Upscaling Documents

James Cummings

January 2010

Automating the markup process

- People don't like typing pointy brackets...
- ... so they do it wrong, or inconsistently!
- What tools can help us automate this process?
  - general transformation tools, e.g. perl
  - word-processor add-ons e.g. for Open Office
  - specifically XML tools, e.g. XSLTPROC, EGE
  - tools for linguistic annotation
  - tools for image annotation

“Plain Vanilla ASCII”

- Search and replace techniques will usually capture
  - Paragraph structure (blank lines)
  - Headings (lines in caps)
  - (sometimes) emphasis (strings between _ or *)
- Watch out for
  - Markup characters in the text
  - Metadata information
- Use:
  - Your favourite editor
  - Perl
  - (once you are well-formed) xslt transforms

Simple example

In a plain text file, which is *completely* regular, we can even write an XSLT stylesheet to do the job!

1. Make the plain text into well-formed XML
2. Use XSLT to enhance its markup

Varney: the input

```
<text>
                          VARNEY, THE VAMPYRE;
                                  or,
                           THE FEAST OF BLOOD.

                             P R E F A C E .

                               ----------

     The unprecedented success of the romance of "Varney the Vampyre," leave
the Author but little to say further, than that he accepts that success and
its results as gratefully as it is possible for any one to do popular favours.

     A belief in the existence of Vampyres first took its rise in Norway and
Sweden, from whence it rapidly spread to more southern regions, taking a firm
hold of the imaginations of the more credulous portion of mankind.

     The following romance is collected from seemingly the most authentic
sources, and the Author must leave the question of credibility entirely to his
readers, not even thinking that he his peculiarly called upon to express his
own opinion upon the subject.

     Nothing has been omitted in the life of the unhappy Varney, which could
tend to throw a light upon his most extraordinary career, and the fact of his
death just as it is here related, made a great noise at the time through
Europe, and is to be found in the public prints for the year 1713.

     With these few observations, the Author and Publisher, are well content
to leave the work in the hands of the public, which has stamped it with an
approbation far exceeding their most sanguine expectations, and which is
calculated to act as the strongest possible ncentive to the production of
other works, which in a like, or perchance a still further degree may be
deserving of public patronage and support.

     To the whole of the Metropolitan Press for their laudatory notices, the
Author is peculiarly obliged.

     _London Sep._ 1847

                                   -+-

    +=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=+
    |   This Varney the Vampyre e-text was entered by members of the     |
```

Varney: the stylesheet

```
<xsl:template name="doParas">

  <xsl:param name="s" select="."/>

  <xsl:element name="p">
    <xsl:choose>
    <xsl:when test="contains($s,$blankLine)">
      <xsl:value-of select="substring-before($s,$blankLine)"/>
     </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="$s"/>
    </xsl:otherwise>
    </xsl:choose>
  </xsl:element><xsl:text>
</xsl:text>
<xsl:if test="contains(substring-after($s,$blankLine),$blankLine)">
  <xsl:call-template name="doParas">
  <xsl:with-param select="substring-after($s,$blankLine)"
                  name="s"/>
  </xsl:call-template>
</xsl:if>

</xsl:template>
```

Varney: the output

```
<!--Output by an XSL transformation--><body>
  <div  n="0">
    <p>
        VARNEY, THE VAMPYRE;
        or,
        THE FEAST OF BLOOD.</p>
    <p> P R E F A C E .</p>
    <p> ----------</p>
    <p> The unprecedented success of the romance of "Varney the Vampyre," leave
        the Author but little to say further, than that he accepts that success and
        its results as gratefully as it is possible for any one to do popular
favours.</p>
    <p> A belief in the existence of Vampyres first took its rise in Norway and
        Sweden, from whence it rapidly spread to more southern regions, taking a firm
        hold of the imaginations of the more credulous portion of mankind.</p>
    <p> The following romance is collected from seemingly the most authentic
        sources, and the Author must leave the question of credibility entirely to his
        readers, not even thinking that he his peculiarly called upon to express his
        own opinion upon the subject.</p>
    <p> Nothing has been omitted in the life of the unhappy Varney, which could
        tend to throw a light upon his most extraordinary career, and the fact of his
        death just as it is here related, made a great noise at the time through
        Europe, and is to be found in the public prints for the year 1713.</p>
    <p> With these few observations, the Author and Publisher, are well content
        to leave the work in the hands of the public, which has stamped it with an
        approbation far exceeding their most sanguine expectations, and which is
        calculated to act as the strongest possible ncentive to the production of
        other works, which in a like, or perchance a still further degree may be
        deserving of public patronage and support.</p>
    <p> To the whole of the Metropolitan Press for their laudatory notices, the
        Author is peculiarly obliged.</p>
    <p> London Sep  1847</p>
```

