual source for image generation.

- What if the font is so old that there is no modern-day equivalent?
- … or what if there is a modern-day equivalent, but it has been developed in the meantime and looks different?
  - Mediaeval (original print font) vs. Dutch Mediaeval (digital font) Letterproef
  - closest match: Cambria (Thanks to Darren Flavelle for his research!)
FONT ISSUES

(a) Mediæval a
(b) Cambria a
(c) Mediæval e
(d) Cambria e
THE SOURCE TEXT APPROACH

• instead of generating artificial training images, we make use of the source text directly
  • no font issues
  • hand-typing is required either way

1. find existing OCR model with similar character set
  • Portuguese

2. run this set on a few pages, hand-validate the resulting box files
  • introduce missing characters manually
  • again: either tesseract command line interface or jTess
Wa'giën tci'liñagandanan i'sin siñidan. Wa'kli'lta'gə l' ya'yələhə 'ańgañan. Wa'giën l' taguna's la l! i'nasins kli'ət l' ya'yələhə 'ańgañan. Li'sluan la l! su'dag l' ga'oswānəñan. Wa'lu l' la'da diyiñidan. Wa'giën Inag
We are expecting:
1) The source text approach to result in more accurate models overall;
2) Those models based on more pages to perform better than those based on fewer pages.
MODEL COMPARISON & QUALITY ASSESSMENT

• first idea: compare character sets for completeness.
  • this didn’t really lead anywhere.

• required: a character-level comparison of OCR output and ground truth.
  • string alignment! (thanks Dustin!)
## STRING ALIGNMENT

### Correct string

```
SAND87-0112*Unlimited_Release*Printed_July_1987*
```

### OCR-generated string

<table>
<thead>
<tr>
<th>Confusion</th>
<th>Insertions</th>
<th>Deletions</th>
<th>Substitutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1V</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>ε</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>rн</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>.</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>v</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>%B</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>2</strong></td>
<td><strong>3</strong></td>
<td><strong>7</strong></td>
</tr>
</tbody>
</table>

\[ n = 48 \]

\[ E = 12 \]

*character accuracy = 75%*

---

Rice 1996, Fig. 3, p. 20
ANALYSIS TOOLS: ISRI TOOLKIT

- once de facto standardized analysis toolkit (*Annual Tests of OCR Accuracy*)
- comes with many tools (character accuracy, word accuracy, confusion matrices...)
- can generate separate reports & aggregate reports, easily “loopable”
ANALYSIS TOOLS: ISRI TOOLKIT

• once de facto standardized analysis toolkit (*Annual Tests of OCR Accuracy*)
• comes with many tools (character accuracy, word accuracy, confusion matrices…)
• can generate separate reports & aggregate reports, easily “loopable”
• can absolutely not cope with Unicode input.
  • replacement ops (extended ASCII, escaped ASCII, character sequences…) all lead nowhere
  • future goal: porting the ISRI toolkit to a more 2015-friendly version
ANALYSIS TOOLS: OCR STAR

• based on swalign python package & code skeleton
• produces accuracy scores & visual alignments for each line in each file
• does not currently produce confusion matrices
• highly customizable, loopable, & handles Unicode well
ANALYSIS TOOLS: OCR STAR - CODE

code skeleton provided with swalign package (all of it)

current version of algorithm implementation (plus ~110 more lines not shown here)
OCR-STAR - An OCR String Aligner
Based on the Smith-Waterman String Alignment Algorithm
isabell.hubert@ualberta.ca - Version v1.2
run at 2015-10-05 15:37:52.631302

### Loading file type *.txt from directory sandbox/testdir/blankline

### your validated base file (REF in algorithm):
 sandbox/testdir/blankline/blanklineTest_val.txt

### files to be compared to base file (QUERY in algorithm):
 sandbox/testdir/blankline/blanklineTest.txt

### The lines in your base text:
### sandbox/testdir/blankline/blanklineTest_val.txt
### stripped of all blank lines are:

nē qā’li i’sin lā’ña l! Lëé’ldaiani. Un
L! Lao’a’si t’a l! djīgā’n iši’n sli’n iš-
dagā’ñane. Wa’gien siñiai’yan.
Wa’gien wa’daa’lēk” na-i “Ai A’ñ’a l’
isā’wane. Lā’wē wēe A’ñ’a l’ isdaí’ane.

