

Digital Weight Watching: Reconstruction of scanned documents

Maarten Marx and Tim Gielissen
ISLA, University of Amsterdam
Science Park 107 1098 XG Amsterdam, The Netherlands
maartenmarx@uva.nl

ABSTRACT

A web-portal providing access to over 250.000 scanned and OCR'd cultural heritage documents is analyzed. The collection consists of the complete Dutch Hansard from 1917 to 1995. Each web document consists of facsimile images of the original pages plus hidden OCR'd text. The inclusion of images for each page yields large file sizes of which less than 2% is the actual text.

The search user interface of the portal provides poor ranking and not very informative document summaries (snippets). Thus users are instrumental in weeding out non-relevant results and for that have to assess the complete documents. This is a time-consuming and frustrating process because of the long download and processing times of the large files. Instead of using the complete document for relevance assessment we propose to use extended document summaries based on the OCR'd text alone. We describe three kinds of summaries of increasing complexity. We elaborate on the most complex summary, a reconstruction of the original document from a purely semantic representation. Evaluation on the Dutch data set shows that documents become two orders of magnitude smaller and still resemble the original to a high degree. In addition they are easier to speed read and evaluate for relevance, due to added hyperlinks and a presentation optimized for reading from a terminal.

We describe the reconstruction process and evaluate the costs, the benefits and the quality.

Categories and Subject Descriptors

H.4.m [Information Systems]: Miscellaneous

Keywords

XML, Information Extraction, Scanned Documents

1. INTRODUCTION

This paper addresses the fact that scanned and OCR'd documents tend to be very large in file size because of the in-

clusion of facsimile¹ images. This makes them expensive to store, expensive to serve over the internet, and cumbersome to handle by users. We present a case-study in which we extracted the content and the structure from a large digitized corpus and reconstructed the documents from scratch “as new”. This yielded much smaller (less than 1% of the original size when stored in gzipped XML, 1.5% when stored as PDF) and far better readable documents which in addition are easier to browse because of added hyperlinked structure.

We present the data, describe our techniques, evaluate the results and generalize them to other cases.

The novelty of our work is located in the way we reconstruct the original files. The reconstruction is based on an almost (we only kept the original separation of the text into pages) purely semantical description of the original data. The reconstruction is done using only two well-described declarative programming languages: XSLT 2.0 and L^AT_EX.

Outline.

The rest of this introduction contains the actual problem we are addressing, the use case we present and related work. Section 2 describes the data; Section 3 analyzes the current search user interface. Section 4 contains the two transformation processes: structure extraction and document reconstruction. We evaluate the quality of these two transformations in Section 5, and conclude in Section 6.

Searching and browsing large collections of OCR'd data.

Much, in particular qualitative, research involving digital documents is done “by hand”. This is especially true in the social sciences and in the humanities. Current large-scale digitization efforts of important collections make more data much more easily available, and offer new technologies, most importantly keyword search over the full (often OCR'd) text.

Large collections are now available at the desktop. A common scientific method of enquiry is searching and browsing through texts and collections of texts. With a good search engine and fast internet access it is easy to go through a large number of documents in a short time, in order to get an overview of the data, or to find specific information.

This process works smoothly if documents are relatively small in size and can be handled by one simple and fast program. But the technical nature of the documents often prohibit fast workflows. As an example, it may very well

¹By a facsimile image, we mean a document that visually resembles the original.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AND2009 Barcelona
Copyright 2009 ACM ...\$5.00.

take more time to open a PowerPoint presentation then to decide that it can be discarded as non-relevant. Unfortunately, digitization processes create large documents, even if the originals are relatively simple text documents. If the original layout and look-and-feel is important, or if mistakes caused by OCR are unwanted, scanned documents must be presented using facsimiles. If they have to be readable these images get very large.

One of the most pressing problems arising from the large file sizes are long download times. For instance, with a fast (12 Mbit/second) internet connection it takes in the optimal case 10 seconds to download an average document (16 Megabyte) from our collection. In reality this speed is often up to 20 times slower. Such long download times prohibit natural workflows.

These long waiting times become a source of frustration when the downloaded document was ranked high by the search engine but found out to be not relevant. Almost all search engines return a list of hits consisting of small document summaries (“snippets”) with a link to the actual document. We propose to add an intermediate level between the snippet and the document, very much like the “Quick Look” button in Apple’s Mail program. This level then is a document summary consisting of the complete OCR’ed text specially presented for quick human scanning. Only when the document turns out to be relevant the large file with all facsimile images needs to be downloaded.

Use case: Dutch parliamentary proceedings.

The Netherlands have parliamentary proceedings since 1814. From 1995 these are available as digitally produced PDF files. The Dutch Royal Library together with the Dutch parliament have scanned and OCR’ed all proceedings from 1814 until 1995 and have thus created the first complete copy of these proceedings. The collection consists of 2.5 million pages which physically span 150 meters. The digital copy needs approximately 30 Terra Byte storage space.

The proceedings are available online at

<http://www.statengeneraaldigitaal.nl>.

Each document is a combination of files: metadata in XML, a JPEG image for each page, the OCR’ed text in an XML wrapper, and an MPEG21-DIDL file describing the connections between all these files [5]. Each document is also available as a PDF file which combines all this information.

In November 2009, all documents from 1917–1995 are available. The complete corpus is planned to be online in the fall of 2010.

