Structure Recognition

Joris Schellekens
Software Engineer - iText

Joris Schellekens
Software Engineer
iText

The iText R&D team

About PDF

- Various **standards**: PDF/A-1, PDF/A-2, PDF/A-3, PDF/UA, PDF/E, PAdES, etc.
- Level of **descriptiveness** (e.g. metadata) varies.
- **Basic** level:
  instructions for how a viewer (e.g. Adobe Reader) should render a document.

```
[a, -28.7356, p, 27.2652, p, 27.2652, e, -27.2652, a, -28.7356, r, 64.6889, a, -28.7356, n, 27.2652, c, -38.7594,
e, 444] TJ
/R10 10.44 Tf
68.16 0.24 Td
[", 17.1965, P, -18.7118, i, -9.35592, l, -9.35592, o, -17.2414, t, -9.35636, ", 17.1965,  , 250] TJ
```

Joris Schellekens
Software Engineer
iText

Expectations of end-users

- People are used to programs like Microsoft® Word™ .
- Reflow (e.g. flowing content around an image).
- Structure is part of the document.
- Export to various formats.
- Extracting data.
- Change appearance of a logical unit (e.g. word, line, or paragraph).

Joris Schellekens
Software Engineer
iText

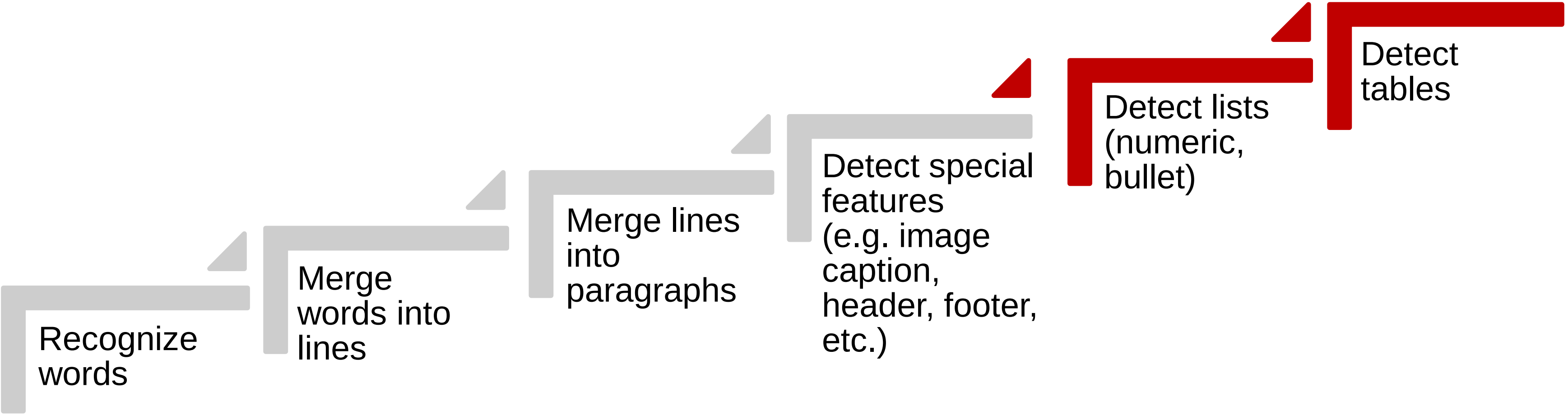
Constraints

- **iText Group nv is an open source company:**
  - **Close to the community.**
  - **Code should be something you want other people to see - accessible (rather than a dirty hack) .**
  - **Code should be something the user can change - contribution.**
  - Submitting pull requests.
  - Tweak and fine-tune to match experience and expertise.

Joris Schellekens
Software Engineer
iText

Technical

Proposed solution



Recognize words

Merge words into lines

Merge lines into paragraphs

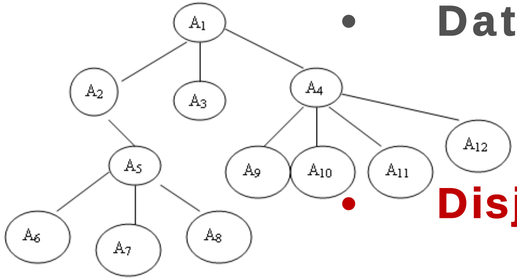
Detect special features (e.g. image caption, header, footer, etc.)

Detect lists (numeric, bullet)

Detect tables

Joris Schellekens
Software Engineer
iText

Efficient merging



- Data structure – evaluating equivalence.
- **Disjoint set** algorithm:
  - Space, find, merge all in *O(α(n)).*
  - Maps nicely to PDF ideas.

