Künstliche Intelligenz manuscript No.
(will be inserted by the editor)

Answer Set Programming in Linguistics

Peter Schüller

Abstract This survey collects scientific works where Answer Set Programming, a declarative knowledge representation and reasoning formalism, is applied to Natural Language Processing and Computational Linguistics.

1 Introduction

Ever since the invention of computing machines, scientists have worked on processing natural language with computers. This task has been tightly associated with artificial intelligence [51].

Answer Set Programming (ASP) [39] is a logic programming formalism for knowledge representation and reasoning. We here survey projects where ASP has been applied to Linguistics. These applications can be split into two areas: (a) *Natural Language Processing*, which is about processing text and its content, and (b) *Computational Linguistics*, which is about language properties and relationships between languages.

ASP, different from classical logic, permits *nonmonotonic* reasoning: the absence of beliefs can be used to make inferences, and the addition of beliefs can prevent inferences. This allows the usage of ASP for defining default inferences that can be blocked once more information becomes available. In analogy, ambiguous words (e.g., 'bank') permit several interpretations, and encountering additional words (e.g., 'money') can rule out certain interpretations. Nonmonotonic reasoning , the possibility to represent linguistic ambiguities as nondeterministic guesses, and the existence of efficient evaluation tools (i.e., solvers) that support search with respect to hard constraints and optimization with respect to weak constraints, make ASP attractive for Linguistics applications.

Peter Schüller
Institute für Logic and Computation, Technische Universität Wien
Favoritenstraße 9–11, 1040 Vienna, Austria
E-mail: ps@kr.tuwien.ac.at

2 ASP in Natural Language Processing

Text is also called *unstructured information* because its content is easily accessible to humans, but it is not formally structured so that it would be easily accessible to machines. Natural Language Processing [33] often has the primary aim to annotate a given text with its structure in order to make the content accessible to computers. Structural analysis can be done on several levels, here we will separate *Syntax*, which concerns the sequence of words and phrases in a sentence and their structures, *Semantics*, which concerns the meaning of sentences, and *Pragmatics*, which is about interpreting language in spatial, cultural, and other kinds of context.

2.1 Syntactic Parsing

Text is often analyzed using a *Grammar*, which is a set of rules that characterize those sequences of words that are sentences in the language under consideration. *Parsing* is the task of finding one or all structures for a given sentence, while *Membership* of a sentence in a language can sometimes be decided without finding an explicit structure [21].

Depending on the expressivity of rules in the grammar, classes of languages of varying descriptive power have been identified, for example Regular and Context Free languages [21]. While it is known that human language is more expressive than Context Free languages [28], for many applications it is sufficient to use less expressive formalisms.

In the area of syntactic parsing, Drescher and Walsh [15] used ASP to decide language membership for Context Free and Regular Languages [21], based on a representation of the CYK algorithm [21]. Lierler and Schüller [31, 41] performed parsing with ASP for the Combinatory Categorial Grammar [47] formalism, also using a CYK representation.

Inductive Logic Programming (ILP) [36] has been used to learn grammars from examples, usually based on Prolog. Muggleton et al. [37] used ASP as an alternative to Prolog for grammar learning with ILP, and found that ASP performs better than Prolog in cases that have small instantiation but a large search space, while Prolog performs better in the reverse case. Kazmi and Schüller [26] applied ASP-based ILP to chunking (a shallow form of parsing), and improved the scalability of the XHAIL [38] approach for ASP-based ILP to make this application feasible.

2.2 Semantic Parsing

Semantic parsing creates *representations* of text in a mathematical formalism that allows for *reasoning* about the content of the parsed text.

Baral et al. [4, 5, 52] created the NL2KR[1] system which translates Combinatory Categorial Grammar parse trees into an ASP representation using Lambda Calculus terms that have been learned from corpora. Importantly, the NL2KR GUI allows non-expert users to provide feedback to the system, and the accuracy of NL2KR improves by learning from this feedback.

