TableTrans, MultiTrans, InterTrans and TreeTrans: Diverse Tools Built on the Annotation Graph Toolkit

Steven Bird, Kazuaki Maeda, Xiaoyi Ma, Haejoong Lee, Beth Randall, and Salim Zayat

Linguistic Data Consortium, University of Pennsylvania
3615 Market Street, Suite 200, Philadelphia, PA 19104-2608, USA
{sb, maeda, xma, haejoong, brandall, zayats}@ldc.upenn.edu

Abstract

Four diverse tools built on the Annotation Graph Toolkit are described. Each tool associates linguistic codes and structures with time-series data. All are based on the same software library and tool architecture. TableTrans is for observational coding, using a spreadsheet whose rows are aligned to a signal. MultiTrans is for transcribing multi-party communicative interactions recorded using multi-channel signals. InterTrans is for creating interlinear text aligned to audio. TreeTrans is for creating and manipulating syntactic trees. This work demonstrates that the development of diverse tools and re-use of software components is greatly facilitated by a common high-level application programming interface for representing the data and managing input/output, together with a common architecture for managing the interaction of multiple components.

1. Introduction

Annotation graphs provide an efficient and expressive data model for linguistic annotations of time-series data. We have developed a complete open-source software infrastructure supporting the rapid development of tools for transcribing and annotating time-series data, the Annotation Graph Toolkit (AGTK). This general-purpose infrastructure uses annotation graphs as the underlying model and allows developers to quickly create special-purpose annotation tools using common components. An application programming interface, an input/output library supporting a variety of annotation file formats, and several graphical user interfaces have been developed.

This paper describes four annotation tools based on AGTK which have been developed at the Linguistic Data Consortium. TableTrans is for observational coding, using a spreadsheet whose rows are aligned to a signal. MultiTrans is for transcribing multi-party communicative interactions recorded using multi-channel signals. InterTrans is for creating interlinear text aligned to audio. TreeTrans is for creating and manipulating syntactic trees. The toolkit and tools are distributed under an open source software license.

2. The Annotation Graph Toolkit (AGTK)

An annotation graph (Bird and Liberman, 2001) is a directed acyclic graph where edges are labeled with fielded records, and nodes are (optionally) labeled with time offsets. The annotation graph model, a generalization of the Tipster model used in text retrieval (Grishman, 1997), is capable of representing virtually all types of linguistic annotation (e.g. phonetic, orthographic, part-of-speech, syntactic, discourse, intonational). This development has opened up an interesting range of new possibilities for creation, maintenance and search, and has lead to new annotation tools with applicability across the text, audio and video modalities.

The Annotation Graph Toolkit (AGTK) is a collection of software for the development of annotation tools, instantiating the annotation graph model. AGTK includes application programming interfaces (APIs) for manipulating annotation graph data and importing data from other formats, wrappers for scripting languages, graphical user interface (GUI) components specialized for annotation tasks, and demonstration applications.

Maeda et al. (2002) describes the toolkit in detail. Ma et al. (2002) describes collaborative annotation using a shared database server.

3. Use of Third-party Software

AGTK depends on a variety of third-party software, as detailed below.