Joris Schellekens
Software Engineer
iText

When to merge

- Disjoint set tells us **how** the merging happens, not when.

- When depends on **what is being merged** - 2 approaches:

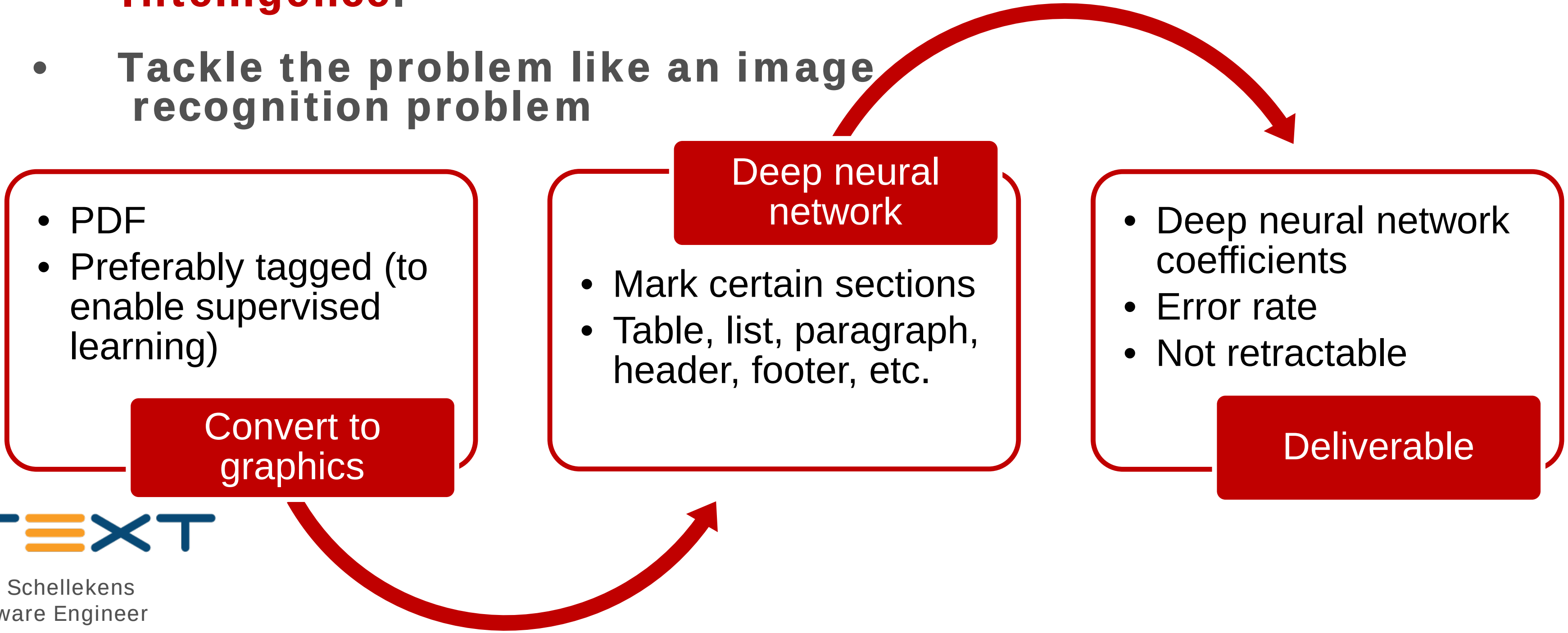
1. (Human) logic

  - Chunks are merged into words based on distance.
  - Words are merged into lines based on distance and global page layout.
  - Lines into paragraphs based on distance and visual cues.

Joris Schellekens
Software Engineer
iText

When to merge (alt)

- Alternative (bends the constraints) – use **Artificial Intelligence.**
- Tackle the problem like an image recognition problem



- PDF
- Preferably tagged (to enable supervised learning)

Convert to graphics

Deep neural network

- Mark certain sections
- Table, list, paragraph, header, footer, etc.

