Manuel Burghardt

Visualization as a Key Factor for the Usability of Linguistic Annotation Tools

Abstract Linguistic annotation is an important means of adding information to corpora of spoken or written language. While some less complex annotation tasks can be performed automatically, a great number of annotation tasks require manual annotation, which is typically very time-consuming and tedious. As a consequence, tools for manual annotation tasks should provide a user-friendly interface that makes the annotation process as convenient and efficient as possible; in other words, *usability* should play an important role in the design of such tools. This article contributes to the field of "visual linguistics" by investigating the role of *visualization* in linguistic annotation tools with regard to good and bad usability practices. While there are several studies that are dedicated to visualizing linguistic results, visualization in the context of linguistic annotation has so far been largely neglected. Accordingly, a heuristic walkthrough evaluation study with 11 annotation tools was conducted to find out about typical usability problems. It showed that many of the usability issues identified during the evaluation are related to aspects of interaction design. However, there are also a large number of usability issues that are directly connected to aspects of visualization and visual design. These aspects of good and bad visualization are discussed by means of existing usability heuristics, which can be used to illustrate and explain how and why visualization influences the usability of linguistic annotation tools.

1. Introduction

Digital annotations are an important means to make the daily flood of information manageable, as they allow us to add "invisible intelligence" (Ruecker et al. 2011, 27) to a text, thus making implicit information explicitly available for computer-based analyses[1]. Linguistic annotation constitutes a specific type of digital annotation. Leech (1997, 2) defines it as "the practice of adding interpretative,

1 The work presented in this article is part of a PhD project finished in 2014 (cf. Burghardt 2014). This article reuses some of the passages from the original PhD thesis. The

Published in: Noah Bubenhofer, Marc Kupietz: Visualisierung sprachlicher Daten. Heidelberg: Heidelberg University Publishing, 2018
DOI: https://doi.org/10.17885/heiup.345.474

linguistic information to an electronic corpus of spoken and/or written language data". Linguistic annotation can be carried out manually, automatically, or semi-automatically (i.e. automatic annotation with manual correction) (McEnery and Hardie 2012, 30). Automatic annotation, however, is limited to fields of manageable degrees of complexity (hence it is also called shallow annotation), including simple text processing tasks such as tokenization and sentence segmentation, or simple tagging and parsing tasks such as part of speech tagging or syntactic phrase detection / categorization (Brants and Plaehn 2000, 1). More sophisticated types of annotation cannot be fully automated, but rather need to be carried out by human annotators (cf. Brants and Plaehn 2000; Dandapat et al. 2009). As manual annotation is a laborious task, computer-based annotation tools need to provide a user-friendly interface that makes the annotation process as convenient and efficient as possible. The important role of *usability*[2] in the domain of linguistic annotation tools is also stressed by a large body of related work (cf. Burghardt and Wolff 2009; Burghardt 2012; Dybkjaer, Berman, Bernsen, et al. 2001; Dipper et al. 2004; Reidsma et al. 2004; Eryigit 2007; Dandapat et al. 2009; McEnery and Hardie 2012, 33; Palmer and Xue 2010; Hinze et al. 2012).

In this article, I will focus on the aspect of *visualization* in linguistic annotation tools and discuss how it influences good and bad usability practices. While there are several studies that are dedicated to visualizing linguistic results (cf. e.g. Wattenberg and Viégas 2008; Culy and Lyding 2010), visualization in the context of linguistic annotation has so far been largely neglected. I present the results from a large-scale usability evaluation study (Burghardt 2014) of existing annotation tools, which illustrate that an adequate visualization is a key requirement for user-friendly annotation tools.

2. Evaluating the usability of linguistic annotation tools

As most of the existing linguistic annotation tools struggle to implement a user-friendly interface, I conducted an evaluation study with 11 annotation tools to find out not only about typical usability problems, but also about positive aspects of the different tools. The evaluated annotation tools are:

focus of this condensed article lies on the aspect of visualization and its implications for the usability of linguistic annotation tools.

