Simple Annotation Tools for Complex Annotation Tasks: an Evaluation

Stefanie Dipper, Michael Götze, Manfred Stede

University of Potsdam
Dept. of Linguistics, Computational Linguistics
D-14415 Potsdam, Germany
{dipper,goetze,stede}@ling.uni-potsdam.de

Abstract

This paper presents a comparative evaluation of ready-to-use, XML-based tools for annotating linguistic data. We start by describing our research project that deals with the creation and annotation of empirical data related to information structure. Based on the requirements of this project and the data, we develop a set of evaluation criteria and apply them in the evaluation of five selected annotation tools.

1. Introduction

Linguistic research based on real-life data has become more and more prominent during the last years. As a consequence, the need for corpora that are (i) large and (ii) richly annotated has grown as well. First, corpora that consist of real-life data such as newspaper texts or recorded dialogues must be large enough to offer enough instances of the phenomena under study. Second, many linguistic phenomena involve factors of different linguistic domains; for instance, word order in German is supposed to depend, among other things, on grammatical functions (syntax), thematic roles (semantic), information structure, and intonation (phonetics). Investigations of such phenomena require corpora that are annotated with detailed information at various linguistic levels.

The creation of large and richly annotated corpora is a time-consuming and expensive task. Whereas morphological and syntactic annotation may be supported, if not taken over, by trained taggers and parsers, the situation is different for the annotation of, e.g., semantic or discourse-related properties. Here, informed linguists have to perform all (or large parts) of the annotation task. Hence, people tend to restrict the data they are going to annotate to *relevant* data, i.e., data featuring the phenomenon in question. The resulting corpora are rather small but may be richly annotated. In such scenarios, the creation of a corpus is a side issue, which should not take up much time or effort. Hence, easy-to-use annotation tools that support manual annotation in a suitable way are desirable. Since the development of such tools is extremely time-consuming and expensive, reuse of already existing tools is to be preferred.

This paper grew out of our work in the Sonderforschungsbereich (SFB, collaborative research center) on information structure at the University of Potsdam[1]. In the context of this SFB, a lot of data of diverse languages will be collected and annotated on various annotation levels. In order to maximize the benefit of this data, we make use of an XML-based encoding standard to facilitate data exchange and reuse. The XML representation will be fed into a database that offers visualization and search facilities.[2]

This paper presents a survey and evaluation of selected, XML-based tools that can be applied in manual annotation. For the evaluation, we developed a set of criteria, based on the SFB requirements. We believe, however, that these criteria are relevant not only to the SFB but to many projects that deal with complex, multi-level annotation. We then applied these criteria to selected annotation tools.

Based on our user-oriented criteria, we believe that, at least in the short run, ready-to-use tools (i.e., tools which are easy to get used to, especially by users without programming skills) serve the annotator better than complex tool kits, which require adaptations by the user (as also argued by Orăsan, 2003).

The paper is organized as follows. First, we describe the context and requirements of the SFB. We then turn to the presentation of the criteria we have developed. Finally, we present the results of the evaluation and give a summary.

2. Requirements

In this section, we present our research project and describe the annotation scenario. Based on this, we formulate requirements for annotation tools, which we believe to be of relevance for similar annotation efforts.

2.1. The Project Context

The SFB "Information structure: the linguistic means for structuring utterances, sentences and texts" consists of 12 individual research projects from disciplines such as theoretical linguistics, psycholinguistics, first and second language acquisition, typology, and historical linguistics. The overarching objective of these projects is the investigation of information structure (IS). This is an area well-known to be prone to terminological or even conceptual confusion—many different theories of how to partition utterances into IS-relevant segments compete with each other, and, furthermore, there is little agreement on what level(s) of utterance representation IS should be located. In a situation like this, the availability of annotated data, which allows for comparing, sharing, and further developing the underlying ideas, is very important. The collection and distribution of empirical data is thus an important objective in the SFB. This concerns in particular the following projects:

[1] `http://www.ling.uni-potsdam.de/sfb/`

[2] The XML encoding standard and the database are under development. Our current work focuses on the annotation task, including the choice of annotation tools and the development of annotation standards.

Semantic annotation The project "A2: Quantification and information structure" examines the relation of quantifier scope and IS and will annotate semantic features such as quantifier scope, identifiability, and definiteness.