- Deep neural network coefficients
- Error rate
- Not retractable

Deliverable

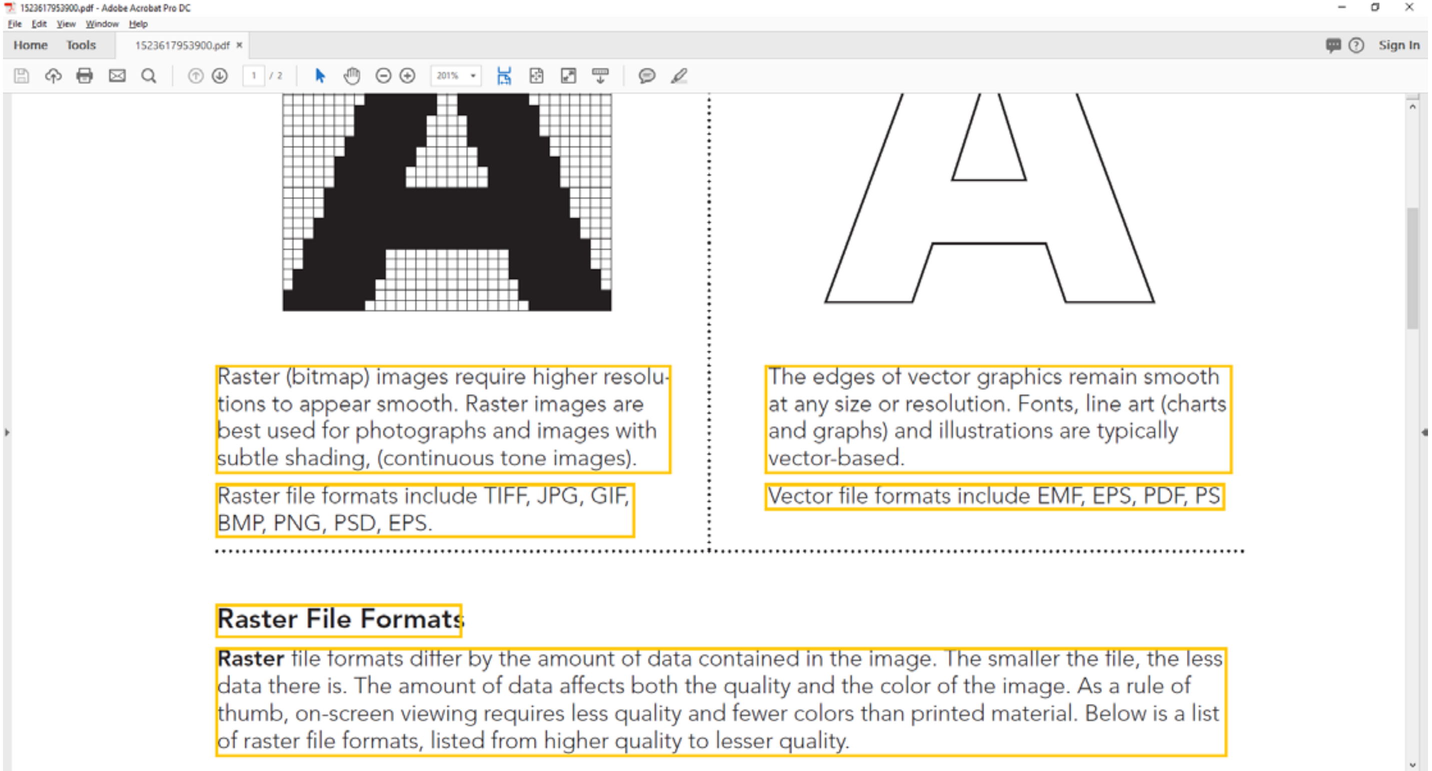
In Depth

Building a training set

- Use of NN requires training data
  - Large volume of (perfectly) tagged PDF documents
- Not readily available
- Build our own?
  - pdfHTML (convert random crawled HTML to PDF)
  - Gutenberg
  - Crawl the internet for PDFs
  - Industry contacts

Joris Schellekens
Software Engineer
iText

Convert training set


1523617953900.pdf - Adobe Acrobat Pro DC
File Edit View Window Help
Home Tools
1523617953900.pdf ×
1 / 2
201%
Sign In

Raster (bitmap) images require higher resolutions to appear smooth. Raster images are best used for photographs and images with subtle shading, (continuous tone images).

Raster file formats include TIFF, JPG, GIF, BMP, PNG, PSD, EPS.

The edges of vector graphics remain smooth at any size or resolution. Fonts, line art (charts and graphs) and illustrations are typically vector-based.

Vector file formats include EMF, EPS, PDF, PS

Raster File Formats

Raster file formats differ by the amount of data contained in the image. The smaller the file, the less data there is. The amount of data affects both the quality and the color of the image. As a rule of thumb, on-screen viewing requires less quality and fewer colors than printed material. Below is a list of raster file formats, listed from higher quality to lesser quality.

