

Annotating basic text files

We can annotate basic text files by adding column data, and this can even include structural data

- ▶ We will first look at some examples of this
- ▶ Then we will look at a tool designed to help with corpus annotation, namely MMAX2

TnT format

A way to simply encode POS tags:

```
%% Comments appear after beginning-of-line '%%'
```

```
%% s1 ...
```

```
The      DT
```

```
man      NN
```

```
ran      VB
```

```
.        .
```

```
%% s2 ...
```

CHAT (CHILDES) format

*MAR: I wanted a toy.

%mor: PRO|I&2S V|want-PAST DT|a&INDEF N|toy.

*MOT: well go get it!

%spa: \$IMP \$REF \$INS

%mor: ADV|well V|go&PRES V|get&PRES PRO|it!

SUSANNE format

A01:0010.03	-	YB	<minbrk>	-	[Oh.Oh]
A01:0010.06	-	AT	The	the	[O[S[Nns:s.
A01:0010.09	-	NP1s	Fulton	Fulton	[Nns.
A01:0010.12	-	NNL1cb	County	county	.Nns]
A01:0010.15	-	JJ	Grand	grand	.
A01:0010.18	-	NN1c	Jury	jury	.Nns:s]
A01:0010.21	-	VVDv	said	say	[Vd.Vd]
A01:0010.24	-	NPD1	Friday	Friday	[Nns:t.Nns:t]

...

Fulton County is a phrase of category Nns.

CoNLL format

1	Cathy	Cathy	N	N	eigen ev neut	2	su	_	_
2	zag	zie	V	V	trans ovt 1of2of3 ev	0	ROOT	_	_
3	hen	hen	Pron	Pron	per 3 mv datofacc	2	obj1	_	_
4	wild	wild	Adj	Adj	attr stell onverv	5	mod	_	_
5	zwaaien	zwaai	N	N	soort mv neut	2	vc	_	_
6	.	.	Punc	Punc	punt	5	punct	_	_

Cathy, *hen*, and *zwaaien* are all dependents of *zag*.

Negra format

```
#BOS 614 0 1091718495 1 %% @SB2AV@
Nach  nach  APPR  --          AC  500
der   der   ART   Dat.Sg.Fem  NK  500
Wende Wende  NN    Dat.Sg.Fem  NK  500
wollte wollen VMFIN 3.Sg.Past.Ind HD  505
Bonn  Bonn  NE    Nom.Sg.Neut SB   505
...
#500  --    PP    --          MO  505
...
#505  --    S     --          --  0
#EOS 614
```

Nach der Wende forms a PP, which is itself a daughter of S.

Using a pre-built tool

With a small script, it is easy to convert text files to XML format, or to any format you might need

With a text file, though, it is sometimes difficult to maintain consistency, and, with more complex annotation, it is often difficult to visualize what you are actually doing

- ▶ So, we'll look at MMAX2, one particular tool used to aid in corpus annotation
- ▶ MMAX2 is fairly easy to obtain and install; simply download and unpack the appropriate files at:
<http://mmax2.sourceforge.net/>
 - ▶ For documentation, see the doc/ folder, as well as the paper available at this site

Loading a text file

Assumptions:

- ▶ You're using a plain text file.
- ▶ We'll assume one word per line, obtained by some process of tokenization.
 - ▶ Some HTML-style convention: e.g., need to convert & to &

1. Run `./startmmax.sh` (unix) or `startmmax.bat` (windows)

2. Tools → Project Wizard

2.1 Text Input File: Pick file and click on *Analyse File*

2.2 Tokenization: select “one token per line” and click on *Tokenize*

2.3 Markable level: Click on *Add level* for each level to be added

- ▶ Make word level
- ▶ Can make POS level (or POS can be an attribute of the word level)
- ▶ Make a level for error annotation (?)

2.4 .MMAX Project: Pick a project path; you'll likely want `basedata`, `scheme`, etc. as daughter directories of this path.

