Solutions and findings from the CJK search and discovery enhancement project in Stanford University's SearchWorks

Mieko Mazza
Japanese Technical Services Librarian
East Asia Library
Stanford University Libraries

Acknowledgement

- Naomi Dushay (Senior Programmer/Analyst, Digital Library Systems and Services (DLSS), Stanford University Libraries)
- Charles Fosselman (Access & Digital Information Services Librarian, East Asia Library, Stanford University Libraries)
- Vitus Tang (Head, Data & E-resources Control, Metadata Department, Stanford University Libraries)

Presentation outline

- Background
- Goal
- Method
- Timeline
- Challenges and issues
- Summary

Background

- Known issues
  - Multiple scripts used
  - Traditional vs. simplified Chinese
  - Traditional vs. modern Japanese
  - Hangul and Hancha
  - Automatic language detection
  - word division, white space between characters vs. no space

Background (cont.)

- CJK language materials (the 6th, 8th and 14th largest language collections at Stanford)
- Solr
- Blacklight
- CJK librarians submitted CJK acceptance test results (August 2012)

Goal

- Improve simplified vs traditional character matches
- Improve variant Japanese scripts (hiragana, katakana, kanji, and romaji) character searches
- Improve translation table
- Improve index
- Improve cross-mapping among the CJK scripts
- Improve word division, white space issues
- Improve character variant search

(Stanford University Consul – CJK Searching)

Method

- CJK script search tests by CJK librarians through VPN test site
- Behavior-driven development
- Feedback link
- Different length query strings
- All types of searches
- Different scripts
- Phrases (using quotes)
- Queries with and without whitespace

(Stanford University Consul – CJK Testing)

Timeline

- Meeting with DLSS/SW development team (August 2013)
- Testing by CJK Librarians (August 2013-September 17, 2013)
- CJK phase 1 in production (September 19, 2013)
- CJK phase 2 testing with additional enhancements
- Phase 2 improvements to production (November 11, 2013)

CJK Tweak 5: CJKFoldingFilter

```
public void testAddlHanVariants() throws Exception
{
        checkOneTerm(analyzer, "嶽", "岳"); // 嶽 5DBD (std trad) => 岳 5CB3 (simp) -
        checkOneTerm(analyzer, "国", "國"); // 国 56EF (variant) => 國 570B (std trad)
        checkOneTerm(analyzer, "戲", "戯"); // 戲 6231 (variant) => 戯 6232 (std trad)
        checkOneTerm(analyzer, "教", "教"); // 教 654E (variant) => 教 6559 (std trad)
        checkOneTerm(analyzer, "甯", "寧"); // 甯 752F (variant) => 寧 5BE7 (std trad)
        checkOneTerm(analyzer, "研", "研"); // 研 784F (variant) => 研 7814 (std trad)
        checkOneTerm(analyzer, "緒", "緒"); // 緒 7DD6 (variant) => 緒 7DD2 (std trad)
        checkOneTerm(analyzer, "姬", "姫"); // 姬 59EB (modern Japanese) => 姫 59EC (s
}
```

```
<!-- CJK field type: final -->
<fieldtype name="text_cjk" class="solr.TextField" positionIncrementGap="10000" autoGeneratePhraseQueries="false">
  <analyzer>
    <!-- remove spaces among hangul and han chars if there is at least one hangul char -->
    <!-- a korean char guaranteed at the start of the pattern: pattern="(\p{Hangul}\p{Han}*)\s+(?=[\p{Hangul}\p{Han}])" -->
    <charFilter class="solr.PatternReplaceCharFilterFactory" pattern="([\p{InHangul_Jamo}\p{InHangul_Compatibility_Jamo}\p{InHangul_Syllables}][\
    <!-- a korean char guaranteed at the end of the pattern: pattern="([\p{Hangul}\p{Han}])\s+(?=[\p{Han}\s]*\p{Hangul})" -->
    <charFilter class="solr.PatternReplaceCharFilterFactory" pattern="([\p{InHangul_Jamo}\p{InHangul_Compatibility_Jamo}\p{InHangul_Syllables}\p{
    <tokenizer class="solr.ICUTokenizerFactory" />
    <filter class="solr.CJKWidthFilterFactory"/>
    <filter class="edu.stanford.lucene.analysis.CJKFoldingFilterFactory"/>
    <filter class="solr.ICUTransformFilterFactory" id="Traditional-Simplified"/>
    <filter class="solr.ICUTransformFilterFactory" id="Katakana-Hiragana"/>
    <filter class="solr.ICUFoldingFilterFactory"/> <!-- NFKC, case folding, diacritics removed -->
    <filter class="solr.CJKBigramFilterFactory" han="true" hiragana="true" katakana="true" hangul="true" outputUnigrams="true" />
  </analyzer>
</fieldtype>
```

Name: [redacted]
Email: [redacted]@stanford.edu
Affiliation: Stanford Staff
Comment:
" 江戶 (traditional)" + "Search Everything" + "Limit to Japanese" retrieves (relevant, best matches on top) 1856 hits in both SW CJK Test & SW Phrase. However, "江戸 (modern)" only retrieves 2 in both SW CJK Test & SW Phrase site. 戶 <=> 戸 mapping is still not working.

