A Artificial Interactions

An artificial data generator was designed and implemented to simulate doctor-system interactions, with sentence *templates* defining the skeleton of each entry in order to maintain high-quality sentence structure and grammar. We used a context free grammar to implement a *template language* that can specify a virtually unlimited number of templates and generate as many examples as desired. Below we show a simplification of three sample rules from the grammar:

$$\langle S \rangle \rightarrow \text{maximum heart rate on } \langle P \rangle \text{ today?}$$
$$\langle P \rangle \rightarrow \text{the day before } \langle P \rangle$$
$$\langle P \rangle \rightarrow \text{the Monday after}$$

A sample derivation using these rules is "maximum heart rate on the Monday after today?".

The implementation allows for the definition of any number of non-terminals, which we call *types*, and any number of *templates*, which are the possible right hand sides of the starting symbol S. The doctor-system interactions can be categorized into three types: *questions*, *statements*, and *clicks*, where templates can be defined for each type, as shown in Table 1. Given the set of types and templates, a virtually unlimited number of sentences can be derived to form the artificial dataset. Since the sentence generator chooses each template randomly, the order of sentences in the output dataset will be random. However, two important aspects of the real interactions are context dependency and coreference. To achieve context dependency, the implementation allows for more complex *combo templates* where multiple templates are forced to come in a predefined order. It is also possible to specify groups of templates, via tagging, and combine groups rather than individual templates. Furthermore, each NL sentence template is paired with a LF template, and the two templates are instantiated jointly, using a reference mechanism to condition the logical form generation on decisions made while deriving the sentence.

Table 1 shows examples of how artificial sentences and their logical forms are generated given templates and types. Most types are defined using context free rules. There are however special types, such as [clocktime] and [range()], which are dynamically rewritten as a random time and a random integer from a given range, as shown in Examples 3 and 4, respectively. Note that most

Example types.
week_days→Monday | Tuesday | ... | Sunday
daily_intervals→morning | afternoon | evening | night
daily_intervals_logic→ Morning | Afternoon | Evening | Night
any_event→heart rate | bolus | blood glucose level
any_event_logic→HeartRate | Bolus | BGL

Example 1: a statement, involving referencing

Let's go to [week_days]. $\rightarrow DoSetDate([\$1])$
Possible derivations:
- Let's go to Monday. $\rightarrow DoSetDate(\text{Monday})$
- Let's go to Tuesday. $\rightarrow DoSetDate(\text{Tuesday})$

Example 2: a combo statement capturing temporal dependence

[[let's/please/we can]/can we] turn the [any_event] off[\$1:./?]
$\quad DoToggle(\text{Off}, [\$2 : any\_event\_logic])$
...and the [any_event] too.
$\quad DoToggle(\text{Off}, [\$1 : any\_event\_logic])$
Possible derivations:
- please turn the bolus off. $\rightarrow DoToggle(\text{Off}, \text{Bolus})$
  and the heart rate too. $\rightarrow DoToggle(\text{Off}, \text{HeartRate})$
- can we turn the blood glucose level off?
  $\quad\quad\quad\quad\quad \rightarrow DoToggle(\text{Off}, \text{BGL})$
  and the bolus too.$\rightarrow DoToggle(\text{Off}, \text{Bolus})$

Example 3: a click, involving the special type clocktime

$Click(e) \wedge e.type == [any\_event\_logic]$
$\quad\quad\quad \wedge e.time == [clocktime]$
A possible derivation:
- $Click(e) \wedge e.type == \text{Bolus} \wedge e.time == \text{12:36 PM}$

Example 4: a question, involving the special type range

is there [a/any][valued_event]
$\quad\quad\quad\quad$ [more/less] than [range(-500,500)]?
$Answer(Any(d.value[\$3 :>/<][\$4]$
$\quad\quad\quad\quad \wedge d.type==[\$2:valued\_event\_logic]))$
One possible derivation:
- is there any heart rate less than 250?
$\quad Answer(Any(d.value < 250 \wedge d.type == \text{HeartRate}))$

Table 1: Examples of generation of artificial samples.

examples use referencing, which is a mechanism to allow for dynamic matching of terminals between the NL and LF derivations. In Example 1, $1 in the logical form template refers to the first type in the main sentence, which is [week_days]. This means that whatever value is substituted for [week_days] should appear verbatim in place of $1. In case a coordinated matching from a separate list of possible options is required, such as in Example 2, another type can be selected. In Example 2, [$2:any_event_logic] will be option i from the type [any_event_logic] when option i is chosen in the main sentence for the second template, which is [any_event].