LaBB-CAT: an Annotation Store

Robert Fromont
NZILBB, University of Canterbury
Private Bag 4800
Christchurch, New Zealand
robert.fromont@canterbury.ac.nz

Jennifer Hay
NZILBB, University of Canterbury
Private Bag 4800
Christchurch, New Zealand
jen.hay@canterbury.ac.nz

Abstract

"ONZE Miner", an open-source tool for storing and automatically annotating Transcriber transcripts, has been redeveloped to use "annotation graphs" as its data model. The annotation graph framework provides the new software, "LaBB-CAT", greater flexibility for automatic and manual annotation of corpus data at various independent levels of granularity, and allows more sophisticated annotation structures, opening up new possibilities for corpus mining and conversion between tool formats.

1 Introduction

"ONZE Miner" (Fromont & Hay 2008) was a browser-based, searchable database tool for time-aligned transcripts of speech produced using Transcriber, a transcription and annotation tool developed by Barras et al. (2000). It has been used for a variety of research projects in various universities and labs, primarily for sociophonetic research.

ONZE Miner's original data structure was designed to closely mirror that of Transcriber, so transcripts are divided into topic-tagged *sections*, which contain speaker *turns*, divided up into utterance *lines* containing text and other 'event' annotations such as noises, comments, etc. In order to allow automatic annotation of lexical data from CELEX (Baayen *et al.*, 1995), and to facilitate storage for forced-alignments produced by the Hidden Markov Model Toolkit, HTK (Young et al. 2006), *lines* were tokenized into *words* that were stored as separately annotatable units, which could be further divided into *segments* for storage of phones produced by HTK.

For researchers with large collections of recordings and Transcriber transcripts, ONZE Miner was very useful for mining corpus data, but it had certain limitations related to its data structures, which are explained below.

1.1 Other Formats

Many corpora exist, or are being produced, using tools other than Transcriber. For example the Buckeye corpus (Pitt et al. 2007) includes aligned transcription files in the Xwaves (Hawkins 2008) format, and transcriptions for various other corpora are available as Praat (Boersma & Weenink 2005) textgrids.

For ONZE Miner, support for these was only available via conversion from these formats to Transcriber files before importing the data. The problem was that, in many cases, the data was not structured in a way that was compatible with the Transcriber model. For example, some formats include much finer-grained synchronisation than is typically available with Transcriber.

Simultaneous speech also presented problems for data conversion. In Transcriber, overlapping speech is modelled using a 'simultaneous speech' turn – i.e. a single turn that has multiple speakers attached to it, and multiple corresponding transcriptions. For example, if a second speaker started their turn before a first speaker finished theirs, this would be modelled as three turns:

1. a turn containing words spoken while only the first speaker is speaking,
2. a 'simultaneous speech' turn containing words spoken by both speakers, during the time that they are both speaking, and
3. a turn containing words spoken by the only second speaker, once the first speaker is no longer talking.

Robert Fromont and Jennifer Hay. 2012. LaBB-CAT: an Annotation Store. In *Proceedings of Australasian Language Technology Association Workshop*, pages 113−117.

However, when researchers transcribe using other tools, they often treat cases like this as being two turns that overlap:

1. a turn containing words spoken by the first speaker, from the time they start talking to the time they stop, and
2. a turn containing words spoken by the second speaker, from the time they start talking to the time they stop, this turn having a start time earlier than the end time of the previous turn.

For ONZE Miner, the only option when importing non-Transcriber data was to convert this second model to the first model (i.e. break the two turns into three), which would have involved an inevitable loss of accuracy when trying to create the middle 'simultaneous speech' turn.

1.2 Different Annotation Granularities

Transcriber has facility for topic-tagging sections of transcripts, and for marking named entities, but beyond this, little facility for independent annotation of multiple words.

This meant that ONZE Miner couldn't be used to store annotations for multiple, independent, and possibly overlapping sets of such annotations. As a result, it was impossible to simultaneously have, for example, topic tags and speaker attitude tags dividing up the transcript in different ways, and also impossible to make more finely-grained multi-word annotations, e.g. phrasal verbs, genitive phrase constructions, syntactic parses, etc.