Silk purses and sows ears

- TEI is an XML vocabulary
- Word and Open Office both use XML vocabularies
- so converting one to the other is a simple XSLT transformation, right?

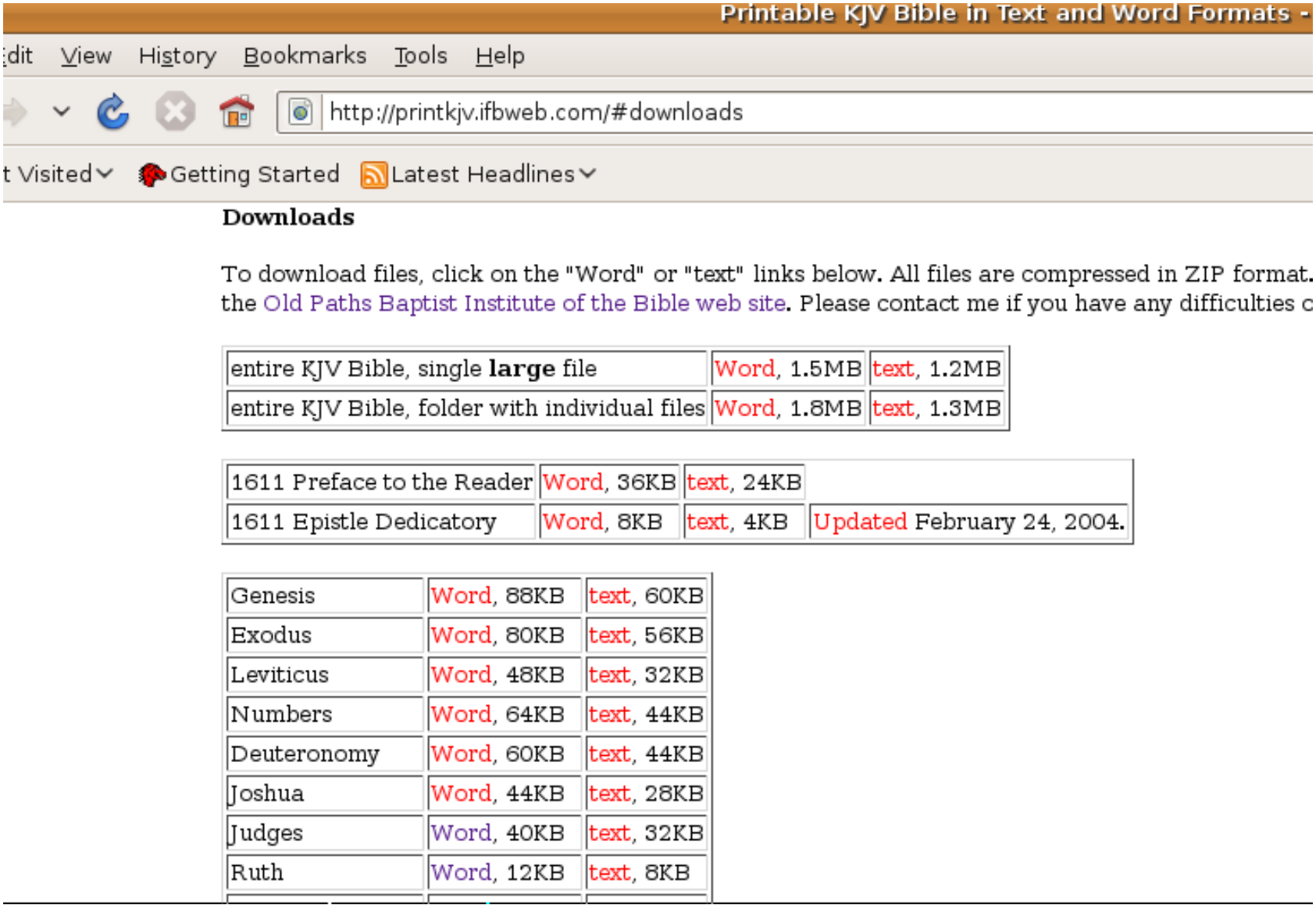
