Evaluation of Transcription and Annotation tools for a Multi-modal, Multi-party dialogue corpus

Saurabh Garg[†]**, Bilyana Martinovski, Susan Robinson**
Jens Stephan, Joel Tetreault[*]**, David R. Traum**

USC Institute for Creative Technologies
13274 Fiji Way
Marina del Rey, CA 90292, USA
{martinovski, robinson, stephan, traum}@ict.usc.edu

Abstract

This paper reviews nine available transcription and annotation tools, considering in particular the special difficulties arising from transcribing and annotating multi-party, multi-modal dialogue. Tools are evaluated as to the ability to support the user's annotation scheme, ability to visualize the form of the data, compatibility with other tools, flexibility of data representation, and general user-friendliness.

1. Introduction

As the range and variety of language resources develop, there is an increasing need for broader and more flexible tools to support the unique needs of multiple domains. The Mission Rehearsal Exercise Corpus (MREC) (Robinson et al., 2004) presents particular challenges to existing transcription and annotation tools, as it consists of primarily multi-party military training simulation dialogues including human-human radio dialogue, and dialogue between human and multiple virtual agents in the MRE scenario. The MREC is multi-modal in two senses: both in the sense of audio and visual data incorporating gesture, as well as in the sense of different modalities within a scenario (radio and face to face conversation). (Traum, 2001). As other reviews have focused on the former issues of multi-modality (c.f. (Bernsen et al., 2002)), we focus this review on the special issues and problems of the latter.

The following transcription and annotation tools were evaluated: Praat, Transcriber, TASX, Anvil, MMAX, DialogueTool, ILSP, the NITE Workbench and DAT. We evaluated each of these tools as to suitability for several tasks, including: transcription from audio or video, annotation of speakers and addressees, several types of dialogue acts, and dependent reference. We evaluated each tool along several dimensions, including: Input/Output flexibility, Portability, Source Code availability, Flexibility in Coding Scheme, Range of Markables, Audio/Visual Playback, Visual Interface, and User support. In addition to the tools' performance under our basic criteria, the nature of the corpus presents some special problems which were not easily solved in any of the tools reviewed, including many speakers within an interaction, and large amounts of overlapping speech and dialogues.

Part of MREC includes radio simulation data consisting of multiple channels of speech including a large number of participants (over 35) engaged in a common overall mission. While some participants have frequent contributions, a number contribute only occasionally. This number and variation in speakers (as well as the occasional challenge of identifying speakers) presents problems for tools that allocate individual tiers for each speaker, as the number of tiers either grows unwieldy or is limited by the tool. While the ability to deal with overlaps in some form seems basic to any spoken dialogue, the more frequent potential for overlaps in multiparty dialogue increases the necessity it be done gracefully, and that it be able to include any associated annotation. Furthermore, multi-party dialogue presents the further problem of overlapping dialogues (where individual overlaps between speakers do not conflict, because of different participants involved in the different dialogues).

2. Annotation Requirements

In our study we want to transcribe and annotate audio sessions of simulations in the MRE (human-computer interaction) and MRE-lite environments (human-human interaction). There are two goals of annotating the sessions: first, to construct a large corpus with which to test different theories and second, to use the corpus and theory testing to improve the performance of the MRE system.

Several factors influence the choice of annotation tool. First, the tool must be able to support the user's annotation scheme. Second, the tool must be user-friendly and possibly compatible with other tools. For our purposes we require a set of tools that can aid an annotator with transcribing data from audio files and possibly even video files. After a file is transcribed it needs to be annotated. In our study we want to annotate dialogue acts and reference between entities. Annotating dialogue acts involves recognizing and marking (or "tagging") utterances with different codes which represent the actions performed in the conversation. Often there is more than one code per utterance. Reference is the study of the relationships between entities in a discourse. Thus annotating reference involves recognizing and marking these entities (usually noun phrases) and then marking the relationship between an entity and a past entity. Reference resolution is important to a dialogue system because failure to resolve entities correctly can lead to confusion between the speakers.

[†]NCEAS, Suite 300, 735 State Street, Santa Barbara, CA 93101-3351,sgarg@ecoinformatics.org

[*]Department of Computer Science, University of Rochester, Rochester, NY, 14627, USA, tetreaul@cs.rochester.edu

Consider the dialogue fragment below from our MRE corpus. The Lieutenant and Sergeant are standing in front of a crash site with a damaged car, a Humvee, a boy lying on the ground, and a woman hunched over him:

14	Sgt	This *woman* and *her son* came from *the side street* and *our driver* didn't see *them*
15	LT	can *we* medevac *him* out of *here*?