3.1. Scripting languages: Python and Tcl

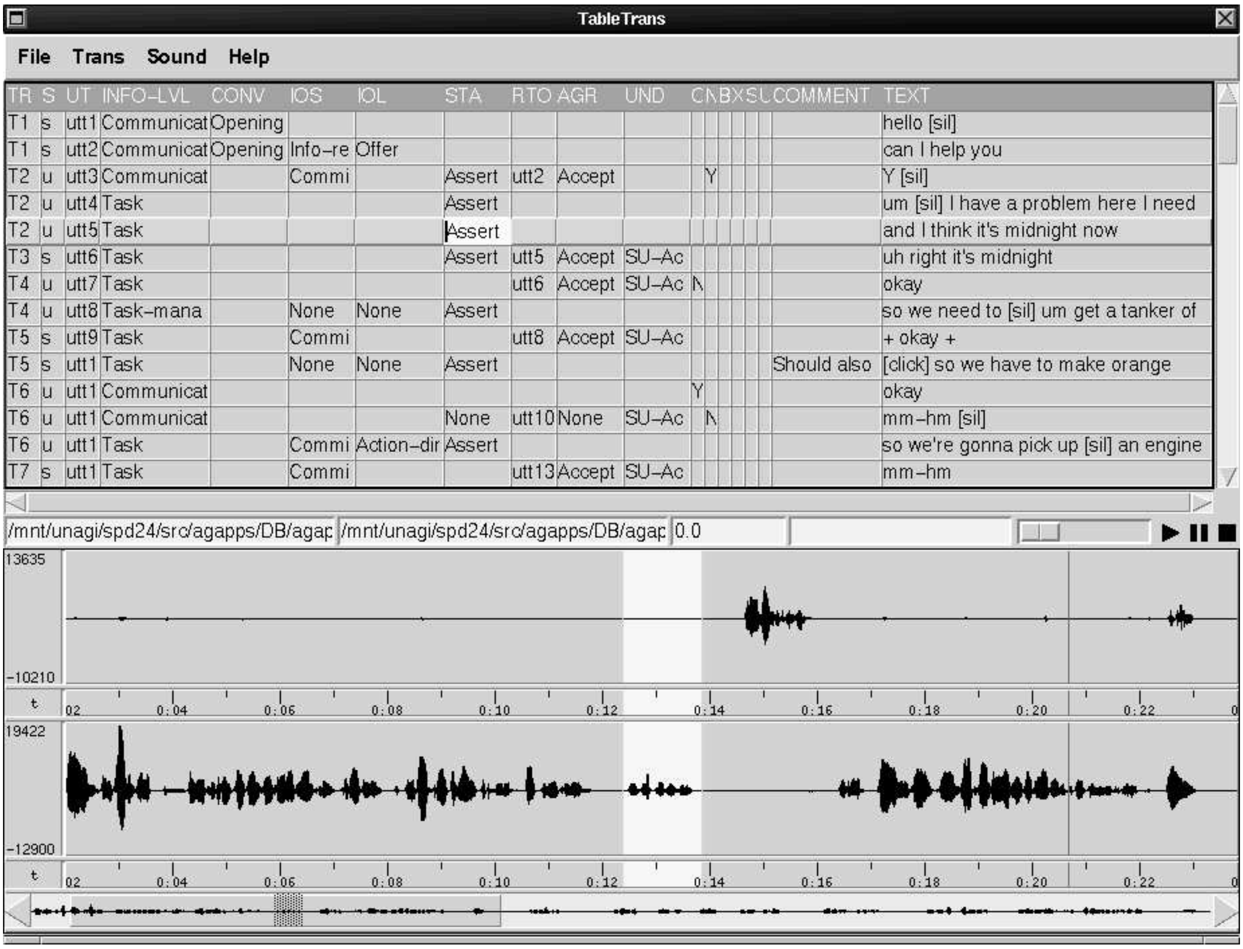
AGTK provides interfaces to its libraries for both Tcl and Python. Tcl is a scripting language with a native GUI toolkit called Tk. Python is a newer object-oriented scripting language. The early AGTK applications were developed using Tcl, but more recent applications have been developed in Python. Both sets of applications depend on the Tk GUI library.

3.2. Audio waveform display: WaveSurfer

WaveSurfer (Sjölander and Beskow, 2000) is open source software for displaying and manipulating audio data, developed by Kåre Sjölander and Jonas Beskow of KTH. WaveSurfer requires Snack, also developed by Sjölander. WaveSurfer is written in Tcl/Tk and its widget, called *wsurf*, can be embedded in any Tcl/Tk application. There is a Python/Tkinter interface for wsurf that permits it to be embedded in a Python/Tkinter application.

3.3. Video display: QuickTime

QuickTime Tcl [`http://hem.fyristorg.com/matben/qt/`] can be embedded in a Tcl/Tk application. We have written a simple wrapper in order to embed QuickTime Tcl in a Python/Tkinter application. QuickTime Tcl requires the QuickTime player distributed by Apple Corporation. [`http://www.apple.com/quicktime/download/`].



