Structuring data in electronic format

Ciprian-Virgil Gerstenberger

University of Tromsø
Norway

DOBES Winter School in Saami Language Documentation and Revitalization
1.–12. March 2010, Bodø, Norway
Outline

Introduction

Basic notions

Working with structure

CSV in 5 Minutes

XML in 5 Minutes

Conclusion
Outline

Introduction

Basic notions

Working with structure

CSV in 5 Minutes

XML in 5 Minutes

Conclusion
Outline

Introduction

Basic notions

Working with structure

CSV in 5 Minutes

XML in 5 Minutes

Conclusion
Outline

Introduction

Basic notions

Working with structure

CSV in 5 Minutes

XML in 5 Minutes

Conclusion
Outline

Introduction

Basic notions

Working with structure

CSV in 5 Minutes

XML in 5 Minutes

Conclusion
Outline

Introduction

Basic notions

Working with structure

CSV in 5 Minutes

XML in 5 Minutes

Conclusion
Introduction
Basic notions
Working with structure
CSV in 5 Minutes
XML in 5 Minutes
Conclusion

Motivation: Why?

Computer-assisted language learning (CALL)
⇒ Computer as useful helper

- rapid grow of users and purposes of electronic devices
  ⧳ → ease any ‘stupid’ work by automatizing it (Let the computer do it!)
  ⧴ → learn the technology behind it (Learn the computer how to do it!)

- globalization via internet
  ⧳ → the whole world in my private home
  ⧴ → my private home in the whole world
Goal: What?

Developing a small demo version of Oahpa for the language of your choice

- Leksa: exercising basic forms of words (lexemes)
- MorfaC: exercising inflected words in an appropriate context (lexemes + morpheme)

Let the computer do it!

😊
Way: How?

Learning how to structure lexical data in different formats

- Leksa: XML format (Extensible Markup Language)
- MorfaC: CSV format (Comma-Separated Values)

Learn the computer how to do it!

😊
Machine-readability

- Which machines?
- What is machine-readable?
- What is machine-readable by which machines?

Be careful with technical terms!
Containers vs. Tools

- computers
- operating systems
- hardware vs. software
- directories (≡ folders)
- files
- databases
- programs
- versions

Do you know what these notions mean?
Operations

- read
- write
- execute

What operates on what?

Who is allowed to operate on what?
→ permissions
Concepts

- iterativity (repetition)
- recursivity (embedding)
- bracketing (open – close)
- uniqueness
- scope
Material: Linguistic Data

- collecting, categorizing, sorting, structuring
- annotating (annotation depth?)
- storing/archiving
- searching/retrieving

- What do you can do? Which tools?
- What do you want to do? Which goals?
Container: Files

• **name:** gersti.txt, gersti.xml, gersti.jpg, gersti.html, gersti

• **extension:** .txt, .xml, .jpg, .html

• **type:** text file, xml file, image file, etc.

• **header:**
  % PDF-1.4
  <?xml version="1.0" encoding="UTF-8" ?>
  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" ... >

  name vs. content
  format vs. structure
Tools: Editors

- simple vs. complex
- working with databases
- operating on different types of files with different operations
- text editors, XML editors, sound editors, image editors, etc.

What is your favourite editor/tool?
Exercises

How well do you know your favourite editor/tool?

• open different file formats with your editor
• open a new file, write some data in your usual way and try to save it in different formats
• open the resulting formats with other editors/tools
• change the extension of a file (for instance, *txt* into *gif*) and try to open with an appropriate editor
• what about invizibile characters (white spaces, tabs, etc.), can you visualise them with your editor?
• what about special characters, diacritics?
• what about coding (ASCII, UTF-8, etc.)?
• try different codings and different settings of your editor!
Which structure?

Structure is (almost) everywhere:

- text files
- FileMaker files
- HTML files
- XML files
- Excel Sheets
- Word document

⇒ find an appropriate (lexical) data structure!
Excel Sheets

# Form

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Excel Sheets

### Content

<table>
<thead>
<tr>
<th>Sáme</th>
<th>Grammatical</th>
<th>English (American)</th>
<th>English (British)</th>
<th>English (Australian)</th>
<th>English (Canadian)</th>
<th>English (Other)</th>
<th>Synonyymat</th>
<th>Finnish</th>
<th>Syllables</th>
</tr>
</thead>
<tbody>
<tr>
<td>absurda</td>
<td>adj.</td>
<td>absurd</td>
<td>start to understand</td>
<td>grandchild; great-grandchild (to a grandfather or great-grandfather)</td>
<td>advertisement, ad</td>
<td>advert</td>
<td>lapsenlapsi</td>
<td>mainos</td>
<td>agás</td>
</tr>
<tr>
<td>áddegøehtit</td>
<td>v.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>áddiut</td>
<td>s.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>advertisa</td>
<td>s.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>agiheaví</td>
<td>adv.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ankábiddu</td>
<td>s.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Excel Sheets

Structure = Form + Content
Excel Sheets: Exercise

- open a new file
- write some lexical data (entry, part-of-speech, translations)
- save it in different formats
- compare the formats
Comma Separated Values (CSV)

Excel Sheets are not the ideal format for a further data processing!