Discourse annotation "A3: Rhetorical structure in spoken language: modeling of global prosodic parameters" investigates the correlation between rhetorical and prosodic structure of spoken discourse. Data consist of radio news and newspaper commentaries.

Focus annotation in African languages The projects "B1: Focus in Gur and Kwa languages" and "B2: Focussing in African Tchadic languages" examine the phenomenon of focus in Western African languages. Both projects carry out field studies.

Diachronic data The project "B3: The role of information structure in the development of word order regularities in Germanic" investigates the evolution of the verb-second phenomenon, which occurred in certain Germanic languages only (e.g., in Modern German as opposed to Modern English). Based on language data of Old High German and Old English, the role of IS in this evolution will be studied.

Typology of information structure "D2: Typology of information structure" focuses on the development of a typology of the means for expressing IS. In close cooperation with the other projects, a questionnaire will be developed, which will serve as a basis to collect language data relevant for IS from typologically diverse languages.

One of the main objectives of the SFB is to determine the factors that play a role in IS. Hence, it is highly desirable that each project can profit from the data collected and annotated by the other projects. This presupposes compliance to certain standards, (i) an annotation standard and (ii) an encoding standard. First, the annotated data must be understandable and comparable. Therefore, SFB-wide working groups are defining an SFB Annotation Standard with tagsets and annotation guidelines for morphosyntax, prosody, semantics/pragmatics, and information structure. Second, we are developing an SFB Encoding Standard, an XML-based stand-off representation of the data, which will serve as the common exchange format within the SFB and thus support the standardization process.

Figure 1 gives an overview of the data flow in the project: a number of different projects will collect and annotate data according to the common SFB Annotation Standard, using a small set of annotation tools. The annotated data will be mapped to the SFB Encoding Standard, which serves as the common basis for further processing. This includes a web-based linguistic database, which provides visualization and retrieval of the SFB data, both by the members of the SFB and the research community.

The circumstances of the annotation differ: parts of the annotation will be done under conditions of fieldwork (as in the projects B1 and B2). Some tagsets to be applied are available, others will have to be created or developed further. Common to all of the projects are the limited resources available for the annotation task: annotation represents only one aspect of the project work and is usually not the main focus. Furthermore, some projects have no or little experience with annotating data at the levels mentioned.

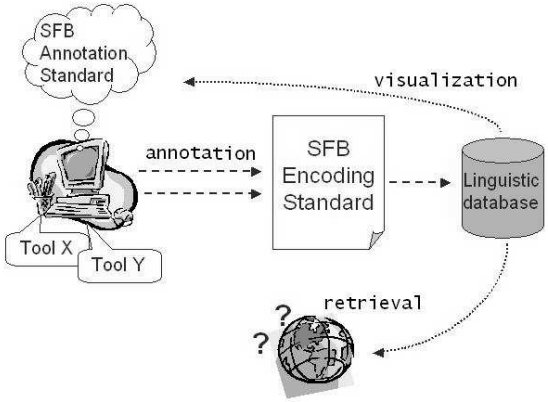

Figure 1: Data flow in the SFB

2.2. Requirements for Annotation Tools

The described scenario is typical for cross-language research based on empirical data and focusing on the investigation of phenomena that require annotation on multiple linguistic levels. Based on the analysis of the needs of the SFB, we define the following list of requirements for annotation tools.

Diversity of data Language data to be annotated differs with respect to modality (written vs. spoken language, monologue vs. dialogue) and basic unit (sentence vs. discourse). In addition, special character sets (e.g., for Kwa languages) must be supported.

Multi-level annotation A very central requirement is support of annotation on multiple levels, each level representing one type of information, e.g. morphemic transcription, grammatical functions, pitch accents, etc.

Diversity of annotation Data types of the annotated information range from attribute-value pairs to set relations (e.g., for annotating co-reference), directed relations/pointers (e.g., for annotating anaphoric relations), trees, and graphs. Furthermore, it might be desirable to allow for annotations relating different levels ("cross-level annotation").

Simplicity The annotation tools must be simple tools, for several reasons. Users of annotation tools in the described scenario usually have little or no prior knowledge about annotation tools. Moreover, for less-studied languages, the researcher has to collect the data during field studies, which means that often there will be no technical support available. Finally, annotating data forms only a small part of the researcher's tasks; hence, using annotation tools should be as simple and intuitive as possible.