TR	S	UT	INFO-LVL	CONV	IOS	IOL	STA	RTO	AGR	UND	C	N	B	X	S	L	COMMENT	TEXT
T1	s	utt1	Communicat	Opening														hello [sil]
T1	s	utt2	Communicat	Opening	Info–re	Offer												can I help you
T2	u	utt3	Communicat		Commi		Assert	utt2	Accept			Y						Y [sil]
T2	u	utt4	Task				Assert											um [sil] I have a problem here I need
T2	u	utt5	Task				Assert											and I think it's midnight now
T3	s	utt6	Task				Assert	utt5	Accept	SU–Ac								uh right it's midnight
T4	u	utt7	Task					utt6	Accept	SU–Ac	N							okay
T4	u	utt8	Task–mana		None	None	Assert											so we need to [sil] um get a tanker of
T5	s	utt9	Task		Commi			utt8	Accept	SU–Ac								+ okay +
T5	s	utt1	Task		None	None	Assert										Should also	[click] so we have to make orange
T6	u	utt1	Communicat								Y							okay
T6	u	utt1	Communicat				None	utt10	None	SU–Ac		N						mm–hm [sil]
T6	u	utt1	Task		Commi	Action–dir	Assert											so we're gonna pick up [sil] an engine
T7	s	utt1	Task		Commi			utt13	Accept	SU–Ac								mm–hm

/mnt/unagi/spd24/src/agapps/DB/agap /mnt/unagi/spd24/src/agapps/DB/agap 0.0

Figure 1: TableTrans With WaveSurfer

4. TableTrans

TableTrans is a spreadsheet-based annotation tool for audio signals. It was primarily intended for linguistic annotations, but it can be used for many kinds of observational coding, such as widely practiced in ethology. The tool allows users to annotate regions of signal with arbitrary features, such as speaker identifiers, utterance identifiers and transcriptions. Each row of the spreadsheet corresponds to a region of the signal. Each column corresponds to a user-defined feature, and has a user-defined width. For example, one could define the width of 5 characters for the speaker identifier, and the width of 40 for the transcription. One could also choose to define ten or more columns as shown below, and use it for more structured coding.

Figure 1 shows TableTrans with a waveform display based on WaveSurfer. Figure 2 shows TableTrans with a video module based on QuickTime and QuickTimeTcl.

Setting Current Region Swiping a region in the waveform display while holding down the left-mouse button highlights the region. This region is called the *current region*. In the video version of TableTrans, the current region can be set with two operations: pressing *control-s* at the beginning of the region, and pressing *control-e* at the end of the region.

Create Annotation Pressing the *Return* key inserts a blank annotation row in the spreadsheet. If the current region of the waveform is chosen, the start and end times of the current region are inserted.

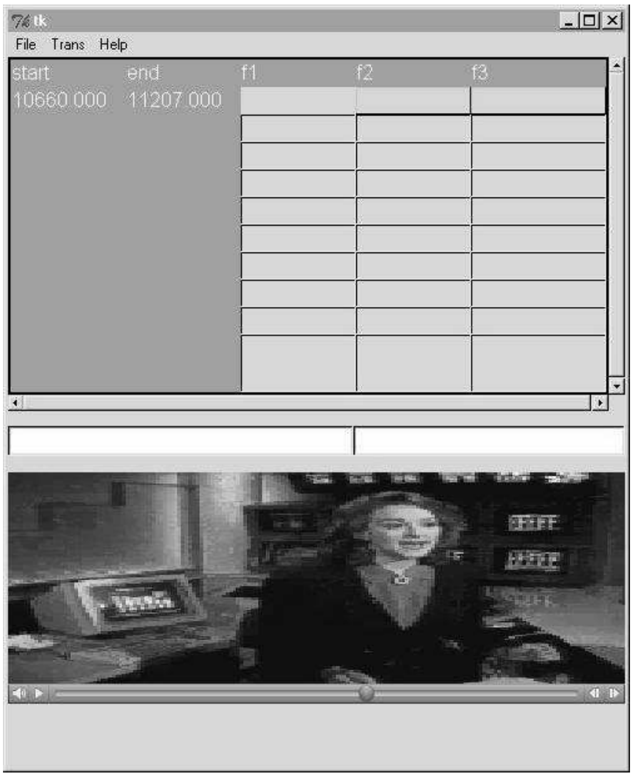
Delete Current Annotation Pressing *Control-d* deletes the currently highlighted annotation (the *current annotation*).

Update Current Region Pressing *Control-g* updates the start and end times of the current annotation using the current region in the waveform.

Navigate in Spreadsheet The *Tab* key moves the cursor one cell to the right, if possible. The right, left, up, and down arrows move to the neighboring cells in the corresponding directions.

Navigate Within Current Cell The *Shift-Right* and *Shift-Left* keys move the insertion point within the current cell.