⇒ What about, for instance, Comma Separated Values?
Comma Separated Values (CSV)

Form-building elements (= separators):

- newlines
- semicolons
Comma Separated Values (CSV)

Problem 1:
- no difference between column description and column content

<table>
<thead>
<tr>
<th>Sámeigella;Grammatical class;English (American);English (British);English (Australian) ;English (Canadian) ;English (Other) ;Synonyymat;Finnish;Swedish;Class;Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>absurda;adj.;absurd;;;;absurdi;;;</td>
</tr>
<tr>
<td>áddegooahtit;v.;start to understand ;;;;;;;;;;;;;;;;;;</td>
</tr>
<tr>
<td>ádjut;s;&quot;grandchild; great-grandchild (to a grandfather or great-grandfather) &quot;;;;;lapsenlapsi;;</td>
</tr>
<tr>
<td>advertiisa;s;advertisement, ad;advert;;;;mainos;;(Does this include commercials, etc.?</td>
</tr>
<tr>
<td>agibeavái;adv.;forever, for good, for life, in perpetuity ;;;agás;;;</td>
</tr>
<tr>
<td>aähkábiddu;s;angler, monkfish, goosefish, anglerfish, fishing frog, frogfish ;;;;;(Lophius piscatorius)</td>
</tr>
</tbody>
</table>
Comma Separated Values (CSV)

Problem 1:
- no difference between column description and column content

"Solution":
- by convention, the first line contains the description of each column

<table>
<thead>
<tr>
<th>Sámegiella;Grammatical class;English (American);English (British);English (Australian);English (Canadian);English (Other);Synonyymat;Finnish;Swedish;Class;Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>absurd;adj.;absurd;;;absurdi;;</td>
</tr>
<tr>
<td>áddegoahtit;v.;start to understand ;;;;;;</td>
</tr>
<tr>
<td>áddjut;s.;&quot;grandchild; great-grandchild (to a grandfather or great-grandfather) &quot;;;;;;lapsenlapsi;;</td>
</tr>
<tr>
<td>advertiisa;s.;advertisement, ad;advert;;;mainos;;(Does this include commercials, etc.?)</td>
</tr>
<tr>
<td>agibeavái;adv.;forever, for good, for life, in perpetuity ;;;;agás;;</td>
</tr>
<tr>
<td>áhkábiddu;s.;angler, monkfish, goosefish, anglerfish, fishing frog, frogfish ;;;;;(Lophius piscatorius)</td>
</tr>
</tbody>
</table>
Comma Separated Values (CSV)

Problem 2:
- separators might occur as content, too
Comma Separated Values (CSV)

Problem 2:
- separators might occur as content, too

"Solution":
- mask them in this context, for instance, using quotation marks
Comma Separated Values (CSV)

Problem 3:
- no discrimination between the end of a content and the beginning of the next one
Problem 3:
- no discrimination between the end of a content and the beginning of the next one

Possible "Solution":
- Use XML!
CSV: Exercise 1

- open a new file
- write some lexical data (entry, part-of-speech, translations)
- compare the format to that generated out of the Excel
CSV: Exercise 2

- open a new file
- write some lexical data (entry, part-of-speech, morphosyntactic description, word form)

Example

```
biigá+N+Sg+Nom  biigá
biigá+N+Sg+Gen  biiggá
biigá+N+Sg+Acc  biiggá
biigá+N+Sg+Ill  biigái
biigá+N+Sg+Loc  biiggás
biigá+N+Sg+Com  biiggáin
biigá+N+Pl+Nom  biiggát
biigá+N+Pl+Gen  biiggáid
biigá+N+Pl+Acc  biiggáid
biigá+N+Pl+Ill  biiggáide
biigá+N+Pl+Loc  biiggáin
biigá+N+Pl+Com  biiggáiguin
```
Extended Markup Language (XML)