Joris Schellekens
Software Engineer
iText

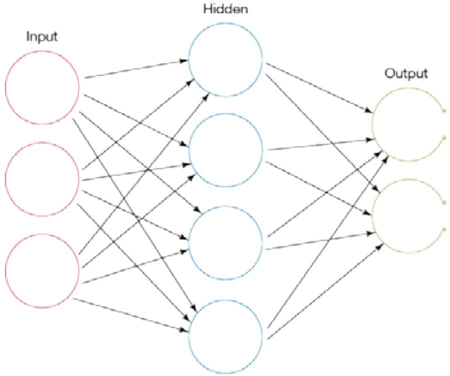
Convert training set (2)

- Feature selection
  - X
  - Y
  - Width
  - Height
  - Fontsize
  - Bold?
  - Italic?
  - Underline?
  - ΔX
  - ΔY

Joris Schellekens
Software Engineer
iText

Feed the beast

41	56	10	26	16	52	56	8	26	16	1	0	1
52	56	8	26	16	41	56	10	26	16	1	0	1
52	56	8	26	16	60	56	9	26	16	0	0	1
60	56	9	26	16	52	56	8	26	16	0	0	1
60	56	9	26	16	70	56	5	26	16	1	0	1
70	56	5	26	16	60	56	9	26	16	1	0	1
70	56	5	26	16	76	56	8	26	16	1	0	1
76	56	8	26	16	70	56	5	26	16	1	0	1
76	56	8	26	16	84	56	14	26	16	0	0	1
84	56	14	26	16	76	56	8	26	16	0	0	1
84	56	14	26	16	99	56	9	26	16	1	0	1
84	56	14	26	16	276	190	17	865	16	178	134	0



Joris Schellekens
Software Engineer
iText

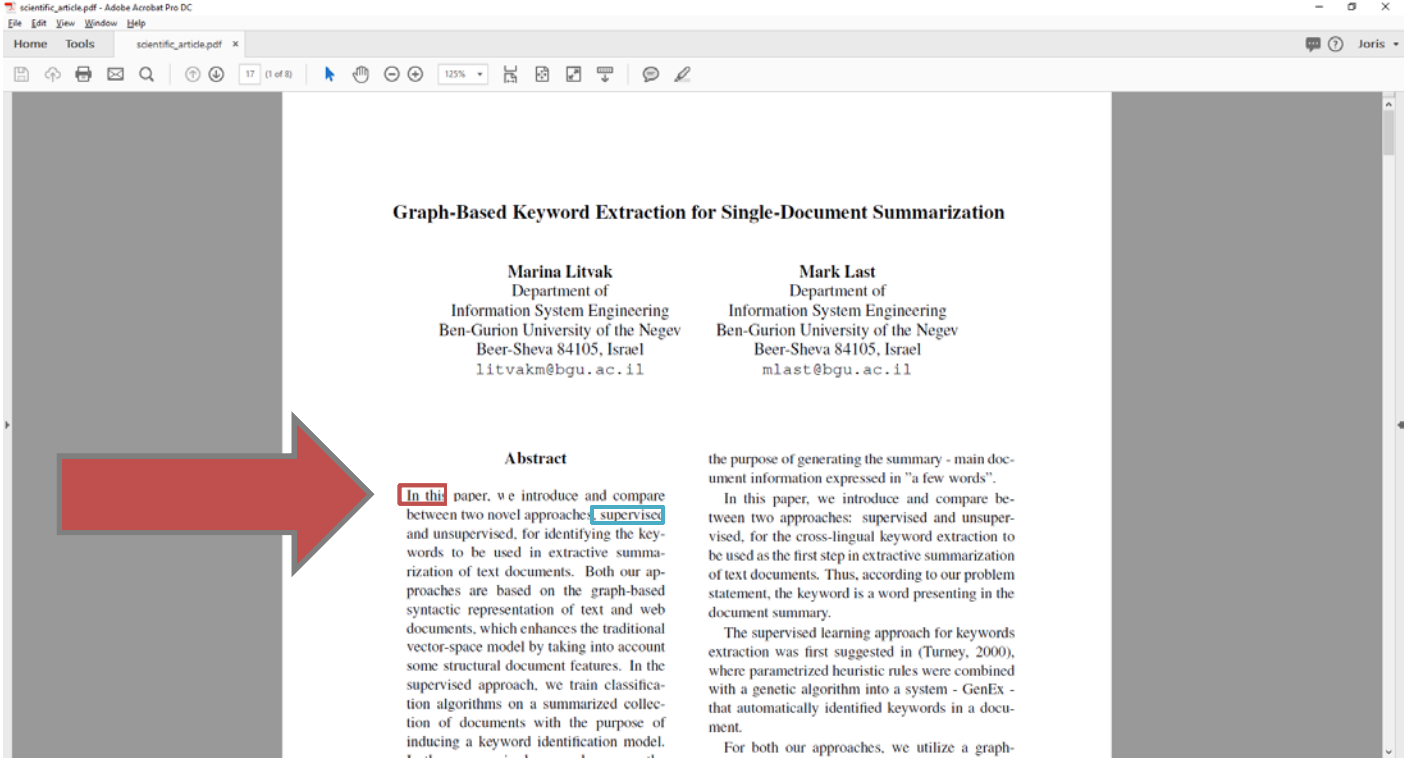
```
private void tagUsingAI(InputStream wekaModel, int confidence, List<StructureNodeImpl> nodes, boolean converge){
    IMerger nn = new WekaModelMerger()
            .load(wekaModel)
            .setThreshold(confidence / 100.0);

    // converge
    if(converge) {
        IMerger cv = new Convergence(nn);
        cv.apply(nodes);
    }
    else{
        nn.apply(nodes);
    }
}

public interface IMerger {

    void apply(List<StructureNodeImpl> l);

}
```

