Designing Focused and Efficient Annotation Tools

D. Reidsma, D. Hofs, N. Jovanović
Human Media Interaction research group, Centre for Telematics and Information Technology, Enschede, Netherlands

Abstract

The creation of large, richly annotated, multimodal corpora of human interactions is an expensive and time consuming task. Support from annotation tools that make the annotation process more efficient is required, especially if the annotation effort involves really large amounts of data. Therefore we investigated how different properties of specific annotation tasks can have an impact on the design of a tool focused on that general class of tasks. In this paper we present our view on the considerations that should drive the design of new tools geared to specific tasks. The main dimensions that we consider are: observation vs interpretation, explicit and implicit input layers, segmentation, feedback, constraints, relations and the content of the annotation elements.

Keywords

large corpora, annotation, tools, reusable design

1 Introduction: Why design focused annotation tools?

Shriberg et al. [12] report an efficiency of 18xRT on annotation of dialog act boundaries, types and adjacency pairs on meeting recordings (i.e. annotation takes 18 times the duration of the video). Simple manual transcription of speech usually takes 10xRT. For more complicated speech transcription such as prosody 100-200xRT has been reported in Syrdal et al. [13]. The cost of syntactic annotation of text (PoS tagging and annotating syntactic structure and labels for nodes and edges) may run to an average of 50 seconds per sentence with an average sentence length of 17.5 tokens (cf. Brants et al. [1], which describes syntactic annotation of a German newspaper corpus). As a final example, Lin et al. [8] report an annotation efficiency of 6.8xRT for annotating MPEG-7 metadata on video using the VideoAnnEx tool (correction of shot boundaries, selecting salient regions in shots and assigning semantic labels from a controlled lexicon). It may be obvious that more complex annotation of video will further increase the cost.

Many large projects face the challenge of annotating a really large amount of data for many different modalities. Given the amount of work needed for each hour of recorded data it still seems useful to invest some effort in designing annotation tools that reduce the work. Tools which are highly efficient for one annotation task are not necessarily so for other tasks. Making a single tool for all conceivable annotations results in a monolithic, unwieldy tool. Nevertheless, designing annotation tools from scratch for each different annotation task is not efficient. This paper presents some ideas which may help reuse design and implementation considerations, which have been applied to the development of several new tools.

2 User types

The users of annotation tools may be divided into the groups described below [2].

- **Annotators:** Users who need a tool for their annotation task. They should not be bothered about data representations, internal design, or API design. A tool should help them work as quickly and efficiently as possible.
- **Corpus Consumers:** Users who want to use annotated data for all kinds of reasons, e.g. theory testing, evaluation and training of models, finding relations between phenomena. They have needs for querying and browsing annotated data.
- **Corpus developers**: Users responsible for corpus design and maintenance (e.g. design of new annotation schemas or altering existing ones, understanding of data representation supported by the tool and mapping of their data to the existing structures).

Since this paper is mostly about designing tools that help reduce annotation effort, we will focus only on the *annotators* in the rest of this paper.

3 Requirements for annotation tools

In the course of this work we also collected tool requirements from a few selected reviews [5,6,7,11]. These reviews together outline most of the criteria used in many papers to rate existing annotation tools or to design new tools. The chosen evaluations are performed from different perspectives reflecting different evaluation goals. The aim of the ISLE Natural Interactivity and Multimodality Working Group report [6] is to provide a survey of world-wide tools which support annotation of natural interactivity and multimodal data. As a result it outlines the most important overall user needs reflected in the tools and projects which created them. The aim of the evaluations presented in [5,7,11] is to select a tool or set of tools based on analysis of research project needs. The reviews follow the same evaluation procedure which consists of two steps. First, based on the analysis of the project needs, a list of requirements for annotation tools is defined (e.g. simplicity, quality assurance, compatibility with other tools, customization of the annotation scheme, etc.). Next, the 'evaluation criteria' are derived. Table 1 lists a reduced version of the collected criteria (the full version can be found in [10]).

The requirements for statistical data analysis and display are supported by software packages that a new tool would hardly displace. Furthermore, the requirements for input/output flexibility, flexibility in coding schemes and querying annotated data are covered by using a stand-off XML data format with a good API such as AGTK, NXT or ATLAS. In this paper we focus only on the annotators as target group and the requirements related to the efficiency of creating annotations such as an easy-to-use interface, marking, audio/ video interfaces, the annotation process and visualization.