Related with semantic parsing is the area of knowledge *mining* where the aim is to extract a specific type of semantic content from text in a shallow analysis, i.e., without representing the whole meaning of the text at hand. Tari et al. [48] applied ASP to the automatic discovery of drug-drug interactions in bioinformatics. Their approach is based on retrieving candidate interactions from parse trees of publication abstracts followed by extending and cleaning the set of candidates using ASP reasoning. Liu et al. [32] used ASP to realize syntax-based extraction of aspects in opinion mining. They emphasize the flexibility and efficiency of ASP for this task. (Examples for aspects of, e.g., a cellphone model, are weight and camera quality.) A further shallow semantic analysis based on ASP is the work of Kazmi and Schüller [25] about the similarity of sentences.

2.3 Pragmatics and Natural Language Understanding

To *understand* natural language text, it is necessary to go beyond semantics and to *interpret* the text to discover the *intention* of the writer. This field is called Pragmatics [22]. Two relevant benchmark tasks are Recognizing Textual Entailment (RTE) [14] which is about deciding whether a given text is a consequence of another given text, and the Winograd Schema (WS) Challenge [29] which is about resolving a pronoun (or adjective) using common sense knowledge.

Lierler and Lifschitz [30] compared First-Order Logic and ASP as representations for RTE, in particular they used the tool boxer [9] which creates a semantic representation of text according to Discourse Representation Theory [24]. They conclude that FOL and ASP have unique strengths for RTE: First-Order Logic allows more complex formulas in representations, while ASP provides nonmonotonic reasoning. In another work about interpreting natural language, Schüller [43] realized the abductive language interpretation framework of Hobbs et al. [20] in ASP.

Apart from the above approaches for interpreting natural language, the Winograd Schema Challenge has received special attention in several works based on ASP. Schüller [42] proposed an approach based on graph matching and relevance theory. Bailey et al. [1] proposed the *correlation calculus* where correlations of belief about logical statements help to resolve WS. The most general approach was proposed by Sharma et al. [45], who created a system that finds, retrieves, and parses text snippets that seem relevant for resolving a given WS. They extract knowledge from these snippets and then use it in a graph matching step that determines how to resolve the pronoun. Importantly, this approach does not require manually encoded knowledge.

2.4 Human Computer Interaction

Several applications of ASP in Linguistics include more than analyzing natural language input: engaging in a dialog with a user or providing natural language responses to queries.

Question Answering (QA) [53] aims to exceed classical keyword search and answer natural language questions about textual content. QA usually involves a representation for text and queries, and reasoning over both representations. The relationship between QA and knowledge representation/reasoning is described by Balduccini et al. [3] and includes several pointers to projects based on ASP. Baral et al. [7] perform QA in the travel domain using ASP for word sense disambiguation and for counterfactual temporal reasoning. They apply ASP together with Link Grammar [46] and WordNet [34]. Todorova et al. [49] answer natural language questions about simple travel stories in the presence of incomplete information. Mitra and Baral [35] used the XHAIL [38] ILP system for learning ASP rules that answer questions about natural language stories in the Facebook bAbI dataset [54] with a much higher accuracy than machine learning approaches.

ASP has been used in large-scale AI projects in the areas of *education* and *robotics* which also include natural language processing. Chaudhri et al. [11] describe Inquire Biology, a project where an ASP representation of a biology textbook is used to answer natural language questions of users with natural language explanations, and to generate natural language questions about the book for educational purposes. Within Inquire, ASP is used for query answering [12] and for ontological reasoning with defaults and inheritance in a frame-based representation [6, 13]. Khandelwal et al. [27] describe the BWIBots robot platform which combines ASP and probabilistic reasoning for natural language interaction with users.

Controlled Natural Language (CNL) avoids ambiguity and missing lexicon entries by defining fragments of natural language with a unique syntactic and semantic interpretation. CNL usually provides parsing (input) as well as generation

[1] `http://nl2kr.engineering.asu.edu/`

(output) within the language fragment. Fuchs et al. [18] describe the Attempto Controlled English (ACE) CNL and the AceRules tool for interactive reasoning based on a transformation from ACE to ASP and vice versa. Erdem and Oztok [17] created a formalism, tool, and CNL for querying biomedical databases and for explaining the results of these queries based on ASP. Guy and Schwitter [19] created the PENG^{ASP} CNL that produces ASP programs as output. They also provide an authoring tool that predicts sentence continuations and allows for extending the lexicon.

3 ASP in Computational Linguistics

ASP has also been applied for reasoning about properties of language(s) and for reasoning about linguistic theories.