Example: a lexical entry in an OAHPA lexicon

```xml
<entry>
  <lemma>giedahallat</lemma>
  <pos class="V"/>
  <translations>
    <tr xml:lang="nob">à bearbeide</tr>
    <tr xml:lang="nob">à behandle</tr>
    <tr xml:lang="fin">käsitellä</tr>
    <tr xml:lang="eng">to treat</tr>
  </translations>
  <valency>
    <val class="TV"/>
  </valency>
  <semantics>
    <sem class="WORK/ECONOMY-V"/>
  </semantics>
  <stem class="bisyllabic" gradation="yes" diphthong="no" soggi="a"/>
  <dialect class=""/>
  <sources>
  </sources>
</entry>
```
Basics

- header
- root
- element
- attribute-value
Dictionary example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xml-stylesheet type="text/css" href="../..//scripts/gt_dictionary.css"/>
<xml-stylesheet type="text/xsl" href="../..//scripts/gt_dictionary.xsl"/>
<!DOCTYPE r PUBLIC "-//XMLmind//DTD smenob//SE" "../..//scripts/gt_dictionary.dtd">
<r>
  <e usage="more">
    <lg>
      <l pos="v" context="mun">čalmmustahttit</l>
    </lg>
    <mg>
      <tg>
        <t pos="v">fokusere på</t>
        <t pos="v">gjøre synlig</t>
      </tg>
    </mg>
  </e>
  <e usage="more">
    <lg>
      <l pos="v" context="mun">skuvlet</l>
    </lg>
    <mg>
      <tg>
        <t pos="v">undervise</t>
      </tg>
    </mg>
  </e>
</r>
```
Well-formedness vs. Validity

- well-formed (↔ ill-formed) XML file
  ⇒ general conventions
- DTD valid (↔ DTD invalid) XML file (Document Type Definition)
  ⇒ special conventions
Dictionary DTD

```
<!ELEMENT r (e+) >
<!ELEMENT e (lg,mg+) >
<!ATTLIST e
  usage (mt | dic | ped | other) #IMPLIED
  src (nj | sk | SvSt | other) #IMPLIED
>
<!ELEMENT lg (l,lc*) >
<!ELEMENT l (#PCDATA) >
<!ATTLIST l
  pos (a | n | npl | v | adv | po | pr ... | num | cc | cs | ) #REQUIRED
  nr (sg | pl) #IMPLIED
  context CDATA #IMPLIED
  type (rel | pers | pre | post) #IMPLIED
  class (I | II | III | IV | V | VI) #IMPLIED
  p3p CDATA #IMPLIED
  minip (notSg1 | onlyPl) #IMPLIED
>
<!ELEMENT lc (#PCDATA) >
<!ELEMENT mg (tg+) >
<!ELEMENT tg (re?,t+,xg*) >
```
XSL and XSLT

XSL: EXtensible Stylesheet Language
XSLT: XSL Transformations
EXtended Markup Language (XML)

Advantages:

- any piece of content can be described by the name of the element as its container
- arbitrarily iterable (as well as Excel Sheets and CSV)
- arbitrarily embeddable (unlike Excel Sheets and CSV)
  → arbitrarily structurable!
- from XML, different kind of formats can be generated
  → XSL (EXtensible Stylesheet Language)
  → XSLT (XSL Transformations)
- from different kind of formats, XML can be generated
Extended Markup Language (XML)

Disadvantages:
- to work with XML requires much more technical knowledge than to work with text files or Exel Sheets
- learning XML is worth but it costs time
- people using XML as an exchange format have to agree upon a specific structure of the XML file, usually described in a Document Type Definition (DTD) or an XML Schema
XML: Exercise

- open a new file, write some lexical data (entry, pos, etc.)
- compare the format to that generated out of the Excel
- compare your XML format to your CSV format

```xml
<lexicon xml:lang="rom">
  <entry>
    <lemma>copil</lemma>
    <pos class="N"/>
    <translations>
      <tr xml:lang="ger">Kind</tr>
      <tr xml:lang="eng">child</tr>
    </translations>
    <semantics>
      <sem class="HUMAN"/>
    </semantics>
    <dialect />
  </entry>
  <entry>
    <lemma>roşu</lemma>
    <pos class="A"/>
    <translations>
      <tr xml:lang="ger">rot</tr>
      <tr xml:lang="eng">red</tr>
    </translations>
    <semantics>
      <sem class="COLOR"/>
    </semantics>
  </entry>
</lexicon>
```
THE Question

Why to know the whole stuff?
- **collecting** (row) data
- **processing**: sorting, categorizing, annotating, formatting
- **using**: applications (e.g., CALL), for other (theoretical?) linguists, for the speaker community
Your actual task:
- Find the best suited structure for describing your data!
Modelling Linguistic Data

- **Quality**: what?
- **Quantity** how often?
- **Size**: how long (≡ span, scope)?

⇒ **Structure**