Toggle Play and Stop The *F1* key toggles playback and pausing of the recording. If the current region is chosen, it plays the current region. If a single point is selected in the waveform, *F1* starts playback from the point, and annotations corresponding to the audio cursor are highlighted (aligned playback).



start	end	f1	f2	f3
10660.000	11207.000			

Figure 2: TableTrans With the QuickTime Player

Sort Annotations Double-clicking the left mouse button on a feature name (a column heading) of the spreadsheet sorts the annotations according to the values in the column. For example, double clicking on the cell titled ’f1’ will sort all the annotations according to the values in the column. Sorting according to start and end times can also be done from the menu items, `Trans->Sort->Sort Annotations by Start Time` and `Trans->Sort->Sort Annotations by End Time`.

Find The menu item `Trans->Find` or the *Control-f* key combination brings up a dialog window to search a string in the spreadsheet. If a matching string is found in the cell it is highlighted and the annotation row becomes the current annotation.

Control View of Annotations The menu item `Trans->View->Show Select Rows` lets the user specify a feature and a value so that only the annotations having this value as the feature will be displayed. The other menu items in `Trans->View` are for displaying all annotations and hiding all annotations in the spreadsheet.

Open Sound/Movie File The menu item `File->Open Sound File` loads a sound file into the waveform panel. It automatically adjusts the number of waveforms according to the number of channels in the sound file. All sound file formats supported by WaveSurfer are supported. In the video version of TableTrans, the `File->Open Movie File` will open a movie file.

Open Annotation File The menu item `File->Open Annotation File` provides the users with options to load an annotation file in the following formats: XML (AIF), Table Format (csv, etc.) and LCF (the LDC Callhome Format).

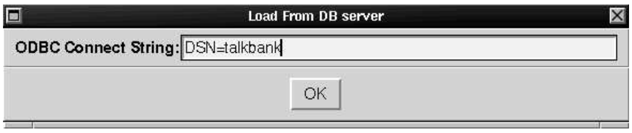
The application opens appropriate dialog windows for each format. First, the user is prompted for the file. Then, for a Table format, a window for specifying names of features and a delimiter is opened. The names specified here are used as feature names and column headers.

Save Annotations in File If a file name and a format type have been already chosen, the menu item `File->Save` will save all the annotations in the file. If they have not be chosen, the user needs to use one of the `File->Save Annotations As` menu items.

Save Current Sound Region in File The menu item `Sound->Save Current Region in File` lets the user save the current region of the sound data into a file.

Column configurations Column headers and width may be changed interactively, or may be specified in a configuration file.

Database support TableTrans supports access to the database component of AGTK. Figure 3 shows the dialog window for entering a ODBC connect string. Table 1 shows some of the parameters used in a connect string. For a complete list, please see [`http://www.mysql.com/doc/M/y/MyODBC_connect_parameters.html`].



ODBC Connect String: DSN=talkbank

Figure 3: ODBC connect string dialog

DSN	Registered ODBC Data Source Name.
SERVER	The hostname of the database server.
UID	User name as established on the server. SQL Server this is the logon name.
PWD	Password that corresponds with the logon name.
DATABASE	Database to connect to. If not given, DSN is used.