We will refer to the collection as the *SGD* corpus.

Related work.

Within digital curation research XML is seen as an important data format for storing data for long periods of time. The National Archives of Australia intend to store all they have in XML and developed a software tool for this, XENA (<http://xena.sourceforge.net>). For the use of XML as a format for governmental documents, see [25] and the publications of the W3C eGov working group [3, 4]. Another development in this direction is the UVC (Universal Virtual Computer) developed by IBM and the Dutch Royal Library [27].

The transformation of scanned and OCR’ed documents into XML is not the topic of this paper and has been de-

scribed in [9]. This transformation is a text extraction task (as in the TAC and MUC conferences [1, 2]), combined with a document layout analysis task [6]. The extra step we take in this paper –automatically reconstructing the original document from the XML-version as a lightweight truly electronic file– is, up to our knowledge, not yet described in the literature. In this paper, we focus on the ‘look-and-feel’ of the documents. An extensive body of research on sustainable digital preservation of properties exists under the name of *significant properties* [17].

2. DESCRIPTION OF THE DATA

Our experiments are based on six years of proceedings of plenary meetings of the Dutch Parliament (both Houses), from 1980 to 1985. Each document contains the meeting notes of one day. Table 1 lists statistics on the sizes.

On average one document contains 51 thousand words, is 50 pages long and has a file size of 16.5 Megabyte. Because each document represents the meeting notes of one complete day, on an average day in Dutch parliament some 50 thousand words are officially spoken.

The largest document is 49 MB (151 thousand words) and the smallest is just 1 page, with 382 words. At the meeting of this one page document less than half of the Dutch MP’s were present and then by law the meeting cannot start.

Figure 1 shows the facsimile of a typical page.

In Section 4 we describe how we transform the scanned and OCR’ed PDF documents into XML. This leads to a large reduction in size. Whereas the original set of 843 documents was 13,62GB, the same set in XML takes only 295 MB, a reduction of 2 orders of magnitude. A further reduction to 88.3 MB can be obtained using gzip, yielding a total reduction of 99,4%. Note that zipping the original PDF files has hardly any effect (less than 5% reduction in size). The reconstructed PDF files takes 205 MB to store.

3. ANALYSIS OF THE SEARCH INTERFACE

We provide an analysis of the search user interface at <http://www.statengeneraaldigitaal.nl>, the portal that now serves the Dutch scanned proceedings. Unfortunately, the ranking of the results is rather poor. Thus the user is instrumental in weeding out non-relevant documents. We analyse the tools provided by the search user interface which help in this task. Based on the recommendations in [12] we give a list of positive and negative aspects of the interface.

The search-interface has a standard three-layer architecture: a (optionally extended) search page which leads to a result page (often called “SERP”) with “ten blue links” which lead to the actual documents. We discuss each layer.

The extended search page facilitates formulation of precise queries by offering selections on three natural metadata for this collection. One can select the House, make a date-restriction and choose among three document types (meeting notes, written questions and answers, and documents sent to the Parliament). These three attributes come back in various parts of the search interface. For each document type, specific further restrictions are possible.

The result page shows the first ten hits with the option to view the next ten results, as usual. Besides the ten results, there is aggregation information and faceted search machinery [11]. The page shows the total number of hits and the distribution of the hits over the values of the three main

year	Number of documents	Total file size	Number of words	Number of pages
1980	143	2,52 GB	7.943.904	7.709
1981	124	1,76 GB	5.613.432	5.405
1982	133	1,76 GB	5.552.685	5.544
1983	150	2,50 GB	7.714.903	7.612
1984	147	2,54 GB	7.870.318	7.750
1985	146	2,56 GB	8.251.097	8.081
Total	843	13,62GB	42.946.339	42.101

Table 1: Description of the corpus used for experiments: proceedings of plenary meetings of the Dutch Parliament (both Houses), from 1980 to 1985.

metadata: document-type (3 values), House (4 values) and parliamentary year (at present over 70 values). Clicking on an attribute–value combination restricts the search and the aggregates are recomputed.

[12] calls the hits “document surrogates” in order to highlight their function: help the user to understand the primary object. In particular:

The quality of the document surrogate has a strong effect on the ability of the searcher to judge the relevance of the document.

The document surrogates of SGD consist of three pieces of information:

- A title with a hyperlink to the document. The title provides the values on the three main metadata attributes plus some additional information (e.g. date in case of proceedings). It is query-independent.
- The number of pages of the document.
- An extract from the retrieved document (“snippet”). The snippet consists of pieces of text taken from various parts of the document and concatenated. These snippets appear mostly query independent.

For several queries the query term does not occur on the first result page. On a sample of 617 snippets less than one third contained the query term. The snippets are rather long: on average 383 characters (N=612). This is more than twice the length of the snippets at Google.

Query terms are not highlighted in the snippets.

We now discuss the presentation of the documents. SGD is a vertical search engine only serving its own documents. It thus can determine their presentation. Each document is shown in a special viewer which allows for some sort of entry-point retrieval [26] system: the user is brought to the first page in the document in which the search term occurs. From there, the user can go backwards and forward through the pages and jump to following and preceding entry-points. The user views a JPEG image of the page on which the search term is highlighted. It is also possible to view the OCRed text of the page as an HTML document.

3.1 Evaluation

We evaluate the three layers of the search system with respect to the ease and speed in which the following task can be performed:

From the list of retrieved documents, find those that are relevant.