2 ISO 9241-11 (1999) definition for usability: The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.

- Analec (http://www.lattice.cnrs.fr/Telecharger-Analec?lang=fr)
- Brat (http://brat.nlplab.org/)
- CATMA (http://www.catma.de/)
- Dexter (http://www.dextercoder.org/)
- GATE (https://gate.ac.uk/)
- Glozz (http://www.glozz.org/)
- Knowtator (http://knowtator.sourceforge.net/docs.shtml)
- MMAX2 (http://mmax2.sourceforge.net/)
- UAM Corpus Tool (http://www.wagsoft.com/CorpusTool/index.html)
- WebAnno (https://code.google.com/p/webanno/)
- WordFreak (http://wordfreak.sourceforge.net/index.html)

I used the heuristic walkthrough method (Sears 1997) to discover a total of 207 usability problems and 84 positive aspects for the 11 tools. It showed that many of the usability issues identified during the evaluation are related to aspects of interaction design. There are, however, also a large number of usability issues that are directly connected to aspects of visualization and visual design.

In the following section, I will discuss aspects of good and bad visualization by means of existing usability heuristics. Usability heuristics – sometimes also called guidelines, rules, recommendations or best practices – are meant to capture and promote good design in a generic way (Johnson 2010, xi). There are many examples for such generic heuristics[3] and they often seem to overlap or even appear redundant. This is largely because most of these heuristics share a common basis and origin, which is knowledge about human psychology, for instance perception, reasoning, memory, etc. (Johnson 2010, xiii). The following set of 10 usability heuristics is among the most widely used heuristics (detailed descriptions taken from Nielsen 1994, p.30, Table 2.2):

- *H1 Visibility of system status:* The system should always keep users informed about what is going on, through appropriate feedback within a reasonable time.
- *H2 Match between system and the real world:* The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
- *H3 User control and freedom:* Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted

3 Cf. Johnson (2010, xi) for an overview of some of the most prominent guidelines and heuristics in the field of human-computer interaction (HCI).

function without having to go through an extended dialogue. Support undo and redo.

- *H4 Consistency and standards:* Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
- *H5 Error prevention:* Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
- *H6 Recognition rather than recall:* Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
- *H7 Flexibility and efficiency of use:* Accelerators – unseen by the novice user – may often speed up the interaction for the expert user so that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
- *H8 Aesthetic and minimalist design:* Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
- *H9 Help users recognize, diagnose, and recover from errors:* Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
- *H10 Help and documentation:* Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

3. Visualization in linguistic annotation tools – A usability perspective

As a result of the usability evaluation of linguistic annotation tools, a large number of usability issues that are related to different aspects of visualization were identified. These issues are structured into the following three subsections: (1) visualization of primary data, (2) visualization of annotation schemes and its items, and (3) visualization of the actual annotations, including parallel annotations as well as relational annotations (e.g. coreference annotation). The different visualization aspects will be discussed from a usability perspective by means of

Nielsen's (1994) ten heuristics, which were introduced in the preceding section. Whenever appropriate, I will also refer to related interaction design patterns by Jenifer Tidwell (2011), which describe generic solutions to recurring usability issues in interface and interaction design.

(1) Primary data

During the annotation process, the primary data is typically not read sequentially from beginning to end, but rather scanned for certain text fragments that can be used as an anchor for a specific annotation. The standard visualization of primary data often does not support such episodic scanning and reading. There are several features that can be implemented on the visualization level to enhance the readability of primary data:

a) The *page* metaphor allows the user to break down very long documents into smaller units that are familiar to the user (cf. Figure 1, left).
   - *Heuristic*: Match between system and the real world
   - *Pattern*: Pagination (Tidwell 2011, 224)
b) The use of two different colors helps to distinguish alternating lines from each other (cf. Figure 1, right).
   - *Heuristic*: Flexibility and efficiency of use
   - *Pattern*: Row striping (Tidwell 2011, 220)
c) Numbered lines facilitate the navigation through the primary data document (cf. Figure 1, right).
   - *Heuristics*: Recognition rather than recall, flexibility and efficiency of use
d) Facilitated orientation in primary data by means of a macro-view and positional syncing (cf. Figure 2): a thumbnailversion of the document (macro-view)

Figure 1: Pagination (left), row striping (right) and numbered lines (right) in the Brat annotation tool.