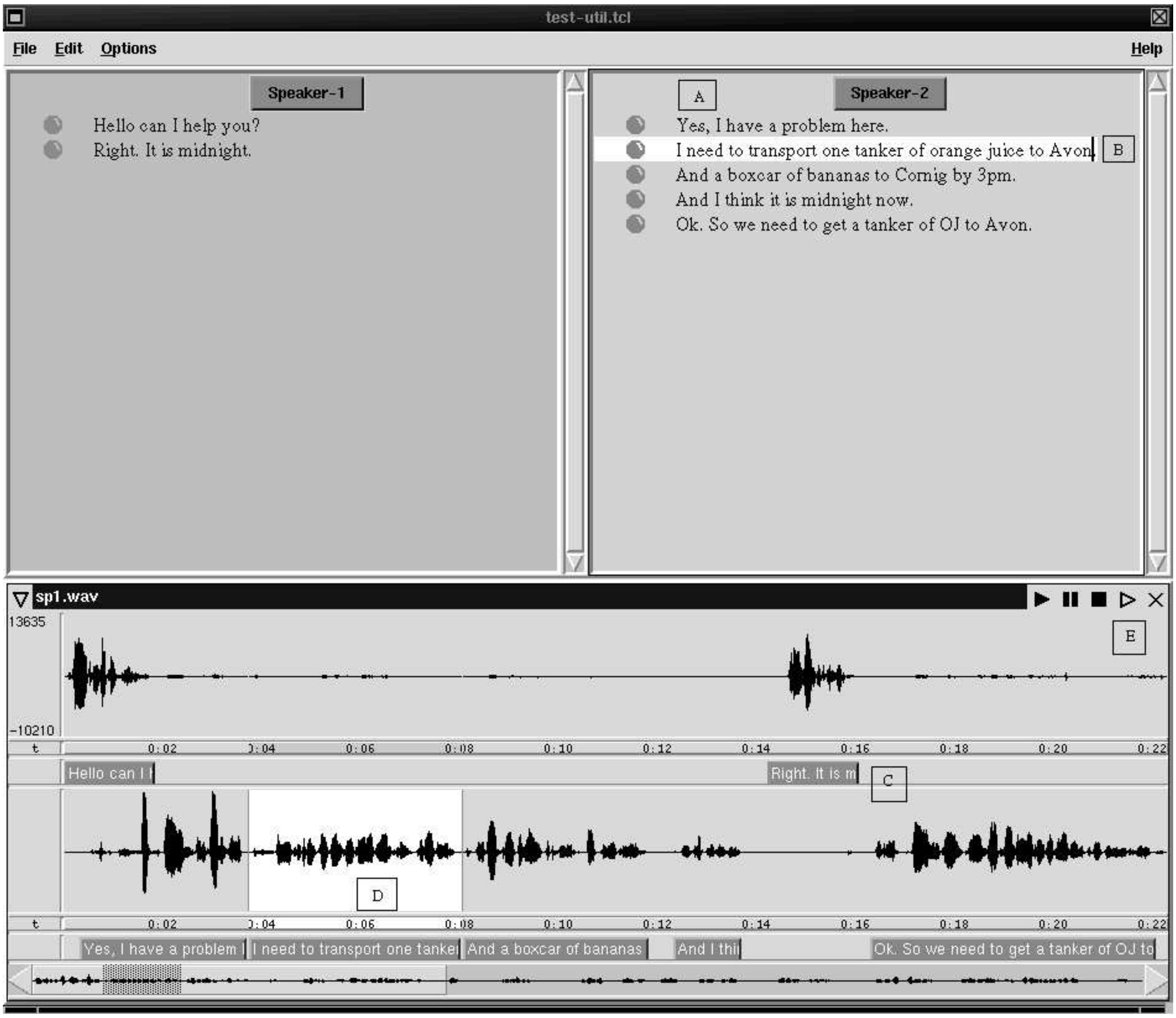
Table 1: ODBC Connect String Parameters

5. MultiTrans

MultiTrans is a transcription tool for transcribing multi-party conversations in multi-channel audio signals. The user interface is similar to Transcriber (Barras et al., 2001), but MultiTrans has one transcription panel corresponding to each channel in the signal.

Figure 4 contains a screenshot of MultiTrans with a two-channel audio signal. The left transcription panel corresponds to the first channel in the audio signal, and the right transcription panel corresponds to the second channel. The boxes labeled from A to E in the figure illustrates the following points.

A: The text panel for speaker 2. The channel associated with speaker 2 is the current channel.



Speaker-1

- Hello can I help you?
- Right. It is midnight.

Speaker-2

- Yes, I have a problem here.
- I need to transport one tanker of orange juice to Avon
- And a boxcar of bananas to Cornig by 3pm.
- And I think it is midnight now.
- Ok. So we need to get a tanker of OJ to Avon.

Figure 4: MultiTrans, for Transcribing Multi-Party Conversation

B: This is the second annotation for speaker 2. It is also the current annotation, and is highlighted.

C: A segment for speaker 1. This shows some of the transcript for this annotation and the associated region of the waveform.

D: The highlighted region of the waveform for the second annotation. This is the current annotation and its entire waveform region is highlighted.

E: The hollow play button. This button will play the current channel only, muting other speech channels.

There are two ways to create an annotation, both are explained below. All annotations are inserted into the text panel, sorted by their starting position.

Create Annotation (Specific) The first way to create an annotation is to explicitly highlight a region in the waveform and press the *Return* key. This will create a bullet in the appropriate text panel that designates the created annotation. Also, a region is created below the waveform itself to designate the created annotation. Only once an annotation is created can the transcription of that region begin.