Up to a point, Lord Copper

Converting from Word DOC to TEI XML

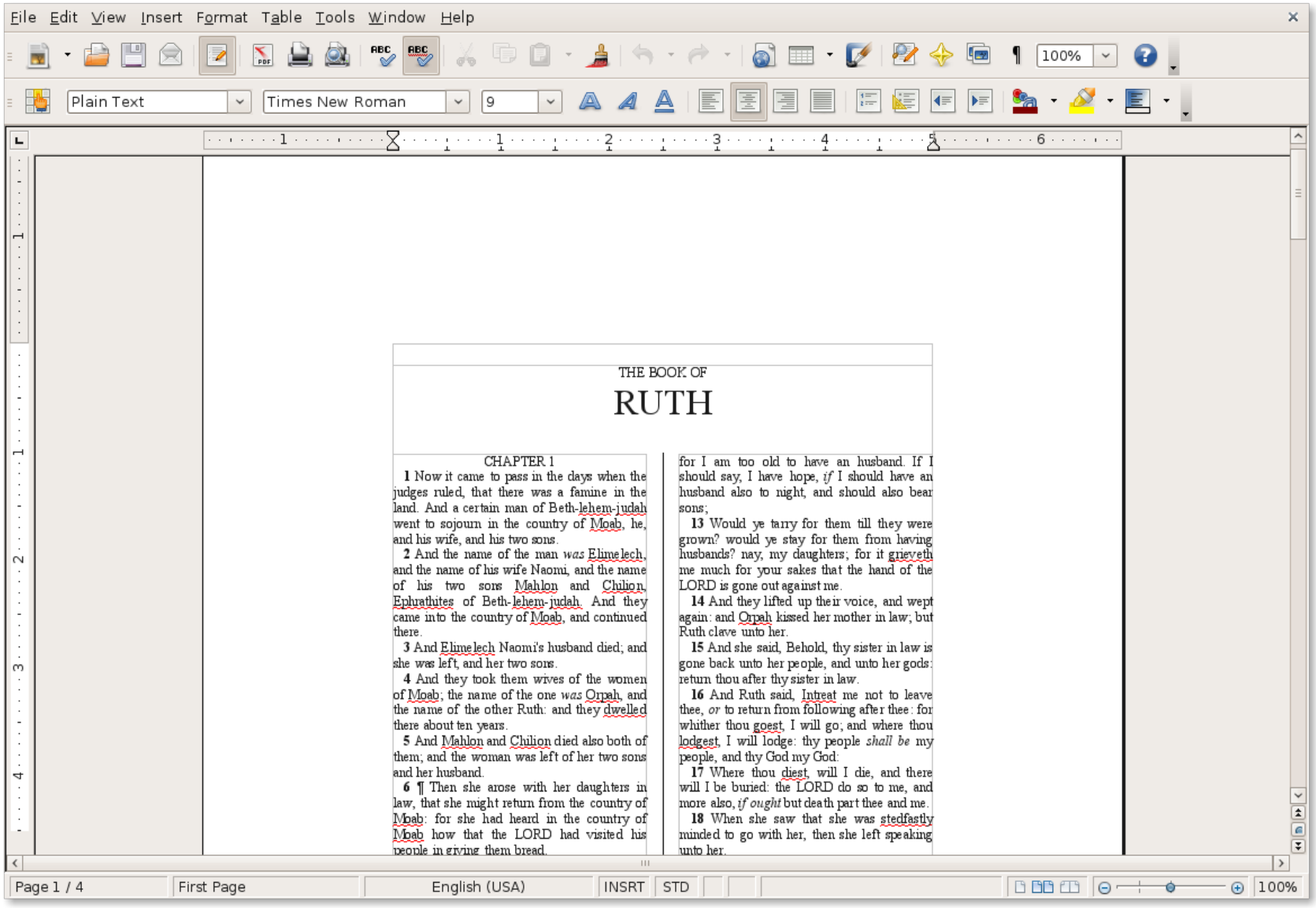
One simple way is to use the TEI XML filter supplied for Open Office, (teioop5.jar), downloadable from Sourceforge