We first list aspects which have a positive effect on the task, followed by those which have a negative effect. We conclude with proposals for improvement. These will be further developed and evaluated in the rest of the paper.

Positive features.

- (++) The extended search interface makes stating precise queries possible and easy. This is especially useful when the user has already good knowledge about the desired document(s) (as in known-item search).
- (+) The faceted search machinery is useful for browsing the collection and quickly zooming in on parts of the results. The temporal facet suffers from a large amount of values, overcrowding the page and resulting in flat frequency distributions. Following [11] a hierarchical faceted design is to be preferred. One level above parliamentary years are the legislative periods (in The Netherlands maximally 4 years). These can be grouped into political eras. For the period 1918–1995, this would yield six eras each having between 2 and 8 legislative periods².
- (++) Entry point retrieval with search term highlighting provides direct access to the potentially relevant parts in the often very long documents. With frequent words in long documents it would be desirable to have a ranking of relevant pages within one document as well [26, 19].

Negative features.

- (--) There is no apparent ranking of the search results. It is not possible for users to specify orderings or groupings on metadata.
- (--) The document summaries are for the most part query independent. This makes them basically useless for relevance assessment of the underlying document [7]. That paper also suggests that snippets should pass a simple readability test. The long SGD snippets consisting of sentences from different parts of the document are often unintelligible.
- (---) Retrieving a document takes a long (download) time. Browsing through the document takes a long (download) time. Reading the scanned images is difficult and hence time consuming because of small

²<http://www.parlement.com/>: Kabinetten per tijdvak

Den Uyl e.a.

Nogmaals, ik zie volkomen het belang van de b.t.w. in, ik wil op geen enkele manier daarop afdringen, maar wij komen m.i. in een volstrekt onmogelijke situatie als wij deze zaak nu gaan behandelen zonder dat de Kamer en het land weten waaraan zij ten aanzien van het loonbeleid toe zijn.

De Voorzitter: De heer Den Uyl stelt voor, eerst de nota inzake het te voeren loon- en werkgelegenheidsbeleid en daarna het wetsontwerp inzake de b.t.w. te behandelen.

Naar mij blijkt, wordt dit voorstel voldoende ondersteund.

De heer Schmelzer (K.V.P.): Mijnheer de Voorzitter! Ik zal niet zeggen dat de Kamer voor een gemakkelijke taak staat - dat wisten wij allemaal - maar ik zou desalniettemin uw voorstel willen ondersteunen. De derde nota van wijzigingen, die ons zojuist heeft bereikt, is voor een niet onbelangrijk deel een antwoord op vragen, die althans van onze kant in het debat over de b.t.w. zonden worden gesteld. Onze fractie ziet bepaald wel kans om op een verantwoorde wijze een oordeel ook daarover in de ons toech niet zo krap toegemeten tijd, die ons rest voor de behandeling van de b.t.w., te vormen.

De heer Bakker (C.P.N.): U hebt verleden week ook al met de Regering kunnen spreken.

De heer Schmelzer (K.V.P.): Op zich zelf is het heel nuttig eens een keer met de Regering te spreken. Dat is ook wel vaker gebeurd. Er zijn zelfs voorstanders van een nog veel nauwer contact tussen Regering en leden van het parlement dan de voorstanders van het dualisme nog wel eens ten beste geven.

De heer Den Uyl (P.v.d.A.): U moet nu wel oordelen over de vraag of de gehele Kamer in de positie is om op een verantwoorde wijze het ontwerp te behandelen. Dat moet u nu beoordelen.

De heer Schmelzer (K.V.P.): Ja, maar dat staat geheel buiten enig contact van enige Minister met enig lid van mijn fractie.

De heer Den Uyl (P.v.d.A.): Dat is theorie.

De heer Schmelzer (K.V.P.): Dat geeft volstrekt niet meer informatie of inzicht in oordeelsvorming dan wanneer dat niet zou hebben plaatsgevonden.

De heer Berg (P.v.d.A.): Iedereen kon van deze Regering wel vermoeden, dat er zo iets zou komen.

De heer Schmelzer (K.V.P.): Wat zou komen?

De heer Berg (P.v.d.A.): Die derde nota van wijzigingen.

De heer Schmelzer (K.V.P.): Ik heb wel meegemaakt van andere kabinetten, dat er nog tijdens de behandeling nota's van wijzigingen kwamen. Wij menen, dat het zeer belangrijke vraagstuk van de sociale compensatie - daar ging het de heer Den Uyl voor een groot deel om - bepaald uit dit debat over de b.t.w. zal moeten komen op een verantwoorde wijze, want ook wij hebben daaraan de grootste betekenis. Intussen heeft het mij wel verbaasd, dat uitgekend de heer Den Uyl om middel van de behandeling van de b.t.w. vraagt, want ik had begrepen uit een televisiepraatje van de heer Berg, dat de fractie van de P.v.d.A. haar standpunt al had bepaald.

Nu het tweede punt van de heer Den Uyl, nl. dat hij niet wil beslissen over de b.t.w., voordat over de loon- en werkgelegenheidsbeleid is beslist. Ik meen, dat uw voorstel, mijnheer de Voorzitter, aan die zorg juist goed tegel-

Zitting 1967-1968

Schmelzer e.a.

moet komt, want wanneer wij dinsdag een loonbeleid zouden houden en wij zouden woensdag, eventueel volgende dagen hierover verder spreken - ik heb begrepen, dat het niet ondenkbaar is, dat wij zelfs begin juni nog stemmingen moeten houden - dan is het heel wel mogelijk, bij ons eindoordeel volledig mee in de koop te nemen de uitkomsten van het debat over loon- en werkgelegenheid. Op die grond wil ik dus ook uw voorstel ondersteunen.