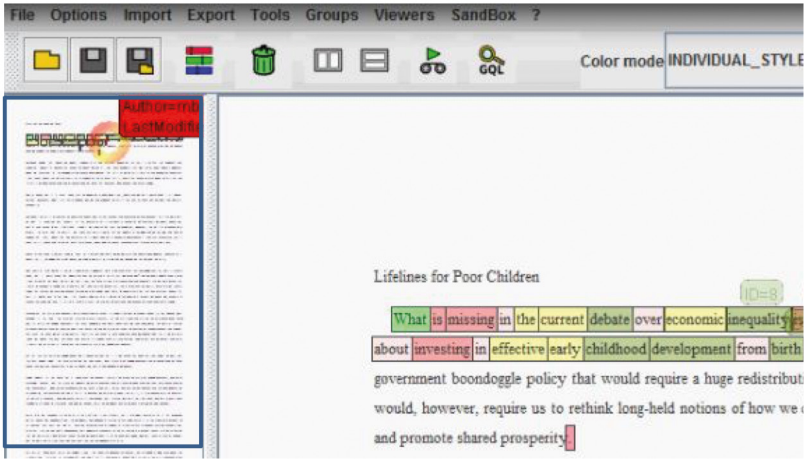


Figure 2: Macro-view of document and positional syncing in the Glozz annotation tool.

allows users to quickly navigate through long documents. The whole text can be accessed via a scrollbar or by clicking into a macro-view of the whole document on the left side. Whenever the mouse cursor is moved somewhere in the document, the position is highlighted in the macro-view (and vice versa). Good visualization of the primary data increases its readability and thus accelerates the overall annotation process

— *Heuristic*: Flexibility and efficiency of use
— *Pattern*: Overview plus detail (Tidwell 2011, 296)

(2) Annotation scheme

The creation of an annotation scheme that defines different levels of annotation as well as concrete annotation items on each level is a crucial task in any annotation project. Typically, annotation schemes are defined by means of document grammars known from markup languages like XML or SGML. Users without technical knowledge about markup languages will have difficulties in creating a scheme in XML syntax. User-friendly annotation tools should provide a visualization of the annotation scheme that can also be understood by markup novices. Adequate visualizations rely on well-known metaphors for the creation of hierarchical structures, e.g. ordered lists and file-trees (cf. Figure 3).

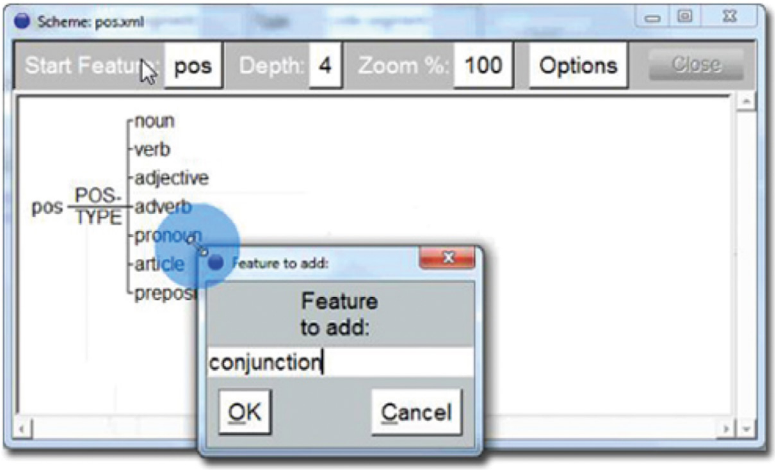


Figure 3: Hierarchical scheme editor in the UAM CorpusTool.