Customizability Usually, the development of suitable tagsets (including annotation guidelines) and the actual annotation are not independent tasks but affect each other. Suitability of tagsets and guidelines has to be proven in practice, i.e., by successful, consistent annotation. This

means that in the beginning phase, tagset definitions may change quite often. Tools should therefore allow for easy customization.

Quality assurance The annotated linguistic data is a central resource for the SFB research on IS. Hence, high-quality annotation is an important issue. Quality concerns consistency and completeness of the annotation as well as compliance to the SFB Annotation and Encoding Standards. The annotation tools should support these aspects in annotation.

Convertibility Support of data conversion is important for several reasons: First of all, this facilitates reuse of existing linguistic resources (e.g., treebanks or speech corpora). In addition, it supports standardization, since data usually have to be transformed into common standard formats (in our project: the SFB Encoding Standard). Finally, data convertibility is a prerequisite for applying specialized tools to the same data: tool X for transcription of the data, tool Y for annotation on multiple levels, and tool Z for for posing complex cross-level queries.

Tools may support convertibility in two ways: (i) by providing a standardized input and output format, which allows the user to easily convert the data; (ii) by providing ready converters from/to other tools.

It is important to note that the individual requirements might be of different relevance to different annotation projects. Their relative importance might also change over time: for instance, users could gather experience and would like to use more elaborate tools; the need for customizability might decrease; etc.

At the current stage of the SFB, the requirements of Simplicity, Quality Assurance, and Convertibility represent the most crucial needs. Later, issues like the support of more complex annotation (such as cross-level annotation or the annotation of ambiguous phenomena) will pose further challenges.

We now move to the tool evaluation criteria, which we derive from the requirements presented in this section.

3. Criteria

In this section, we first present the criteria we applied in choosing candidate tools ("selection criteria"). We then define the criteria for evaluating the individual tools, in the form of a feature checklist.

In line with the suggestions of standardization groups working on software evaluation (ISO, 2001; EAGLES, 1996), our evaluation starts from the user's needs. That is, both the choice of tools to be evaluated as well as the choice of evaluation criteria are guided by the user requirements that have been described above.

Note that our criteria do not test for highly detailed tool features (compared to, e.g., the feature checklist for translation memory by EAGLES (EAGLES, 1996)). This is because the tools we are comparing have been developed for different purposes and therefore exhibit many differing features, which we compare at a quite abstract level only.

3.1. Selection Criteria

We regard the following criteria as highly relevant for the SFB's annotation scenario and thus use them to restrict the set of tool candidates that we evaluate.

XML-based The tools must provide for an XML-based export and import format. This eases the data transfer between the annotation tools and the SFB-internal Encoding Standard.

Maintenance Maintenance of the tools should be guaranteed, hence we focus on tools that are being actively supported.

Ready and easy to use At the present stage, we consider tools that are ready and easy to use, i.e., installation and use of the tool must not require advanced programming skills. The end user (the annotator) should be able to apply the tool with little or no support.

Linguistic tools For similar reasons, we restrict the evaluation to tools developed and tailored specifically for linguistic purposes. That is, we exclude general-purpose tools such as XML editors.

Portability The tools must run on any platform and must be easy to install.

Cost The tools must be available free of charge for research purposes.

3.2. Evaluation Criteria

We developed a checklist of features to evaluate the tools one by one. These features can be classified according to the quality characteristics proposed by the ISO 9126-1 standard (ISO, 2001). Our features exemplify the ISO characteristics of "Functionality" and "Usability".

Functionality The aspect of Functionality concerns the presence or absence of functions that are relevant for a specified task. Roughly speaking, Functionality concerns the relation tool–task.

In our context, the ISO subcharacteristics of "Suitability" and "Interoperability" (which belong to the more general aspect of Functionality) are relevant.

- Suitability indicates whether a tool provides appropriate functions for the specified task.
- Interoperability concerns the capability of the tool to interact with other systems.

Usability In contrast to Functionality, Usability takes user aspects into consideration by evaluating the effort needed for use; i.e., it concerns the relation tool–user. In the SFB context, the following ISO subcharacteristics of Usability are important:

- "Learnability": Is the tool easy to learn?
- "Attractiveness": Does the user enjoy using the tool?
- "Documentation" measures the availability and quality of documentation.
- "Compliance": Does the tool adhere to standards/conventions relating to usability? For instance, for tasks such as text editing: does the tool provide features known from common text editors?
- "Operability": Is the tool easy to operate?