De heer De Goele (D'66): Mijnheer de Voorzitter! Wat ons vandaag overkomt, is een herhaling van wat gebeurde in november jl. bij het belastingdebat. U herinnert zich, dat ik het toen een weinig slagenige benadering van de zijde van de Regering ten opzichte van het parlement vond, dat zelfs tijdens - niet vóór - de beradslaging een nota verscheen om ons meder te informeren omtrent het punt van de loon- en werkgelegenheid, dat toen aan de orde was. Ik heb toen gesteld, dat, als het parlement zich zelf wilde restreteren, het die behandeling zo niet mocht laten doorgaan. Ik heb toen een oordeel gegeven om dat stuk van de behandeling los te koppelen totdat wij gelegenheid zouden hebben gehad, dat madere stuk te bespreken. Ik kan niet nalaten er even op te wijzen, dat toen ook de woordvoerder van de P.v.d.A., de heer Van den Bergh, zich daarbij niet heeft aangesloten. Ik vind het nu wat vreemd, dat, hoewel de heer Den Uyl volstrekt gelijk heeft met zijn benadering vandaag, ik die het vorige jaar in die zin heb gemist. Niettemin vind ik het juist wat de heer Den Uyl heeft gezegd. Ik protesteer scherp tegen deze behandeling van de zijde van de Regering ten opzichte van de Kamer om dit soort essentiële informatie om eerst nu te doen toekomen. Mijn benadering van het voorstel van de heer Den Uyl hangt af van het volgende. Gelet op uw voorstel, mijnheer de Voorzitter, waartoe ik geneigd ben om erin mee te gaan, dus om dinsdag het loonbeleid te houden, wil ik u vragen of in ieder geval, wanneer er replieken worden gehouden - morgen of op een later tijdstip - zoveel ruimte kan worden geschapen, dat wij terzake over deze andere zaken kunnen spreken en dat er ook een mogelijkheid voor een derde termijn nu reeds wordt geopend, zodat wij deze zaken zo goed kunnen bespreken, dat wij terzake de beradslaging niet behoeven op te schorten. Dat zou onnodig tijdsverlies betekenen. Ik ondersteun dus uw voorstel - ik durf niet te zeggen: op voorwaarde dat - in de loop, dat er bij de replieken en door het aanwezig van een derde termijn zoveel ruimte wordt geschapen, dat wij deugdelijker over deze informatie kunnen spreken dan ons nu mogelijk is.

De Voorzitter: Ik wil ter nadere informatie van de leden mededelen dat ik mij de gang van zaken als volgt heb voorgesteld. Vandaag zal aanvangende de behandeling van het wetsontwerp inzake de b.t.w., zoals op de agenda is vermeld. Dit betekent, dat de Kamer ongeveer 9 uur zal spreken. Het kan ook dicht bij de 10 uur liggen. De Kamer zal morgen in de namiddag haar bespreking in eerste termijn kunnen beëindigen. Na een pauze zal dan de Regering antwoorden.

Ik kan nu nog niet zeggen, of de Kamer moegenavond op het antwoord van de Regering zal kunnen replieken en of de Regering kan dupliceeren. Is dit laatste niet mogelijk, dan zullen de replieken en dupliceeren, wanneer mijn voorstel door de Kamer wordt aanvaard, pas nu de behandeling van de nota inzake het te voeren loon- en werkgelegenheidsbeleid kunnen plaatsvinden, dus op zijn vroegst woensdag, tenzij de suggestie, die de heer Den Uyl deed, wordt gevolgd en wij a.s. maandag gaan vergaderen. Doeze zaak stel ik liever later aan de orde, mid over het principe een beslisning is genomen.

Het voorstel van de heer Den Uyl strekt ertoe, deze week niet te beginnen met de behandeling van de ontwerp³⁾ Wet op de inkomstenbelasting 1968, doch eerst te waken op dinsdag 23 mei a.s. - respectievelijk maandag 27 mei a.s. - wanneer hierover een beslissing is genomen - het debat over de nota inzake het te voeren loon- en werkgelegenheidsbeleid te houden en daarna

TWEDE KAMER

3.2 Extended document summaries

We propose to add a fourth layer to the search architecture in between the result page with the ten snippets and the actual documents. This functionality is comparable to the "Quick Look" button in Apple's Mail program. The fourth layer contains an approximation of the original document, based on the OCR'd text and the original layout. Its sole purpose is to provide a fast interface to the complete document in order to make a quick relevance assessment.

We discuss and evaluate three systems for creating these document summaries. They differ in the amount of semantics that is explicit in the markup.

