TELIDA: A Package for Manipulation and Visualization of Timed Linguistic Data

Titus von der Malsburg, Timo Baumann, David Schlangen
Department of Linguistics
University of Potsdam, Germany
{malsburg|timo|das}@ling.uni-potsdam.de

Abstract

We present a toolkit for manipulating and visualising time-aligned linguistic data such as dialogue transcripts or language processing data. The package complements existing editing tools by allowing for conversion between their formats, information extraction from the raw files, and by adding sophisticated, and easily extended methods for visualising the dynamics of dialogue processing. To illustrate the versatility of the package, we describe its use in three different projects at our site.

1 Introduction

Manual inspection and visualization of raw data is often an important first step in the analysis of linguistic data, be that transcripts of conversations or records of the performance of processing modules. Dialogue data or speech processing data in general are typically temporally aligned, which poses additional challenges for handling and visualization. A number of tools are available for working with timed data, each with different focus: as a small selection, Praat (Boersma, 2001) and Wavesurfer (Sjölander and Beskow, 2000) excel at acoustic analysis and are helpful for transcription work, Anvil (Kipp, 2001) helps with the analysis of video material, Exmaralda (Schmidt, 2004) offers a suite of specialized tools for discourse analysis.

We developed TELIDA (TimEd LInguistic DAta) to complement the strengths of these tools. TELIDA comprises (a) a suite of Perl modules that offer flexible data structures for storing timed data; tools for converting data in other formats to and from this format; a command-line based interface for querying such data, enabling for example statistical analysis outside of the original creators of transcriptions or annotations; and (b) a lightweight but powerful visualization tool, *TEDview*, that has certain unique features, as will be described in Section 2.3. TELIDA is available for download from http://www.ling.uni-potsdam.de/~timo/code/telida/.

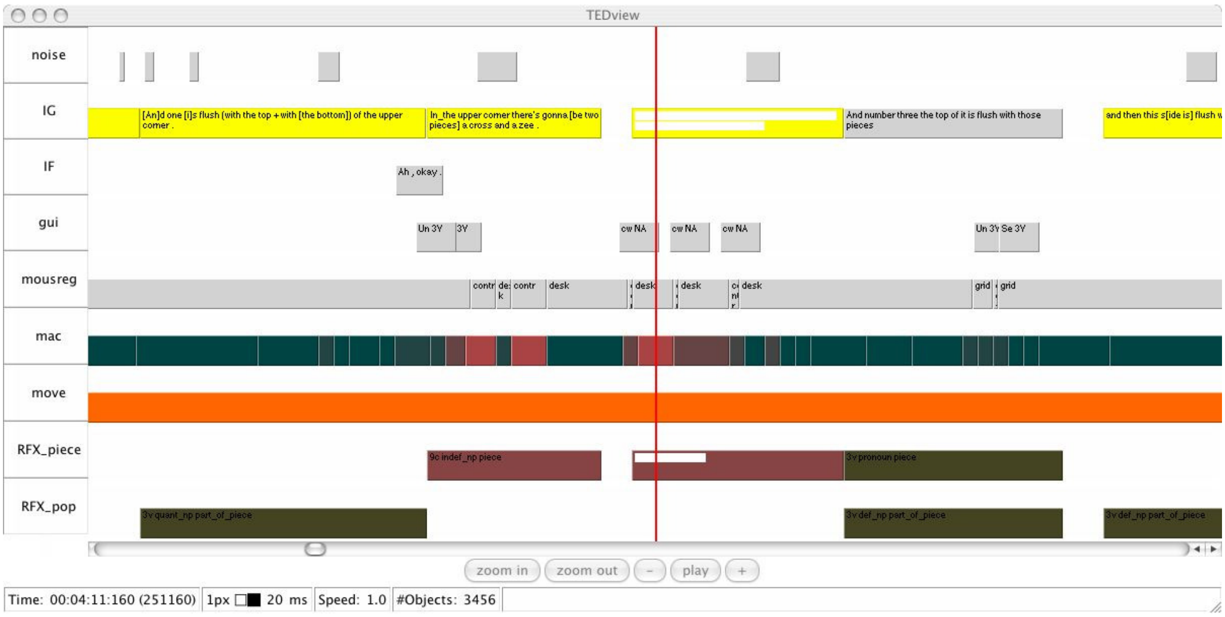
2 Overview of TELIDA

2.1 Data Structures

Like the tools mentioned above, we handle timed data as discrete *labels* which span a certain time and contain some data. To give an example, in a word-aligned transcription of a recording, a single word would correspond to one label. Sequences of (non-overlapping) labels are collected into what we call *alignments*. In our example of the word-aligned transcription, all words from one speaker might be collected in one alignment.

