Managing Unconstrained Natural Language Challenges & Ideas for Deploying CNL Systems

Rogan Creswick | GF Summer School | August 29, 2013 creswick@galois.com



Natural Language @ Galois



Frequent Requests

Parse Documents

Explain Results

Translate Documents to Formal Language:

- W3C Guidelines to Tests
- Policy (eg. HIPAA) to XML

Formal Language to NL:

- Audit logs
- Query results
- Output to different audiences:
 - Developers
 - Legal
 - SME

Example: Executable Specifications

To example, in this do not use the threading.		
cause the content of a (non-table header)	cell to b	
displayed centered and in bold.	W/2C accessibility	
Techniques for checkpoint 5.4		
5.5 Provide summaries for tables. [Priority 3]		
For example, in HTML, use the "summary	/" attribute duidelines	
of the TABLE element.	3	
Techniques for checkpoint 5.5		
5.6 Provide abbreviations for header labels. [Priority 3]		
For example, in HTML, use the "abbr" att	ribute on	
the TH element.		
Techniques for checkpoint 5.6		
also to checkpoint 10.3	<pre>\$5(!table!) noSubElt(!th!) each(function () /</pre>	
siso to checkpoint 10.5.	report.error('Table does not have column headers <th>', this).</th>	', this).
	});	
	<pre>\$5('table').noSubElt('caption').noAttr('title').each(function () {</pre>	
	report.warning('Table has no caption or title attribute', this);	
	<pre>});</pre>	
Executable tests	<pre>\$('table').noAttr('summary').each(function () {</pre>	
	<pre>report.warning('Table has no summary attribute', this);</pre>	
(Eivol III)	n:	
(FIVEOI)		
	<pre>\$5('th').noAttr('abbr').each(function () {</pre>	
	report.advisory(' <th> has no abbrevation attribute', this);</th>	has no abbrevation attribute', this);

NL Interfaces to Formal Languages

galois



Application Grammars in General





Linearization Works Great!



Input is problematic

Phrasebook> p "I'd like pizza"

The parser failed at token "I'd" Phrasebook> p **"I would like pizza"** The parser failed at token "would"

Phrasebook> p "I want pizza"

The parser failed at token "pizza"

Phrasebook> p "I want pizza, please"

The parser failed at token "pizza,"

Phrasebook> p "I want a pizza, please"

The parser failed at token "pizza,"

Phrasebook> p "I want a pizza"

Linearization can raise expectations

Language generation is very compelling

Convincing the audience that the system is fragile is hard.

Demonstrated parsing is also very convincing

• We (GF developers) know the rough edges.

Using the parser is quite difficult

• Minor typos cause the parse to fail.



How can we relax the parser for a given concrete language?



Relaxing the parser



Relaxing the parser

Admit that there will be errors. May introduce ambiguity. ulletMapping will not be precise. Most input should map to some valid string. Interact with the user to resolve ambiguity. (eg: "Did you mean?" interface) Identify/implement heuristics.



Relaxing the parser

• Admit that there will be errors. Limit the domain. 2. Semantic nuance is often irrelevant. The user knows the domain, just not the syntax.



Relaxing the parser



Insights

Concrete syntaxes already add irrelevant details (compared to abstract syntax)

• Eg: Italian gender alignment in Foods.

Accept:

- synonyms/hyponyms
- Misspellings
- Incorrect case
- Etc...

Leverage SMT techniques

• Account for distortion, fertility, insertion

A (more) formal description

Given a grammar G

• s.t. G accepts L

galois

Define a grammar transform t_g

Define an input transform t_i

•
$$t_i(L'') \in L$$





Solution? Create a DSL!

Example

```
transform = do
 trToParser $ do
    addSynonyms [ ("wine", "vino")
                 , ("that", "thar")
                 , ("that", "the")
                 , ("is", "be")
                 , ("is", "are")
    addHyponyms [ ("cheese", "cheddar")
                , ("cheese", "brie") ]
    addHypernyms [ ("cheese", "food")
                 , ("cheese", "snack")
                 , ("wine", "drink")
                 , ("wine", "beverage") ]
```

Example

```
Foods> p -lang=Eng "that cheddar is warm" | l
that cheese is warm
quel formaggio è caldo
```

```
Foods> p -lang=Eng "thar drink be warm" | 1
that wine is warm
quel vino è caldo
```

```
Foods> p -lang=Eng "thar fresh Italian vino be warm" | 1
that fresh Italian wine is warm
quel vino italiano fresco è caldo
```

DSL Features (in progress)

$$t_g(pgf)$$
 Modify PGF

- addSynonym
- addHyponym
- addTerm
- setProbabilities

$t_i(input)$ Modify a parser (and possibly PGF)

- matchStems, spellcheck, ignoreCase
- addWordNgram, dropWordNgram
- dropWords, wordsWords
- replaceInput

Ideas/Future work

Finish Implementation!

• It is currently very slow.

Map categories between grammars

- Addresses word-order issues
- "How large is..." vs. "What is the size of ..."
- "that wine is warm" vs. "the warm wine is over there"

WordNet / FrameNet (?)

Robust parser

Thanks!

Rogan Creswick

creswick@galois.com

Based in Portland, Oregon