1. The simplest transformation preserves physical layout using absolute positions of each line of text and font information. This can be achieved efficiently and effectively with the `pdftohtml` package³. The only structural element that is preserved is the page. The size of resulting documents is on average 9% of the original size.
2. [20] describes a system for transforming PDF files into an XML format in which page, paragraph, page number, page header and page footer information is preserved. The program also attempts to detect the reading order of the multi-column input and outputs the text in reading order. We evaluated its effectiveness on 11 randomly chosen proceedings files from the period 1928-1966. The results are in Table 2. The SGD system also provides a separation of the pages into paragraphs. On the same data that system has an accuracy of 44%. The size of resulting documents is on average 6% of the original size. The difference with the previous transformation is due to the fact that here position information is only retained for each paragraph, not for each line of text.
3. The system described in this paper. This system first does an extensive text analysis and produces semantically rich XML without any layout information except for paging. From that XML file a uniform looking PDF file according to the style of one specific period (the 80ties) is created. The size of these PDF's are 1.5% of the original size.

We now describe this last system.

4. DESCRIPTION OF THE TRANSFORMATION

The transformation of a scanned and OCR'd PDF document into a PDF document that closely resembles the original, but without the facsimile images, involves two steps. First, the structure that is implicit in the document is made explicit using a variety of text extraction techniques [9]. Next, the resulting XML document is transformed into a PDF document without facsimile images using XSLT and \LaTeX . In this section we describe both steps.

4.1 Making structure explicit: from PDF to XML

All documents contain structure. Often we can easily recognize titles, paragraphs and page numbers on the basis of

³<http://pdftohtml.sourceforge.net/>

Figure 1: A typical page of the Dutch parliamentary proceedings. Page 2077 of the meeting of May 21, 1968. Available at http://resources.sgd.kb.nl/SGD/19671968/PDF/SGD_19671968_0000410.pdf (22MB).

font-size, unfamiliar fonts, poor quality of the scan, and layout designed for paper printing.

Possible improvements.

The improvements we suggest are based on the premise that we cannot change the ranking except for orderings on metadata. Thus the user still has to do most of the relevance assessments. The above discussion yields three clear cases for improvement:

1. Improve aggregated search results and facets [22].
2. Offer (reverse) chronological ranking, possibly combined with result grouping [12].
3. Offer query dependent document summaries with keyword highlighting [7].
4. Drastically improve download times and fast browsability of documents.

Of these improvements only the last is specific for noisy-data collections. The only way to drastically reduce download times is to postpone serving large and images as long as possible.

Semantic feature	Accuracy
page header	100%
page footer	89%
reading order	67%
paragraphs	91%

Table 2: Evaluation of the quality of the transformation from PDF to XML described in [20]. Accuracy measures the number of times a feature is correctly extracted. For reading order, this means that the XML document order of the text on one complete page is the reading order of a the PDF page (usually in multiple columns), except for special text-blobs like page-headers, footers, captions, etc.

their layout (e.g., position or size) and their content (e.g., a number) for example. This structure is often not explicit in digital form, especially not when the document is scanned and OCRed. If the structure of documents in a corpus is standardized to some degree, the structure can be made explicit in digital form automatically [8].

The Dutch parliamentary proceedings shows this kind of standardization of document structure. All elements of a proceeding, for example topics and speakers, are represented in their own distinct way. This enables the automatic recognition of these elements. We programmed the text and structure extraction as an Extract-Transfer-Load process [23] consisting of eight steps.

First we extract the text from the PDF using the open source program `pdftohtml` with the `-xml` option. This yields an XML file with for each line of text four coordinates which indicate the bounding box of that text. Multiple columns are detected and preserved. Some font and layout information is preserved but not all. The XML structure is simple and flat:

```

root   → (page)*
page   → (text)*
text   → (#PCDATA,b,i,span)*

```

The second step involves cleaning the output from `pdftohtml`. We ensure that the output is well-formed XML and we solve problems with diacritics. In this step, we also fix the most common OCR errors. We found one specific error quite often: the OCR inserts a space before the last letter of a word. For example, the token `wij` is OCRed as `wi j`. A regular expression designed to fix this problem matches 3.1% of all the lines in the text corpus (i.e., one line in one column in the original PDF as in Figure 1). A sample of 100 matches taken at random indicated that this regular expression fixes the problem in 93% of the cases and makes it worse in 7% of the cases (by incorrectly adding a single letter to the word in front of it).

The values indicating the position and size of the bounding box of the text are normalized in the third step as we found they can differ among different devices.

In the fourth step, we analyze the document’s layout. The margins, the number of columns and the header and footer are detected and marked. For this, we use the position of the text elements and the (deducted) position of the whitespace. Using this information, we sort the text of the body (i.e., not belonging to the header or the footer) in reading order in the fifth step.

During the sixth and seventh step, different markers are placed in the text to signal the start of different elements in the document. In the sixth step we place markers indicating the position of text elements in the document. We place markers on places where paragraphs begin (they are indented), where there is whitespace and when a new column starts. This information is used in the seventh step where markers are placed based on the content of the document. In the seventh step we use regular expressions to recognize standardized structure. The start of a statement for example, is represented as follows:

Mevrouw **Swenker** (VVD):

This adheres to the following structure: title, last name, party name within parenthesis, colon. This is then followed by the statement this person made. The start of a statement can only begin after a whiteline marker or the start of a new column. So in our XML, we convert this to:

```
<speechstart speaker='Swenker' party='VVD' ... />
```

with the `...` containing additional information.

We now have an XML document that is flat and contains markers that mark the beginning of structural elements, but not the end. In the last step we replace the markers by XML tags that enclose the entire element. This is done by performing a cascade of groupings starting with the elements which need to be most deeply nested: the paragraphs `p`. XSLT 2.0 has a useful command for this task: `xsl:for-each-group`.