This so far is a conceptualization that is common to many tools. In Praat for example, our alignments would be called a *tier*. TELIDA adds a further, novel, abstraction, by treating alignments as belief states that can have a time (namely that of their formation) as well. Concretely, an incremental ASR may hypothesize a certain way of analyzing a stretch of sound at one point, but at a later point might slighlty adapt this analysis; in our conceptualization, this would be two alignments that model the same original data, each with a time stamp. For other applications, timed belief states may contain other information, e.g. new states of parse constructions or dialogue manager information states. We also allow to store several of such *alignment sequences* (= successive belief states) in parallel, to represent n-best lists.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 302–305,
Queen Mary University of London, September 2009. ©2009 Association for Computational Linguistics



TEDview

[An]d one [i]s flush (with the top + with [the bottom]) of the upper comer .

In_the upper comer there's gonna [be two pieces] a cross and a zee .

Ah , okay .

And number three the top of it is flush with those pieces

and then this s[ide is] flush w

Time: 00:04:11:160 (251160) | 1px □■ 20 ms | Speed: 1.0 | #Objects: 3456

Figure 1: TEDview Showing Annotated Dialogue Data

A *document* finally can consist of collections of such alignments that reference the same timeline, but model different aspects of the base-data. For example, we may want to store information about turns, how they decompose into words, and into phonemes; or, for dual-channel dialogue, have separate alignments for the different speakers.

2.2 Data Manipulation Tools

In order to process timed linguistic data, we implemented a Perl library and command-line tools, *TGtool* and *INtool* for non-incremental and incremental data respectively. They facilitate handling (showing, merging, editing, . . .) and processing (search-and-replace, hypothesis filtering, . . .) of data and interface to TEDview for interactive visualization.

2.3 TEDview

TEDview is the visualization component of TELIDA. It organizes the different sources of information (i.e., alignments or alignment sequences) in horizontal tracks. Similar as in many of the abovementioned tools, time progresses from left to right in those tracks. The content of tracks consists of events that are displayed as bars if they have a temporal extent or as diamonds otherwise. TEDview uses a player metaphor and therefore has a cursor that marks the current time and a play-mode that can be used to replay recorded sequences of events (in real-time or sped-up / slowed-down). Unlike in other tools, TEDview has a steady cursor (the red line in the Figures) across which events flow, and this cursor can be moved, e.g. to give a configuration where no future events are shown.

Information encapsulated by events is displayed in two different ways:

a) *Labels* are represented as bars, with the label information shown as text. (Figure 1 shows a configuration with only labels.)

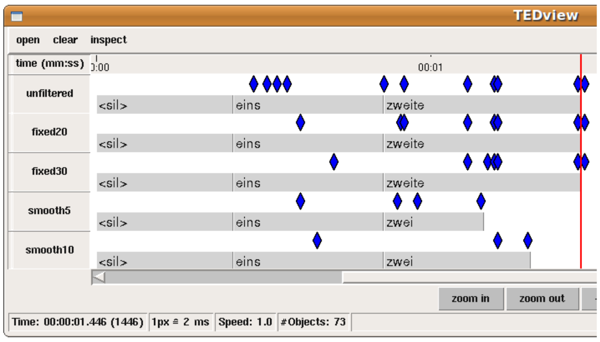
b) Events without duration are displayed as diamonds at the appropriate time (all other Figures). Such events can carry a “payload”; depending on its type, different display methods are chosen:

• If the payload is an alignment, it is displayed on the same track, as a sequence of labels.

• In all other cases TEDview determines the data type of the information and selects an appropriate plug-in for displaying it in a separate inspector window. These data types can be syntax trees, probability distributions, etc.

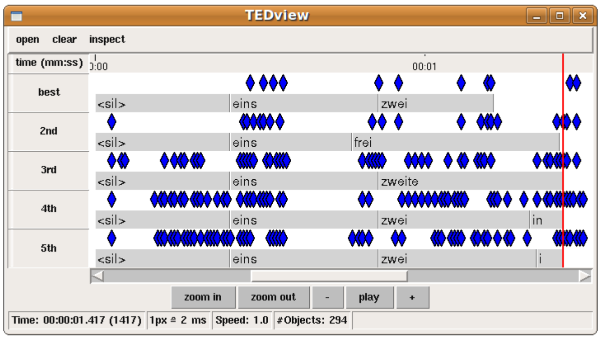
To avoid visual clutter, only the information contained in the diamonds that most recently passed the cursor are displayed. In this way, TEDview can elegantly visualize the dynamics of information state development.

