Demonstration of the UAM CorpusTool for text and image annotation

Mick O'Donnell
Escuela Politécnica Superior
Universidad Autónoma de Madrid
28049, Cantoblanco, Madrid, Spain
michael.odonnell@uam.es

Abstract

This paper introduced the main features of the UAM CorpusTool, software for human and semi-automatic annotation of text and images. The demonstration will show how to set up an annotation project, how to annotate text files at multiple annotation levels, how to automatically assign tags to segments matching lexical patterns, and how to perform cross-layer searches of the corpus.

1 Introduction

In the last 20 years, a number of tools have been developed to facilitate the human annotation of text. These have been necessary where software for automatic annotation has not been available, e.g., for linguistic patterns which are not easily identified by machine, or for languages without sufficient linguistic resources.

The vast majority of these annotation tools have been developed for particular projects, and have thus not been readily adaptable to different annotation problems. Often, the annotation scheme has been built into the software, or the software has been limited in that they allow only certain types of annotation to take place.

A small number of systems have however been developed to be general purpose text annotation systems, e.g., MMAX-2 (Müller and Strube 2006), GATE (Cunningham et al 2002), WordFreak (Morton and LaCivita 2003) and Knowtator (Ogren 2006).

With the exception of the last of these however, these systems are generally aimed at technically advanced users. WordFreak, for instance, requires writing of Java code to adapt to a different annotation scheme. Users of MMAX-2 need to edit XML by hand to provide annotation schemes. Gate allows editing of annotation schemes within the tool, but it is a very complex system, and lacks clear documentation to help the novice user become competent.

The UAM CorpusTool is a text annotation tool primarily aimed at the linguist or computational linguist who does not program, and would rather spend their time annotating text than learning how to use the system. The software is thus designed from the ground up to support typical user workflow, and everything the user needs to perform annotation tasks is included within the software.

2 The Project Window

In the majority of cases, the annotator is interested in annotating a range of texts, not just single texts. Additionally, in most cases annotation at multiple linguistic levels is desired (e.g., classifying the text as a whole, tagging sections of text by function (e.g., abstract, introduction, etc.), tagging sentences/clauses, and tagging participants in clauses. To overcome the complexity of dealing with multiple source files annotated at multiple levels, the main window of the CorpusTool is thus a window for project management (see Figure 1).

Proceedings of the ACL-08: HLT Demo Session (Companion Volume), pages 13–16,
Columbus, June 2008.