4 Characterizing Annotation Problems

Different annotation problems, such as transcription, video labeling and text mark-up, each have their own properties. The properties may have an impact on how the requirements from the previous section are to be interpreted and fulfilled. This section gives an overview of those properties and discusses how they can influence the design of efficient tools for annotation.

4.1 Observation vs interpretation

A specific layer of annotation in a corpus may pertain to *direct observations* of events in the physical world, such as certain movements, speech or gaze directions, or to *interpretations* of those observations, such as emotional states, dialog acts or complex semantic annotations. The interpretations involve deducing information about the internal mental state of the persons involved in the observation, about their beliefs, desires and attitudes [9].

Interpretation takes a lot more time than straightforward observations. Aiming for a real time coding process may be sensible while one is coding observations. When coding interpretations, this may be less feasible. If an annotation is part observation and part interpretation, it may be a good idea to split it up in two different tasks.

4.2 Input layers

Every annotation layer is based on certain sources of input. The most basic layers are based only on the audio and/or video (labeling of head nodding, transcription, hand tracking). More complex layers may also be based on other layers (e.g. dialog acts based on transcriptions, interpretation of gestures for their communicative function). Sometimes the reference from annotation elements to elements in input layers is made explicit, such as dialog acts referring to text fragments. Sometimes this relation is implicit, such as the relation between dialog acts and video or audio: though the explicit input is the speech of the participant, the video and audio offer valuable input for determining the exact dialog act (facial expression, intonation, etc).

Explicit and implicit input layers determine what should be displayed in the tool. An annotation tool should preferably display only the explicit and implicit input layers and the created annotations. Anything else would be a distraction. The explicit input layer should be displayed in a way that clearly shows its relation to the created annotation elements. The explicit input layers also influence the *selection mechanisms* of the tool.

4.3 Segmentation of the input layer

The segmentation properties of an annotation have a major impact on the design of the GUI. The segmentation determines which fragments of the explicit input layer(s) an annotation element can refer to. A list of possible characteristics of the segmentation is given below.

- Segments may or may not relate to overlapping parts of the explicit input layers.
- Segments may or may not interleave with each other.
- Segments may or may not be discontinuous.
- Each segment may be annotated with only one, or more than one, element.
- The segmentation may or may not fully cover the input layer.
- The size of segments differs from problem to problem: single words, sentences, arbitrary fragments, etc.

These properties determine how the selection mechanism should be designed, but also whether semi-automatic support is possible for segmentation and selection. If, for instance, a tool is being developed for manual coding of part-of-speech, the segmentation properties suggest that the tool might perform segmentation automatically and present the segments (i.e. words) one by one for labeling. For dialog acts, the segmentation is not obvious, so it should be done by the annotator.

4.4 Labeling vs complex information

Some annotation layers contain annotation elements that are just labels from a (possibly very complex) set or ontology. Other annotation layers have more complex structures as their constituent elements, such as the multiple labels in MRDA [4].

When the information per annotation element consists of a single label, one can for example decide to map labels to keystrokes or a set of GUI buttons. If the information is more complex, a separate edit panel for annotation elements is probably more suitable.

4.5 Relations between annotated elements

Some annotation elements may define relations between/to other annotation elements. As far as the annotator is concerned, there are two types of relations. One of the related elements may be considered an attribute of the other element, or their relation may be seen as an annotation element in its own right, stored in a separate layer. Depending on how complex the relational structures are, one may consider making the relational coding a separate task.

4.6 Feedback

There are several types of feedback: feedback about the contents of the annotation as the video of the observation is replayed, feedback about which elements and values are currently being annotated and feedback about the ‘whole annotation up till now’. All three types of feedback should be present in an annotation tool, though they need not necessarily be given by the same components.

4.7 Constraints

There may be constraints on element contents and relations (e.g. an answer belongs to a question, certain combinations of tags are not allowed). The tool may help maintain integrity by enforcing those constraints, so limiting the choices of the annotator.

4.8 Default values

A special type of ‘constraint’ is a default value. If a default value for a certain attribute can be defined, the tool can support faster coding by pre-filling the attribute. Syrdal et al. show that, in some cases, default suggestions can speed up manual labeling without introducing too much bias in the annotation [13].

