Logic Programming: from NLP to NLU?

Paul Tarau

Department of Computer Science and Engineering
University of North Texas

AppLP'2016

NLP: one of the original motivations for LP

- Colmerauer's Metamorphosis Grammars
- Pereira and D.H.D Warren's Definite Clause Grammars
- Montague Grammars, implemented in Prolog by D.S. Warren
- Veronica Dahl: logic representations for more realistic fragments of natural languages
- DCGs + backtrackable assumptions: "parsing as hypothetical reasoning"
- ⇒ a penchant of Logic Programming towards the higher objectives of Natural Language Understanding (NLU).

LP: the potential for disruptive NLP→NLU evolutions

- today's search engines (Google, Bing etc.) are becoming NLU-enabled Question Answering systems!
- ⇒ high hopes for approaching some NLU objectives
- inhibited during the "AI-winter" - but it is Spring time now (or maybe already Summer time :-))
- today: prevalence of statistical NLP, ML and emergence of "deep learning"
- but ultimately: we still want a observable behavior and human-understandable output
- which is the closest formal mechanism mimicking natural language?
  - sentence $\sim\sim$ proposition
  - verb $\sim\sim$ predicate
  - noun $\sim\sim$ constant (well, if we forget about λ-terms)
  - adjective $\sim\sim$ unary predicate/property
  - who/which/what/where/when $\sim\sim$ (logic) variables
- ⇒ (some form of) logic-based representation

A (rich) choice of logic forms and tools for using them

- Logic forms
  - neo-davidsonian: sentences as “event variables” with dependents hierarchically describing semantic relations
  - Montague-style lambda term semantics
  - DRT: discourse representation theory
  - AMR: abstract meaning representations
  - conceptual graphs
  - approximations: shallow/simplified logic representations
- LP tools
  - classic Prolog
  - constraint solvers
  - theorem provers
  - SAT/SMT/ASP solvers
  - Inductive LP, Probabilistic LP

Our choices: we go for the most practical ones :-)

our favorite LP tool: good old vanilla Prolog (it is now in the same league as Scala, higher than Erlang or Scheme) in the Tiobe index!)

- scalable, rich ecosystem (SWI-Prolog)
- LeanProlog (Java-based)/SWI-Prolog hybrid
- ⇒ easy access to third party NLP-tools (mostly in Java)
- SD-drives and/or memory-based Hadoop variants are fast enough for large scale file-based data flows in the tool chains

our favorite NLP-tools:

- state of the art, statistically trained parsers that are “logic-friendly”
- we like lexicalized representations (going back to Montague’s λ-terms)
- ⇒: Combinatorial Categorial Grammars (CCGs) e.g., the `C&C` parser
- semantic representations as Prolog terms: the `boxer.pl` program
- other resources: WordNet, VerbNet, PropBank, CCGBank, dependency parsers, AMR tools

Sketch of a CCG-reducer in Prolog

```
:-op(400,xfx,(/)).
:-op(400,xfx,(\)).

red(Xs):-red(Xs,s). % reduce a sentence to root symbol s.

red([S],S).
red([X/Y,Y|Xs],S):-red([X|Xs],S).
red([Y,X\Y|Ys],S):-red([X|Ys],S).
red([X/Y,Y/Z|Xs],S):-red([X/Z|Xs],S).
red([Y\Z,X\Y|Xs],S):-red([X\Z|Xs],S).
```

Some help from DCGs

DCGs can be used to build the CCGs representation of a sentence as in:

```
the -->[np/n].
cat -->[n].
chased -->[(s/np)\np].
dog -->[n].
playful --> [n/n].
quiet -->[n/n].
quick -->[n/n].
and --> [X/X].

sent-->the,quick,and,playful,dog,chased,the,quiet,cat.
```

a lexicalized representation: categories associated to each lexical element

```
?- sent(S,[]),red(S).
S = [np/n, n/n, n/n, n/n, n,  (s/np)\np, np/n, n/n, n] .
```

real-life parsers: CYK or A^* search, large training sets, thousands of lexical categories

Graph-based NLP

- about 3-4 citations a week: the TextRank algorithm (joint work with Rada Mihalcea)
- very simple, but extremely effective idea: connect words and sentences where they occur in a graph
- run PageRank-like algorithms on the graph
- collect the highest ranked sentences as a *summary*
- collect the highest ranked words as *keywords*
- possibly using richer graphs (e.g., with WordNet's synset relations)
- 1300++ enhancements and applications are now out there, but none seem to bring in significant NLU elements
- our new refinements: can we enhance TextRank by using a richer graph based on logic representations?
- some obvious things: dependency graphs centered around nouns and named entities seem to give intuitively very nice results

Prolog-based natural language-enabled agents

- an NLU-minded application we have worked more than a decade or ago: Prolog-based natural language-enabled agents
- they interacted with the Prolog version of WordNet and Google's metasearch API to bring in knowledge distributed over the internet
- logic inferences using a Prolog representation (as dynamic clauses or backtrackable assumptions) of the agents' short-term memory
- shared virtual worlds and interactive story telling systems developed on top of them
- these days, fields like interactive story telling have become an integral part of computer games (e.g., Minecraft Story Mode)
- voice-enabled software agents are part of major mobile phone platforms (e.g., Siri, Cortana, Google Ok)
- home automation systems (e.g., Alexa) and more generally in the upcoming Internet-of-Things platforms all need predictable and debuggable inference mechanisms

Revisiting logic programming-based NLP/NLU tools

- today LP-based tools can benefit from access to improved metasearch
- massive online knowledge repositories like Wikipedia
- constraint programming libraries, now part of most widely used Prolog systems can improve the speed and the accuracy of WSD, an important NLU component
- SAT-solvers and ASP-based systems can help narrowing down some of the heavily combinatorial aspects related to the inherent ambiguity of natural language
- they can also help dealing with incomplete or noisy information streams one faces in voice and image recognition tasks

Some future steps

- interaction at word level between symbolic and connectionist representations
- lexicalized grammar and logic representations can be more easily interfaced with deep-learning tools (e.g. word2vec and similar “distributed” word representations)
- graph-based NLP leverages “holistic” statistical language properties that can bring meaningful logic representation beyond sentence level
- a very interesting document type: scientific papers
  - usable as gold standards for training (author provided keywords and abstracts)
  - a more precise document structure: sections, related work, conclusions
  - simpler, closer to logic sentence structure
  - available domain ontologies and related-work databases
  - impact information (e.g., Google Scholar rankings)

Conclusions

- NLU-focused NLP is a promising application field for LP tools and techniques
- big data analytics need more and more effective NLU for accuracy and usefulness to humans
- NLU-enhanced tools like voice based assistants (Alexa, Google Ok, Siri) are becoming widely used (and might have billions invested in their progress)
- interactive fiction and related game types are ready for realistic natural language dialog
- shared virtual reality sites (e.g., Minecraft, Roblox based) are ready for NLU-enabled agents
- and most importantly: search engines are evolving into NLU-enabled question answering systems

Questions?

Picasso:

Questions tempt you to tell lies, particularly when there is no answer.