In the following paragraphs, we relate the ISO characteristics to the SFB requirements presented in 2.2. and define concrete criteria that instantiate these characteristics.

3.2.1. Functionality (I): Suitability

This aspect concerns the presence/absence of appropriate functions. Referring to the SFB requirements, Suitability indicates the tool's appropriateness with regard to the requirements of Diversity of data, Multi-level annotation, and Diversity of annotation.

The concrete criteria that we define to measure suitability of the tools concern source data and annotated data:

Primary/source data This criterion covers properties of the source data (i.e., the data that are input to the tool).

(1) Modality: Which input formats does the tool allow for?
   (a) discourse (sequence of sentences)
   (b) speech/audio
   (c) video
   (d) monologue
   (e) dialogue

(2) Preprocessing: Does the primary data need any preprocessing before annotation can start (e.g., is tokenization necessary)?

(3) Unicode: Does the tool support Unicode for the representation of special characters?

Secondary data This criterion concerns properties of the annotations.

(4) Markables (segments): The basic units referenced by the annotation are defined by inclusion/embedding (e.g., <markable>...</markable>) vs. specifying a start and end point (e.g., <markable span="id_2..id_4"/>).[3]

(5) Data structure: Secondary data consist of:
   (a) atomic features of a markable (e.g., part-of-speech tags)
   (b) relations between markables: (undirected) relations, pointers
   (c) dominance relations: bracketing, trees/graphs
   (d) conflicting hierarchies (e.g., overlapping markables or trees can be defined)

(6) Metadata: Can meta-information be annotated?
   (a) header: meta-information relating to the entire document (e.g., header data such as the author of an input text)
   (b) comments: referring to specific basic units or annotations

(7) Unicode: Does the tool support Unicode in the secondary data?

[3] Only tools that specify markables by their start and end point may represent conflicting hierarchies, see below.

3.2.2. Functionality (II): Interoperability

The aspect of Interoperability relates to interface properties, including the interaction with other tools. (All selected tools provide for an XML-based export and import format.) This feature covers the SFB requirement of Convertibility. We define the following criteria:

(8) Export and import
   (a) is stand-off representation supported?
   (b) can annotation schemes (see below) be imported/exported and if yes, in which format?

(9) Converters: Are converters from/to other tool formats provided?[4]

(10) Plug-ins: Is it possible to attach other tools?

3.2.3. Usability (I): Learnability/Attractiveness

The aspects of Learnability and tool Attractiveness—people should as much as possible enjoy annotation—relate to the SFB requirement of Simplicity. Since they are of central importance in our context, we performed a separate study of these issues, see Section 4.3.

3.2.4. Usability (II): Operability

We consider the aspect of Operability ("Is the tool easy to operate?") to cover the SFB requirements Simplicity, Customizability[5], and Quality assurance.

The criteria we define to measure Operability cover tool features that are tailored to the actual task of annotation. They concern features related to annotation schemes and the annotation process.

Specifying annotation schemes This criterion concerns tool features that allow the user to restrict the format and/or content of the annotation data (secondary data); it covers important aspects of Customizability.

(1) Annotation levels: Can levels be defined as obligatory, optional?

(2) Annotation tagsets: Can tagsets (i.e., admissible tag values) be specified? If yes, can the tagsets be structured, i.e., is it possible to define interdependencies between tag specifications? (For instance, the user is prompted to annotate the type of anaphoric reference only if the markable in question is marked as being anaphoric.)

(3) Specification: Are annotation levels or tagsets defined by external files or within the tool?

[4] Some tools provide APIs for further processing of the data, including conversion to other formats. However, the use of APIs requires programming skills, which we do not expect the user to have. Hence, we do not take API support into account.

[5] The requirement of Customizability could just as well be considered as reflecting the ISO characteristic of "Maintainability" (King, 2001).

Annotation process This criterion concerns properties of the annotation process.

(4) Automatic annotation: Does the tool support some kind of automatic annotation? (For instance, based on previously annotated data, the tool makes suggestions the annotator can accept or reject.)

(5) Selection-based: Does the tool support selection-based annotation? (For instance, only tags and tag values that are defined by annotation schemes are presented to the user.)

(6) Visualization: How is the annotated information presented?