Brooks et al. [10] have used ASP to compute trees of historical relationships between languages, based on evolutionary changes of words. Erdem et al. [16] extended these trees to graphs that can also represent direct exchange of lexicon entries between languages. Scherl et al. [40] use ASP to extract information about relationships between conversation participants. Inclezan [23] represented a second language acquisition theory in ASP and described a tool for designing second language teaching material. Schüller [44] created a tool for consolidating inconsistent coreference annotations from human annotators based on ASP. Baumgartner and Burchardt [8] describe how the FrameNET [2] knowledge base and reasoning about frames can be represented in ASP. Toivanen et al. [50] used ASP for analyzing and generating Finnish poems.

4 Conclusion

ASP has several properties that make it interesting for Linguistics applications: reasoning about defaults, actions, and change; explainable and decidable reasoning over recursive structures; and the existence of hard as well as soft constraints. These make ASP a powerful formalism for reasoning about natural language. ASP has been extended in various ways to incorporate properties such as probabilistic reasoning which is useful for dealing with the variability of human language.

Acknowledgements This is a pre-print of an article published in KI - Künstliche Intelligenz. The final authenticated version is available online at: `http://dx.doi.org/10.1007/s13218-018-0542-z` .

References

1. Bailey, D., Harrison, A., Lierler, Y., Lifschitz, V., Michael, J.: The Winograd Schema Challenge and Reasoning about Correlation. In: CommonSense, pp. 17–24 (2015)
2. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet Project. In: ACL/COLING, pp. 86–90 (1998)
3. Balduccini, M., Baral, C., Lierler, Y.: Knowledge Representation and Question Answering. In: Handbook of Knowledge Representation, pp. 779–819. Elsevier (2008)
4. Baral, C., Dzifcak, J., Gonzalez, M.A., Gottesman, A.: Typed answer set programming lambda calculus theories and correctness of inverse lambda algorithms with respect to them. Theory and Practice of Logic Programming **12**(4-5), 775–791 (2012)
5. Baral, C., Dzifcak, J., Gonzalez, M.A., Zhou, J.: Using Inverse Lambda and Generalization to Translate English to Formal Languages. In: IWCS, pp. 35–44 (2011)
6. Baral, C., Liang, S.: From knowledge represented in frame-based languages to declarative representation and reasoning via ASP. In: KR, pp. 413–423 (2012)
7. Baral, C., Tari, L.: Using AnsProlog with Link Grammar and WordNet for QA with deep reasoning. In: ICIT, pp. 125–128 (2006)
8. Baumgartner, P., Burchardt, A.: Logic programming infrastructure for inferences on FrameNet. In: JELIA, pp. 591–603 (2004)
9. Bos, J.: Wide-coverage semantic analysis with Boxer. In: Conference on Semantics in Text Processing, pp. 277–286 (2008)
10. Brooks, D., Erdem, E., Erdoğan, S., Minett, J., Ringe, D.: Inferring Phylogenetic Trees Using Answer Set Programming. J Autom Reasoning **39**, 471–511 (2007)
11. Chaudhri, V.K., Cheng, B.H., Overholtzer, A., Roschelle, J., Spaulding, A., Clark, P., Greaves, M., Gunning, D.: Inquire Biology: A Textbook that Answers Questions. AI Magazine **34**(3), 55–72 (2013)
12. Chaudhri, V.K., Heymans, S., Wessel, M., Son, T.C.: Object-Oriented Knowledge Bases in Logic Programming. In: ICLP, TC (2013)
13. Chaudhri, V.K., Son, T.C.: Specifying and reasoning with underspecified knowledge bases using answer set programming. In: KR, pp. 424–434 (2012)
14. Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment challenge. In: Machine Learning Challenges, pp. 177–190. Springer (2006)
15. Drescher, C., Walsh, T.: Modelling grammar constraints with answer set programming. In: ICLP, TC, *LIPIcs*, vol. 11, pp. 28–39 (2011)
16. Erdem, E., Lifschitz, V., Ringe, D.: Temporal Phylogenetic Networks and Logic Programming. Theory and Practice of Logic Programming **6**(5), 539–558 (2006)
17. Erdem, E., Oztok, U.: Generating explanations for biomedical queries. Theory and Practice of Logic Programming **15**(1), 35–78 (2015)
18. Fuchs, N.E., Kaljurand, K., Kuhn, T.: Attempto Controlled English for Knowledge Representation. In: Reasoning Web, pp. 104–124 (2008)