The result is an XML file with the same text as the original document but with explicit structure. The file is valid with respect to a rich Relax NG schema, constraining both the structure of the XML-tree as well as the values of many attributes. The structure can be used for many purposes, e.g. to analyse the structure of the debates [14].

4.2 Reconstruction of the originals: from XML to PDF

Because the structural elements of the documents are made explicit in the XML file, it is possible to reconstruct the original PDF with high resemblance. We use a combination of XSLT 2.0 and \LaTeX for this process. We created a XSLT stylesheet that transforms the XML file described in the previous subsection into a \LaTeX file. We briefly describe this transformation.

First, the stylesheet writes a general preamble that loads all necessary packages and specifies layout information for the document. Some values are copied from the XML file, like the value that indicates the number of columns.

After the preamble, the document itself begins. For every element specified in the Relax NG schema, we defined a template (for more information about the schema, see [9]). These templates are nested according to the structure of the proceeding. First we have a template for the topics, the highest level of the structure of the proceedings. The stylesheet writes the necessary layout information for the topic and then places the text from the XML file in the \LaTeX file. Next, it applies all the templates for the elements within the topic. This way, we cycle through all the elements in the deep structure of the XML and create the \LaTeX file step by step. If the stylesheet encounters a pagebreak, it

redefines the page style (including header and footer) for the next page.

When the \LaTeX file is created, the next step is to create the PDF document with `pdflatex`. Creating the \LaTeX file with XSLT and compiling the PDF file takes about one to two seconds, depending on the size of the file.

An alternative approach for reconstructing the files is to use XSL-FO to produce a PDF document directly with XSLT. We used \LaTeX because it appeared to be easier to obtain fast results.

5. EVALUATION

We give a technical, an information-theoretic and an economic evaluation. The technical evaluation describes the reduction in file sizes obtained and the processing times needed for the reduction. In the economic evaluation we compare the efforts invested in creating the transformation scripts with the benefits of smaller files and explicit markup information. In the information-theoretic evaluation we look at the quality of the transformations: we evaluate whether information was lost or distorted.

5.1 Benefits of the reconstruction

Size reduction.

Our first goal was the reduce the file sizes. This was achieved with a reduction of 2 orders of magnitude. Table 3 lists the results.

	Size in MB	% of original
Original corpus	13.620	
Reconstructed PDF	205	1.5
gzipped XML	88	0.6

Table 3: Total file sizes of the test corpus described in Table 1.

Processing times.

Transforming the whole corpus of 13,62 GB into reconstructed PDF's took 25 hours and 50 minutes, an average of 1.8 minutes per document. The table below shows the percentage of the total time that was spent on each of the elements of the entire pipeline, which were described in the section: "Description of the transformation".

<code>pdftohtml</code>	PDF-2-XML (without <code>pdftohtml</code>)	XML-2-PDF
40%	58%	2%

Table 4 shows the processing speeds of the two main steps of the transformation in pages per minute. (Recall that a typical document has about 50 pages.) These times were measured on an Apple MacBook with 2.4 GHz CPU running the OS X 10.5.6 operating system.

Both transformations can be done offline and we need only store the reconstructed PDF on the webserver. The transformation from XML to PDF is fast enough to perform at query time. Recall that an average document is 50 pages. Thus this can be transformed from a gzipped XML file into PDF in two seconds. This transformation can be further optimized as a streaming transformation [18].

PDF-2-XML	XML-2-PDF
27,6 pages/minute	1503,6 pages/minute

Table 4: Processing speeds of the two main transformations.

Digital sustainability.

We aimed to make our transformations in such a way that they can be performed again after minimally 100 years with relative ease. Thus we wanted a minimal dependency on specific software and hardware, and transparent and reproducible transformations [10]. Our goal was to have all steps of the transformation written in a declarative language with a precisely defined software-independent semantics. XPath 2.0 and XSLT 2.0 meet these standards [15, 16].

For the conversion from PDF to XML we reached this goal except for the first step of the transformation and the final validity check. In the first step we used `pdftohtml-xml` (<http://pdftohtml.sourceforge.net>) to transform the PDF to XML. Unfortunately this program may produce non-well formed XML and it has problems with certain diacritics. We repaired these with a `perl` script and checked for XML well-formedness with `xmlint`. This was also used in the final step to check validity with respect to the schema specified in Relax NG.

The conversion from XML back to PDF is done by an XSLT 2.0 script which produces a \LaTeX source file.

Economics.

We calculate what can be saved using the file size reduction based on the prices for storage and bandwidth set by Amazon, <http://aws.amazon.com/s3/#pricing> at the time of writing (Spring 2009). The fixed costs for storing the test collection are still very modest: \$ 30 per year. The variable costs are determined by the number of users and the number of documents they want to see: downloading at Amazon costs \$0.17 per GB. Table 5 lists the costs per year for 3 user models, and three ways of serving: the original collection (as it is now served at <http://www.statengeneraaldigitaal.nl>), serving the transformed PDF's (as described in this paper), or serving the gzipped XML files only. The monetary savings are two orders of magnitude.

It is hard to quantify the amount of time needed to create the transformation scripts. This depends very much on the complexity and the regularity of the documents and the use of (commercial) ETL tools [23].

5.2 Quality of the reconstruction

Quality of the data: from PDF to XML.