- Install the jar file in your copy of Open Office
- Open the Word file
- Choose Save As and select TEI P5 from the list
- Use XSLT to improve on the tagging you get back

Word to TEI example (1)



Word to TEI example (1)



Word to TEI example (1)

File Edit Options Buffers Tools XML TEI UniChar Help

```
      <revisionDesc>
         <change>
            <name>Bruce Wilcox</name>
            <date>2003-11-13T22:53:00</date>
         </change>
      </revisionDesc>
   </teiHeader>
   <text>
      <body>
         <p>THE BOOK OF</p>
         <p>RUTH</p>
         <p>CHAPTER 1</p>
         <p>
            <hi>1</hi> Now it came to pass in the days when the judges r
uled, that there was a famine in the land. And a certain man of Beth-leh
em-judah went to sojourn in the country of Moab, he, and his wife, and h
is two sons.</p>
         <p>
            <hi>2</hi> And the name of the man <emph>was</emph> Elimelec
h, and the name of his wife Naomi, and the name of his two sons Mahlon a
nd Chilion, Ephrathites of Beth-lehem-judah. And they came into the coun
try of Moab, and continued there.</p>
         <p>
            <hi>3</hi> And Elimelech Naomi's husband died; and she was l
eft, and her two sons.</p>
         <p>
            <hi>4</hi> And they took them wives of the women of Moab; th
e name of the one <emph>was</emph> Orpah, and the name of the other Ruth
: and they dwelled there about ten years.</p>
         <p>
            <hi>5</hi> And Mahlon and Chilion died also both of them; an
d the woman was left of her two sons and her husband.</p>
         <p>
            <hi>6</hi> ¶ Then she arose with her daughters in law, that
she might return from the country of Moab: for she had heard in the coun
```

A case study: adding POS tagging to Punch

- Q. What do we mean by POS tagging?
- A. Something like this:
  `http://ucrel.lancs.ac.uk/claws/trial.html`
- See
  `http://www-nlp.stanford.edu/links/statnlp.html#Tagger`
  for a long list of such things.

We like treetagger...

IPP annotation workflow

For each file,

1. extract the `<text>` element only and pass it throgh treetagger
2. post-process the treetagger output to make it well formed XML
3. combine that with the header of the file

IPP-POS example -1

```
<div  type="div2">
  <head>RESOLUTIONS.</head>
  <lg>
    <l>I will not breakfast in my bed</l>
    <l>With downy cushions at my head;</l>
    <l>That would be very wrong—and so</l>
    <l>Away the eggs and bacon go!</l>
  </lg>
  <lg>
    <l>I will not read in bed at night</l>
    <l>And burn the dear electric light;</l>
    <l>Nor buy another costly hat;</l>
    <l>Oh no!  I'm much too good for that.</l>
  </lg>
  <lg>
    <l>But I will rise before the dawn</l>
    <l>And weed and cut and roll the lawn;</l>
    <l>My border I will plant with veg,</l>
    <l>Abundantly from hedge to hedge.</l>
  </lg>
  <lg>
    <l>And all the day I'll practise thrift</l>
    <l>And no more happily will drift</l>
    <l>In deeper debt, as once, alas!</l>
    <l>—But what an awful year I'll pass.</l>
  </lg>
  <milestone  rend="hr"  unit="rule"/>
</div>
```