Benefits

Tagging an untagged document

```
[...]
      <MCID text="e" x="409.0" y="686.0" width="11.0" height="25.0" />
      <MCID text="r" x="421.0" y="686.0" width="8.0" height="25.0" />
      <MCID text="l" x="429.0" y="686.0" width="5.0" height="25.0" />
      <MCID text="a" x="435.0" y="686.0" width="11.0" height="25.0" />
      <MCID text="n" x="446.0" y="686.0" width="12.0" height="25.0" />
      <MCID text="d" x="459.0" y="686.0" width="12.0" height="25.0" />
      <MCID text=" " x="472.0" y="686.0" width="5.0" height="25.0" />
   </Span>
</P>
<P lang="nl" x="72.0" y="295.0" width="468.0" height="109.0">
   <Span x="72.0" y="295.0" width="182.0" height="14.0">
      <MCID text="m" x="72.0" y="295.0" width="9.0" height="14.0" />
      <MCID text="e" x="81.0" y="295.0" width="5.0" height="14.0" />
      <MCID text="t" x="87.0" y="295.0" width="3.0" height="14.0" />
      <MCID text=" " x="91.0" y="295.0" width="2.0" height="14.0" />
[...]
```

Reflow


scientific_article.pdf - Adobe Acrobat Pro DC
File Edit View Window Help
Home Tools scientific_article.pdf ×
17 (1 of 8)
125%
Joris

Graph-Based Keyword Extraction for Single-Document Summarization

Marina Litvak
Department of
Information System Engineering
Ben-Gurion University of the Negev
Beer-Sheva 84105, Israel
litvakm@bgu.ac.il

Mark Last
Department of
Information System Engineering
Ben-Gurion University of the Negev
Beer-Sheva 84105, Israel
mlast@bgu.ac.il

Abstract

In this paper, w e introduce and compare between two novel approaches, supervised and unsupervised, for identifying the keywords to be used in extractive summarization of text documents. Both our approaches are based on the graph-based syntactic representation of text and web documents, which enhances the traditional vector-space model by taking into account some structural document features. In the supervised approach, we train classification algorithms on a summarized collection of documents with the purpose of inducing a keyword identification model.

