# Optical Character Recognition in Multilingual Text: A Brief Survey

Christa Mabee
University of Washington
Seattle, WA

May 27, 2012

## Abstract

This is an investigation of several state-of-the-art Optical Character Recognition (OCR) systems in the spring of 2012. An example from Open-Source software is considered alongside a leading commercial software package. Finally, some other promising Open-Source technology is briefly discussed.

## 1 Introduction

**This work was completed as part of an internship project sponsored by the Utilika Foundation.**

**The corpus used for this experiment is a collection of scans created for the project, 4,627 jpg images of multi-lingual dictionary pages containing four scripts. The images were scanned at 600dpi using a desktop scanner.**

**The dictionaries in the corpus are as follows:**

- A New English Thai Dictionary [16]
- A Tibetan English Dictionary [8]
- A Tibetan English Dictionary [7]
- English Persian Dictionary [4]
- English-Persian Dictionary [5]
- Pocket Thai Dictionary [6]
- New Thai-English, English-Thai Compact Dictionary [1]

## 2 Tesseract

### 2.1 About

**Tesseract was originally developed by Hewlett-Packard and UNLV in the 1990s. In 2005 it was released as open source, and in 2006 became sponsored by Google[14]. Today, as of version 3 (released in October of 2011), Tesseract can support multiple-language documents and already has language packs of varying quality for more than 30 languages, even including some resource-poor languages such as Cherokee and classic Quechua. Only six of these are officially supported, but the open source community contributes language packs. As of the time of writing, 3.02 is not available as a binary but must be compiled**

from the source accessible from the code repository.

Tesseract now has some rudimentary page layout analysis capabilities, which do extend to recognizing when text is in multiple columns on the page. Accuracy since 1995 has increased around 1%, and can be expected on English text to approach 97% character-level recognition accuracy. The system was unique in its day for implementing component analysis before attempting segmentation, which enabled it to detect sections with inverse-color text. [13]

### 2.2 Installation

Tesseract 3.02 is supported on Windows and Ubuntu. It may be run on Mac OS X, though the maintainers do not test this platform. On Windows, an installer is available. On Linux, one must compile the source code. This process completed for the writer on Ubuntu 12.04, 64bit, without any issues.

### 2.3 Training

Training Tesseract is a fair amount of work. For a detailed guide, see TrainingTesseract3[17]. The details for each font and character set must be specified, but most of the work is in generating "boxfiles", or text files that specify the coordinates of the boundaries of each character on an associated page image. The documentation for Tesseract[17] specifies that characters should have between 10 and 20 training examples. The examples are processed into feature vectors used by the classifier. As described in Smith 2009[12]:

> In training, a 4-dimensional feature vector of (x, y-position, direction, length) is derived from each element of the polygonal approximation, and clustered to form prototypical feature vectors. (Hence the name: Tesseract.)

The instructions for generating training data stress that it is vital to have characters spaced out on the training page. Overlapping boxes result in features being created or dropped inappropriately in the training stages. The suggested method is to print out the desired characters on paper, so that you have an appropriate number of characters and so that they are spaced easily. That is not possible with this dictionary project, as many of the fonts used are either proprietary or reproductions of handwritten signs.

In order to generate valid boxfile/image sets, I created pages with the desired characters essentially copy/pasted out of the corpus, cleaned up so that features from overlapping characters are removed. This made it possible to generate clean boxes, but means any baseline (in the sense of physical spacing) data is lost. The results were not impressive compared to the default language pack which was not trained on the given fonts, and so I suspect that either the importance of the baseline data or the sheer number of training examples

far outweighs the benefit of having font-specific training data. Additionally, creating training data this way was very time-consuming.

Creating boxfiles took over 60 hours for the seven fonts I prepared, and several of the fonts had characters not presented frequently enough to provide the recommended 15 examples. As a result I do not recommend attempting to collect font-specific training data for use in Tesseract unless the digital font is available. In addition to being incredibly tedious, it did not result in improved accuracy for any of the tested fonts. Bootstrapping a new character set would probably be better done by finding similar fonts and printing suitable training pages. (See: Future Work.)

There are multiple boxfile editing GUIs available; I tried several of them. The moshpytt[9] interface was in my opinion the best of the installable ones. Unfortunately, it suffers from not allowing the user to drag box boundaries, instead you have to use the directional arrow keys to move or resize them pixel by pixel. This slows down work considerably. A better interface, in some respects, is the AJAX based web interface Tesseract OCR Chopper by Dino Beslagic[15]. It allows resizing boxes with the mouse. Unfortunately it doesn't always process the entire page, and will sometimes skip over entire columns or paragraphs. It would be useful to this community to have a multi-platform boxfile editor with draggable box boundaries, in my opinion.