As ad hoc modifications of the annotation scheme are part of the typical annotation process, a good visualization for annotation schemes speeds up the overall annotation process, increases the learnability of the annotation tool and decreases the number of potential errors that may occur when novices are forced to translate linguistic annotation schemes into formal markup languages.

— *Heuristics*: Aesthetic and minimalist design, error prevention

(3) Annotations

Linguistic annotations consist of three basic elements: *body*, *anchor* and *marker* (Marshall 2010, 42ff.). The body of an annotation is the actual content that is added to a text. It is connected to an anchor that denotes the scope of a portion of text an annotation relates to. The marker is the actual visualization of the anchor. For the case of linguistic annotation tools, typical visualizations for anchors are colored underlines or highlights (cf. Figure 4).

In linguistic annotation scenarios, a single anchor is typically annotated with multiple values, which results in parallel annotations (cf. Figure 5).

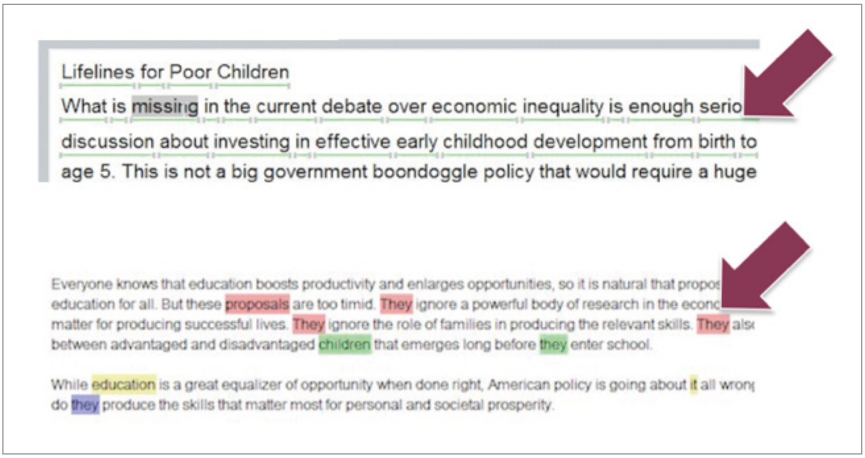


Figure 4: UAM CorpusTool uses underlining (top), GATE uses highlighting (bottom).

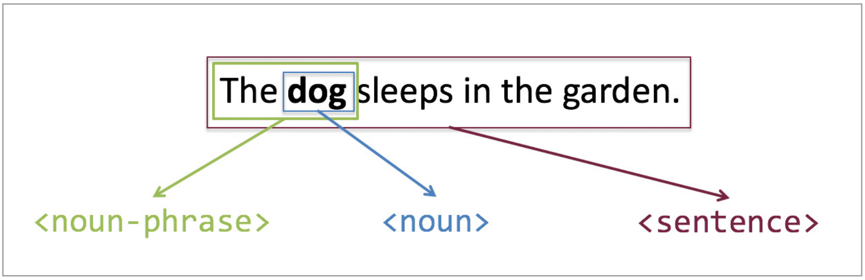


Figure 5: The anchor “dog” is annotated on three different annotation levels.

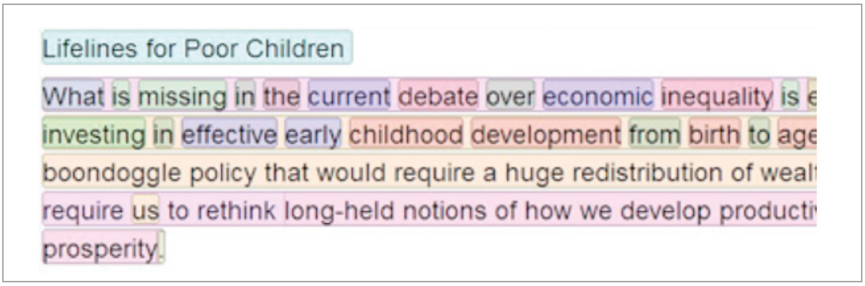


Figure 6: Colored highlights in the Dexter annotation tool overlap one another.