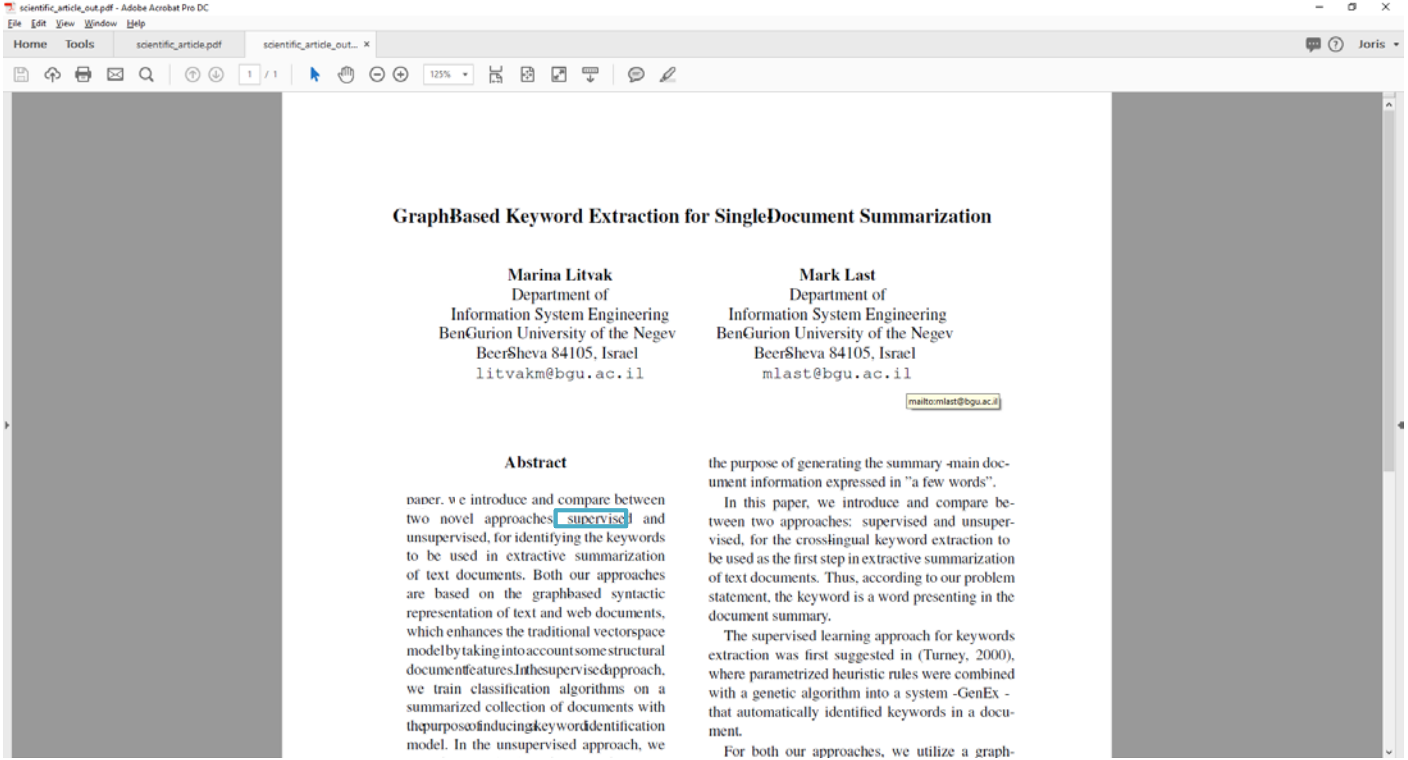
the purpose of generating the summary - main document information expressed in "a few words".

In this paper, we introduce and compare between two approaches: supervised and unsupervised, for the cross-lingual keyword extraction to be used as the first step in extractive summarization of text documents. Thus, according to our problem statement, the keyword is a word presenting in the document summary.

The supervised learning approach for keywords extraction was first suggested in (Turney, 2000), where parametrized heuristic rules were combined with a genetic algorithm into a system - GenEx - that automatically identified keywords in a document.

For both our approaches, we utilize a graph-

Reflow


scientific_article_out.pdf - Adobe Acrobat Pro DC
File Edit View Window Help
Home Tools
scientific_article.pdf
scientific_article_out... ×
Joris
1 / 1
125%
mailto:mlast@bgu.ac.il

GraphBased Keyword Extraction for SingleDocument Summarization

Marina Litvak
Department of
Information System Engineering
BenGurion University of the Negev
BeerSheva 84105, Israel
litvakm@bgu.ac.il

Mark Last
Department of
Information System Engineering
BenGurion University of the Negev
BeerSheva 84105, Israel
mlast@bgu.ac.il

Abstract

paper. w e introduce and compare between two novel approaches supervised and unsupervised, for identifying the keywords to be used in extractive summarization of text documents. Both our approaches are based on the graphbased syntactic representation of text and web documents, which enhances the traditional vectorspace model by taking into account some structural documentfeatures.Inthesupervisedapproach, we train classification algorithms on a summarized collection of documents with thepurposeofinducingakeywordidentification model. In the unsupervised approach, we

the purpose of generating the summary -main document information expressed in "a few words".

In this paper, we introduce and compare between two approaches: supervised and unsupervised, for the crosslingual keyword extraction to be used as the first step in extractive summarization of text documents. Thus, according to our problem statement, the keyword is a word presenting in the document summary.

The supervised learning approach for keywords extraction was first suggested in (Turney, 2000), where parametrized heuristic rules were combined with a genetic algorithm into a system -GenEx - that automatically identified keywords in a document.

