Logic Programming: from NLP to NLU?

Paul Tarau
Department of Computer Science and Engineering
University of North Texas
paul.tarau@unt.edu

Natural Language Processing (NLP) was one of the original motivations leading to programming in logic [1] back in the seventies, with Colmerauer's Metamorphosis Grammars [2] and with Pereira and D.H.D Warren's Definite Clause Grammars [3], enhanced later with mechanisms for hypothetical reasoning [4]. Montague Grammars (implemented in Prolog by D.S. Warren [5]) and work by Veronica Dahl on defining logic representations for more realistic fragments of natural languages (including long distance dependencies and anaphora resolution) [6] have all shown a penchant of Logic Programming towards the higher objectives of Natural Language Understanding (NLU).

After being long delayed (and partly frozen by the AI winter) recent progress in NLU promises to bring disruptive paradigm shifts in human-computer interaction and several directly and indirectly related industries. In fact, hopes for more logic-based NLU are high again, partly due to the possibility of sharing successful technologies and tools with successful fields like deep-learning neural-networks, machine learning, statistical parsers and graph-based NLP.

This brings us to the obvious question: what new role can logic programming play in this new context?

When trying to sketch an answer, after more than a decade spent on other research topics, we became aware that one can benefit today from the extended logic programming ecosystem, consisting of classic tools like Prolog, constraint solvers, Answer-Set Programming and SAT/SMT systems, as well as machine learning techniques implemented on top of inductive and probabilistic logic programming [7].

One of the research directions we have worked on in the past, that turned out to be remarkably successful, is graph-based NLP. It has originated in a Prolog program that was using WordNet's semantic links to improve word-sense disambiguation (WSD) [8], by building a graph connecting words, sentences with their semantic equivalence classes (synsets) and then guessing the most likely sense associated to a word, by running the PageRank algorithm [9] on the graph. A few months later, an unsupervised version of the algorithm [10] based on graphs connecting sentences to word occurrences has been shown to extract high quality summaries and keywords. It later became one of the most popular techniques for the task [11], implemented in virtually all widely used programming languages and several NLP libraries.

While recently revisiting the topic, it became clear that involving logic programming tools is likely to enhance graph based NLP. The use of logic-based implementations of Combinatorial Categorial Grammars (CCGs) looks especially appealing as it provides lexicalized representations easier to correlate with words and word phrases. The NLU-component coming from extracting logic representations is likely to make the links in the graph structure more meaningful.

A simple reducer for a subset of CCG rules looks as follows:

```
:-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).
```