Events can be fed to TEDview either from a file, in a use case where pre-recorded material is replayed for analysis, or online, via a network connection, in use cases where processing components are monitored or profiled in real-time. The format used to encode events and their encapsu-



Time: 00:00:01.446 (1446) | 1px ≙ 2 ms | Speed: 1.0 | #Objects: 73

Figure 2: TEDview showing different filtering strategies for incremental ASR: Diamonds correspond to edits of the hypothesis.



Time: 00:00:01.417 (1417) | 1px ≙ 2 ms | Speed: 1.0 | #Objects: 294

Figure 3: TEDview showing 5-best incremental ASR hypotheses.

lated information is a simple and generic XML format (which the data manipulation tools can create out of other formats, if necessary), i.e. the format does not make any assumptions as to what the events represent. For this reason TEDview can be used to visualize almost any type of discrete temporal data. Intervals can be adorned with display information, for example to encode further information via colouring. Plug-ins for special datatypes can be written in the programming language Python with its powerful library of extension modules; this enabled us to implement an inspector for syntax trees in only 20 lines of code.

3 Use Cases

To illustrate the versatility of the tool, we now describe how we use it in several projects at our site. (Technical manuals can be downloaded from the page listed above.)

3.1 Analysis of Dialogue Data

In the DEAWU project (see e.g. (Schlangen and Fernández, 2007)), we used the package to maintain transcriptions made in Praat and annotations made in MMAX2 (Müller and Strube, 2006), and to visualize these together in a time-aligned view. As Figure 1 shows, we made heavy use of the possibility of encoding information via colour. In the example, there is one track (*mac*, for *mouse activity*) where a numerical value (how much the mouse travels in a certain time frame) is visualized through the colouring of the interval. In other tracks other information is encoded through colour as well. We found this to be of much use in the “getting to know the data” phase of the analysis of our experiment. We have also used the tool and the data in teaching about dialogue structure.

3.2 Analysis of SDS Performance

In another project, we use TELIDA to analyze and visualize the incremental output of several modules of a spoken dialogue system we are currently developing.

In incremental speech recognition, what is considered the best hypothesis frequently changes as more speech comes in. We used TEDview to analyze these changes and to develop filtering methods to reduce the jitter and to reduce edits of the ASR’s incremental hypothesis (Baumann et al., 2009a). Figure 2 shows incremental hypotheses and different settings of two filtering strategies.

When evaluating the utility of using n-best ASR hypotheses, we used TEDview to visualize the best hypotheses (Baumann et al., 2009b). An interesting result we got from this analysis is that typically the best hypothesis seems to be more stable than lower-ranked hypotheses, as can be seen in Figure 3.

We also evaluated the behaviour of our incremental reference resolution module, which outputs distributions over possible referents (Schlangen et al., 2009). We implemented a TEDview plug-in to show distributions in bar-charts, as can be seen in Figure 4.

3.3 Analysis of Cognitive Models

In another project, we use TEDview to visualize the output of an ACT-R (Anderson et al., 2004) simulation of human sentence parsing developed by (Patil et al., 2009). This model produces predictions of parsing costs based on working-memory load which in turn are used to predict eye tracking measures in reading. Figure 5 shows an example where the German sentence “*Den Ton gab der Künstler seinem Gehilfen*” (the artist gives the clay to his assistant) is being parsed, taking

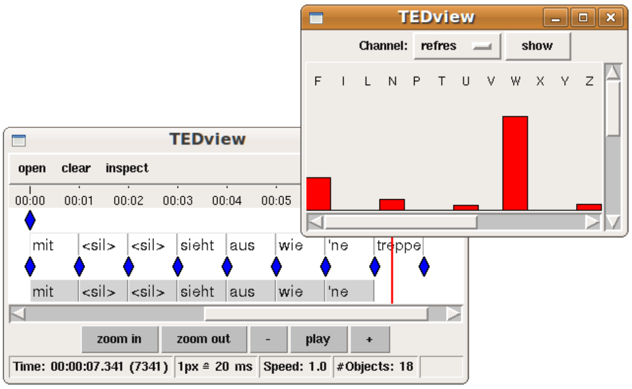

TEDview
Channel: refres
show
F I L N P T U V W X Y Z
TEDview
open clear inspect
00:00 00:01 00:02 00:03 00:04 00:05
mit <sil> <sil> sieht aus wie 'ne treppe
mit <sil> <sil> sieht aus wie 'ne
zoom in
zoom out
-
play
+
Time: 00:00:07.341 (7341) 1px ≙ 20 ms Speed: 1.0 #Objects: 18