As a further note on training, when I began this project multi-language support was not in place yet and so I attempted to create a custom language that essentially just combined the resources for English and Thai. This was a complete failure. The resulting text was mostly Latin characters with a smattering of Thai diacritics superimposed, in place of the Thai script. I suspect that this is because the English resources outnumbered Thai resources by a fair margin (the system does not use grammar rules). In *Adapting the Tesseract open source OCR engine for multilingual OCR* [12] Ryan Smith mentions Thai specifically as being particularly ambiguous and so introducing multilingual recognition difficulties.

One can optionally include a wordlist when training Tesseract. The wordlist, if present, is one of the few linguistically informed features and will be used during word segmentation.[13]

Tesseract does not output confidence data by default, and although it can be modified to do so the numbers it uses do not appear to be very useful.

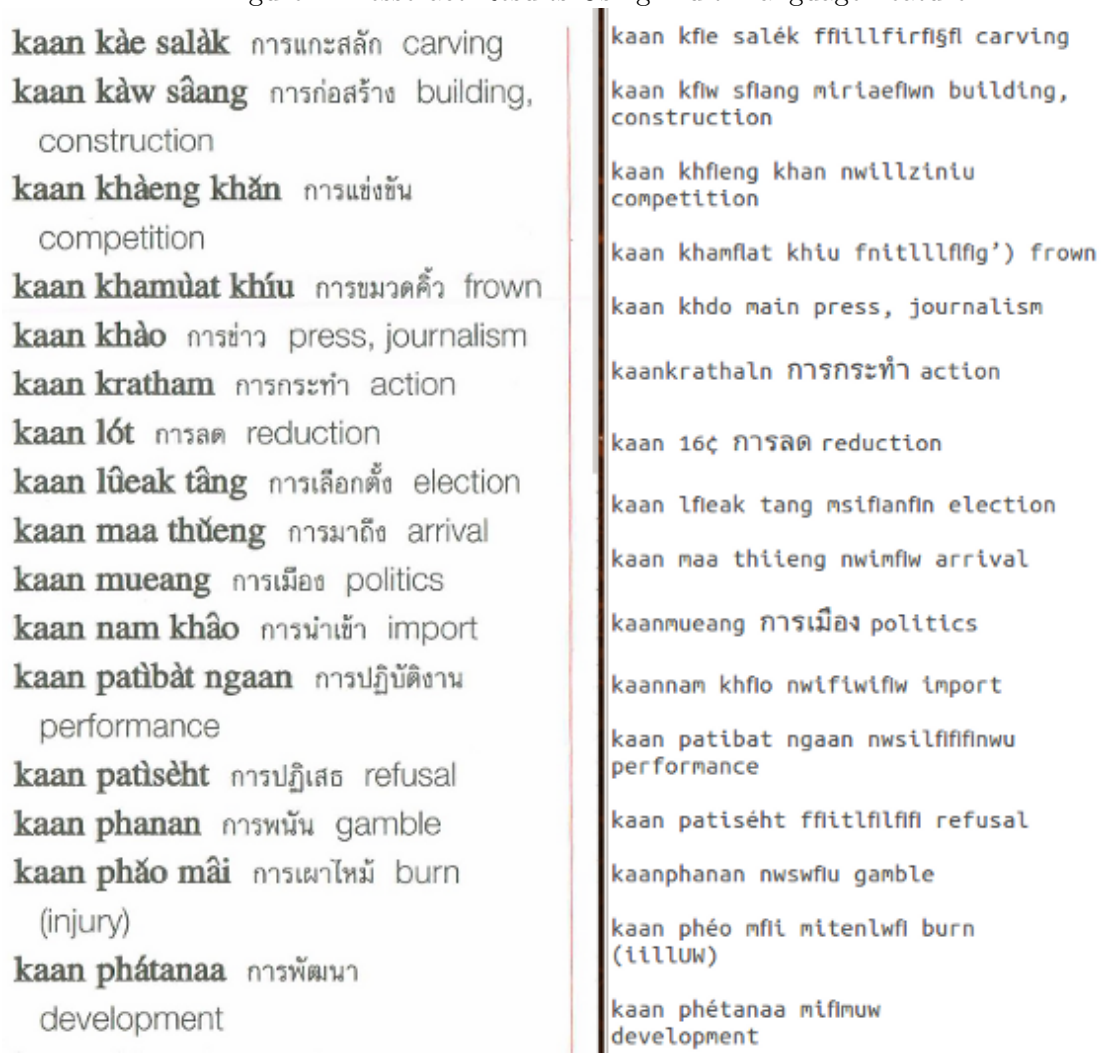
### 2.4 Results

Overall, I was not very impressed with Tesseract's performance. The accuracies observed in this corpus do not resemble those reported by the project. Figure 1 shows a sample of these results for Thai.