There are a number of dialogue acts performed: utterance 14 is an assertion and 15 is an information request. But 14 also ends the turn, while 15 is an acknowledgment of 14. At the reference level, the entities for both utterances are listed in italics. In 14, *Her son* is a bridging reference back to *this woman*. *Our driver* refers to the Humvee in the scene and *them* refers back to the compound *woman and her son*. In 15, *we* refers inclusively to the LT and Sgt (and perhaps some of the units to which they belong). *Him* is ambiguous, though most likely to refer to the injured boy in view. *Here* refers to the local setting.

Given the requirements for each type of annotation and the need to make things as easy for annotators as possible, we outlined several issues to rate each annotation tool.

Input/Output flexibility What is the data format for the tool's input? For audio files this may require converting our files to a different format just to use the tool. Also, is the output from the tool compatible with other tools and/or easily readable so one can check the annotators' work?

Portability Can the tool be used on different operating systems (such as Windows or Linux)? Also, does the tool require special packages?

Source Code Does the tool come with source code so it can be altered? This is useful for possibly extending or modifying the coding scheme offered by the tool, or altering the display to make it more user-friendly.

Audio/Visual interface Does the tool offer an easy-to-use method for playing sections of audio (or video) and segmenting sections? Can large audio files be handled by the tool? Also, can it play back a sentence's corresponding audio file to check for intonation.

Comments Does the tool allow the user to make comments or notes on their annotation?

Flexibility in Coding Scheme Does the tool require using its own scheme or can one specify a different one?

Marking (Annotation Only) How much can the tool mark? Just words, or also groups of words or acts? Can it also mark segments of sentences or just an entire sentence? Is it possible to mark discontinuous acts (with material in the middle that is not part of the act)?

Viewing work Does the tool have a large enough display to show the current work and the corresponding codes in a clear manner? Is earlier work visible as well?

User manual Does the tool come with a user manual? Having a manual saves on training time and allows annotators a quick reference if needed.

Optimally, one would want a tool that could do both transcription and annotation but we did not find one that one was flexible enough to use. In fact, simply finding a tool that can do two different schemes of annotations as we require is quite difficult. Thus we break up our review of tools into two sections: transcription tools and annotation tools. For each tool, we give a brief description and a list of its main advantages and disadvantages.

3. Transcription Tools

These are tools used for transcribing a recorded session of audio and/or video data. The result is usually either a simple text or XML file with a time-ordered list of the dialogue between the session participants. Typical information annotated for each sentence (or sub-sentential unit when possible) is the speaker, the start and end time of the sentence, and any comments about the marking. In this section we review three transcription tools: Praat, Transcriber, TASX and Anvil.

3.1. Praat (v4.0.43)

Praat [1] is a phonetics tool used for speech analysis and synthesis. It was not originally intended for text transcription but rather for editing and analyzing sound files. It involves taking in an audio file, then clicking on the waveform for marking start and end segments then typing in the words for that segment. After transcription is done, the transcribed text along with the time stamp info is saved in a text file.

Praat is one of the best developed and flexible transcription tools in that it can run on both Windows and Linux platforms, its output is a simple text format, it is compatible with the video annotation tool Anvil, and most of all, it offers an easy segmentation interface. On the other hand, since it is primarily a phonetics tool as opposed to a discourse markup tool, previously transcribed segments are hard to see, and long segments are also difficult to see in their entirety. It also cannot handle overlapping speech, when two or more speakers speak at the same time. It is possible to transcribe each speaker onto a separate track, but the tool does not merge each track. Adding a field to write comments would be helpful.

Praat has undergone several changes since the version we tested to make it more user-friendly but most of the problems we cited above have not been addressed. However, Praat offers good support as well as source code and the authors say that they can tailor the tool to your needs if necessary.

3.2. Transcriber (v1.4)

Transcriber[2]. is a transcription tool developed in Tcl/Tk and C extensions. It runs on various Unix systems as well as Windows operating systems. It was originally intended for use in transcribing broadcast news recordings so it makes a good match with our domain.

Using the tool involves first segmenting the audio file. This is done by playing the file and hitting a key upon hearing a segment break. This is easier than in Praat since it

[1] Praat: http://www.fon.hum.uva.nl/praat/ for more details

[2] Transcriber: http://www.etca.fr/CTA/gip/Projets/Transcriber/

involves a lot less clicking and segmentation can be performed while listening continuously. Each segment is then transcribed by clicking on its waveform and entering the text in the top segment of the display panel.