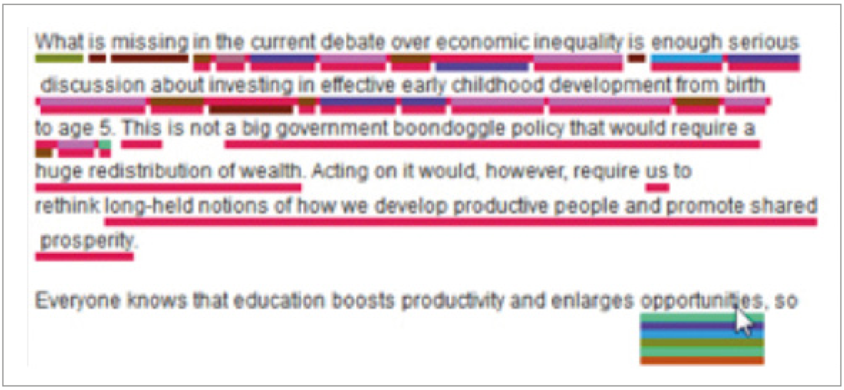


Figure 7: Stacked, colored underlines in the CATMA annotation tool.

It shows that some visualizations are better suited for parallel markers than others. Colored highlights do not work well, as they cannot be stacked, but rather overlap one another (cf. Figure 6).

In contrast, underlines are better suited, as they can be stacked without problems (cf. Figure 7).

- *Heuristics*: Error prevention, aesthetic and minimalist design

Parallel annotation also poses challenges for the visualization of the annotation body, as several competing bits of information – on different levels of annotation – need to be displayed in an adequate way. A user-friendly tool visualizes parallel annotations in a context menu that is displayed next to the respective anchor. The annotation values are displayed as text strings in the context menu (cf. Figure 8). Alternatively, they may be displayed in a separate window or pane rather than in a context menu. Another way to visualize parallel annotation values is by means of a stack view that displays an anchor and (optionally) some of its left and right textual context in the horizontal dimension. In the vertical dimension, parallel annotation values are displayed as a stack of different annotation levels (cf. Figure 9). By visualizing multiple, parallel annotations for one anchor, users have more control about the annotation process and are therefore less likely to produce annotation errors.

- *Heuristics*: Visibility of system status, aesthetic and minimalist design
- *Pattern*: Datatips (Tidwell 2011, 300)

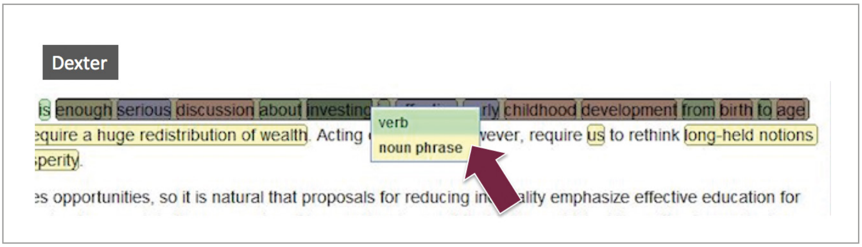
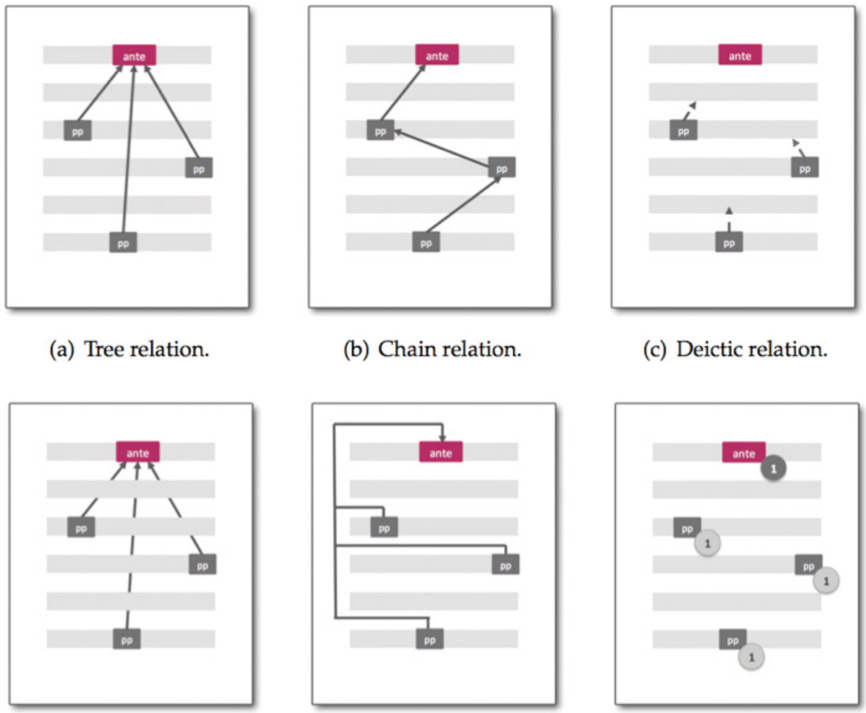


