How to Optimize OCR Quality

Ivan Gravanov
Technical Project Manager

ABBYY Europe,
November 2010
Agenda

- What is OCR Quality?
- Image Quality for OCR
  - Scanning Settings
  - Image Pre-Processing
- Layout Analysis
  - Document Analyzers
- Tuning Text Recognition
  - Text Types
  - User Patterns
  - Languages
  - Dictionaries
  - Voting API
- Questions & Answers
What is OCR Quality?

- Document layout analysis (text, barcode and picture blocks)
- Text recognition rate (character confidence)
- Document synthesis (font family, size and style, hyperlinks, etc.)
- Layout retention in the export (object positions and coordinates)

Questions:

- Why optimize?
- What optimize?
When Optimize?

- Is there something to optimize?
When Optimize?

- No, there isn’t. Nothing comes from nothing.
Image Quality for OCR
Scanner Settings – Understanding the Resolution

- **Resolution** – the number of distinct pixels in each image dimension

<table>
<thead>
<tr>
<th>Stage</th>
<th>Resolution</th>
<th>Image Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Pixel</td>
</tr>
<tr>
<td>Before Scanning</td>
<td>96 dpi</td>
<td>200 dpi</td>
</tr>
<tr>
<td>After Scanning</td>
<td>200 dpi</td>
<td>96 dpi</td>
</tr>
</tbody>
</table>

- Image Interpolation (re-sampling) – the pixel number change on the image

96 dpi **ABBYY** Constant print size 200 dpi
Image Quality for OCR

Scanner Settings

- Scanning resolution
  - **300 dpi** – typical texts (font size >= 10 pt)
  - 400-600 dpi – small font texts (font size <= 9)

- Which color mode?
  - B&W – good quality documents
  - Grayscale – medium and poor quality
  - Color – if color is required in export

- Optimal brightness
  - **brightness** – suitable for recognition brightness level
  - **brightness** – too high level makes characters “torn” and very light
  - **brightness** – too low level makes characters distorted and stuck together
Image Quality for OCR

Image Pre-Processing – Distortions Removal

- Automatic rotation
- Automatic deskewing
- Cropping
- Page splitting
- Lines straightening
- Colour filtering

- New V10 binarization and pre-processing algorithms
Layout Analysis
Using Document Analyzers (DA)

- Accurate document analysis – Better recognition quality

- General document analyzer
  - Extracts text, tables, graphics (pictures), barcodes & patch codes, lines (separators)

- Document analyzer for full-text indexing
  - Extracts text, tables, graphics (pictures), barcodes & patch codes, lines (separators) and text inside of pictures and diagrams

- Document analyzer for invoice processing (small fonts)
  - Extracts text, tables as plain text, barcodes & patch codes, lines (separators), text inside of pictures and diagrams
Image Quality for OCR
Confident Barcodes Recognition

- Separate barcodes from text by wide white gap
- Barcode height should be higher than double height of text lines around
- Barcode length should be bigger than his height
- Width of thinnest bar should be $\geq$ 3-5 pixels
- Don’t use lossy compression like JPEG (it makes bar edges fuzzy)
- If possible don’t skew barcodes
Tuning Text Recognition
Defining Text Types

- **Text type** – the art of the characters models used during recognition

- Set up an appropriate text type

  - Typographic
  - Typewriter
  - Matrix
  - Handprinted
  - Index

- OCR A
- OCR B
- E13B
- CMC-7
- Gothic

ABBYY Europe Developers Conference, Munich 2010
Tuning Text Recognition
Defining Text Types

- Typefaces instead of specific fonts support
  - Serif typefaces
  - Sans serif typefaces
  - Monospaced typefaces (similar to both above)
  - Other typefaces (at a lower quality)

  http://en.wikipedia.org/wiki/Typeface

- “Fast auto-detection set” of text types
  - Typographic
  - Typewriter
  - Matrix
  - OCR A
  - OCR B
Tuning Text Recognition
Character Recognition – Pattern Training

- When train and use own user-defined patterns?
  - Texts with decorative fonts
  - Texts with unusual characters (e.g. mathematical symbols)
  - Texts of very poor print quality

- Pattern training for
  - single characters
  - ligatures (characters “stuck” together)

  fidget  fidget
  waffle  waffle
  fluff    fluff
Tuning Text Recognition
Context Recognition – Using Languages

- **Recognition language** – a combination letter sets and optional dictionaries, which may be applied during recognition

- 198 pre-defined (built-in) recognition languages
  - Natural non-hieroglyphic and hieroglyphic languages
  - Formal languages (e.g. programming languages)

```csharp
// Set predefined text language
pageProcessingParams.RecognizerParams.SetPredefinedTextLanguage( "English,German" );
```

- Multi-language recognition (not more than 3-5 at once)

- Custom user-defined recognition languages

```csharp
// Create new recognition language
TextLanguage textLanguage = makeTextLanguage();
// Set custom user-defined recognition language
pageProcessingParams.RecognizerParams.TextLanguage = textLanguage;
```

- Language auto-detection per character
Tuning Text Recognition
Context Recognition – Using CJK Languages

- Hieroglyphic pre-defined recognition languages
  - Chinese Simplified
  - Chinese Traditional
  - Japanese
  - Korean
  - Korean (Hangul)

- CJK recognition direction (automatic, vertical, horizontal)

- Hieroglyphic multi-language recognition
  - Combinations of hieroglyphic languages
  - Combinations of hieroglyphic and non-hieroglyphic languages

```c
// Set pre-defined text language
```
Tuning Text Recognition
Context Recognition – Working with Dictionaries

- Why use dictionaries?
  - Scanning noise and artifacts
  - Low print quality of documents
  - Pre-processing loss effects

![Diagram showing the use of dictionaries in text recognition.](image)
Tuning Text Recognition
Context Recognition – Working with Dictionaries

- Multiple dictionary types
  - Standard dictionaries – built-in extendable dictionaries (binary file sets)
    *.amd, *.amm, *.amt or *.ame files
  - User dictionaries – own pre-defined dictionary content (binary files)
  - Regular expression dictionaries – based on regular expressions (memory data)
    e.g. e-mail regular expression: 
  - External dictionaries – programming interface for creating own dictionary types
    (memory data or external resources e.g. databases)
  - Cache dictionaries – small dictionaries (~100 words) with on-the-fly changeable
    content (memory data)

ABBYY Europe Developers Conference, Munich 2010
Tuning Text Recognition
Context Recognition – Language & Dictionary API Overview

- FineReader Engine languages and dictionaries object model

- Built-in languages

- Text language (recognition language)

- Dictionaries
Internal structure and limitations?

- Organized as a tree
- Multiple branches (sub-trees)
- Nodes are character pairs
- 128K data per branch (sub-trees)

- Have huge dictionaries (100,000+ words)?
- Split them up into several user dictionaries
- Interested in a tool for splitting? Just ask...
Tuning Text Recognition
Voting API – Recognition Variants

- What is the Voting API?
  Programming interface, which provides access to different hypotheses of character or word recognition with corresponding weight values

- Recognition variants (hypotheses)
  - Word recognition variants and confidence
  - Single character recognition variants with weight

- When the recognition variants can be useful?
  - Several recognition engines
  - Different recognition results
  - Result checks with own algorithms or against databases

ABBYY Europe Developers Conference, Munich 2010
Any questions?

Thank you for your attention!

Ivan Gravanov
gravanov@abbyy.com