Abbyy Finereader v10 - Training Material | WP3

Author: Jesse de Does. INL Internal Review: Katrien Depuydt

Introduction

ABBYY FineReader is a widely used, well-documented commercial product for text recognition in images. A complete overview of all members of the FR product line would be too much here, but the most relevant options are listed in table 1. It would not make sense, of course, to try to replace the existing end-user documentation; we can only give some first pointers.

Product	Description	Comment
ABBYY FineReader 11 Professional Edition (http://finereader.abbyy.com/professional/)	Creates editable, searchable files and e-books from scans of paper documents, PDFs and digital photographs	Desktop product, not suitable for digitization workflows
ABBYY FineReader 11 Corporate Edition (http://finereader.abbyy.com/corporate/)	Solution for streamlining document processing among workgroups in business, government and academic environments.	Desktop and small office product, not suitable for digitization workflows
ABBYY FineReader online (http://finereader.abbyyonline.com/en)	Online web client for FR.	Does not support batch processing
ABBYY Cloud OCR SDK: http://www.abbyy-developers.eu/en:onlineocrsdk:start ; pricing: http://ocrsdk.com/plans-and-pricing/ , api documentation at: http://ocrsdk.com/documentation/; examples for using the Cloud OCR SDK at https://github.com/abbyysdk/ocrsdk.com	Cloud-based online OCR service. The difference between this and the Engine SDK is that this has a web API instead of a local library interface.	Should be relatively easy to fit into any workflow system working with web service API's. Many parameters can be set, but control is less than in in the engine (e.g. external dictionaries or pattern training during recognition are not possible).
ABBYY Recognition Server 3.5 (http://www.abbyy.com/recognition_server/)	Automated OCR server and document capture software. Server-based OCR solution designed for mid- to high-volume document processing tasks.	Manages aspects like distributed processing and job scheduling.
ABBYY FineReader Engine 11 for Windows (http://www.abbyy.com/ocr_sdk_windows/)	Software Development Kit (SDK) for integrating ABBYY's multilingual recognition and conversion technologies into external applications.	Allows control over all that can be controlled, but requires highly technical implementation work.

Table 1: ABBYY finereader products

Which OCR Product do we need?[1]

Which product best suits your needs depends, of course, on your organization. The first three products (professional, corporate, FineReader online) are targeted towards end users and smaller organizations; we can omit them from further discussion because the XML output formats most suitable in a digital library context are not supported. The other three options (recognition server, cloud SDK and engine SDK) may be used for digitization workflows in a digital library

[1] This section does not address the issue of choosing between products from different manufacturers, for instance ABBYY versus Omnipage or Tesseract OCR.

context.

Engine vs Recognition Server

The recognition server manages aspects like distributed processing and job scheduling; the engine allows full control over all configurable processing options, but using it requires software development. Development of custom applications involving OCR requires the use of the engine SDK. It seems hard to give a general rule for use in digitization workflows, but, for instance, service providers or libraries targeted on running high-volume standardized processes might well use the recognition server, whereas organizations with skilled technical staff and specific requirements might favour the engine.
In terms of technical staff competence, deployment of the recognition server leans on systems administration, whereas the engine SDK leans more heavily on software developers.

Cloud SDK vs. Engine SDK

For a comparison between the FineReader Engine SDK and the cloud SDK, see http://www.abbyy-developers.eu/en:onlineocrsdk:comp_onlinesdk-fre. See also http://forum.ocrsdk.com/questions/133/ocr-sdk-versus-finereader-engine:

> “FineReader Engine has a bigger set of tuning options compared to Cloud SDK. For example, using Engine you can specify your own dictionaries for recognition, you can get detailed information about recognition variants, you can setup advanced export parameters like encryption for created PDFs and so on. Engine has dozens of options, Cloud SDK provides access only to most useful of them.
> Cloud SDK is already using the best Engine's settings for general images, but in some cases it is possible to achieve better recognition results using FineReader Engine by tuning it according to you image source. For example you may set up a specific dictionary, or apply image preprocessing filters, or tell recognizer that there is no italic. This is not a simple task and requires a lot of testing.”

For which Tasks do we need the Engine SDK?

The engine SDK is the only product allowing for full control over a number of features, for instance

1. Use of custom dictionaries
2. Accessing recognition variants
3. We want to do medium to large scale in-house processing; we have our own workflow requirements or other reasons not the use the recognition server

Install Software

All mentioned ABBYY products come with easy to use graphical installers. Please refer to the manuals in case of problems.