Figure 8: A context menu shows all existing, parallel annotations for a selected anchor in the Dexter annotation tool.

Figure 9: Parallel annotations are displayed in a stack view with different layers in the GATE annotation tool.

Another major challenge, with regard to adequate visualization, is posed by relational annotations, which frequently occur in linguistic annotation scenarios, e.g. for the annotation of coreference relations between two or more anchors. In a related study on human handwritten annotations in a linguistic context, we observed a number of different visualizations for relational annotations (cf. Figure 10). The study participants were asked to create coreference annotations between an antecedent (*ante*) and several corresponding personal pronouns (*pp*). In most cases, lines or directed arrows were used to establish a relation between the separated constituents. The direction of the arrows was mostly pointing toward the antecedent. The lines and arrows either reached directly from the pronouns to the antecedent, thus creating a tree-like structure (a), or they were connected in some sort of chain, where only the first pronoun pointed to the antecedent and the other pointed to the preceding pronoun (b). In some cases, short arrows were used as deictic devices that indicate the direction and position of the antecedent (c). Some participants chose to draw their arrows and lines directly through the text, while others tried to interrupt the lines so they would not obscure the text (d). One participant even tried to draw the lines around the text using the margins of the page (e). Another way to establish a relation between different constituents is by means of an indexing system (f).

For the case of linguistic annotation tools, relational annotations should be realized by means of arrows or connecting lines (cf. the “chain relation” visualization in Figure 10b and Figure 11) between the participating anchors, as these can be understood by the users in a natural and intuitive way.



(a) Tree relation. (b) Chain relation. (c) Deictic relation.

(d) Relation "behind" text. (e) Margin relation. (f) Indices relation.

Figure 10: Examples for different realizations of relational annotation.

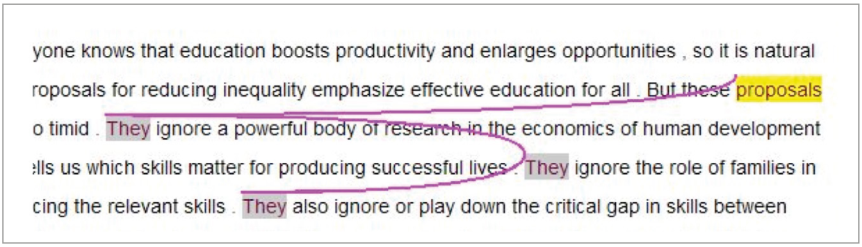


Figure 11: Relational annotation visualization in the *MMAX2* annotation tool.