Create Annotation (Non-Specific) Another way to create an annotation is during speech playback. When the playback of speech is begun the user may press the *Return* key to insert an anchor in the current channel. When *Return* is pressed a small black bar will appear below the waveform. This designates the starting position of the current annotation. When *Return* is pressed a second time the end anchor for the annotation is inserted and the annotation is created. Ending speech playback destroys any start anchors that do not yet have an associated end anchor.

Delete Current Annotation An annotation can be deleted using *Control-d*. To delete an annotation one must first select the annotation to be deleted. The current annotation can easily be distinguished from others because its transcription and waveform regions are highlighted. When an annotation is deleted its transcription and its association with the waveform region are also deleted.

Change Current Annotation Once an annotation is created its region in the waveform can be changed without deleting the entire annotation. This is done by selecting the desired annotation (with a click in the text panel or segment below the waveform), and moving either the end point or start point of the annotation and pressing the *Return* key to register the change.

Split Current Annotation Large annotations can be split into smaller annotations using the split current annotation command. First, the area in the text transcription where the split is to occur is selected. Next, the area in the waveform where the split is to occur is selected. Finally, the *Return* key is pressed and the old annotation is split into two new annotations, each associated with a different waveform region.

Join Current Annotation Joining an annotation is the opposite of splitting an annotation. This is done by selecting an annotation and pressing the *Shift-BackSpace* key combination. This will merge the currently selected annotation region and transcription with the one that occurs immediately before it.

Squeeze Current Annotation When an annotation is squeezed its starting boundary is pushed to the ending boundary of the previous annotation. This is used when annotations are desired to be separate, but one is to begin as soon as the other ends. This is done by selecting an annotation and pressing the *Control-Shift-BackSpace* key combination.

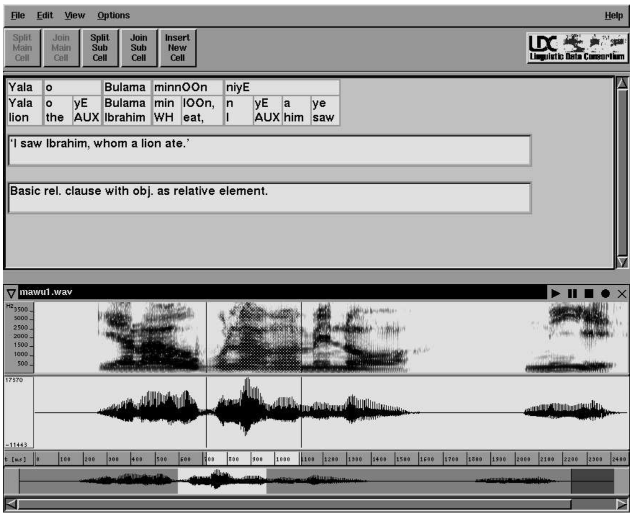
Toggle Speech Playback There are several ways to begin speech playback. Pressing the *Tab* key will toggle speech playback of the current annotation, or the entire speech file if there is not a current annotation. Playback can also be initiated by pressing the solid play button in the waveform panel. Either of these commands will play all channels in the speech file. Pressing the hollow play button located in the waveform panel will play the current channel only, muting all other channels. This is useful when there are several channels that make speaker distinction difficult.

6. InterTrans

InterTrans is an interlinear text editor. Interlinear text is a kind of text in which each word is annotated with some combination of phonological, morphological and syntactic information (displayed under the word) and each sentence is annotated with a free translation. For an extended discussion of interlinear text, and how to model it using annotation graphs, see (Maeda and Bird, 2000).

InterTrans permits interlinear transcription aligned to a primary audio signal, for greater convenience, accuracy and accountability. Whole words and sub-parts of words can be easily aligned with the audio. Clicking on any part of the annotation causes the corresponding extent of audio signal to be highlighted. As an extended recording is played back, annotated sections are highlighted (both waveform and interlinear text displays). Figure 5 contains a screenshot of InterTrans.