1.3 Development of a New Data Structure

As a result of these limitations, we decided to develop a new system, using ONZE Miner as a basis, keeping all of ONZE Miner's features and interfaces, but introducing new capabilities. The new system, LaBB-CAT (**La**nguage, **B**rain and **B**ehaviour – **C**orpus **A**nalysis **T**ool), adopts a different underlying data model, "annotation graphs", which is described in section 2. How annotation graphs solve the above problems, and introduces new possibilities, is discussed in section 3.

2 Annotation Graphs and LaBB-CAT

Bird and Liberman (1999 a&b) proposed a framework for modelling linguistic annotations, which seemed to provide potential solutions for the limitations faced by ONZE Miner. A new annotation storage tool was developed, called LaBB-CAT, which would maintain ONZE Miner's general way of working with recordings, transcripts, annotation, and search via web browser, but use a new data model based on annotation graphs.

2.1 Annotation Graphs

Bird and Liberman proposed a model for linguistic data which they claimed could encompass a wide variety of types of linguistic annotation. The commonality that Bird & Liberman saw between all approaches to linguistic annotation is that annotations are always:

1. some kind of contentful label, and
2. each label is usually 'anchored' to some portion of a 'source' (e.g. the recording).

They model this using digraphs, which consist of nodes that are joined by directional arcs. In their model:

1. labels are arcs, and
2. anchors are nodes.

In order to be specifically useful for linguistic annotation, there are some extra features to the model:

Arcs can have:

- a 'label' which represents the 'content' of the annotation (e.g. the orthography, the part of speech, the phonemic transcription, etc.)
- a 'type' which categorises the label (e.g. as being an 'orthography', or a 'part of speech', or a 'phonemic transcription', etc.)
- an optional 'class' which provides a mechanism for linking distal annotations by membership to an equivalence class.

In addition, nodes can have an 'offset' which represents the temporal position of the anchor (e.g. number of seconds since the beginning of the recording), but the offset is optional, so that annotations having no precise position in time can be represented.

By virtue of being a digraph, every arc has a start and end node, meaning that every annotation has a start point and an end point. However, these may have the same offset, to represent annotations of instants rather than intervals in time.

Annotations may share anchors, thereby reflecting a logical relationship between two annotations and conversely, two annotations may use two different anchors that have the same offset, thereby reflecting the lack of logical

relationship between the annotations despite coincidence in time.

2.2 LaBB-CAT Implementation

The relational database schema we designed for LaBB-CAT is not dissimilar to that proposed for annotation graphs by Ma et al. (2002), but with some changes to enhance performance and meet specific needs for the time-aligned transcription data and annotations already stored using ONZE Miner.

In particular, both anchors (nodes) and annotations (arcs) carry a 'status' field that allows automatic annotations and alignments to be distinguished from manual ones. This is used, for example, to prevent HTK forced-alignment from overwriting alignments that have already been hand-corrected.

In addition, annotation records are kept in separate layer tables instead of a single table, and have a number of extra fields that allow, for example, a word's turn annotation to be immediately identified, without having to traverse the graph structure to find it (thus avoiding a constraint that the graph be connected between words and turns to make such a traversal possible).

These departures boost the performance of LaBB-CAT, both for searching and automatic annotation. However, they impose on the data an 'ontology' that isn't formally present in Bird & Liberman's original proposal. Essentially LaBB-CAT assumes that there are speaker turns, words, and sub-word segments.

In Bird & Liberman's definition anchor offsets are optional. In contrast, LaBB-CAT anchors are always given an offset. Where an accurate offset is not known, the offsets are computed by linear interpolation. These anchors are marked as having 'default' offsets using their status field, so they can be easily identified if required for data export, but having an approximate offset has two advantages:

- The anchors can always be sorted in relation to surrounding anchors, to help internal operations like displaying the transcript to the user.
- It provides research assistants a starting point to work with if they need to do manual alignment from scratch.

3 Advantages of Annotation Graphs

Having implemented essentially an annotation graph store, LaBB-CAT overcomes the limitations of ONZE Miner described is section 1, and supports a number of new possibilities for annotation creation and refinement.

3.1 Importing Data

Bird & Liberman's aim was to facilitate linguistic data exchange, and they demonstrated how annotation graphs could be used to model data from a number of linguistic tools.