Being specifically tailored for dialogue transcription, Transcriber avoids several of the drawbacks of using Praat for such a task. It is able to accommodate different speakers and has two windows – a large one to see the entire dialogue and a smaller one to do the transcribing. Transcriber also allows multiple tracks for each speaker, and offers a way to annotate speaker information as well as a field for comments. The tool does not have as a rich a manual as Praat does, but it is straightforward enough to use without one.

3.3. TASX (v. alpha 1)

Another annotation tool, TASX[3], is a Java program that handles both audio and video, and works for Linux and Windows. It was originally intended to study prosody acquisition. Like the previous tools, the primary display window consists of different tiers or tracks. One can treat each track as a speaker and segment the track by clicking the start and end points. Input and output are in XML form.

We tested an early version of TASX that did not come with good documentation or a lot of the advantages the current version has. The main disadvantages of the version we tested involved ease of use. We found that it was hard to view what you were working on and view past work. In addition, our annotators found that the segmentation operation was unwieldy. However, the latest version seems to address many of these disadvantages. Other advantages include ability to link with Praat and source code availability.

3.4. Anvil (v3.6)

Anvil[4] (Kipp, 2001) is a JAVA tool for Windows, Unix, and Mac that was developed for video analysis of gesture research. Its primary advantages are a rich tier system in which transcribers can specify relationships between tiers and its ease in marking different types of annotations. For our purposes these abilities were more than required, which was simple text transcription. On the other hand, with Anvil it is difficult to view past work and the unused functionality has the potential to confuse annotators. However, if one uses Anvil for transcription, it could double as a minor annotation tool in that different attributes can be specified in each tier. We used these to annotate dialogue acts. Anvil could be improved for this task with a larger display window to view words from a segment.

4. Annotation Tools

After transcription, the next step is annotating the texts with our two different coding schemes: dialogue acts and references. Our perfect tool would be one which could work on input from our transcription tool, allow user-specified coding schemes so one could use the same tool for both coding schemes, and finally present the different annotations in a readable manner. Another attribute of a good tool is the use of “standoff marking” which means that there are different files for each code in a scheme which makes it easy to check individual codes in a file. In this section we describe MMAX, DialogueTool, the ILSP tool, NITE, and DAT.

4.1. MMAX (v0.9)

MMAX[5] (Müller and Strube, 2001) runs in both Java 1.3 and 1.4 and has been tested successfully in Windows but has had problems in Linux. It was originally conceived for reference annotation but it can be altered to handle utterance level annotations as well. The underlying concept is that you can highlight anything in the text - a word, series of words, sentences, parts or groups of sentences, or some combination of the above (called a markable) and assign properties to that markable (entity). The annotation scheme is user specified.

The greatest advantage of this tool is that it can support both of our annotation schemes so we only have to use one tool. This is easier for annotators since they only have to learn and use one tool. Other advantages of MMAX are that it has good support from its creators and a decent manual.

4.2. DialogueTool

DialogueTool (Hardy et al., 2003), from the University of Albany, runs on Windows machines and is intended for annotating at the utterance level and cannot be used for annotating words or sub-sentential entities. The window of the tool is split into two sections - the dialog display which highlights the current sentence being annotated, and a panel of drop-down coding menus. As in MMAX, it is possible to annotate the same sentences with multiple codes, however in DialogueTool the only thing you can mark up are sentences and not words. Clauses can be marked but only after the sentence has been segmented, a function that DialogueTool offers.

If one were to use this tool for our purposes, it would be used for strictly annotating all sentence-level codes. All reference and sub-sentential codes would have to be done in MMAX or some other tool. Like MMAX, the coding scheme is user-specified. The input to the tool is a little simpler than MMAX in that all that is required is a simple text file with each segment annotated with speaker information. DialogueTool could be improved by being able to tag groups of utterances

4.3. ILSP Tool

Another tool especially made for reference annotation is the ILSP tool[6]. It uses the MATE reference annotation scheme, which is a subset of our scheme. There are two main drawbacks to the tool, first is the lack of a reference manual since the tool is not so straightforward to use and second is that you have to use their coding scheme, it is not possible to alter it as in MMAX or DialogueTool.