Figure 4: TEDview showing the output of our incremental reference resolution module. Distributions are shown with a bar-chart plug-in.

about 3 seconds of simulated time. The items in the channel labeled "Memory" indicate retrievals of items from memory, the items in the channel labeled "Parse" indicate that the parser produced a new hypothesis, and the inspector window on the right shows the latest of these hypotheses according to cursor time. The grey bars finally in the remaining channels show the activity of the production rules. Such visualizations help to quickly grasp the behaviour of a model, and so greatly aid development and debugging.

4 Conclusions

We presented TELIDA, a package for the manipulation and visualization of temporally aligned (linguistic) data. The package enables convenient handling of dynamic data, especially from incremental processing, but more generally from all kinds of belief update. We believe that it can be of use to anyone who is interested in exploring complex state changes over time, be that in dialogue annotations or in system performance profiles.

Acknowledgments This work was funded by a grant from DFG in the Emmy Noether Programme.

References

J.R. Anderson, D. Bothell, M.D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. 2004. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060.

Timo Baumann, Michaela Atterer, and David Schlangen. 2009a. Assessing and Improving the Performance of Speech Recognition for Incremental Systems. In *Proceedings of NAACL-HLT 2009*, Boulder, USA.

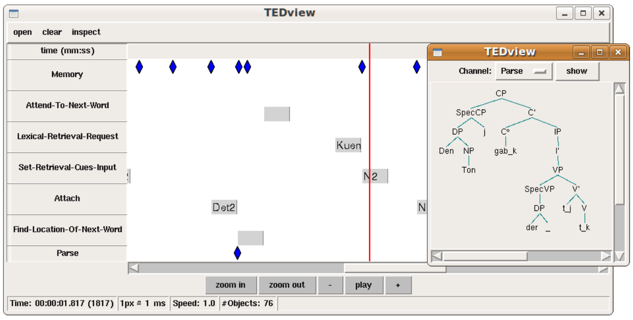


Figure 5: TEDview visualizing the dynamics of an ACT-R simulation, including the current parse-tree.

Timo Baumann, Okko Buß, Michaela Atterer, and David Schlangen. 2009b. Evaluating the Potential Utility of ASR N-Best Lists for Incremental Spoken Dialogue Systems. In *Proceedings of Interspeech 2009*, Brighton, UK.

Paul Boersma. 2001. Praat, a system for doing phonetics by computer. *Glot International*, 5(9–10):341–345.

Michael Kipp. 2001. Anvil - a generic annotation tool for multimodal dialogue. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*, pages 1367–1370, Aalborg, Denmark.

Christoph Müller and Michael Strube. 2006. Multi-level annotation of linguistic data with MMAX2. In *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, pages 197–214. Peter Lang.

Umesh Patil, Marisa Ferrara Boston, John T. Hale, Shravan Vasishth, and Reinhold Kliegl. 2009. The interaction of surprisal and working memory cost during reading. In *Proc. of the CUNY sentence processing conference*, Davis, USA.

David Schlangen and Raquel Fernández. 2007. Speaking through a noisy channel - experiments on inducing clarification behaviour in human-human dialogue. In *Proceedings of Interspeech 2007*, Antwerp, Belgium.

David Schlangen, Timo Baumann, and Michaela Atterer. 2009. Incremental Reference Resolution: The Task, Metrics for Evaluation, and a Bayesian Filtering Model that is Sensitive to Disfluencies. In *Proc. of SigDial 2009*, London, UK.

Thomas Schmidt. 2004. Transcribing and annotating spoken language with exmaralda. In *Proceedings of the LREC-Workshop on XML based richly annotated corpora, Lisbon 2004*, Paris. ELRA. EN.

K. Sjölander and J. Beskow. 2000. Wavesurfer—an open source speech tool. In *Sixth International Conference on Spoken Language Processing*, Beijing, China. ISCA.