### Lines from this text will show up as REF in the algorithm below.
RESULTS OF SMITH-WATERMAN ALIGNMENT ALGORITHM

## COMPARING...

**base file (REF):**
sandbox/testdir/blankline/blanklineTest_val.txt

**with OCRed file (QUERY):**
sandbox/testdir/blankline/blanklineTest.txt

### COMPARING LINE 1

**Query:** ɨ Nê Qâ’î ɨ’SiN Lâ’ña Ɂ! Ɂ!e’ Ɂ!DAIANI. DN 55

| Score: 85 |
| Matches: 47 (83.9%) |
| Mismatches: 9 |
| CIGAR: 31M1D24M |

**Ref:** ɨ Nê Qâ’î ɨ’SiN Lâ’ña L! Ɂ!e’ Ɂ!DAIANI. UN 56

### COMPARING LINE 2

**Query:** Ɂ! L! LÂO Ɂ!A’SI T!LAɁ! L! DJiG–A’N ɨSi’N Ɂ! S–L!Ɂ! Ɂ!S– 56

| Score: 94 |
| Matches: 51 (86.4%) |
| Mismatches: 8 |
| CIGAR: 18M1113M2D15M1D9M |

### COMPARING LINE 3
ANALYSIS TOOLS: OCR\text{EVALUATION}

- not easily loopable; does produce aggregate reports though
- visually appealing & interactive output
  - e.g. hovering & highlighting
- command line & GUI
  - GUI is very very buggy though
ANALYSIS TOOLS: OCREEVALUATION

• not easily loopable; does produce aggregate reports though
• visually appealing & interactive output
  • e.g. hovering & highlighting
• command line & GUI
  • GUI is very very buggy though
• no matrices, but character lists + character & word accuracies
• handles Unicode well

• current tool of choice
Recognition on some pages is really bad.

- about 30 pages total had abysmal recognition rates (~ 15%).
- issues:
  - skew of up to 1 degree
    - tesseract has a line finding algorithm that should be able to deal with this...? Smith 2007
    - skew of less than 1 degree typically causes no problems…? Rice et al. 1994
  - curved baseline
  - black edge (of varying thickness and darkness) around the sides of the page
RECOGNITION ACCURACIES

We expect:

1) The source text approach to result in more accurate models overall;

2) Those models based on more pages to perform better than those based on fewer pages.
The Influence of Page Rotation on CRA

Character Recognition Accuracy in %

rotated
FALSE
TRUE
model_type
por
tif

96.51%
The Influence of Page Rotation on WRA

Word Recognition Accuracy in %

rotated
FALSE
TRUE

model_type
por
tif

Number of Pages

85.43%
### HOW DO OUR MODELS STACK UP?

<table>
<thead>
<tr>
<th>Language</th>
<th>CRA in %</th>
<th>WRA in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>99.53</td>
<td>93.6</td>
</tr>
<tr>
<td>Italian</td>
<td>99.46</td>
<td>94.59</td>
</tr>
<tr>
<td>Russian</td>
<td>99.33</td>
<td>94.43</td>
</tr>
<tr>
<td>Hebrew</td>
<td>96.8</td>
<td>89.42</td>
</tr>
<tr>
<td>Northern Haida</td>
<td>96.51</td>
<td>85.43</td>
</tr>
<tr>
<td>Japanese</td>
<td>95.74</td>
<td>81.28</td>
</tr>
<tr>
<td>Vietnamese</td>
<td>94.94</td>
<td>80.61</td>
</tr>
<tr>
<td>Thai</td>
<td>78.69</td>
<td>19.47</td>
</tr>
</tbody>
</table>

Smith 2013, p.11
HOW DO OUR MODELS STACK UP?

<table>
<thead>
<tr>
<th>Error Rate</th>
<th>Time Period</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1%</td>
<td>28</td>
<td>106</td>
<td>169</td>
<td></td>
</tr>
<tr>
<td>1-30%</td>
<td>101</td>
<td>55</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>30-100%</td>
<td>81</td>
<td>18</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

adapted from Mihov et al. 2005, Table 1, p.3
• The one, once de-facto standardized analysis tool, the ISRI toolkit, does not work in 2015.

• [related] We currently have no character confusion matrices.
  • Port ISRI tools so they accept Unicode input to deal with all languages. We are on that.

• The process still requires some knowledge of the command line.
  • Automate process further to make the system immediately accessible to language communities.

• Do we keep line breaks and hyphenation at those line breaks? Preserve layout – or focus on corpus?

• How to deal with (obvious) transcription errors in the source text?
IDEAS FOR THE FUTURE*

• see whether an FST could improve recognition accuracy
  • word lists not usually too helpful in morphologically rich languages, but
  • integrating an FST doesn’t seem too straightforward either (Miikka & Antti)
• potential use of transcriptor to transform Swanton orthography into modern standard
• potential expansion to another variety of Haida

* these are beyond the scope of izzy’s GP 😊
CONCLUSIONS & CONTRIBUTIONS

• OCR research has focused on English/Latin scripts for too long
• there are no standardized measures for OCR quality control
  • but: there is a growing interest

• the resulting OCR model will be the first for a NA indigenous language
  • can serve as basis for OCR for other indigenous languages
• the resulting electronic corpus of Northern Haida will be the largest to date
• learning materials can be constructed based on:
  • the quantitative makeup of the language
  • large amounts of natural language data
THANK YOU!

Any questions? Thoughts?

send love & hate mail to: isabell.hubert@ualberta.ca
REFERENCES


jTessBoxEditor 1.4. (2015).


Isabell Hubert - isabell.hubert@ualberta.ca - ALT Lab, 4 Dec 2015