The linguistic levels of information such as *Free Translation (FT)*, *Word* (`WD`) and *Morphome* (`MP`) are called *types*. The relationships among types are defined in a configuration file. For example, if the type `WD` is defined to *dominate* the type `MP`, operations described in this section, such as split and join, on any cell of the type `WD` will cause corresponding cell of the type `MP` to split or join as well. It is also possible to define two or more types as in an equivalent class. In the implementation, this is done by using multi-attribute labels of arcs. For example, if the types *Morpheme* and (`MP`) and *Morphemic Gloss* (`MP-GLOSS`) are to be used, these can be defined as types of an equivalent class. Then, operations, such as *Split* on any of the cells of the type `MP` will cause the corresponding cell of the type `MP-GLOSS` to split as well.



Yala	o		Bulama	minnOOn		niyE			
Yala	o	yE	Bulama	min	lOOn,	n	yE	a	ye
lion	the	AUX	Ibrahim	WH	eat,	I	AUX	him	saw

'I saw Ibrahim, whom a lion ate.'

Basic rel. clause with obj. as relative element.

Figure 5: InterTrans, Interlinear Text Transcription

Insert New Cell A new cell can be inserted after the current cell by clicking on the *Insert New Cell* button. A normal procedure for creating interlinear text is to use this function first, and then split the sub cells as necessary.

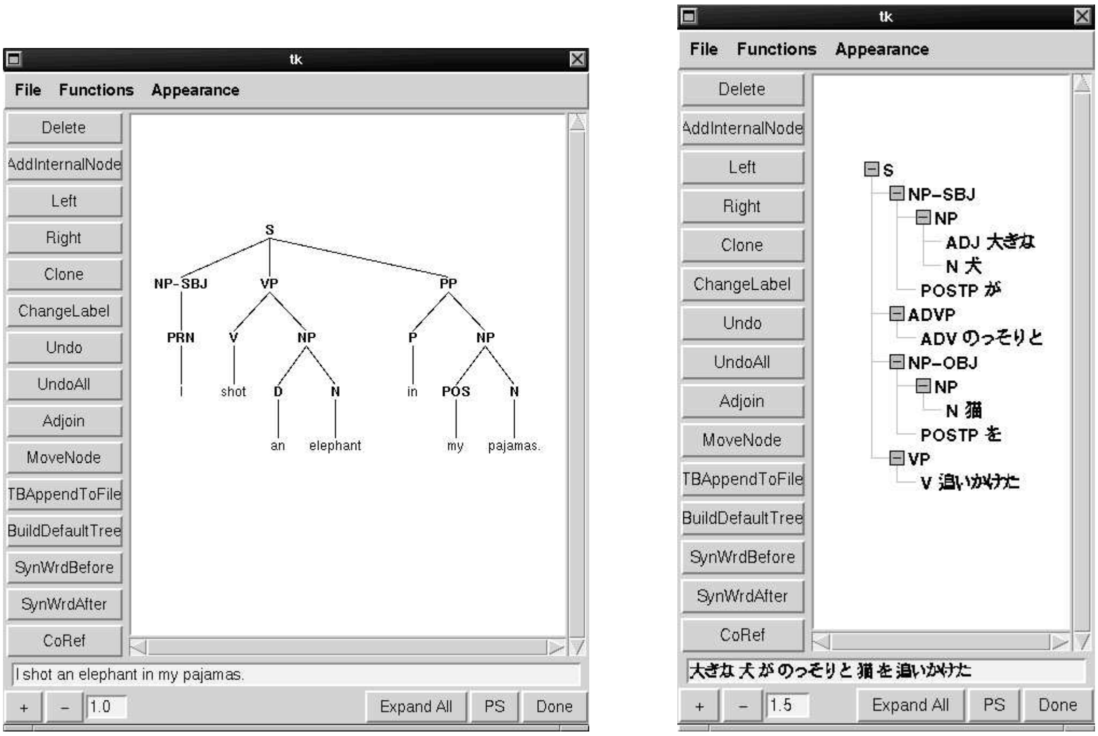
Delete Cell Deleting a cell can be done in one of the following ways after selecting the current cell; use the *Control-d* key combination, or click on the *Delete Cell* button.

Split Cell Splitting a cell in InterTrans can be done in a similar manner to MultiTrans. First, the current cell should be selected. Optionally, the splitting point in the waveform can be chosen if the current cell is associated with a region in the waveform. Pressing the *Return* key, or clicking on the *Split Cell* button will split the current cell.[1]

Join Cell Joining two adjoining cells is simple. First, the current cell should be chosen. Then clicking on the *Join Sub Cell* button or pressing the *Control-j* key combination will join the current cell and its preceding cell.