Figure II shows a sample of the results using the custom resources generated for

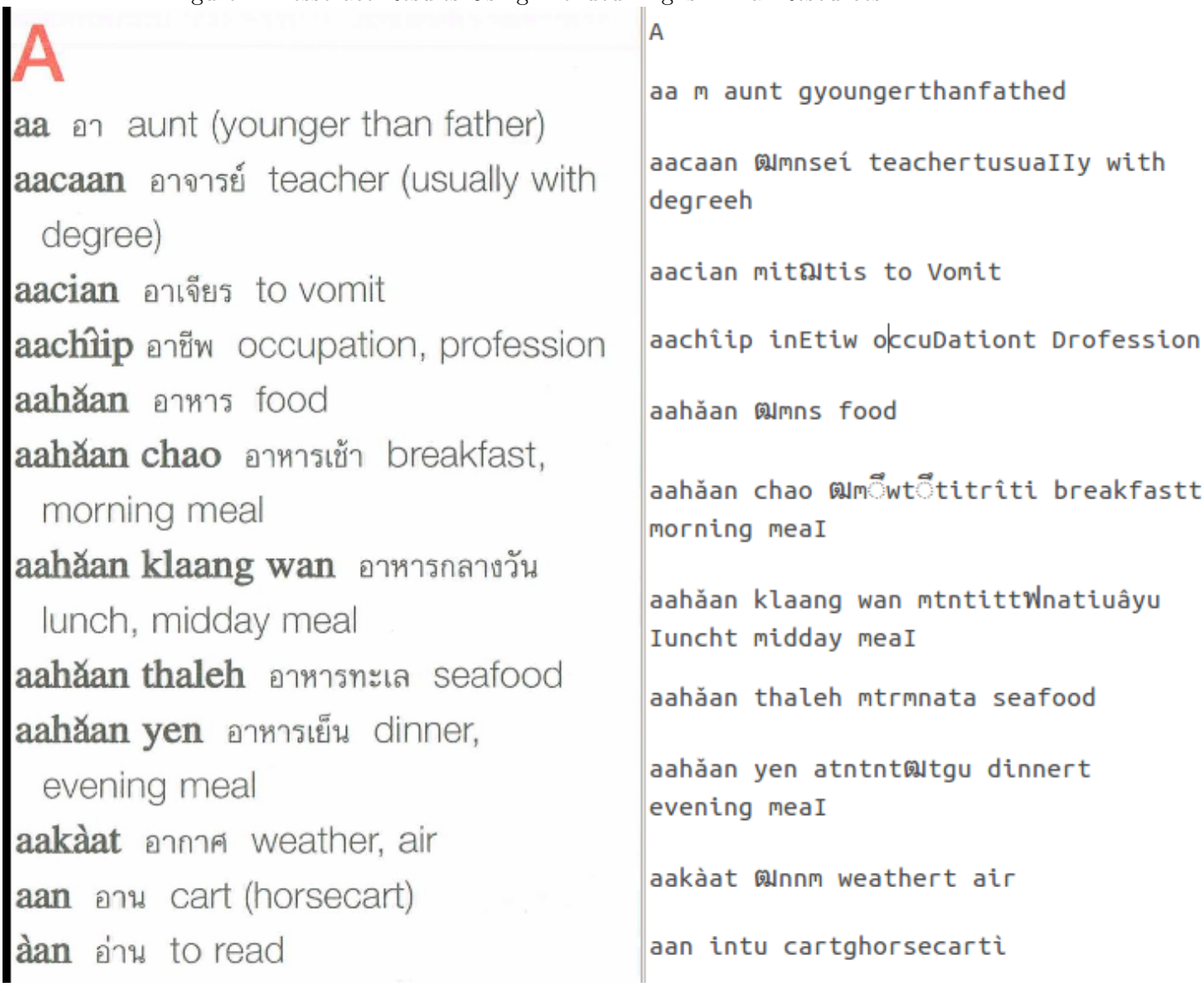
**a blended Thai/English single language.**