  (a) scope: the annotated information is visible for all markables vs. only for the currently active markable (= the markable "in focus")

  (b) style: how is the annotated information displayed? (annotation as, e.g., text, XML source, or menu/radio button)

  (c) additional highlighting: does the tool provide further means to visualize the annotated information? (e.g., by coloring, font size/type, brackets, etc.)[6]

  (d) reference units of additional highlighting: do the additional highlightings in (c) refer to features or feature values? (e.g., all markables that are annotated for the feature "case" are highlighted vs. only markables with a specific case feature, e.g., "case = ergative", are highlighted)

  (e) user adaptation: can the visualization be changed dynamically by the user (e.g., by temporarily hiding certain annotation levels, by modifying coloring, font size, etc.)?

  (f) user definition: can the visualization be defined by the user?

(7) Search: Does the tool integrate a simple search facility (for primary and/or secondary data)?

3.2.5. Usability (III): Documentation

The aspect of Documentation relates to the SFB requirement of Simplicity. It refers to the availability and quality of:

(8) general documentation

(9) help (problem-specific documentation)

(10) example files, which can be loaded and modified

(11) tutorial (detailed walk-through)

3.2.6. Usability (IV): Compliance

Compliance ("Does the tool adhere to standards?") again relates to the requirement of Simplicity. We define criteria that concern features known from common document processing tools:

(12) Mouse vs. keyboard: Are there shortkeys for all (important) actions?

(13) Editing etc.: Does the tool provide undo/redo/autosave/...

(14) Unicode: Is there any input support for Unicode?

4. Evaluation

This section presents the results of the evaluation. First, however, the tools selected for evaluation are shortly described. Then the results of the feature checklist are given. In addition, we present results of a questionnaire focusing on tool usability. Finally, we present implications that our evaluation might have for the choice of annotation tools.

4.1. The Evaluated Tools

Given the selection criteria outlined above, we found the following tools to be suitable candidates.

TASX Annotator[7] 'Time Aligned Signal data eXchange Format' (Milde and Gut, 2002). The TASX Annotator allows transcription and annotation of speech and video data on multiple levels.

EXMARaLDA[8] 'EXtensible MARkup Language for Discourse Annotation' (Schmidt, 2001). EXMARaLDA aims at the multimodal transcription and analysis of discourse.

Since the TASX Annotator and EXMARaLDA specialize for speech annotation, the annotated information is represented by tiers and refers to segments ("events") that are defined with respect to a common timeline.

MMAX[9] 'Multi-Modal Annotation in XML' (Müller and Strube, 2001). MMAX is a tool for annotation of text and dialogue, following a strongly relation-based annotation paradigm.

PALinkA[10] 'Perspicuous and Adjustable Links Annotator' (Orăsan, 2003). PALinkA is an annotation tool that has been employed in several discourse-related tasks.

Systemic Coder[11] The Systemic Coder was initially developed in the context of a discourse analysis project.

The ready-to-use criterion excludes multi-purpose tool kits such as the Annotation Graph Toolkit[12] (AGTK), the NITE XML Toolkit[13], and CLaRK[14]. The issue of customizing a powerful toolkit to the needs of the SFB projects might be reconsidered at a later stage, when standards, formats, annotation and retrieval procedures in the SFB have matured.

[6] For the focus-based tools MMAX and PALinkA, additional highlighting concerns the annotation of all markables, not just the markable "in focus"—in contrast to (b).

[7] http://tasxforce.lili.uni-bielefeld.de/

[8] http://www.rrz.uni-hamburg.de/exmaralda/index.html

[9] http://www.eml-research.de/english/Research/NLP/Publications

[10] http://clg.wlv.ac.uk/projects/PALinkA/

[11] http://www.wagsoft.com/Coder/

[12] http://sourceforge.net/projects/agtk/

[13] http://sourceforge.net/projects/nite/

[14] http://www.bultreebank.org/clark/

4.2. Results of the Feature Checklist

The detailed results of the feature checklist evaluation are presented by the tables in Figures 3 and 4 . Figure 3 presents criteria measuring Functionality, Figure 4 lists criteria measuring Usability. The criteria are numbered according to Section 3.2.; '+' means: "feature (as defined in Section 3.2.) is available", '–' means "feature is not available".

These are the prominent findings (focusing on the SFB-relevant criteria):

Simplicity

- 'Ready-to-use': Here, the TASX Annotator and EXMaRaLDA perform best: They do not require any data preprocessing; no tagsets must (and can) be defined. The copy-and-paste function (see footnote [1] in Figure 3) allows for a quick start.