For both our approaches, we utilize a graph-

Joris Schellekens
Software Engineer
iText

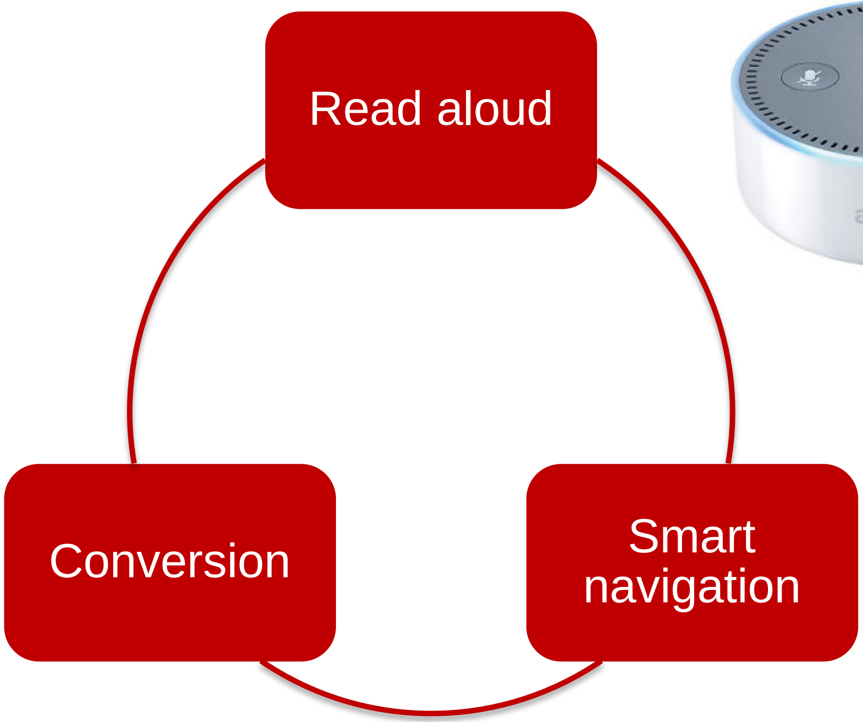
2018-05-14

Responsiveness

- Extreme case:
  - Layout content of A4 PDF on A3 canvas.