Figure 1: Tesseract Results Using Multi-Language Feature




**As you can see, character recognition is decent so long as Tesseract recognizes that language of the word. However, it frequently fails to recognize that the word is in Thai and not English; here this error rate is about 80%. It appears that Tesseract 3.02's new multi-language recognition feature is not ready for practical use just yet. Perhaps the disparity in confidence between the Latin and Thai characters could be corrected so that the system prefers Thai at a lower confidence threshold. (See Future Work.)**

Figure 2: Tesseract Results Using Blended English-Thai Resources



A

**aa** อา aunt (younger than father)

**aacaan** อาจารย์ teacher (usually with degree)

**aacian** อาเจียร to vomit

**aachîip** อาชีพ occupation, profession

**aahăan** อาหาร food

**aahăan chao** อาหารเช้า breakfast, morning meal

**aahăan klaang wan** อาหารกลางวัน lunch, midday meal

**aahăan thaleh** อาหารทะเล seafood

**aahăan yen** อาหารเย็น dinner, evening meal

**aakàat** อากาศ weather, air

**aan** อาน cart (horsecart)

**àan** อ่าน to read



**This figure illustrates how the system strongly preferred Latin character interpretations for many characters which were actually Thai.**

# 3 Abbyy

## 3.1 About

Abbyy is headquartered in Moscow, Russia. They produce a range of language products, such as digital dictionaries, translation software and services, and data capture/document processing software. One of their products, Abbyy FineReader, is a particularly powerful document processing engine that includes their SDK. Through their representative Mark Smock I obtained a 60-day trial license for the FineReader SDK.

## 3.2 Installation

The SDK comes with a GUI installer. While FineReader 9.0 is available for Linux, Windows and Mac, the 10.0 version's Linux distribution is not available yet, and so I used the 10.0 Windows version. The SDK exposes document analysis features, grammar features, and character recognition data.

## 3.3 Training

No training was required for Thai or Farsi. Abbyy provides a GUI-based training tool to generate data files for new character sets. Unfortunately, I did not have enough time to evaluate this functionality, and have been so far unable to launch the interface due to runtime errors.

Starting with one of the SDK examples, I wrote a fairly simple Java program to analyze the page, using the languages specified through the command line, and output data in raw text and XML format. This program and accompanying bash script are on the project Sourceforge page[2], though for licensing reasons I cannot also provide the jarfile for the engine itself, which is required to run the program. The program records the uncertainty and unrecognizable characters as reported by FineReader to generate a results file. The results from sample sets can be seen in Table I.

Following in the footsteps of Doermann et all[3], I randomly selected two pages from two of the books and manually prepared ground truth data for them. I then compared the output of the FineReader-based program. I used Microsoft Word's "Compare Document" feature to see differences in the text. Table II has the results, if all whitespace characters are considered equal (tab vs space).

Thai fared considerably better than Farsi, the problem with both Farsi texts is that the transliteration column is written in a partially cursive English script that FineReader performed poorly on.

## 3.4 Results

Table I: Character Recognition Confidence

| Book | Total | Failed | Suspicious |
|---|---|---|---|
| Pocket Thai | 50900 | 57 | 2301 |
| Thai English | 62645 | 40 | 2825 |
| | | 0.08% | 4.5% |
| English Farsi | 77461 | 785 | 26868 |
| | | 1.01% | 34.68% |

| Table II: FineReader Sample Accuracy | |
|---|---|
| Book | Sample Characters Correct |
| Pocket Thai | 99.56% |
| Thai English | 99.9% |

# 4 Other Systems, Future Work

## 4.1 OCRopus

**My evaluation of OCRopus, from a quick installation and test run on English text was that it may become relevant if work on the project continues, but at the moment offers no off-the-shelf functionality that Tesseract on its own does not. Furthermore, work on the project seems to have stalled.** [1]

**The documentation for OCRopus[10] does not cover training of its built-in recognizers, but Tesseract may still be used as the recognizer and its training I have already described in this paper. In the plans posted to the project website in 2010, the site maintainers state that training scripts will become available in version 0.5.**

## 4.2 Future Work

**As there is no Tibetan language pack for Tesseract yet, and I've gone to the trouble of learning to train Tesseract, I plan to work on that over the summer and contribute it back to the Tesseract project. Given my experience with the Thai language pack, I am going to try using existing Tibetan fonts to generate training pages in the suggested fashion (printing spaced-out characters and scanning the printed pages) rather than continuing with the attempt to collect all the examples from the existing pages.**

**It will be interesting to see how the two sets of samples compare in terms of recognition accuracy, and what the real threshold is for sufficient character examples in this script.**