江戶(戸) = Edo (former name of Tokyo)

Message sent from: http:/[redacted]/?utf8=%E2%9C%93&q=+%E6%B1%9F%E6%88%B6&search_field=search&utf8=%E2%9C%93&f%5Blanguage%5D%5B%5D=Japanese

A corresponding relevancy test might be:

```
context 'Edo (old name for Tokyo)', :jira => ['VUF-2726', 'VUF-2770'] do
  # Second char of traditional doesn't translate to second char of modern with ICU traditional->simplified
  it_behaves_like "both scripts get expected result size", 'everything', 'traditional', '江戶', 'modern', '江戸', 1900, 2000
  it_behaves_like "matches in vern short titles first", 'everything', '江戶', /(江戶|江戸)/, 100  # trad
  it_behaves_like "matches in vern short titles first", 'everything', '江戸', /(江戶|江戸)/, 100  # modern
  # exact match
  it_behaves_like "matches in vern short titles first", 'everything', '江戶', /^(江戶|江戸)[^[[:alnum:]]]*$/, 3  # trad
  it_behaves_like "matches in vern short titles first", 'everything', '江戸', /^(江戶|江戸)[^[[:alnum:]]]*$/, 3  # modern
  # starts w
  it_behaves_like "matches in vern short titles first", 'everything', '江戶', /^(江戶|江戸)/, 12  # trad
  it_behaves_like "matches in vern short titles first", 'everything', '江戸', /^(江戶|江戸)/, 12  # modern
end
```

```
1  ‶    [半]二重引用符(始)
2  ”    [全]二重引用符(終)
3  “    [全]二重引用符(始)
4  "    [全]濁点
5  ″    〔単位〕秒
6     ‶ 環境依存文字
7  ‟    環境依存文字
8     ” 環境依存文字
```

```
describe "replace_special_quotes" do
  it "should replace any special characters with quotes in user params" do
    q = 'query'
    user_params = {:q => "«#{q}» ‘#{q}’ ‚#{q}‘ “#{q}” „#{q}“ ‹#{q}› 「#{q}
                          『#{q}』 〝#{q}〟 ﹁#{q}﹂ ﹃#{q}﹄ "#{q}" 「#{q}」"}
    controller.stub(:params).and_return(user_params)

    controller.send(:special_quote_characters).each do |character|
      user_params[:q].should     include character
    end
    controller.send(:replace_special_quotes)
    controller.send(:special_quote_characters).each do |character|
      user_params[:q].should_not include character
    end
    # there are 14 instances of the q variable in the user_params[:q] string above
                                                  \"#{q}\"").length.should eq 14
```

```
class CatalogController < ApplicationController
  include Blacklight::Catalog
  include ModsDisplay::ControllerExtension
  before_filter :check_callnum_search, :replace_special_quotes,
                :check_dh_az_search, :check_hidden_context
```

```
def replace_special_quotes
  unless params[:q].blank?
    special_quote_characters.each do |char|
      params[:q].gsub!(/#{char}/, '"')
    end
  end
end
```

```
def special_quote_characters
  ["\u00AB", "\u00BB", "\u2018", "\u2019", "\u201A", "\u201B", "\u201C",
   "\u201D", "\u201E", "\u201F", "\u2039", "\u203A", "\u300C", "\u300D",
   "\u300E", "\u300F", "\u301D", "\u301E", "\u301F", "\uFE41", "\uFE42",
   "\uFE43", "\uFE44", "\uFF02", "\uFF62", "\uFF63"]
end
```

Before the project

	A	B	C
1			SW
2	***CHINESE***		
3	中国经济政策	simplified, title search	0
4	中國經濟政策	traditional, title search	1
5	中国　经济　政策	simplified, title search	0
6	中國　經濟　政策	traditional, title search	0
7			
8	***JAPANESE***		
9	まんが	hiragana, title search	3
10	マンガ	katakana, title search	13
11	漫画	modern, title search	17
12	漫畫	traditional, title	3
13			
14	***KOREAN***		
15	한영우	author search	3
16	한 영우	author search	29

After the project

	A	B	C
1	中国经济政策 (simp)	title	66
2	中國經濟政策 (trad)	title	66
3	中国 经济 政策 (simp)	title	84
4	中國 經濟 政策 (trad)	title	84
5			
6	まんが (hiragana)	title	70
7	マンガ (katakana)	title	140
8	漫畫 (traditional)	title	237
9	漫画 (modern)	title	237
10			
11	한영우	author	34
12	한 영우	author	38