5 Designing annotation tools

Using the characteristics described in the previous section, several components and modules were developed that can be used to develop new annotation tools targeting specific annotation problems with only little extra effort. Two classes of annotation tasks were taken into consideration: discourse labeling and labeling of events in audio or video with a time and duration. Due to space constraints only one of these tasks is discussed partially. It can be seen as an illustration of how the dimensions presented above can help with the design of annotation tools.

The actual modules and tools were developed using the Nite XML Toolkit, an open source toolkit for heavily annotated corpora [3]. The actual annotation tools which have been developed in the course of this work are freely available as part of the Nite XML Toolkit, downloadable from http://www.sourceforge.net/projects/nite

5.1 Discourse labeling

Many annotation tasks involve the labeling of discourse. In the AMI project, in the context of which this work has been done, this means labeling the transcriptions of multi-party dialog. Examples of such tasks are named entity annotation and dialog act annotation. Annotations of this class share several properties along the dimensions described above, and may be different in other properties.

Input layers. For this class of problems, one explicit input layer is the transcription. Other explicit or implicit input layers may contain all kinds of codings which have already been defined on top of that transcription. Therefore it would be useful to have a customizable view that can show a multi-party discussion (dependent on the corpus structure), enhanced with mark-up from existing annotations. Since there are many ways in which the existing annotations can be related to the transcriptions, the mark-up should be highly configurable.

Feedback. Feedback about the annotation can be provided as soon as the transcription view allows visualization of existing mark-up.

Segmentation. With respect to segmentation, discourse labeling tasks may have widely diverging properties. Different types of discourse labeling involve single words or phrases, may or may not span multiple transcription segments, may or may not contain *partial* phrases, may or may not allow overlap between segments, etc. It is very useful if a transcription visualization component is able to support these different types of segmentation explicitly, by allowing or disallowing certain types of selection. The module that has been implemented allows a broad range of selection restrictions to reflect this.

Relations. Relational codings defined on top of a discourse labeling are very common. A module for relational annotation was therefore developed which does not depend on the exact structure of the discourse elements that are related but which nevertheless allows visualization which is integrated with the marked-up transcription.

Result. The result of such reflections is an annotation tool for discourse labeling which can be adapted to many different tasks, either through configuration settings, or, for more complex adaptations, through the extensive API of the modules and components. The tool is centered on a configurable transcription view and an audio/video viewer with time aligned highlighting of the transcription. Using a number of configuration settings, the tool can be used on any corpus defined in the NXT stand-off data format. The tool has already been used for annotation of RST information, dialogue acts and named entities.

6 Conclusions

In this paper we take the position that, to meet the annotation requirements for very large corpora, it may be necessary to develop annotation tools that are specialized to reduce the time effort for creating the annotations. We present our view on the considerations that should drive the design of new tools geared to specific annotation tasks. The main dimensions that we consider are: observation vs interpretation, explicit and implicit input layers, segmentation, feedback, constraints, relations and the content of the annotation elements. Finally, we discuss one example class of annotation problem for which we have designed actual annotation tools, modules and components, using information about these dimensions.

This work was partly supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811, publication AMI-100).

References:

1. Brants, T; Skut, W; Uszkoreit, H (2003). Syntactic annotation of a German newspaper corpus. *Building and using Parsed Corpora*. Kluwer, Dordrecht (NL).
2. Carletta, J; Isard, A; Klein, M; Mengel, M; Moller, M.B (1999). The MATE Annotation Workbench: User Requirements. *Proc of ACL-99 Workshop 'Towards Standards and Tools for Discourse'*.
3. Carletta, J; Evert, S; Heid, U; Kilgour, J; Reidsma, D; Robertson, J; The NITE XML Toolkit. (submitted).
4. Dhillon, R; Bhagat, S; Carvey, H; Shriberg, E (2003). Meeting Recorder Project: Dialog Act Labeling Guide. *Report, ICSI Speech Group, Berkeley, USA*.
5. Dipper, S; Goetze, M; Stede, M (2004). Simple Annotation Tools for Complex Annotation Tasks: an Evaluation. *LREC2004 Workshop on XML-based Richly Annotated Corpora*, Lisboa, Portugal.
6. Dybkjaer, L; Berman, S; Kipp, M; Olsen, M.W; Pirelli, V; Reithinger, N; Soria, C (2001). Survey of existing tools, standards and user needs for annotation of natural interaction and multimodal data. *ISLE NIMM Working Group Deliverable D11.1*
7. Garg, S; Martinovski, B; Robinson, S; Stephan, J; Tetreault, J; Traum, D.R (2004). Evaluation of Transcription and Annotation tools for a Multi-modal, Multi-party dialog corpus. *LREC2004*, Portugal.
8. Lin, C.Y; Tseng, B.L; Smith, J.R (2003). Video Collaborative Annotation Forum: Establishing Ground-Truth Labels on Large Multimedia Datasets. *Proc of the TRECVID2003*, Gaithersburg, Md., USA.
9. Reidsma, D; Rienks, R.J; Jovanović, N (2004). Meeting modeling in the context of multimodal research. *MLMI'04*, Martigny, Switzerland.
10. Reidsma, D; Jovanović, N; Hofs, D (2004). Designing Annotation Tools based on the Properties of Annotation Problems. *Report, Centre for Telematics and Information Technology.*
11. Rydeman, B (2003). Using Multimodal Annotation Tools in the Study of Multimodal Communication Involving Non speaking Persons. *Report, Goteborg University.*
12. Shriberg, E; Dhillon, R; Bhagat, S; et al (2004). The ICSI Meeting Recorder Dialog Act (MRDA) Corpus. *HLT-NAACL SIGDIAL Workshop*. Boston, USA.
13. Syrdal, A.K; Hirschberg, J; McGory, J; Beckman, M (2001). Automatic ToBi prediction and Alignment to speed manual labeling of Prosody. *Speech communication,* 33, 135-151.