We now evaluate the transformation from the original PDF file to XML, which was described in 4.1. For each part in the proceedings that we wanted to extract and mark up by XML tags, we scored whether the start- and end-tags were placed correctly. We evaluated two complete days (50 pages). Table 6 shows the percentage of correctly marked elements for 7 typographical features.

These are promising initial scores. The semantically im-

Daily use	Original Collection	Transformed Collection	Transformed Collection Client side XML-2-PDF
10 users 10 docs each	\$100	\$2	\$0.64
50 users 10 docs each	\$500	\$11	\$3
100 users 10 docs each	\$1000	\$21	\$6

Table 5: Costs per year in dollars (rounded) for downloading documents stored at Amazon (prices April 2009). Average document sizes are used.

Feature	Score	Comments
Topics	77,8%	All recognized, but in 22.2% included too much text
Blocks	100%	
Speakers	88.7%	Caused by OCR errors
Paragraphs	93.5%	
Header	91.5%	Caused by OCR errors
Footer	92.5%	Caused by OCR errors
Stage directions	73.5%	

Table 6: Percentage correctly extracted structural elements.

portant XML elements, topic and block, were all recognized. The topic description sometimes included too much text (32.2% of the topics), but they were correctly marked as topics. Most of our mistakes were caused by OCR errors which did not let our extraction rules fire. E.g. for recognizing the start of a speech element (and the title, name and party of the speaker) we use the pattern `[title] [name] [(party)] [:]` as in `[De heer] [Van der Spek] [(PSP)] [:]`.

But sometimes this string is wrongly tokenized by the OCR as `DeheerVanderSpek(PSP) :`. OCR mistake repairing, using e.g. the TICL technique [24], will improve our scores considerably.

Quality of the look and feel: from XML to PDF.

Our goal was to keep the look and feel of the structure of the documents. This was achieved with great success, see Figure 2 for an indication of the results.

We wanted to preserve the layout of each page as much as possible, but not be overly restrictive. For instance, old and new pages should contain exactly the same characters in the same order, but the words may be broken differently over the lines of texts. Table 7 contains the most important typographical features of these documents and an assessment of the quality of our reconstruction. We tested the similarity each time on ten randomly chosen documents. We score whether the element was visually indistinguishable throughout the complete document. Figure 3 contains examples of visual (in)distinguishability. In the top of the Figure we see two footers (on the left the original). These were scored as being “the same”. Below that we see twice the wording of a *motie* (the original is on the left). The header *Motie* between the two horizontal lines indicates an important structural feature of the text, which is not present in the reconstruction. For that reason these two are scored as different.

An error analysis indicated that almost all mistakes are due to OCR errors or to inconsistent layout in the original.

5.3 Additional advantages

23 februari 1999 EK 20	23 februari 1999 EK 20
De voorzitter: De motie-Witteveen-Hevinga (26570, nr. 3) is in die zin gewijzigd, dat zij thans luidt:	De voorzitter: De motie-WitteveenHevinga (26570, nr.3)isindie zin gewijzigd, dat zij thans luidt: Motie
Motie	De Kamer,
De Kamer,	gehoord de beraadslaging,
gehoord de beraadslaging,	overwegende, dat er op provinciaal c.q.

Figure 3: Example of two features scored as the same (top) and two scored as different (bottom). At the left is the original, at the right the reconstructed document.

Feature	Score
Header	9/10
Footer	7/10
Individual speeches	6/7
Stage directions	
- Members present	4/4
- Start of agenda topics	6/10

Table 7: Score of visual similarity on 10 typographical features. The score k/m indicates that on k of the m documents containing these elements, all elements in that document were visually the same as in the original.

Having the data in XML creates numerous analysis possibilities which are impossible to do automatically from the original PDF files. [19] and [14] give a number of examples related to search, in particular focused retrieval and result aggregation. Here we look at several possibilities that become available when making a new PDF file with L^AT_EX from XML input.

Table of contents The Dutch proceedings do not contain a table of contents. The table of contents acts as an agenda of the meeting and hence is very valuable. It can be created automatically from the extracted structure and accurately reflects the order of the meeting.

Select from PDF Users can select text from the PDF file.

What you see is what you get The PDF files available at <http://statengeneraaldigitaal.nl> have a logic which is not obvious to average users. If opened in a PDF reader, one looks at the scanned images. But it is possible to search with Control F, and that yields highlighted hits. However, the search takes place in the OCRred text which may contain errors. Thus words which occur in the text may not be found due to OCR-errors. This can be confusing for users. Also, seeing



Figure 2: Similarity of layout between original (right) and our copy (left).

the OCR errors gives users the opportunity to broaden their search terms and still retrieve what they look for.

Wikification and hyperlinking Names of persons who speak may be hyperlinked to pages with their biographical information [21]. References to other parliamentary documents may be hyperlinked to these documents.

Back of the book index Using machine learning and keyword extraction techniques [13] useful index terms can be extracted and indicated in the running text. Creation of a back of the book index is then automatic using L^AT_EX's `makeindex` command.

6. CONCLUSION

We have shown that reconstructing PDF documents from scanned and OCR'd data is feasible and leads to size reductions of two orders of magnitude. The quality of the reconstructed PDF files is very good, and in several aspects better than the original. The most important gain of this exercise is the reduction in download time from unacceptably slow to instantaneous. For instance, with a fast (12 Mbit/second) internet connection it takes in the optimal case 10 seconds to download an average document (16 Mb) from our collection. In reality this speed is often up to 20 times slower. But the achieved size reduction to just over 1% makes downloading immediate, even in unfavourable cases.

