Community Discussion: OCR Correction

HTRC UnCamp 2012
Sept. 11, 2012

***Attendees*:**
Loretta Auvil (UIUC), Travis Brown (UMD), Tom Burton-West (UM), Tim Cole (UIUC), Micah Cooper (Texas A&M), Andrew Filer (UW), Paul Fogel (UC) - moderator, Kirk Hess (UIUC), Andy Mardesich (UC) -note taker, Peter Organisciak (UIUC), Maryam Rahnemoonfar (IU), Daniel Tracy (UIUC), Michael Twidale (UIUC), Ted Underwood (UIUC), LaTesha Velez (UIUC), Sean Wilner (UIUC)

OCR correction is important. …But as a caveat, not all people need OCR cleaned up for the work they're doing.

Two things Tim Cole sees of importance:

- How to track changes that have been done?
- How to handle OCR corrections from a page that has been rescanned?

The HT Metadata Management System has a shadow file to allow a history and access to previous version of metadata file.

We would want to archive the cleaning processes.

Crowdsourcing and Machine Learning
You can feed crowdsourcing output into machine learning. A correction profile can be created depending on what era the book is from (since books from different eras bring up different types of OCR errors). It's not hard to see typical errors from a given period.

Crowdsourcing could handle correcting OCR and verifying mechanized corrections of OCR - For instance, the long/medial s (ſ).

Crowdsourcing however cannot be captured as a process. Crowdsourced output should be fed separately into training data for machine learning.

Data such as 'date' and 'scanning tool' can be coordinated together into machine learning problem.

Versioning
We need to track which things are getting OCR correction updates in Google.

HTRC can log changes. This can be HTRC's responsibility.

Tom Burton-West recalls that Google goes through whole corpus every 2 years. Problem areas can be addressed by Google per request (i.e. Arabic).

Path forward seems to be that versions would have to come to the HathiTrust Research Center; repository overwrites old OCR with any newer (possibly updated) OCR.

One difficult area in the OCR correction process: You may wrongly "fix" things that are already correct.

We will need to rank things by easiest to fix to separate out problem sets and coordinate how to address each. It's easy to rank things by frequency, to handle forms that don't occur in dictionary.

How can we version some materials?
How do you merge human corrections with machine corrections?
Thing you want to avoid: things corrected and then fixed wrongly again

Archiving Transformations
How do you define transformations you keep and reapply them?

One-gram transformation: proper nouns not corrected
Contextual corrections: You can't make a transformation rule in the case of 'six' or 'fix', but you can contextually spell check them using 2-grams (i.e. how often does it say "I sixed" something?). And this is useful to have this data.

You don't want to be doing corrections at the generality of "this is error => this is correction".
It has to be implemented in a rule based language. But you confront multiple possible corrections in any word longer than 3-4 characters. You need an algorithm that understands that the problem has multiple different answers.

The level to archive it at is: the transformation rules and the replacement rules (word to word)
It can be broken apart to include a choice algorithm that makes a selection. It could output a suggestion.
There's no simple standard for how this should be done.

Improving OCR Engines

- Certain OCR Errors can be found consistent with the OCR engine it was generated from.
- Travis Brown has worked with OCRopus and Tesseract: retraining Tesseract
- Texas A&M used Tesseract & Gamera, have run things to automatically choose selection.
- If you combine the output of 4 diff't OCR engines, you will get a better product.
- There's lots of copied of stuff in HathiiTrust. You can collate the OCR and compare. Google's already doing this on Public Domain material.

Conclusions:

How might we accommodate changes into corpus?

It's about applying corrections to something dynamic.

There's not one standard way corrections can be applied. How might they be shared in a research center?

How could you select from selection models?

And a post of Ted's I found relevant to this discussion:

"The obvious thing we're lacking" argues for a way to identify the best possible copy of a volume to save researchers much pain. Do we need to create an ontology for our OCR?