System requirements

Recognition server

The recognition server system requirements are documented at http://www.abbyy.com/recognition_server/specifications/. The recognition server setup is distributed and allows for the distribution of functions over distinct machines. A number of stations with different functionalities are listed: Server Manager, Scanning Station, Processing Station, Indexing Station, Remote Administration Console, COM-based API, Web Service, Google Search

Appliance Connector, Microsoft Isearch Filter, each of which have rather modest minimal system requirements .
Of course, for high-volume processing, requirements will be less modest.

Engine

Minimal system requirements for working with the FR Engine are rather modest (cf. http://www.abbyy.com/ocr_sdk_windows/technical_specifications/) , PC with x86-compatible processor (1 GHz or higher), for processing multi-page documents — minimum 1 GB RAM, recommended 1,5 GB RAM. In practice, combining OCR processing with development work will be reasonably smooth with 4G ram and at least 2 cores.
Again, requirements for high-volume processing will be less modest. We quote a recent example: The Europeana newspaper project[2] OCR workbench in Innsbruck runs 32 parallel FineReader SDK processes to achieve processing of between 10.000 and 100.000 newspaper pages per day[3].

Development tools which can be used with the Engine SDK

We quote the technical specifications:

> “The ABBYY FineReader Engine application programming interface conforms to the COM standard and can be easily used in C/C++, Visual Basic, .NET, Delphi, Java or any development tool supporting COM components. The Engine can be adapted for use in scripting languages like VBS, JS, Perl.”

C++ or C# development work requires, in practice, an installation of Microsoft Visual Studio. We cannot supply detailed information on which version of Visual Studio is compatible with which version of the SDK, but the free express version of this development tool (Visual Studio Express 2013 for Windows Desktop) can be used. For other languages, development does not specifically require certain tools. We have tested, for instance, java development with Eclipse.
A caveat for java developers: Java development is possible, but does not support all engine options:

1. Methods which have out parameters or work with byte arrays are not supported
2. Callback interfaces and events are not supported

These limitations entail, among others, that deploying external dictionaries using the external dictionary interface is not possible unless special Java Native Interface wrappers are produced.

Cloud SDK

Since the processing load takes places at ABBYY’s servers, system requirements are determined by the processing requirements of your own workflow management and development tools. Any development tools providing support for implementing web service clients can be used. The code samples include examples for php, ruby, python, javascript, java, .net, asp.net and even bash using cURL.

Setup & getting started

First steps for working with the recognition server are documented at: http://knowledgebase.abbyy.com/article/701.
Code samples for the cloud API are at http://ocrsdk.com/documentation/.
Code samples for the Engine SDK are included with the engine distribution.

[2] http://www.europeana-newspapers.eu/
[3] cf. http://eod2013.techlib.cz/files/download/id/22/gunter-muhlberger.pdf

Documentation

More information can be found at the ABBYY website (http://www.abbyy.com). All ABBYY products come with extensive documentation.

Input

Input Image Formats

FineReader accepts a wide range of input image formats, among which we list: bmp, dcx, pcx, png, jpeg2000, jpeg, pdf, tiff, gif, djvu, jbig2, wdp, wic. It will not open images larger than 32512*32512 pixels. For dealing with other image formats, you can consult the SUCCEED tool list to find conversion utilities.

Supported Languages

FineReader reads documents in 188 languages (cf. http://finereader.abbyy.com/recognition_languages/) , of which 45 have dictionary support.

Limitations

The fact that your image format is supported and your language is implemented does not necessarily mean that your recognition results will be satisfactory. The main reasons for suboptimal results are

- Poor quality images, for instance low-resolution black and white images from old micro-films
- Degraded documents (warped, unclear printing, damaged, …)
- Font shapes unknown to the engine
- Your language may be listed as supported, but the actual language in your documents may be incompatible with the implemented language support, if it contains specific terminology, historical or regional language.

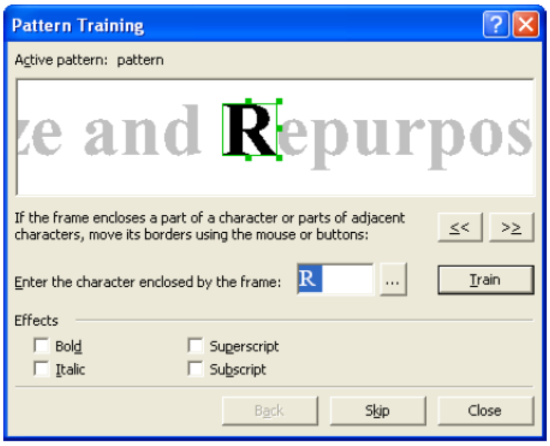
Extending Language Support

A limited amount of words can be added as user dictionary. There is unfortunately no utility to produce Finereader dictionaries from user word lists. A possible approach to implementation of language support is the *External Dictionary* mechanism in the ABBYY SDK, for which we refer to a separate document[4].