CRITERIA	1	2	3	4	Annotator?
Portability		X	X	X	YES
Can the tool be used on different platforms? Does it require any additional packages? [2] Is it easy to install? [4]					
Flexible architecture	X				
Allows extension of the tool by adding new components.					
I/O flexibility	X	X		X	
What are the tool's input formats? Does the input data need any preprocessing? Is the output format compatible with other tools? Are there converters from/to other formats provided? Can annotation scheme be imported/exported and in which format?[4]					
Robustness and stability	X		X		YES
Is the tool robust, stable and fast?					
Audio/video interface		X			YES
Does the tool offer an easy-to-use method for playing audio and/or video and for segmenting it? Does the tool support handling large media files? Does the tool support playing back the media file aligned with an annotation element?					
Flexibility in coding scheme	X	X	X	X	
Does the tool support easy addition of a new coding scheme or altering of the existing one? [1,2,3] Does the tool allow user to restrict format and/or the content of annotation data? [4] Can annotation levels be defined as obligatory or optional? [4] Can tag sets be specified? Can tag sets be structured? [4] Are annotation levels and tag sets defined within the tool or by external files? [4]					
Easy to use interface	X	X	X	X	YES
The interface should support users as much as possible, be intuitive and based on standard interfaces conventions					
Learnability				X	YES
Is the tool easy to learn?					
Attractiveness				X	YES
Does the user enjoy working with tool?					
Transcription support	X			X	YES
Can the tool be used for speech transcription?					
Marking	X	X	X	X	YES
Does the tool support annotations at different levels, of different modalities and annotations across levels and modalities? How much can the tool mark (e.g. just words or group of words; entire sentences or segments of sentences)? Does it allow the marking of discontinuous fragments? [2] Does the tool support simultaneous annotation for several persons? [3]					
Meta-data		X		X	YES
Does the tool support 'meta-data' such as annotators' comments and notes referring to annotations or relating to the entire document?					
Annotation process				X	YES
Does the tool support some kind of (semi) automatic annotation? Does the tool support selection-based annotation where only appropriate the tags are presented to the user?					
Visualization	X	X	X	X	YES
Scope: Is the annotated information visible for all annotation elements or only the currently active element? Style: How are the annotated elements presented? [4] Can the user change visualization dynamically? Can the user define visualization? [1,4] Does the tool support synchronized view of different annotation layers and of different modalities? [1] Does the tool have a large display to show the current works and corresponding data in a clear manner? [2]					
Documentation		X		X	YES
Availability and quality of user manual; on-line help					
Querying, extraction	X			X	
Does the tool support (simple or powerful) search mechanisms and an interface to the search tool? Are the results presented in an intuitive and easy-to use way?					
Data analysis	X				
Does the tool support (statistical) analysis of annotated data?					

1: Dybkjaer et al. [6]
3: Rydeman [11]
2: Garg et al. [7]
4: Dipper et al. [5]

'Annotator?' marks whether the requirement is relevant for the annotator.

Table 1. Collected requirements for annotation tools.