Challenges and issues (Chinese)

1. **Searching by romanization is still the most effective and reliable way to find Chinese-language materials in SearchWorks. Everyone searching for Chinese materials still needs to be familiar with Pinyin romanization rules.**
2. **Simplified Chinese and traditional Chinese are inter-searchable. For example, using the simplified characters 张爱玲 for an author search, SearchWorks will retrieve records with both 张爱玲 and 張愛玲 as the author**
3. **Each Chinese character is indexed as a single word in SearchWorks. No space is needed between characters.**
4. **For a more precise search, double quotation marks can be used for phrase or term search. For example, “当代中国经济” in the keyword search query, records with exact phrases will be retrieved. 当代“中国经济” as the search query, Term 1 is 中国经济, term 2 is 当代. SearchWorks will retrieve records where "中国经济" appears as a phrase and 当代 appears anywhere in the record. Records with term 1 and term 2 in the same field will be ranked the highest. Please note, when you use double quotation marks “ ”, you have to switch input mode to English. Currently, Chinese-style quotation marks “ ” will not work.**
5. **For some technical and historical reasons, a small number of characters with variant forms, for example 研 and 研, 戲 and 戏, 緒 and绪, etc., require extra mapping and are not inter-searchable. Search by romanization if no results or limited results are returned.**
6. **When Chinese character searches result in a mix of Japanese, Chinese and Korean language records, you can limit the search to Chinese records by selecting Chinese from the language field under “Limit Your Result” on the left.**

(http://library.stanford.edu/eal/chinese-collections/searching-chinese-language-materials-searchworks)

Challenges and issues (Japanese)

1. Search in kanji/hiragana/katakana

Titles/authors/keywords can be searched in Japanese script, with two limitations. The first is the lack of character mapping between kana and kanji. For example, when searching for "monogatari" in kanji, only results with the term written in kanji will appear. Likewise, searching for "monogatari" in hiragana will retrieve records in which it is written in kana. The other limitation relates to materials that have not yet been cataloged. Not-yet-cataloged materials can only be found in SearchWorks by using romanized search terms.

2. Search in Romanized Japanese

Romanize the Japanese title/author/keyword following the Library of Congress Japanese Romanization and Word Division rules. Diacritics are not needed when romanizing Japanese for searching in SearchWorks.

3. Search using English keywords

English keywords can also be used to find Japanese language books, since the subject words in the cataloging records are written in English. For example, in order to find books on 仏教文学, keywords “Buddhist Literature” can be used. If you wish to see only the resources written in the Japanese language, you can limit your search results by choosing “Japanese” under “Language” at the left column of SearchWorks.

(http://library.stanford.edu/eal/japanese-collections/searching-japanese-language-materials-searchworks)

Challenges and issues (Korean)

1. Search in Hangul ((한글))/Hancha(한자)

Titles/authors/keywords can be searched in Hangul and/or Hancha (Chinese characters used by the Koreans). Since SearchWorks currently does not do character mapping between Hangul and Hancha, if the title or name of the author or publisher could be written in Hancha on a publication, it is a good idea to perform two separate searches: one in Hangul and another in Hancha. For example, when searching 최인훈, a book titled 徐基源・崔仁勳 will not be retrieved, since 최인훈 is not included in the catalog record of the book. Therefore, you need to perform another search for 崔仁勳.

2. Search in Romanized Hangul

Romanize the Korean title/author/keyword following the Library of Congress Korean Romanization and Word Division rules. Diacritics are not needed when romanizing Hangul for searching in SearchWorks. For example, according to the LC Romanization rules 최인훈 or 崔仁勳 is romanized Ch'oe In-hun. However, when you search, just type Choe In-hun (no need to capitalize any words either).

3. Search using English keywords

English keywords can also be used to find Korean language books, since the subject words in the cataloging records are written in English. For example, in order to find books on 한국의 노동운동, keywords "Korea labor movement" can be used. If you wish to see only the resources written in the Korean language, you can limit your search results by choosing "Korean" under "Language" at the left column of SearchWorks.

(http://library.stanford.edu/eal/korean-collections/searching-korean-language-resources-searchworks)

Summary

- CJK search quality dramatically improved
- Successful collaboration
- Continuing community-based efforts for improvement

Reference

- Dushay, Naomi. “Discovery Grindstone.” Searching in Solr, Analyzing Results and CJK, 22 Jan. 2014, discovery-grindstone.blogspot.com/2014/01/searching-in-solr-analyzing-results-and.html.

Thank You

Mieko Mazza
Japanese Technical Services Librarian
East Asia Library
Stanford University Libraries
Email: miekom@stanford.edu