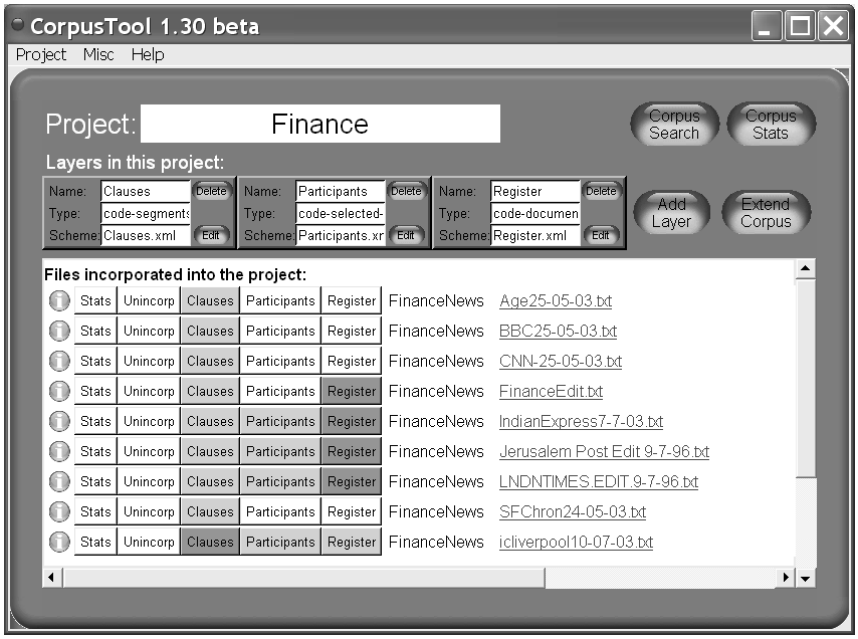

Figure 1: The Project Window of UAM CorpusTool

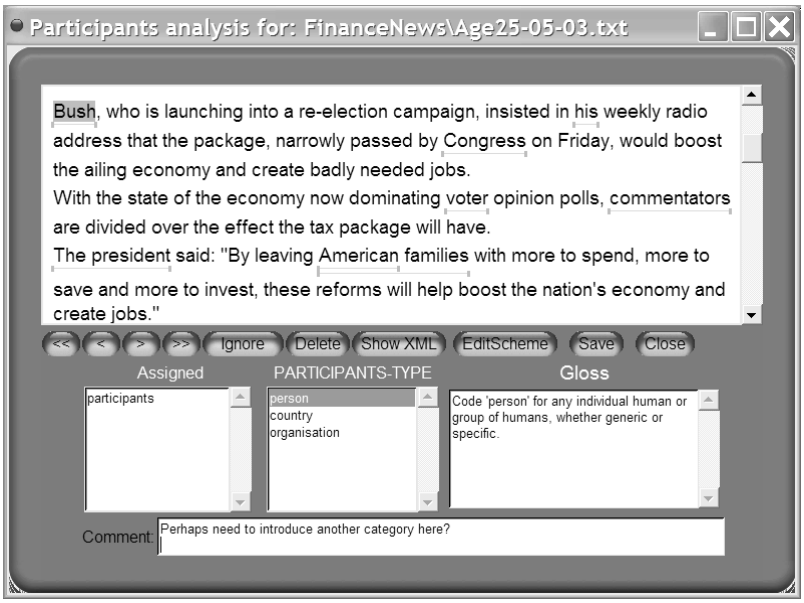

Figure 3: An annotation window for 'Participant' layer.

```
<?xml version='1.0' encoding='utf-8'?>
<document>
  <segments>
    <segment id='1' start='158' end='176'
             features='participant;human' state='active'/>
    <segment id='2' start='207' end='214'
             features='participant;organisation;company'
             state='active'/>
        ...
  </segments>
</document>
```

Figure 4: Annotation Storage Example

This window allows the user to add new annotation layers to the project, and edit/extend the annotation scheme for each layer (by clicking on the "edit" button shown with each layer panel). It also allows the user to add or delete source files to the project, and to open a specific file for annotation at a specific layer (each file has a button for each layer).

3 Tag Hierarchy Editing

Most of the current text annotation tools lack built-in facilities for creating and editing the coding scheme (the tag set). UAM CorpusTool uses a hierarchally organised tag scheme, allowing cross-classification and multiple inheritance (both disjunctive and conjunctive). The scheme is edited graphically, adding, renaming, moving or deleting features, adding new sub-distinctions, etc. See Figure 3.

An important feature of the tool is that any change to the coding scheme is automatically propagated throughout all files annotated at this layer. For instance, if a feature is renamed in the scheme editor, it is also renamed in all annotation files.

The user can also associate a gloss with each tag, and during annotation, the gloss associated with each feature can be viewed to help the coder determine which tag to assign.

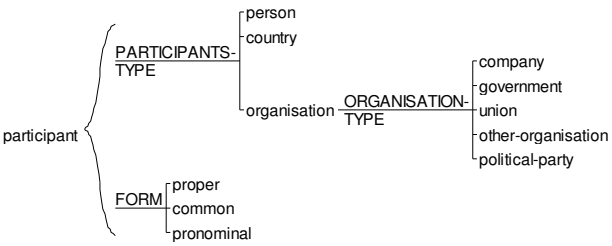

Figure 2: Graphical Editing of the Tag Hierarchy

4 Annotation Windows

When the user clicks on the button for a given text file/layer, an annotation window opens (see Figure 3). This window shows the text in the top panel (with previously identified text segments indicated with underlining). When the user creates a new segment (by swiping text) or selects an existing segment, the space below the text window shows controls to select the tags to assign to this segment. Tags are drawn from the tag scheme for the current layer. Since the tag hierarchy allows cross-classification, multiple tags are assigned to the segment. CorpusTool allows for partially overlapping segments, and embedding of segments.

Annotated texts are stored using stand-off XML, one file per source text and layer. See Figure 4 for a sample. The software does not currently input from or export to any of the various text encoding standards, but will be extended to do so as it becomes clear which standards users want supported.

Currently the tool only supports assigning tags to text. Annotating structural relations between text segments (e.g., co-reference, constituency or rhetorical relations) is not currently supported, but is planned for later releases.