  With the Coder, the user has to specify annotation tagsets before annotation can start; however, the Coder supports tool-internal defining of tagsets.

  Finally, with MMAX and PALinkA, the user must preprocess the input text and define tagsets externally. Both requires an understanding of the XML format that underlies the data and tagset representation, respectively.
- EXMARaLDA offers a tutorial (in German), which allows even unexperienced users to get access to the tools on their own.

Quality assurance

- Predefined tagsets (MMAX, PALinkA, Coder) improve the quality of annotation (at the cost of simplicity), by defining admissible features and/or feature values; this improves consistency of annotation. Moreover, it improves completeness of annotation, by prompting the user to annotate the predefined tagsets. Finally, structured tagsets (MMAX, Coder) can be used for modeling decision trees, which guide the user through the annotation task.
- Good visualization is important. The tier-based tools (TASX Annotator, EXMARaLDA) display the annotated information in a straightforward way. The primary data and annotation layers are presented by horizontal tiers. That is, a sequence of adjacent markables and the associated annotations can be inspected simultaneously. However, only a small part of primary data can be viewed at the same time, which is a disadvantage for the annotation of phenomena that involve larger spans of discourse (such as discourse or anaphoric relations).
  In contrast, the focus-based tools (MMAX, PALinkA, and Coder) allow for concurrent visualization of a large amount of primary data, while annotated information is displayed for only one markable in turn. This drawback is partly compensated by annotation-dependent coloring of the primary data. The search facility provided by MMAX even allows for highlighting markables with feature combinations on different annotation levels (e.g. direct objects marked as topic).

Convertibility

- All of the selected tools offer XML-based import and export formats. Hence, all support convertibility in this aspect.
- In addition, some of the evaluated tools offer good opportunities for working with the same data in several 'special-purpose' tools (tools for annotation, visualization, querying). As the evaluation table shows (see footnotes [5]+[6] in Figure 3), the tier-based tools (TASX Annotator, EXMARaLDA) offer a lot of transformation opportunities.

Multi-level annotation/Diversity of annotation and data

- All tools support multi-level annotation. However, they differ with regard to the data structures of the annotated information. PALinkA and MMAX are the only tools that allow for structural annotation (by pointers, brackets).
- Only the TASX Annotator allows for direct annotation of audio and video data. With the other tools, this kind of data has to be annotated via an intermediate textual representation.

4.3. Results of the Usability Questionnaire

Since Usability is an important aspect for our annotation scenario, we decided to conduct an additional study with the future annotators of the SFB. We therefore provided a one-day tutorial about the annotation tools. After the tutorial, we asked the participants to fill in a questionnaire, reporting about their subjective impressions, covering aspects of Usability such as Attractiveness, Learnability and Operability. Due to time limitations, we considered only three of the tools: EXMARaLDA, MMAX and PALinkA.

A further goal of the tutorial was to get the annotators acquainted with a set of annotation tools and to enable them to work with the tools on their own. We therefore first introduced the basic functionality of each tool by demonstrating and practising segmentation and tag assignment, focusing on a simple annotation task on sentence level. After that, we addressed the process of preparing the primary data (preprocessing, tokenization) and the customization of tagsets (for MMAX and PALinkA only).

The most noteworthy results of the questionnaire are:

- The participants were most satisfied with the visualization in EXMARaLDA, where the annotation of sequences of markables can be inspected simultaneously. The XML-like visualization in PALinkA was criticized because of its poor readability. Apparently, additional means of visualizing annotated information (such as coloring, brackets) did not offer sufficient support. This means that visualization plays a highly important role in the annotation process.
- In the tutorial, we provided scripts for external preprocessing and tokenizing. Nevertheless, the preparation of the primary data remained difficult for the participants.

	TASX	EXMARaLDA	MMAX	PALinkA	Coder
Immediate Annotation	+	+	–	–	–
Consistent Annotation	0	0	+	+	+
Guided Annotation	–	–	+	0	+

Figure 2: Suitability according to the annotation scenario

- Customization of tagsets, which has to be performed tool-externally, was considered to be too complex by most of the participants. Understanding and modifying tagset specification formats requires more than can be expected from many users of annotation tools.

4.4. Implications

How can the findings of this section help the users to decide which annotation tool fits their requirements best? Viewed from the perspective of the purpose of an annotation task, we can distinguish three types of annotation scenarios:

Immediate annotation Immediate annotation implies that the tool allows the user to start the annotation without preparatory work. This requirement may be typical of preliminary, experimental annotations of a small amount of selected data.