4.4. NITE Workbench (v 2)

By design, the NITE Workbench[7] addresses many of our basic criteria, by including both transcription and flex-

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

[4] Anvil: http://www.dfki.uni-sb.de/ kipp/anvil/

[5] http://www.eml-research.de/english/Research/NLP/Downloads

[6] ILSP Tool: http://www.ilsp.gr/

[7] NITE Workbench: http://nite.nis.sdu.dk

Evaluation	Praat	Transcriber	TASX	Anvil	MMAX	DT	ILSP	DAT
Portability	+	+	+	+	+	-	+	+
Source Code	+	+	+	?	-	-	-	+
A/V Interface	+	+	+	+	N/A	N/A	N/A	+
Comments	-	+	+	-	+	+	-	+
Coding Scheme Flexibility	N/A	N/A	N/A	+	+	+	-	+
Viewing Work	-	+	-	-	+	+	+	-
Ease of Use	-	+	-	-	+	+	-	-
Support/Manual	+	-	-	+	+	-	-	+
Overall	+	+	-	-	+	+	-	-

Figure 1: Summary of Tool Evaluation

ible annotation definition and levels. The tested version, however, was buggy and its method of segmenting data is so complex as to render it unusable.

4.5. DAT

The DAT[8] from the University of Rochester allows playing sound files for utterances while annotating, which is very useful for checking intonation and inflection when labeling dialogue acts. However it, it does not allow markup at the sub-sentential level, and also requires that the sound file for a dialog be broken up into one sound file per utterance. Source code (perl/tk) is available so the coding scheme can be changed (though not as easily as with a config file). Input/Output was a negative, since files are in a special SGML format (with no standoff), rather than XML.

5. Summary of Tool Evaluation

Figure 1 provides a quick description of the properties and advantages of the tools we tested for easy comparison. Table entries marked with a "+" indicate that the tool performed well in that category, and a "-" means it could have performed better. In some cases, these categories aren't binary, such as ease of use, so we marked the category based on how well it fit our minimum expectations.

There were many different ways to pick the tools, but in the end, the factors we weighted the highest were ease of use by the annotators and ease of import and export of data. As none of the tools we tested were capable of handling all of our needs, we opted to use Transcriber for general transcription, MMAX for coding, and Praat for prosodic analysis, utilizing Perl scripts to convert data between formats, when necessary. Transcriber was selected because it offered easy playback and segmentation mechanisms. MMAX's ability to support both user-defined annotation schemes and multiple levels of markables made it the obvious choice.

While some of the particular problems of appropriate tool support encountered in the MREC development seem specific to the task domain (mixed radio and face to face dialogues of military scenarios), similar challenges will need to be faced in other domains as human language technologies expand into increasingly realistic discourse situations where multi-party dialogue is common. The common use of communication technology (e.g. widespread use of cellular phones) renders the occurrence of such multi-modal communication as discussed here increasingly commonplace in natural dialogue situations.

Acknowledgments

The work described in this paper was supported by the Department of the Army under contract number DAAD 19-99-D-0046. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the Department of the Army. We would like to thank Mary Harper, Tomek Strzalkowski, Hilda Hardy, Michael Kipp, Cristoph Mueller for advice and help acquiring and using the tools described above. We would also like to thank Damon Davison, Nathan Klinedinst and Despoina Theodorou for additional help in evaluating the tools.

6. References

Bernsen, N.O., L. Dybkjr, and M. Kolodnytsky, 2002. The nite workbench - a tool for annotation of natural interactivity and multimodal data. In *Third International Conference on Language Resources and Evaluation (LREC2002)*.

Hardy, Hilda, Kirk Baker, Helene Bonneau-Maynard, Laurence Devillers, Sophie Rosset, and Tomek Strzalkowski, 2003. Semantic and dialogic annotation for automated multilingual customer service. In *Eurospeech-2003*.

Kipp, Michael, 2001. Anvil - a generic annotation tool for multimodal dialogue. In *7th European Conference on Speech Communication and Technology (Eurospeech)*.

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

Robinson, Susan, Bilyana Martinovski, Saurabh Garg, Jens Stephan, and David Traum, 2004. Issues in corpus development for multi-party multi-modal task-oriented dialogue. In *Fourth International Conference on Language Resources and Evaluation (LREC 2004)*.

Traum, David R., 2001. Ideas on multi-layer dialogue management for multi-party, multi-conversation, multi-modal communication: Extended abstract of invited talk. In *Computational Linguistics in the Netherlands 2001: Selected Papers from the Twelfth CLIN Meeting*.

[8] DAT Tool: http://www.hcrc.ed.ac.uk/ amyi/mate/dat.html