5 Corpus Search

A button on the main window opens a Corpus Search interface, which allows users to retrieve lists of segments matching a query. Queries can involve multiple layers, for instance, `subject in passive-clause in english` would retrieve all NPs tagged as subject in clauses tagged as passive-clause in texts tagged as 'english' (this is thus a search over 3 annotation layers). Searches can also retrieve segments "containing" segments. One can also search for segments containing a string.

Where a lexicon is provided (currently only English), users can search for segments containing lexical patterns, for instance, `clause containing 'be% @participle'` would return all clause segments containing any inflection of 'be' immediately followed by any participle verb (i.e. most of the passive clauses). Since dictionaries are used, the text does not need to be pre-tagged with a POS tagger, which may be unreliable on texts of a different nature to those on which the tagger was trained. Results are displayed in a KWIK table format.

6 Automating Annotation

Currently, automatic segmentation into sentences is provided. I am currently working on automatic NP segmentation.

The search facility outlined above can also be used for semi-automatic tagging of text. To autocode segments as 'passive-clause', one specifies a search pattern (i.e., `clause containing`

'be% @participle'). The user is presented with all matches, with a check-box next to each. The user can then uncheck the hits which are false matches, and then click on the "Store" button to tag all checked segments with the 'passive-clause' feature. A reasonable number of syntactic features can be identified in this way.

7 Statistical processing

The tool comes with a statistical analysis interface which allows for specified sub-sections of the corpora (e.g., 'finite-clause in english' vs. 'finite-clause in spanish') to be described or contrasted. Statistics can be of the text itself (e.g., lexical density, pronominal usage, word and segment length, etc.), or relate to the frequency of annotations. These statistics can also be exported in tab-delimited form for processing in more general statistical packages.

8 Intercoder Reliability Testing

Where several users have annotated files at the same layers, a separate tool is provided to compare each annotation document, showing only the differences between coders, and also indicating total coder agreement. The software can also produce a "consensus" version of the annotations, taking the most popular coding where 3 or more coders have coded the document. In this way, each coder can be compared to the consensus (n comparisons), rather than comparing the n! pairs of documents.

9 Annotating Images

The tool can also be used to annotate images instead of text files. In this context, one can swipe regions of the image to create a selection, and assign features to the selection. Since stand-off annotation is used for both text and image, much of the code-base is common between the two applications. The major differences are: i) a different annotation widget is used for text selection than for image selection; ii) segments in text are defined by a tuple: (startchar, endchar), while image segments are defined by a tuple of points ((startx,starty), (endx,endy)), and iii) search in images is restricted to tag searching, while text can be searched for strings and lexical patterns.

10 Conclusions

UAM CorpusTool is perhaps the most user-friendly of the annotation tools available, offering easy installation, an intuitive interface, yet powerful facilities for management of multiple documents annotated at multiple levels.

The main limitation of the tool is that it currently deals only with feature tagging. Future work will add structural tagging, including co-reference linking, rhetorical structuring and syntactic structuring.

The use of the tool is rapidly spreading: in the first 15 months of availability, the tool has been downloaded 1700 times, to 1100 distinct CPUs (with only minimal advertisement). It is being used for various text annotation projects throughout the world, but mostly by individual linguists performing linguistic studies.

UAM CorpusTool is free, available currently for Macintosh and Windows machines. It is not open source at present, delivered as a standalone executable. It is implemented in Python, using TKinter .

Acknowledgments

The development of UAM CorpusTool was partially funded by the Spanish Ministry of Education and Science (MEC) under grant number HUM2005-01728/FILO (the WOSLAC project).

References

C. Müller, and M. Strube. 2006. Multi-Level Annotation of Linguistic Data with MMAX2. In S. Braun, K. Kohn, J. Mukherjee (eds.) *Corpus Technology and Language Pedagogy. New Resources, New Tools, New Methods (English Corpus Linguistics, Vol.3)*. Frankfurt: Peter Lang. 197-214.

H. Cunningham, D. Maynard, K. Bontcheva and V. Tablan. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *Proceedings of the 40th Meeting of the Association for Computational Linguistics (ACL'02)*. Philadelphia, July 2002.

T.S. Morton and J. LaCivita. 2003. WordFreak: An Open Tool for Linguistic Annotation. *Proceedings of HLT-NAACL*. 17-18.

P.V. Ogren 2006. Knowtator: a plug-in for creating training and evaluation data sets for biomedical natural language systems. *Proceedings of the 9th International Protégé Conference*. 73–76.