Consistent annotation This requirement is important for the creation of high-quality corpora with complex (multilevel) annotation.

Guided annotation The annotation of certain phenomena require detailed and complex annotation guidelines, consisting of decision trees and lists of annotation criteria that the annotator has to check for. In such a scenario, guided annotation may model (parts of) the annotation guidelines.

The table in Figure 2 estimates the suitability of the evaluated tools with regard to these requirements ('+' means 'well suited for the annotation scenario', '–' means 'not suited', '0' is neutral).

5. Conclusion

In this paper, we presented selected XML-based tools that can be applied in manual annotation of language data. Due to the requirements of the SFB, we decided to focus on ready-to-use tools, which would not require programming skills.

On the base of a list of requirements, we developed a set of evaluation criteria for these tools, covering aspects of functionality and usability. Inspecting the results of this evaluation, we can state that these tools fulfill many of the criteria and offer a lot of support for the annotator. That is, the use of a small set of ready-to-use tools can be seen a worthwhile alternative to the application of complex toolkits, even for the multilevel and complex annotations the SFB is aiming at.

However, practice showed that the tools still require considerable effort for many users. The central drawbacks of the evaluated tools concern the visualization of the annotation, preprocessing of primary data, and tagset customization.

Our conclusions are therefore:

- Suitable visualization of the annotated information is highly important.
- A tool-internal preprocessing facility would render the tools more 'ready to use'.[15]
- A tool-internal interface for the specification of own tagsets would be an important step forward.

There is, of course, little value in seeking a "final ranking" for such a comparative evaluation of tools. Instead, it is clear that the annotation scenario determines which tools are suitable and which are not. We have suggested three such scenarios and provided a comparison of the tools along those lines (Figure 2). However, the potential users are encouraged to define their own, specific annotation scenario in terms of the fine-grained features we provided, and then peruse the information in Figures 3 and 4.

6. References

EAGLES, 1996. Evaluation of natural language processing systems. Final report. EAGLES DOCUMENT EAG-EWG-PR.2. Version of October 1996; `http://issco-www.unige.ch/projects/ewg96/ewg96.html`.

ISO, 2001. ISO/IEC 9126-1:2001: Software engineering – product quality – part 1: Quality model. `http://www.iso.org`.

King, Maghi, 2001. Standards work related to evaluation. MTEval Workshop Geneva, Hand-out; `http://www.issco.unige.ch/projects/isle/mteval-april01/maghi-isonew.html`.

Milde, Jan-Torsten and Ulrike Gut, 2002. The TASX-environment: an XML-based toolset for time aligned speech corpora. In *Proceedings of the third international conference on language resources and evaluation (LREC 2002)*. Gran Canaria, Spain.

Müller, Christoph and Michael Strube, 2001. MMAX: A tool for the annotation of multi-modal corpora. In *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*. Seattle, WA.

Orăsan, Constantin, 2003. PALinkA: A highly customisable tool for discourse annotation. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue*. Sapporo, Japan.

Schmidt, Thomas, 2001. The transcription system EXMARaLDA: An application of the annotation graph formalism as the basis of a database of multilingual spoken discourse. In *Proceedings of the IRCS Workshop on Linguistic Databases*. Philadelphia, PA.

[15] Such a facility is provided by the Coder: It enables the import of plain text and its segmentation into sentences, for instance.

Criterion	TASX	EXMARaLDA	MMAX	PALinkA	Coder
(I) Suitability					
Primary data:					
(1) Modality					
(a) Discourse	+	+	+	+	+
(b) Audio	+	–	–	–	–
(c) Video	+	–	–	–	–
(d) Monologue	+	+	+	+	+
(e) Dialogue	+	+	+	–	–
(2) Preprocessing	optional [1]	optional [1]	obligatory	obligatory	optional [1]
(3) Unicode	+	+	+	+	+
.....	.....	.....	.....	.....	.....
Secondary data:					
(4) Markables	start/end	start/end	start/end	inclusion	inclusion
(5) Data structure					
(a) Atomic features	+	+	+	+	+
(b) Relations	–	–	undirected rel., pointer	pointer	–
(c) Dominance rel.	–	–	–	bracketing	–
(d) Conflicting hier.	–	–	+	–	–
(6) Metadata					
(a) Header	+	+	–	+	–
(b) Comments	– [2]	– [2]	– [2]	– [2]	+
(7) Unicode	+	+	–	–	–
(II) Interoperability					
(8) Export/Import					
(a) Stand-off	–	–	+	–	–
(b) Annot. schemes	[3]	[3]	+, XML	+, text	+, text
(9) Converters [4]					
(a) Import	+ [5]	+ [6]	–	–	–
(b) Export	+ [5]	+ [6]	–	–	–
(10) Plug-ins [4]	+ [5]	–	–	–	–