Joris Schellekens
Software Engineer
iText

Accessibility



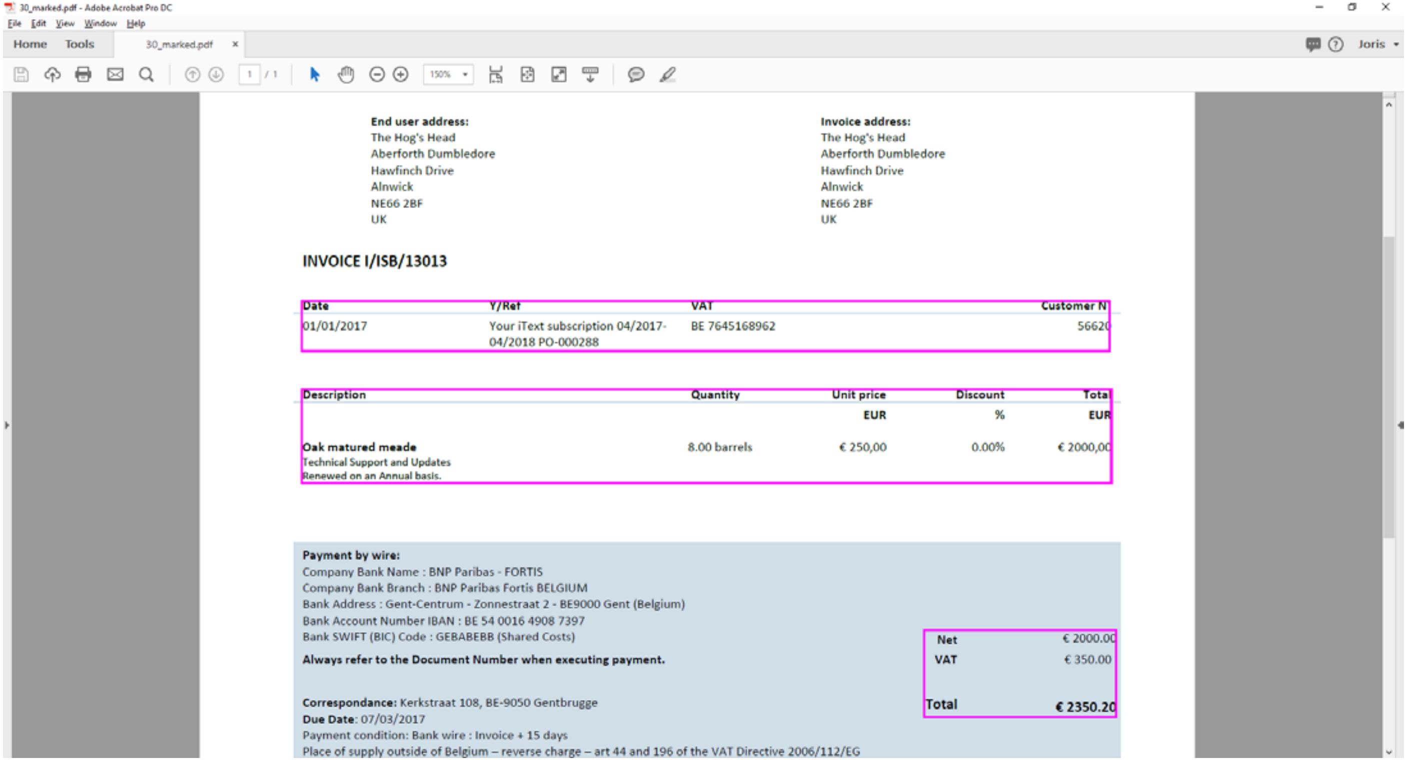
Read aloud

Conversion

Smart navigation

Joris Schellekens
Software Engineer
iText

Data extraction


30_marked.pdf - Adobe Acrobat Pro DC
File Edit View Window Help
Home Tools 30_marked.pdf ×
Joris
1 / 1
150%

End user address:
The Hog's Head
Aberforth Dumbledore
Hawfinch Drive
Alnwick
NE66 2BF
UK

Invoice address:
The Hog's Head
Aberforth Dumbledore
Hawfinch Drive
Alnwick
NE66 2BF
UK

INVOICE I/ISB/13013

Date	Y/Ref	VAT	Customer N
01/01/2017	Your iText subscription 04/2017-04/2018 PO-000288	BE 7645168962	56620

Description	Quantity	Unit price	Discount	Total
		EUR	%	EUR
Oak matured meade Technical Support and Updates Renewed on an Annual basis.	8.00 barrels	€ 250,00	0.00%	€ 2000,00

Payment by wire:
Company Bank Name : BNP Paribas - FORTIS
Company Bank Branch : BNP Paribas Fortis BELGIUM
Bank Address : Gent-Centrum - Zonnestraat 2 - BE9000 Gent (Belgium)
Bank Account Number IBAN : BE 54 0016 4908 7397
Bank SWIFT (BIC) Code : GEBABEBB (Shared Costs)

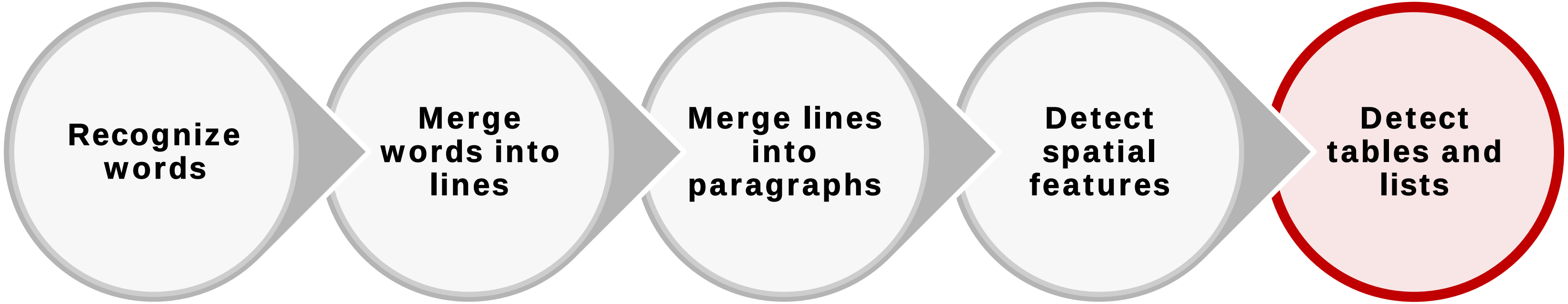
Always refer to the Document Number when executing payment.

Net	€ 2000.00
VAT	€ 350.00
Total	**€ 2350.20**

Correspondance: Kerkstraat 108, BE-9050 Gentbrugge
Due Date: 07/03/2017
Payment condition: Bank wire : Invoice + 15 days
Place of supply outside of Belgium – reverse charge – art 44 and 196 of the VAT Directive 2006/112/EG

iTEXT
Joris Schellekens
Software Engineer
iText

2018-05-14



- Data structures
  - Disjoint set: (Human) logic | Global level | Artificial intelligence
- Benefits:
  - still configurable.
  - able to train network to work well for documents you typically process.
  - ‘soft fail’ mode.

Joris Schellekens
Software Engineer
iText

Thank you! Any questions?

Joris Schellekens
Software Engineer
iText

Get in touch:	joris.schellekens@itextpdf.com
Web site:	www.itextpdf.com
Twitter:	@itext