IPP-POS example -2

```
<div type="div2">
<head>
RESOLUTIONS NNS resolution
.  SENT .
</head>
<lg>
<l>
I PP I
will MD will
not RB not
breakfast NN breakfast
in IN in
my PP$ my
bed NN bed
</l>
<l>
With IN with
downy JJ downy
cushions NNS cushion
at IN at
my PP$ my
head NN head
; :  ;
</l>
<l>
That DT that
would MD would
be VB be
very RB very
wrong—and JJ <unknown>
so IN so
</l>
<l>
```

IPP-POS example -3

```
<div  type="div2">
  <head>
    <s  n="1">
      <w  type="NNS"  lemma="resolution">RESOLUTIONS</w>
      <c  type="SENT">.</c>
    </s>
  </head>
  <lg>
    <s  n="2">
      <lb/>
      <w  type="PP"  lemma="I">I</w>
      <w  type="MD"  lemma="will">will</w>
      <w  type="RB"  lemma="not">not</w>
      <w  type="NN"  lemma="breakfast">breakfast</w>
      <w  type="IN"  lemma="in">in</w>
      <w  type="PP$"  lemma="my">my</w>
      <w  type="NN"  lemma="bed">bed</w>
      <lb/>
      <w  type="IN"  lemma="with">With</w>
      <w  type="JJ"  lemma="downy">downy</w>
      <w  type="NNS"  lemma="cushion">cushions</w>
      <w  type="IN"  lemma="at">at</w>
      <w  type="PP$"  lemma="my">my</w>
      <w  type="NN"  lemma="head">head</w>
      <c  type=":">;</c>
      <lb/>
      <w  type="DT"  lemma="that">That</w>
      <w  type="MD"  lemma="would">would</w>
      <w  type="VB"  lemma="be">be</w>
      <w  type="RB"  lemma="very">very</w>
      <w  type="JJ"  lemma="XXXX">wrong—and</w>
      <w  type="IN"  lemma="so">so</w>
      <lb/>
```

<p> to <u> (1 - Sample XML)

```
<div>
  <p>
    <hi rend="bold">RG</hi> :  Et est-ce que <emph>vous</emph>
pouvez me dire:  <seg>quelque chose</seg> sur votre famille
d'origine ?  </p>
  <p>
    <hi rend="bold">AG</hi> :  Oui, qu'est-ce que
<hi rend="bold">vous</hi> voulez :  savoir ?</p>
</div>
```

<p> to <u> (2 - Mediocre XSLT)

```
<xsl:template match="p/hi[1]"/>
<xsl:template match="p">
 <u who="{concat('#',normalize-space(hi[1]))}">
  <xsl:value-of select="substring-after(., ':')"/>
 </u>
</xsl:template>
<xsl:template match="@*|node()" priority="-1">
 <xsl:copy>
  <xsl:apply-templates select="@*|node()"/>
 </xsl:copy>
</xsl:template>
```

<p> to <u> (2 - Better XSLT)

```
<xsl:template match="p/hi[1]"/>
<xsl:template match="p">
 <u who="{concat('#',normalize-space(hi[1]))}">
  <xsl:apply-templates select="node()[not(hi[1])]"/>
 </u>
</xsl:template>
<xsl:template match="@*|node()" priority="-1">
 <xsl:copy>
  <xsl:apply-templates select="@*|node()"/>
 </xsl:copy>
</xsl:template>
<xsl:template match="text()">
 <xsl:analyze-string select="." regex="\s*:\s*">
  <xsl:matching-substring>
   <xsl:text/>
  </xsl:matching-substring>
  <xsl:non-matching-substring>
   <xsl:value-of select="."/>
  </xsl:non-matching-substring>
 </xsl:analyze-string>
</xsl:template>
```

<p> to <u> (3 - Result)

```
<div>
 <u who="#RG"> : Et est-ce que <emph>vous</emph> pouvez me dire:
<seg>quelque chose</seg> sur votre famille d'origine ? </u>
 <u who="#AG"> : Oui, qu'est-ce que <hi rend="bold">vous</hi>
voulez savoir ?</u>
</div>
```