[1] Primary data may be imported both in tokenized or untokenized format. TASX/EXMARaLDA: If untokenized data (= plain text) is to be imported, the data must be imported via copy and paste. Coder: Plain text files can be imported.

[2] These tools do not provide extra means for encoding comments. However, comments can easily be encoded as an ordinary annotation.

[3] TASX/EXMARaLDA: These tools do not allow for specification of annotation schemes, hence export/import of annotation schemes is not an issue.

[4] The given lists of converters and plug-ins are taken from the TASX and EXMARaLDA documentation. We did not check their functionality.

[5] The TASX Annotator provides import and export converters for Annotation Graphs, EXMARaLDA, Praat-label, ESPS-label, ESPS-freq. In addition, it provides import converters for Anvil, SyncWriter, Transcriber (STM), and export converters for NITE, HTML. Finally, it comes with plug-ins for Praat (Spectrogram, Pitch), sox.

[6] EXMARaLDA provides import and export converters for TASX, Praat TextGrid, ELAN Annotation File. In addition, it provides import converters for HIAT-DOS, ExSync Data, and export converters for AIF (Atlas Interchange Format), HTML partitur, RTF partitur.

Figure 3: Functionality evaluation

Criterion	TASX	EXMARaLDA	MMAX	PALinkA	Coder
(II) Operability					
Specifying annotation schemes:					
(1) Annotation levels	–	– [1]	–	–	–
(2) Annot. tagsets	–	–	+, structured	+	+, structured
(3) Specification	[1]	[1]	external	external	internal, extern.
.....	.....	.....	.....	.....	.....
Annotation process:					
(4) Automatic annot.	+ [2]	–	–	+ [3]	–
(5) Selection-based	–	–	+	+	+
(6) Visualization					
(a) Scope	all	all	focus	focus	focus
(b) Style	text	text	choice menu	XML	text
(c) Additional Highlighting	(+) [4]	coloring, font type, font size	(+) [4]	coloring, brackets [5]	coloring
(d) Reference unit	(feat, value) [4]	feat	(feat, value) [4]	feat	value
(e) User adaptation	tier hiding	tier hiding	–	+	–
(f) User definition	(+) [4]	–	(+) [4]	+	–
(7) Search	prim., second.	prim., second.	secondary	primary	primary
(III) Documentation					
(8) Documentation	+	+	+	(+) [6]	+
(9) Help	+	[7]	[7]	[7]	[7]
(10) Example files	–	+	+	+	+
(11) Tutorial	–	+	–	–	–
(IV) Compliance					
(12) Shortkeys	+	+	–	–	–
(13) Editing etc.					
(a) Undo/redo	undo, redo	–	undo (once)	undo, redo	–
(b) Cut/copy/paste	+	+	(+) [8]	–	(+) [8]
(c) Search/replace	+	+	–	–	–
(d) Autosave	+	–	–	+	–
(14) Unicode	virtual keyb.	virtual keyb.	–	–	–

[1] TASX/EXMARaLDA: These tools do not allow for explicit specification of annotation schemes. Within EXMARaLDA, however, XML elements specifying annotation levels may be added to the input data, thus simulating the definition of annotation levels.
[2] The TASX Annotator provides a completion function for the annotated information, by suggesting word completions (which can be accepted or rejected).
[3] PALinkA provides suggestions for annotation by taking previously annotated data into account.
[4] TASX and MMAX: Feature and value-depending coloring and fonts can be defined by the user. However, the definitions must be done tool-externally, by XSLT stylesheets. MMAX offers a query function that can be used to mark values.
[5] PALinkA allows the insertion of arbitrary, user-defined material to mark encodings visually.
[6] PALinkA's documentation is (at the time of writing) incomplete.
[7] Documentation and help files are identical.
[8] The features are not fully functional.

Figure 4: Usability evaluation