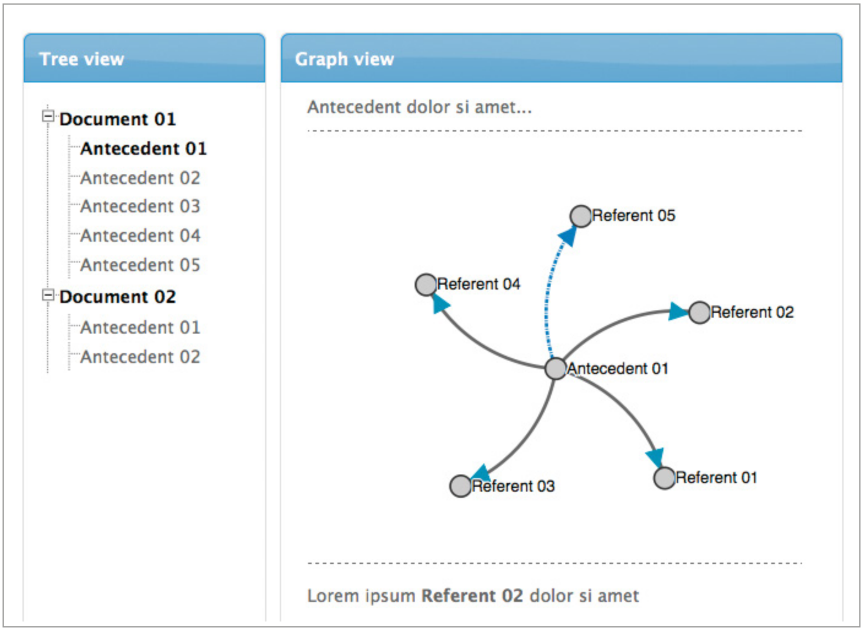


Figure 12: Alternative view for the visualization of coreference annotations according to Witte and Tang (2007).

– *Heuristics*: Match between system and the real world, recognition rather than recall, error prevention

Another good way for the visualization of coreference annotations is described by Witte and Tang (2007). The proposed solution makes use of *Topic Maps* and *OWL ontologies* and can be summarized as follows: relational annotations should be displayed in a separate view that is detached from the primary data view. This view shows all existing relational annotation chains as an integrated graph. Such a graph view even allows users to visualize relational annotations from different documents and thus greatly facilitates navigation in coreference chains and documents (cf. Figure 12).

– *Heuristic*: Aesthetic and minimalist design
– *Pattern*: Alternative views (Tidwell 2011, 66)

4. Conclusion

This article illustrates that visualization plays an important role for the usability of linguistic annotation tools. While there are many competing visualizations, existing usability heuristics can be used to assess and discuss their specific strengths and weaknesses. This kind of assessment is helpful not only for tool developers who design new annotation tools, but also for users of annotation tools, who need to choose from a wide variety of applications and who might want to use "adequate visualization" as a selection criterion.

5. References

Brants, Thorsten, and Oliver Plaehn. 2000. "Interactive corpus annotation." In *Proceedings of Second International Conference on Language Resources and Evaluation (LREC '00)*, 453–459.

Burghardt, Manuel, and Christian Wolff. 2009. "Werkzeuge zur Annotation diachroner Korpora." In *Proceedings of the GSCL-Symposium "Sprachtechnologie und eHumanities"*, edited by Wolfgang Hoeppner. Duisburg: Abteilung für Informatik und Angewandte Kognitionswissenschaft, Universität Duisburg-Essen, 21–31.

Burghardt, Manuel. 2012. "Usability Recommendations for Annotation Tools." In *Proceedings of the ACL 2012, 6th Linguistic Annotation Workshop (LAW '12)*, Stroudsburg, PA: Association for Computational Linguistics, 104–112.

Burghardt, Manuel. 2014. "Engineering Annotation Usability – Toward Usability Patterns for Linguistic Annotation Tools." Doctoral dissertation, Universität Regensburg.