LaBB-CAT modules can be implemented that convert data from the original format directly into LaBB-CAT's annotation graph structure, thereby escaping from any requirement that data be first convertible to a Transcriber file. We have already implemented converters for Transcriber files, Praat textgrids, Xwaves files as used by the Buckeye corpus, and ELAN (Sloetjes & Wittenburg, 2008) annotation files.

Simultaneous speech presented a particular problem for ONZE Miner's Transcriber-centric model. However with annotation graphs, either of the approaches to simultaneous-speech described in section 1.1 can be accommodated.

3.2 Exporting Data

Annotation graphs also allow for conversion of annotation data to a wider variety of formats.

As has already been expressed in the results from 2007 Multimodal Annotation Tools Workshops (Schmidt et al. 2008), and by the TILR2 Working Group 1 (Chochran et al. 2007), this is sometimes necessarily a lossy process, as different tools have different priorities, ontologies, and ways of structuring data (e.g. handling of simultaneous speech, as described in section 1.1). Thus not all of the information that was present in one file format when imported into LaBB-CAT will necessarily still be present when it's exported to a different format.

3.3 Round tripping

A further possibility that is suggested by import/export of data in various formats is that of re-importing data that has been exported and then refined. We have already implemented such round-trip data conversion, using algorithms that allow an annotation graph (or partial graph) to be merged into another:

1. A full Transcriber transcript is uploaded into LaBB-CAT, where an annotation graph is constructed.
2. Then a single utterance from the graph may be exported to Praat as a textgrid.
3. Edits are made in Praat to add, edit, and

re-align annotations.

4. The resulting textgrid can then be re-imported into LaBB-CAT, where it is converted into a graph fragment, which is compared to the full annotation graph stored in the database. The change deltas are then identified, validated, and saved to the database.

This is the kind of scenario presented by the TILR Working Group 1 as being a solution to the inevitable loss of information during conversion mentioned in section 3.2. They call this a "process-based" architecture for providing interoperability between different software tools.

With increased convertibility of LaBB-CAT annotation graphs from/to other formats, it's hoped that similar export/import interfaces can be developed involving other tools, using the annotation graph model as the pivot for annotation refinement. Information loss due to format conversion needn't always be a problem, as the central annotation store retains what gets lost in translation during export, and can thus use it to reconcile changes introduced during re-import, without loss of information.

3.4 Annotation Granularity and Structure

As annotation graphs don't include the compulsory definition of a single set of 'sections' with topic tags, any number of new layers can be created in LaBB-CAT and populated with independent sets of tags for annotating stretches of speech. These might contain annotations over long sections of transcript, or annotate only a few words at a time, or parts of words, e.g. stress-marked syllable annotations computed by combining HTK-produced phones within words and syllabification data from CELEX.

3.5 Syntactic Parses

In ONZE Miner, tree structures could not be modelled, so it was not possible to use readily available parsers like the Stanford Parser (Klein & Manning 2003) to provide syntactic parse annotations over utterances.

For annotation graphs, Bird and Liberman presented a possible technique for modelling trees using annotation graphs[1], where phrases can have their own bounding annotations, together marking the syntactic constituents of utterances.

We have taken this approach in LaBB-CAT, where a layer can be defined as containing trees. A newly created 'Stanford Parser' module can be configured to populate the layer with syntactic parses computed over words from another layer. These are represented in the annotation graph as arcs linking nodes, like all other annotations. We have also implemented an editor that allows these constructions to be viewed and edited using a tree layout more familiar to linguists.

[1] For example Bird & Liberman 1999b §3.2 Figure 10

4 Future Work

We have not yet implemented converters for some other commonly-used tools like Transana (Mavrikisa & Gernaniou 2011), Emu (Bombien et al. 2006), etc. While there will undoubtedly be some nuances to each of these cases, Bird & Liberman have shown that there should be no obstacle in principle to their representation as annotation graphs. Current and future work thus involves identifying tools and formats, both for import and export of data, making LaBB-CAT not only useful to a wider variety of researchers, but also making the data stored by it more shareable.

In addition there are many more possibilities for automatic annotation; lexical databases other than CELEX, other computations that may be useful, e.g. training classifiers for automated topic tagging, etc.