Interestingly, Prolog's 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.
```

When executed it accepts a sentence as follows:

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

More elaborate parsers can be built using CYK or A^* parsers and tools like tabling in Prolog provide the means to do that efficiently. Interestingly, the CCG parsing problem can also be nicely expressed and executed directly with Answer Set Programming tools as shown in [12].

Tools like the `boxer` program [13] can, in combination with a statistically trained CCG parser like [14], build Prolog clauses describing semantically labeled first-order formulas representing input sentences, ready to be further explored with standard and probabilistic logic programming algorithms as well as graph based algorithms exploiting the richer link structure between their underlying concepts.

An emerging field in NLP these days is sentiment analysis, as knowing what the opinion of an author is about a topic is as important and knowing what a document is about. Modalities and negation detection provided by a logic component combining syntactic and semantic parsing can improve sentiment analysis. Figuring out the implicit entailment links important for understanding a story line or the rhetorical structures involved in an argument is also likely to benefit from logic representations.

Another NLU-minded application we have worked more than a decade or ago is the use of Prolog-based natural language-enabled agents [15]. They interacted with the Prolog version of WordNet and Google's metasearch API to bring in knowledge distributed over the internet. The integration of logic inferences and a Prolog representation (as dynamic clauses or backtrackable assumptions [4]) of the agents' short-term memory, have significantly enhanced the quality of the dialog, with shared virtual worlds and interactive story telling systems developed on top of them [16, 17]. These days, fields like interactive story telling have become an integral part of computer games (e.g., Minecraft Story Mode) and voice-enabled software agents are part of major mobile phone (e.g., Siri, Cortana, Google Ok) and platforms are making their way in home automation systems (e.g., Alexa) and more generally in the upcoming Internet-of-Things (IOT) platforms.

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

Finally this brings us to the possible synergies between logic programming and deep-learning neural network technologies [18], credited for the new "AI-Spring", brought by successful applications to popular fields like vehicle automation, vision and internet search. Tools like Google's TensorFlow and word2vec [19] are specifically focused on enabling extensions transforming quantitatively represented meaning fragments into more human-friendly logic representations, ready for inference steps that reveal implicit connections between facts and events. Integrating logic programming components with this family of tools, possibly involving the quantitative means provided by probabilistic logic programming opens the door for being part of this the re-emergence of AI-based techniques in new application domains.

Of special practical interest are logical formalisms based on lexicalized natural language representations (e.g., CCGs) that are likely to enable interaction at word level between symbolic and connectionist representations. As the same lexicalized representations can also enable synergies with graph-based methods in natural language processing [11], we expect a significant practical impact from logic programming tools bringing together these NLP fields.

References

[1] R.A.K. Kowalski. *Logic for Problem Solving.* North-Holland, 1979.

[2] A. Colmerauer. *Metamorphosis Grammars*, volume 63, pages 133–189. Springer-Verlag, 1978.

[3] F.C.N. Pereira and D.H.D. Warren. Definite clause grammars for language analysis - a survey of the formalism and a comparison with transition networks. *Artificial Intelligence*, 13:231–278, 1980.

[4] Veronica Dahl, Paul Tarau, and Renwei Li. Assumption Grammars for Processing Natural Language. In Lee Naish, editor, *Proceedings of the Fourteenth International Conference on Logic Programming*, pages 256–270, MIT press, 1997.

[5] David Scott Warren. Using Lambda-Calculus to Represent Meanings in Logic Grammars. In *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics*, ACL '83, pages 51–56, Stroudsburg, PA, USA, 1983. Association for Computational Linguistics.

[6] V. Dahl. Natural language processing and logic programming. *Journal of Logic Programming*, 12(1):681–714, 1994.

[7] Daan Fierens, Guy Van den Broeck, Joris Renkens, Dimitar Shterionov, Bernd Gutmann, Ingo Thon, Gerda Janssens, and Luc De Raedt. Inference and learning in probabilistic logic programs using weighted Boolean formulas. *Theory and Practice of Logic Programming*, 2013.

[8] Rada Mihalcea, Paul Tarau, and Elizabeth Figa. PageRank on Semantic Networks, with application to Word Sense Disambiguation. In *Proceedings of The 20st International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland, August 2004.

[9] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[10] Rada Mihalcea and Paul Tarau. TextRank: Bringing Order into Texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, Barcelona, Spain, July 2004.

[11] Rada F. Mihalcea and Dragomir R. Radev. *Graph-based Natural Language Processing and Information Retrieval.* Cambridge University Press, New York, NY, USA, 1st edition, 2011.

[12] Yuliya Lierler and Peter Schüller. Parsing combinatory categorial grammar via planning in answer set programming. In *Correct Reasoning*, pages 436–453. Springer, 2012.

[13] Johan Bos. Open-domain semantic parsing with boxer. In Beáta Megyesi, editor, *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015, May 11-13, 2015, Institute of the Lithuanian Language, Vilnius, Lithuania*, pages 301–304. Linköping University Electronic Press / ACL, 2015.

[14] James R. Curran, Stephen Clark, and Johan Bos. Linguistically motivated large-scale NLP with c&c and boxer. In John A. Carroll, Antal van den Bosch, and Annie Zaenen, editors, *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic.* The Association for Computational Linguistics, 2007.

[15] Paul Tarau, Koen De Bosschere, Veronica Dahl, and Stephen Rochefort. LogiMOO: an Extensible Multi-User Virtual World with Natural Language Control. *Journal of Logic Programming*, 38(3):331–353, March 1999.

[16] Elizabeth Figa and Paul Tarau. The VISTA Project: An Agent Architecture for Virtual Interactive Storytelling. In N. Braun and U. Spierling, editors, *TIDSE'2003*, Darmstadt, Germany, March 2003.

[17] Elizabeth Figa and Paul Tarau. Story Traces and Projections: Exploring the Patterns of Storytelling. In N. Braun and U. Spierling, editors, *TIDSE'2003*, Darmstadt, Germany, March 2003.

[18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[19] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1188–1196, 2014.