Our sample of 6 years seems to be representative of the last 80 years of Dutch data, in terms of layout complexity and noisiness of the OCR data⁴. Preliminary investigations abroad (Belgium, Germany) show that our findings

⁴We had access to the data from 1960–1995 and from 1930 and 1931.

generalize to other parliamentary proceedings corpora. Interesting directions of future research are to exploit the rich structural semantics of these documents in ways described in Section 5.3.

Acknowledgements.

Maarten Marx acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the FET-Open grant agreement FOX, number FP7-ICT-233599.

7. REFERENCES

- [1] *Message Understanding Conference Proceedings MUC-7*. National Institute of Standards and Technology (NIST) Gaithersburg, Maryland, USA, 1997.
- [2] *Proceedings of the First Text Analysis Conference (TAC 2008)*. National Institute of Standards and Technology (NIST) Gaithersburg, Maryland, USA, 2008.
- [3] J. Alonso et al. Improving access to government through better use of the web. W3C Interest Group Note 12 May 2009 <http://www.w3.org/TR/egov-improving/>, May 2009.
- [4] D. Bennet and A. Harvey. Publishing open government data (W3C Working Draft 8 September 2009). <http://www.w3.org/TR/gov-data/>, 2009.
- [5] Koninklijke Bibliotheek. Staten-generaal digitaal. <http://www.statengeneraaldigitaal.nl/backgrounds.html>, 2009.
- [6] Th. Breuel. High performance document layout analysis. In D. Doermann, editor, *Proceedings 2003 Symposium on Document Image Understanding*

- Technology*, pages 209–218, 2003.
- [7] Ch. Clarke, E. Agichtein, S. Dumais, and R. White. The influence of caption features on clickthrough patterns in web search. In *Proceedings SIGIR '07*, pages 135–142, 2007.
- [8] A. Doan, R. Ramakrishnan, and S. Vaithyanathan. Managing information extraction: state of the art and research directions. In *Proceedings SIGMOD '06*, pages 799–800, 2006.
- [9] T. Gielissen and M. Marx. Exemelification of parliamentary debates. In *Proceedings of the 9th Dutch-Belgian Information Retrieval Workshop (DIR 2009)*, Twente, The Netherlands, pages 19–25, 2009.
- [10] H. M. Gladney and R. A. Lorie. Trustworthy 100-year digital objects: durable encoding for when it's too late to ask. *ACM Trans. Inf. Syst.*, 23(3):299–324, 2005.
- [11] M. Hearst. Design recommendations for hierarchical faceted search interfaces. In *ACM SIGIR Workshop on Faceted Search*, 2006.
- [12] M. Hearst. *Search User Interfaces*. Cambridge University Press, 2009.
- [13] Anette Hulth, Jussi Karlgren, Anna Jonsson, Henrik Boström, and Lars Asker. Automatic keyword extraction using domain knowledge. In *Proceedings CICLing 2001*, pages 472–482. Springer, 2001.
- [14] R. Kaptein, M. Marx, and J. Kamps. Who said what to whom? Capturing the structure of debates. In *Proceedings SIGIR '09*, pages 831–832, 2009.
- [15] M. Kay. *XPath 2.0 Programmer's Reference*. Wrox, 2004.
- [16] M. Kay. *XSLT 2.0 3rd edition Programmer's Reference*. Wrox, 2004.
- [17] G. Knight and M. Pennock. Data without meaning: Establishing the significant properties of digital research. In *iPRES 2008 Conference Proceedings*, 2008.
- [18] Bertram Ludäscher, Pratik Mukhopadhyay, and Yannis Papakonstantinou. A transducer-based XML query processor. In *Proceedings VLDB '02*, pages 227–238. VLDB Endowment, 2002.
- [19] M. Marx. Long, often quite boring, notes of meetings. In *ESAIR '09: Proceedings of the WSDM '09 Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 46–53. ACM, 2009.
- [20] M. Marx and A. Schuth. DutchParl. A Corpus of Parliamentary Documents in Dutch. In *Proceedings LREC 2010*, 2010.
- [21] Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings CIKM '07*, pages 233–242, 2007.
- [22] V. Murdock and M. Lalmas. Workshop on aggregated search. *SIGIR Forum*, 42(2):80–83, 2008.
- [23] E. Rahm and H.H Do. Data cleaning: Problems and current approaches. *IEEE Techn. Bulletin on Data Engineering*, 23(4), 2000.
- [24] M. Reynaert. Non-interactive OCR post-correction for giga-scale digitization projects. In *Proceedings CICLing (Computational Linguistics and Intelligent Text Processing, 9th International Conference)*, pages 617–630, 2008.
- [25] A. Salminen. Building digital government by XML. In *Proceedings of the Thirty-Eighth Hawaii International Conference on System Sciences*. IEEE Computer Society, 2005.
- [26] B. Sigurbjörnsson. *Focused information access using XML element retrieval*. PhD thesis, University of Amsterdam, 2006.
- [27] J. R. Van Der Hoeven, R. J. Van Diessen, and K. Van Der Meer. Development of a universal virtual computer (uvc) for long-term preservation of digital objects. *J. Inf. Sci.*, 31(3):196–208, 2005.