Alignment with Audio Waveform A partial or complete set of cells can be aligned with the audio signal displayed in the waveform panel. This can be done by selecting the current cell in the interlinear transcription panel, and selecting the corresponding region in the waveform panel. The alignment can be changed later.

[1] We plan to modify the program so that key bindings can be assigned by the user.



I shot an elephant in my pajamas.

大きな 犬 が のっそりと 猫 を 追いかけた

Figure 6: TreeTrans: Syntactic Transcription (Horizontal and Vertical Views)

7. TreeTrans

TreeTrans is a tool for displaying and annotating tree structures.[2] Basic annotation functions are provided, and functions can be easily added or changed for specific annotation projects. Trees can be displayed in three forms: bottom-up, top-down, and vertical. Two views of trees in TreeTrans are shown in Figure 6. Every node of the displayed tree has one of three types: `syn`, a syntactic or internal node; `pos`, a part-of-speech node; or `wrd`, a node containing original text.

To change a displayed tree, users highlight the appropriate node(s) then choose a function from the list of buttons. All the tree manipulation functions preserve the surface order of the word string.

Input/Output TreeTrans can input either Penn Treebank-style bracketed trees or annotation graphs. Likewise, it can store annotated trees in either of these formats.

InsertInternalNode If one node is highlighted, this function adds an internal node as the parent of the highlighted node. If two nodes are highlighted, this function adds an internal node as the parent of the the two highlighted nodes and any intervening material. If the branches of the new internal node would cross any existing branches of the tree, the insertion is not performed.

Delete This deletes the highlighted node. If the highlighted node has type `pos`, its corresponding `wrd` node is also deleted. If the deletion would cause an internal node to have no leaves, the deletion is not performed.

MoveNode When two nodes are highlighted, this moves the first highlighted node and its subtree to become a child of the second highlighted node. When three nodes are highlighted, the first two nodes and any intervening material are moved to become children of the third highlighted node. The move is not performed if it would cause word order to be changed, or if it would result in an internal node with no leaves.

Adjoin This is a syntactic adjoin, which creates a clone of the second highlighted node, then moves the first highlighted node to become a child of the clone.

SynWrdBefore/After This function adds a pair of nodes, one of type `syn`, and one of type `wrd`. It is used for adding traces to the text. The new `syn` node is a previous or following sibling of the highlighted node.

ChangeLabel This changes the label of the highlighted node.

CoRef This gives two highlighted nodes the same trace number.

BuildDefaultTree This builds a basic tree structure using the input sentence as the terminals of the tree.

8. Obtaining the tools

All of the annotation tools described in this paper are distributed under an open source license, and available from [`http://www.ldc.upenn.edu/AG/`].

9. References

C. Barras, E. Geoffrois, Z. Wu, and M. Liberman. 2001. Transcriber: development and use of a tool for assisting speech corpora production. *Speech Communication*, 33:5–22.

[2] For a discussion of how to model trees using annotation graphs, see (Bird and Liberman, 2001; Cotton and Bird, 2002).

S. Bird and M. Liberman. 2001. A formal framework for linguistic annotation. *Speech Communication*, 33:23–60.

S. Cotton and S. Bird. 2002. An integrated framework for treebanks and multilayer annotations. In *Proceedings of the Third International Conference on Language Resources and Evaluation*.

R. Grishman. 1997. TIPSTER Architecture Design Document Version 2.3. Technical report, DARPA. `http://www.nist.gov/itl/div894/894.02/related_projects/tipster/`.

X. Ma, H. Lee, S. Bird, and K. Maeda. 2002. Models and tools for collaborative annotation. In *Proceedings of the Third International Conference on Language Resources and Evaluation*.

K. Maeda and S. Bird. 2000. A formal framework for interlinear text. In S. Bird and G. Simons, editors, *Proceedings of the Workshop on Web-Based Language Documentation and Description*. `http://www.ldc.upenn.edu/exploration/expl2000/papers/`.

K. Maeda, S. Bird, X. Ma, and H. Lee. 2002. Creating annotation tools with the annotation graph toolkit. In *Proceedings of the Third International Conference on Language Resources and Evaluation*.

K. Sjölander and J. Beskow. 2000. WaveSurfer – an open source speech tool. In *Proceedings of the 6th International Conference on Spoken Language Processing*. `http://www.speech.kth.se/wavesurfer/`.