See also p. 22 of the `mmax2quickstart.pdf` file, which walks you through using the wizard.

Markables

What is a markable?

- ▶ A markable is an item from the corpus which can be marked.
 - ▶ For POS annotation, this corresponds to words
 - ▶ For other annotations, this might be more than one word
- ▶ Annotation is either an *attribute* or a *relation* of the markable
 - ▶ An attribute is a property (e.g., POS tag) with a particular value for that markable.
 - ▶ A relation relates one markable to another
 - ▶ Can have MARKABLE_SETs (unordered relations) or MARKABLE_POINTERs (ordered relations)

Preprocessing

So, when we create a word level of annotation, we have word markables that can be annotated

- ▶ Markable files look like the following:

```
<?xml version="1.0" encoding="US-ASCII"?>
<!DOCTYPE markables SYSTEM "markables.dtd">
<markables xmlns="www.eml.org/NameSpaces/word">
<markable mmax_level="word" id="markable_1" span="word_1"/>
<markable mmax_level="word" id="markable_2" span="word_2"/>
...
</markables>
```

- ▶ MMAX2 creates this automatically, but it really isn't that hard to convert data into this format

Scripting your way to a better life

I wrote a short program which can replace the markable file with one which contains annotation:

```
print """<?xml version="1.0" encoding="US-ASCII"?>
<!DOCTYPE markables SYSTEM "markables.dtd">
<markables xmlns="www.eml.org/NameSpaces/POS">"""

# ... Then parse each TnT line and print it in the
# appropriate format ...

print "</markables>"
```

- ▶ Calling the program:

```
python merge.py > \
  /...wsj10000/markable/wsj10000_POS_level.xml
```

Adding annotation

But to properly add the annotation, I also need to change the scheme files

- ▶ Here is what my POS_scheme.xml file now looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<annotationscheme>
<attribute id="tag_level" name="tag" type="freetext">
  <value name="tag"/>
</attribute>
</annotationscheme>
```

Note the use of `freetext` as the type: this allows me to create new POS tags on the fly (but could lead to more errors)

- ▶ Useful slides: <http://homepages.inf.ed.ac.uk/olemon/mullerslides.pdf>

Syntactic annotation?

Constituency annotation:

- ▶ Can create markables of larger length by highlighting them, or by predefining a file with markables listed
- ▶ Constituents can be seen as mappings from strings to labels; thus, a layer of annotation can mark sequences of words as categories
- ▶ One could take a treebank and put it into MMAX2 by scripting

Dependency annotation:

- ▶ MARKABLE_POINTERs are already what we need for dependencies
- ▶ One can then label the relation or the dependent (assuming single-headedness)

Changing displays

When using annotation, it is often useful to change displays

- ▶ You can do this through style sheets and, for things like color, through the customization file.
- ▶ See the `mmax2stylesheets.pdf` documentation.

WordFreak

1. `http://wordfreak.sourceforge.net/`
2. `java -jar wordfreak-2.2.jar`
 - ▶ Look at the help contents for some help.

WordFreak is a bit more limited in its capacity (e.g., it's harder to change tagsets)

Assignment

In groups of approximately 4 people:

- ▶ Acquire a short learner text, e.g.
 - ▶ <http://joeandco.blogspot.com/>
 - ▶ <http://www.eng.ritsumei.ac.jp/asao/lcorpus/>
 - ▶ <http://www.staff.amu.edu.pl/~przemka/rawsmp1.html>
- ▶ Tokenize it
- ▶ Load into MMAX2
- ▶ Design a “toy” learner-appropriate annotation scheme and implement it
- ▶ In a brief (1-2 page) writeup, describe:
 - ▶ How you went about the task and, in particular, any difficulties you encountered.
 - ▶ What errors you were trying to cover and which you were ignoring (and why).
 - ▶ What grammatical properties you would want to include in the future.

This will be due next Wednesday.