**It would be interesting to see if the font-specific data, when added to existing language packs, can improve Tesseract's recognition accuracy. Other work in Tesseract worth examining: finding a way to boost recognition of non-Latin characters on a page with Latin text. I suspect that the ability to give a weighted preference to one language or another could be useful in texts where one script has more ambiguous characters. Ideally this would be a parameter one could provide on the command line to test until a good balance is found.**

**For Abbyy's FineReader, I still wish to examine the training tool. Currently I am unable to get the program to launch and suspect that there were issues with the SDK installation which the installer did not flag.**

**One idea which I briefly experimented with was using the OpenCV[11] (Open Computer Vision) library. While it is easy enough to train a character recognizer, I was unable to construct a working document analyzer (at least casually). It is**

[1] Currently, OCRopus will not compile in Ubuntu versions later than 10.04 due to outdated packages.

**my expectation that OpenCV could do very well at segmenting dictionary entries, given its successes at recognizing and parsing far more complex objects.**

## 5 Conclusions

**In my opinion, while Tesseract and OCRopus are certainly promising projects, Abbyy is the most practical solution for obtaining clean data. More investigation would be required for me to comment on the practicality of training Abbyy.**

## References


[1] Benjawan Poomsan Becker and Chris Pirazzi. *New Thai-English, English-Thai Compact Dictionary for English Speakers with Tones and Classifiers*. Paiboon Publishing, 2009, p. 982. ISBN: 1887521321. URL: http://www.amazon.com/Thai-English-English-Thai-Dictionary-Speakers-Classifiers/dp/1887521321.

[2] *DictionaryReader*. May 2012. URL: https://sourceforge.net/projects/dictreader/.

[3] D Doermann and B Karagol-Ayan. "Use of OCR for Rapid Construction of Bilingual Lexicons". In: (2003). URL: http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstracthttp://www.stormingmedia.us/28/2865/A286554.html.

[4] Dariush Gilani. *English Persian Dictionary*. San Jose: Maple Press Inc, 1983. ISBN: 0936347953.

[5] Sulayman Hayyim. *English-Persian dictionary*. 5th ed. New York: Hippocrene Books, 2006. ISBN: 0781800560. URL: http://books.google.com/books?hl=en\&lr=\&id=OxBMU6P-4S8C\&pgis=1.

[6] Benjawan Jai-Ua and Michael Golding. *Pocket Thai Dictionary*. Singapore: Periplus Editions, 2004, p. 96. ISBN: 079460045X. URL: http://www.amazon.com/Pocket-Thai-Dictionary-Thai-English-English-Thai/dp/079460045X.

[7] H. A. Jaschke. *A Tibetan-English Dictionary (Dover Language Guides)*. Dover Publications, 2003. ISBN: 0486426971. URL: http://www.amazon.com/Tibetan-English-Dictionary-Dover-Language-Guides/dp/0486426971.

[8] H. A. Jaschke. *Tibetan-English Dictionary (With Special Reference to the Prevailing Dialects, to which is added an English-Tibetan Vocabulary)*. Delhi: Motilal Banarsidass Publishers Private Limited, 1998, p. 671. ISBN: 8120803213. URL: http://www.amazon.com/Tibetan-English-Dictionary-Prevailing-English-Tibetan-Vocabulary/dp/8120803213.

[9] *moshpyTT*. Mar. 2012. URL: http://code.google.com/p/moshpytt/.

[10] *ocropus*. May 2012. URL: http://code.google.com/p/ocropus/.

[11] *OpenCV Library*. Mar. 2012. URL: http://opencv.willowgarage.com/.

[12] R Smith and D Antonova. "Adapting the Tesseract open source OCR engine for multilingual OCR". In: *Workshop on Multilingual OCR* Cc (2009). URL: http://dl.acm.org/citation.cfm?id=1577804.

[13] Ray Smith. "An overview of the Tesseract OCR engine". In: *Document Analysis and Recognition, 2007. ICDAR* (2007). URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4376991.

[14] *tesseract-ocr*. Mar. 2012. URL: http://code.google.com/p/tesseract-ocr/.

[15] *Tesseract OCR Chopper*. Mar. 2012. URL: http://pp19dd.com/tesseract-ocr-chopper/.

[16] Wit Thiengburanatham. *A New English-Thai Dictionary*. Ruamsan Publishers, 2000. ISBN: 9742464952. URL: http://www.amazon.com/New-English-Thai-Dictionary-Sentence-Structures/dp/9742464952.

[17] *Training Tesseract*. Mar. 2012. URL: http://code.google.com/p/tesseract-ocr/wiki/TrainingTesseract3.