5 Conclusion

While ONZE Miner enabled several options for automatic and manual annotation of linguistic data, the adoption of an annotation graph framework for LaBB-CAT opens up new possible levels of granularity and sophistication for annotation and search.

The challenges that remain to be addressed reflect this new set of possibilities and the increasing diversity of domains in which LaBB-CAT can be of use as an annotation data store.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments.

References

Baayen, H.; R. Piepenbrock; H. van Rijn, 1995. The CELEX Lexical Database (Release 2, CD-ROM), LDC Catalogue No.: LDC96L14, Linguistic Data Consortium, University of Pennsylvania, Philadelphia.

Barras, C.; E. Geoffrois; Z. Wu; M. Liberman, 2000. Transcriber: development and use of a tool for assisting speech corpora production, Speech

Communication 33 (special issue on Speech Annotation and Corpus Tools) Numbers 1–2.

Bird, Steven; Mark Liberman, 1999a. A Formal Framework for Linguistic Annotation, Technical Report MS-CIS-99-01, Department of Computer and Information Science, University of Pennsylvania, (arXiv:cs/9903003v1 [cs.CL]).

Bird, Steven; Mark Liberman, 1999b. Annotation graphs as a framework for multidimensional linguistic data analysis, in proceedings from the Association for Computational Linguistics workshops "Towards Standards and Tools for Discourse Tagging" workshop, pp. 1-10, (arXiv:cs/9907003v1 [cs.CL]).

Boersma, P.; D. Weenink, 2005. Praat: Doing Phonetics by Computer (Version 4.3.14) [http://www.praat.org/]

Bombien, L.; S. Cassidy; J. Harrington; T. John; S. Palethorpe, 2006. Recent Developments in the Emu Speech Database System, in Proceedings of the Australian Speech Science and Technology Conference, Auckland, December 2006.

Cochran, Michael; Jeff Good; Dan Loehr; S. A. Miller; Shane Stephens; Briony Williams; Imelda Udoh, 2007. Report from TILR Working Group 1: Tools interoperability and input/output formats, Working Group report from the "Toward the Interoperability of Language Resources" workshop, 2007 LSA Summer Institute. [http://linguistlist.org/tilr/]

Fromont, Robert; Jennifer Hay, 2008. ONZE Miner: the development of a browser-based research tool, Corpora, vol. 3, no. 2, pp. 173–193.

Hawkins, Sarah, 2008. Introduction to Xwaves+. [http://www.ling.cam.ac.uk/li9/waves_08.pdf]

Klein, Dan; Christopher D. Manning, 2003. Accurate Unlexicalized Parsing, in proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430.

Ma, Xiaoyi; Haejoong Lee; Steven Bird; Kazuaki Maeda, 2002. Models and Tools for Collaborative Annotation, in proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002), European Language Resources Association, Paris.

Mavrikisa, Manolis; Eirini Geranioub, 2011. Using Qualitative Data Analysis Software to analyse students' computer-mediated interactions: the case of MiGen and Transana, International Journal of Social Research Methodology, Volume 14, Issue 3, pp. 245-252.

Pitt, M.A.; L. Dilley; K. Johnson; S. Kiesling; W. Raymond; E. Hume; E. Fosler-Lussier, 2007. Buckeye Corpus of Conversational Speech (2nd release), Department of Psychology, Ohio State University (Distributor).

[http://www.buckeyecorpus.osu.edu]

Schmidt, T.; S. Duncan; O. Ehmer; J. Hoyt; M. Kipp; D. Loehr; M. Magnusson; T. Rose; H. Sloetjes, 2008. An exchange format for multimodal annotations, in proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008).

Sloetjes, H.; P. Wittenburg, 2008. Annotation by category - ELAN and ISO DCR, in proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008).

Young, Steve; Gunnar Evermann; Mark Gales; Thomas Hain; Dan Kershaw; Xunying (Andrew) Liu; Gareth Moore; Julian Odell; Dave Ollason; Dan Povey; Valtcho Valtchev; Phil Woodland, 2006. The HTK Book (for HTK Version 3.4), Cambridge University Engineering Department.