Culy, Chris, and Verena Lyding. 2010. "Double Tree: An Advanced KWIC Visualization for Expert Users." In *14th International Conference Information Visualisation*. Los Alamitos, CA: IEEE, 98–103. doi: 10.1109/IV.2010.24

Dandapat, Sandipan, Priyanka Biswas, Monojit Choudhury, and Kalika Bali. 2009. "Complex Linguistic Annotation – No Easy Way Out! A Case from Bangla and Hindi POS Labeling Tasks." In *Proceedings of the 3rd Linguistic Annotation Workshop (LAW '09)*. Stroudsburg, PA: Association for Computational Linguistics, 10–18.

Dipper, Stefanie, Michael Götze, and Manfred Stede. 2004. "Simple annotation tools for complex annotation tasks: an evaluation." In *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC '04)*, 54–62.

Dybkjaer, Laila, Stephen Berman, Michael Kipp, Malene Wegener Olsen, Vito Pirrelli, Norbert Reithinger, and Claudia Soria. 2001. *Survey of existing tools, standards and user needs for annotation of natural interaction and multimodal data (Deliverable D11.1). ISLE Natural Interactivity and Multimodality Working Group.*

Eryigit, Gülşen. 2007. "ITU Treebank Annotation Tool." In *Proceedings of the 1st Linguistic Annotation Workshop (LAW '07)*. Stroudsburg, PA: Association for Computational Linguistics, 117–120.

Hinze, Annika, Ralf Heese, Markus Luczak-Rösch and Adrian Paschke. 2012. "Semantic Enrichment by Non-Experts: Usability of Manual Annotation Tools." In *Proceedings of the 11th International Semantic Web Conference (ISWC '12)*. Springer: Berlin, 165–181.

Johnson, Jeff. 2010. *Designing with the Mind in Mind. Simple Guide to Understanding User Interface Design Rules. Children.* Amsterdam: Morgan Kaufman.

Leech, Geoffrey. 1997. "Introducing Corpus Annotation." In *Corpus Annotation. Linguistic Information from Computer Text Corpora*, edited by Roger Garside, Geoffrey Leech, and Tony McEnery. Harlow, Essex: Addison Wesley Longman, 1–18.

Marshall, Catherine C. 2010. *Reading and Writing the Electronic Book*. San Rafael, CA: Morgan, and Claypool Publishers (Synthesis).

McEnery, Tony, and Andrew Hardie. 2012. *Corpus Linguistics: Method, Theory and Practice*. Cambridge: Cambridge University Press.

Nielsen, Jakob. 1994. "Heuristic evaluation." In *Usability Inspection Methods*, edited by Jakob Nielsen and Robert Mack. New York: Wiley, 25–62.

Palmer, Martha, and Nianwen Xue. 2010. "Linguistic Annotation." In *Computational Linguistics and Natural Language Processing Handbook*, edited by

Alexander Clark, Chris Fox, and Shalom Lappin. Oxford: Wiley-Blackwell, 238–270.
Reidsma, Dennis, Nataša Jovanović, and Dennis Hofs. 2004. "Designing Annotation Tools Based in Properties of Annotation Problems." http://doc.utwente.nl/49282/ (accessed 26 August 2011).
Ruecker, Stan, Milena Radzikowska, and Stefan Sinclair. 2011. *Visual Interface Design for Digital Cultural Heritage*. Farnham: Ashgate Publishing.
Sears, Andrew. 1997. "Heuristic Walkthroughs: Finding the Problems Without the Noise." *International Journal of Human-Computer Interaction 9* (3): 213–234.
Tidwell, Jennifer. 2011. *Designing Interfaces.* 2nd ed. Sebastopol, CA: O'Reilly Media.
Wattenberg, Martin, and Fernanda B. Viégas. 2008. "The Word Tree, an Interactive Visual Concordance." *IEEE Transactions on Visualization and Computer Graphics*, 14 (6): 1221–1228.
Witte, René, and Ting Tang. 2007. "Task-Dependent Visualization of Coreference Resolution Results." *International Conference on Recent Advances in Natural Language Processing (RANLP 2007)*, September 27–29, 2007, Borovets, Bulgaria. http://www.rene-witte.net/system/files/075_witte.pdf (accessed 25 February 2014).