Training Character Shapes

It is possible to train the engine for unknown or unusual character shapes. The engine and the desktop user interface have the option to train a “user pattern” during recognition. User patterns may be saved and loaded for recognition jobs.

[4] Jesse de Does, The External Dictonary Interface Documentation.



Pattern Training

Active pattern: pattern

If the frame encloses a part of a character or parts of adjacent characters, move its borders using the mouse or buttons:

Enter the character enclosed by the frame: R

Effects

□ Bold
□ Italic
□ Superscript
□ Subscript

Figure 1: Pattern Training

This option may improve recognition of unusual character shapes, but it is impossible to reach the quality that would be obtained by full training of the engine, which can only be done by ABBYY. One should bear in mind that the applicability of the trained shapes is limited to images with not only the same font shapes, but also the same image parameters (resolution, quality, colour depth). For a really robust extension to different font shapes, you have to contact ABBYY (http://www.abbyy.com/support/).
For a comparison between the Finereader and the Tesseract OCR trainability, cf. for instance the case study http://lib.psnc.pl/dlibra/doccontent?id=358 , included with the training materials.

Output

Output formats

Output formats are: RTF, DOCX, XLS/XLSX, PPTX, PDF, PDF/A, HTML, TXT/CSV,(finereader) XML, ALTO (xml), FB2 (feedbook, an ebook format), EPUB (ebook format), ODT (open office document format).
In digitization workflows, one typically prefers XML-based solutions, which allow flexible handling of the recognized text, and in particular, store information about the location of text regions, lines, words, and (possibly) characters in the source image, thus enabling, for instance, highlighting of search terms in the original image in a retrieval application.
XML export is not possible in the two desktop editions of Finereader, or in the online OCR client. The recognition server, the cloud API and the Engine SDK do support this option. There are two supported XML output schemata:

1. ABBYY xml: this is an XML format defined by the company (cf. for instance http://www.abbyy-developers.com/en:tech:features:xml). This format allows the highest degree of control of the output, with options to store detailed glyph properties and alternative recognitions of words and characters.

2. ALTO xml: this is a widespread standard for optical character recognition results, cf. http://www.loc.gov/standards/alto/. ALTO export is possible in recent releases of the SDK and the extended recognition server, cf http://www.abbyy-

developers.eu/en:tech:features:alto. It is currently not possible to export glyph coordinates in this format.

If alternative formats are required for your digitization workflow, there are two main options:

1. Conversion of ALTO or ABBYY XML to your desired format. Cf for instance http://able.myspecies.info/abbyy-xml-tei-xml .
2. Implementation of an SDK application which directly exports the recognition result. As an example, we can mention the PAGE xml exporter developed in the IMPACT project.

Procedures

Procedures are very dependent on whether you use recognition server, cloud sdk or FR engine, and better explained in the accompanying documents.

Run OCR

This should be self-explanatory for the products with user interface; please refer to the respective product documentation.
For the engine, refer to the programming examples. To run OCR without development, we point the user to the sample application “commandlineinterface” in the code samples folder, residing in a folder like

```
C:\ProgramData\ABBYY\SDK\<version>\FineReader Engine\Samples\Visual C++
(Raw)\CommandLineInterface
```

The compiled version in the subfolder “Release” accepts one or several input images, and can recognize text dependent on a number of parameters implementing many of the options of the engine. Unfortunately, ALTO output is not currently one of the options.

Training Character Shapes

This is supported in the desktop versions and in the SDK. A character training utility sample application is now included with the SDK.

Further Reading

- The ABBYY web site (http://www.abbyy.com)
- The FineReader SDK User Guide (included in the training materials)
- http://www.codeproject.com/Articles/552521/The-Dew-Review-ABBYY-FineReader-Engine-OCR-SDK
- External Dictionary Interface Documentation, Jesse de Does (included in the training materials).
- Pricing information: http://www.abbyy-developers.eu/en:business:pricing (engine), http://ocrsdk.com/plans-and-pricing/ (OCR SDK), http://buy.abbyy.com/content/default.aspx (desktop products). Pricing for the recognition server: contact ABBYY.

- The following IMPACT reports (included in training materials) describing, among others, experiences on using the FR engine on a diverse set of historical documents.

  Use of Computational Lexica for OCR and IR on historical documents – a cross-language perspective (Section 5, page 44 of the included PDF describes the OCR evaluation)
  Lexicon-supported OCR of eighteenth century Dutch books: a case study (also includes a comparison with the tesseract OCR engine)
  The already mentioned *Report on the comparison of Tesseract and ABBYY FineReader OCR engines*

Licensing

Finereader is a commercial product developed by Abbyy (www.abbyy.com).