IORelator: A Graphical User Interface to Enable Rapid Semantic Annotation for Data-Driven Natural Language Understanding

David DeVault and Susan Robinson and David Traum
USC Institute for Creative Technologies
13274 Fiji Way
Marina del Rey, CA 90292
{devault,robinson,traum}@ict.usc.edu

Abstract
This paper describes a new annotation GUI, called IORelator, which is designed to facilitate rapid semantic annotation in support of high-performance data-driven NLU for spoken dialogue systems. We summarize our requirements for rapid NLU annotation, and discuss how the GUI views and operations that IORelator provides meet these needs by enabling thousands of natural utterances to be quickly annotated with their correct semantics.

1 Introduction
This paper introduces a new annotation GUI, called IORelator, that is designed to improve the process of annotating the semantics of large collections of natural language utterances. This new GUI has a number of design features which work together to enable an annotator to focus on relevant parts of a growing annotated corpus, and to add new annotations quickly and in a consistent manner. These design features include:

- The annotator can create one or more views of the data using several kinds of filters. The filters enable similar utterances or similar semantics to be grouped together within a view (a sub-window) in the GUI.
- Several of these filtered views can be positioned simultaneously on screen, and can be connected in a way that allows the annotator to choose the best semantics by highlighting and exploring previous annotations of similar or related utterances.
- Together, these features allow groups of similar utterances to be identified, selected, considered, and quickly annotated.

We begin by motivating our need for rapid semantic annotation.

2 Motivation and Background
In this paper, we consider an annotation task that arises for the natural language understanding (NLU) component in SASO-EN (Traum et al., 2008), a virtual human dialogue system. This system allows users to interact in interactive spoken dialogue with virtual human characters, and is designed to serve as a negotiation training tool, where users learn about negotiation tactics in the context of the culture and social norms of a particular community. The semantic representation for NLU, also used in several previous virtual human systems (Traum, 2003), is an attribute-value matrix (AVM), where the attributes and values are linked to a domain ontology and task model (Hartholt et al., 2008). Figure 1 shows an example AVM for an utterance such as “we can provide you with power generators”. To facilitate statistical NLU, the AVMs are linearized using a path-value (or key-value) notation, as shown in Figure 2. We call these linearized AVMs frames. In SASO-EN, the NLU module takes a spoken utterance (as rec-
Table 1: NLU performance vs. training set size.

<table>
<thead>
<tr>
<th>Training set size</th>
<th>NLU F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.31</td>
</tr>
<tr>
<td>500</td>
<td>0.65</td>
</tr>
<tr>
<td>1000</td>
<td>0.70</td>
</tr>
<tr>
<td>2000</td>
<td>0.74</td>
</tr>
<tr>
<td>3000</td>
<td>0.75</td>
</tr>
<tr>
<td>3826</td>
<td>0.79</td>
</tr>
</tbody>
</table>
theme power-generator.¹ Because only 5 of 115 frames in the output domain match this filter, only 5 frames are visible in the view. The annotator can thereby focus on only these frames. In the figure, the user has used filters to create Utterance View 1 as a view that shows only utterances which have already been linked, and Utterance View 2 as a view that shows only utterances which have not yet been linked. In this way, Utterance View 2 serves as a “todo list” for utterances requiring annotation; the annotator can consult the previous annotations of already linked utterances in Utterance View 1 when desired. Additionally, the user has added a keyword filter (see Utterance Text Filter) to Utterance View 2, so that only the 20 unlinked utterances containing the word generator are displayed. Together, these filters allow the annotator to focus attention on the 20 utterances including the word “generator” and the 5 possibly appropriate frames for these utterances.

Each view in IORelator can be either connected or not connected. When a view is connected, items selected with the mouse cause related items in other connected views to become highlighted. In the figure, the user has marked all the views as connected except for Utterance View 2, which contains unlinked utterances. In this screenshot, the user has selected the bottom-most frame in the Frame View; this selection has been transferred (Selection Transfer) to Utterance View 1 by automatically selecting the utterances in that view that are already linked to that frame. This helps the annotator to consider this frame as a possible annotation for some of the unlinked utterances, as they can be quickly compared to other utterances already linked to that frame.² A second aspect of connectedness between views is that filters applied in connected views automatically filter out unrelated items in other connected views.

The item selection in a view v is determined

¹To support filtering with regular expressions, IORelator requires that all items in its input and output domains be convertible to plain text. This does not preclude the use of structured semantic annotations such as SASO-EN frames in the output domain. Any semantic annotation can be used so long as it can (1) be depicted on screen in views and (2) converted to plain text so that filtering operations can be performed.

²A long-term motivation for this functionality is to lower the learning curve associated with the NLU annotation task, by allowing novice annotators to consult previous expert annotations for similar utterances.
6 Discussion and Future Work

We have found using IORelator to be substantially faster than prior approaches to SASO-EN NLU annotation, which have included linking utterances to frames inside the Protégé GUI (using custom dialogue boxes and forms), or alternatively, using a simple text editor for annotation.\footnote{While there has not been a formal study to quantify exact annotation time for these previous approaches, we estimate that the previous approach to annotation in Protégé, where navigation and searching frames is more cumbersome, required at least 1 minute per utterance for an expert annotator. Several schemes for annotation in a text editor have also been implemented, but we estimate that even the fastest of these required an expert about 20 seconds/utterance.} This has made it easier to annotate enough training data that the NLU module can achieve the high-performance region of Table 1. Our annotators attribute the speed-up to the rapid browsing, filtering, and linking operations discussed in Section 4.

Future directions include evaluation of additional design goals for IORelator. These include evaluating throughput for novice annotators or hybrid teams of novices and experts, efficiency of revision and maintenance as opposed to initial annotation, inter-annotator agreement, and the extent to which the throughput observed here transfers to other NLU annotation schemes.

Acknowledgments

We thank our anonymous reviewers, Ron Artstein, Anton Leuski, Arno Hartholt, Tom Russ, and Kenji Sagae for helpful discussions. The project or effort described here has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References