19. Guy, S.C., Schwitter, R.: The PENG ASP system: architecture, language and authoring tool. Language Resources and Evaluation **51**(1), 67–92 (2017)
20. Hobbs, J., Stickel, M., Martin, P., Edwards, D.: Interpretation as abduction. Artif Intel **63**(1-2), 69–142 (1993)
21. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Pearson (2000)
22. Horn, L.R., Ward, G.: The Handbook of Pragmatics. Blackwell (2006)
23. Inclezan, D.: An application of ASP to the field of second language acquisition. In: LPNMR, pp. 395–400 (2013)
24. Kamp, H., Reyle, U.: From Discourse to Logic: Introduction to Model Theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory. Kluwer (1993)
25. Kazmi, M., Schüller, P.: Inspire at SemEval-2016 Task 2: Interpretable Semantic Textual Similarity Alignment based on Answer Set Programming. In: SemEval, pp. 1109–1115 (2016)
26. Kazmi, M., Schüller, P., Saygın, Y.: Improving Scalability of Inductive Logic Programming via Pruning and Best-Effort Optimisation. Expert Systems with Applications **87**, 291–303 (2017)
27. Khandelwal, P., Zhang, S., Sinapov, J., Leonetti, M., Thomason, J., Yang, F., Gori, I., Svetlik, M., Khante, P., Lifschitz, V., Aggarwal, J., Mooney, R., Stone, P.: BWIBots: A platform for bridging the gap between AI and human-robot interaction research. Int J Rob Res **36**(5–7), 635–659 (2017)
28. Kracht, M.: The Mathematics of Language. De Gruyter Mouton (2003)
29. Levesque, H.J., Davis, E., Morgenstern, L.: The Winograd Schema Challenge. In: KR, pp. 552–561 (2012)
30. Lierler, Y., Lifschitz, V.: Logic Programs vs. First-Order Formulas in Textual Inference. In: IWCS, pp. 340–346 (2013)
31. Lierler, Y., Schüller, P.: Parsing Combinatory Categorial Grammar via planning in Answer Set Programming. In: Correct Reasoning, pp. 436–453. Springer (2012)
32. Liu, Q., Gao, Z., Liu, B., Zhang, Y.: A logic programming approach to aspect extraction in opinion mining. In: Intl Conf Web Intelligence, pp. 276–283 (2013)
33. Martin, J.H., Jurafsky, D.: Speech and Language Processing. Pearson (2000)
34. Miller, G.A.: WordNet: a lexical database for English. Commun ACM **38**(11), 39–41 (1995)
35. Mitra, A., Baral, C.: Addressing a Question Answering Challenge by Combining Statistical Methods with Inductive Rule Learning and Reasoning. In: AAAI, pp. 2779–2785 (2016)
36. Muggleton, S., De Raedt, L., Poole, D., Bratko, I., Flach, P., Inoue, K., Srinivasan, A.: ILP turns 20: Biography and future challenges. Machine Learning **86**(1), 3–23 (2012)
37. Muggleton, S.H., Lin, D., Pahlavi, N., Tamaddoni-Nezhad, A.: Meta-interpretive learning: application to grammatical inference. Machine Learning **94**(1), 25–49 (2014)
38. Ray, O.: Nonmonotonic abductive inductive learning. Journal of Applied Logic **7**, 329–340 (2009)
39. Schaub, T., Woltran, S.: Answer set programming unleashed! Künstliche Intelligenz, this issue (2018)
40. Scherl, R., Inclezan, D., Gelfond, M.: Automated inference of socio-cultural information from natural language conversations. In: SocialCom/PASSAT, pp. 480–487 (2010)
41. Schüller, P.: Flexible Combinatory Categorial Grammar Parsing using the CYK Algorithm and Answer Set Programming. In: LPNMR, pp. 499–511 (2013)
42. Schüller, P.: Tackling Winograd schemas by formalizing relevance theory in knowledge graphs. In: KR, pp. 358–367 (2014)
43. Schüller, P.: Modeling Variations of First-Order Horn Abduction in Answer Set Programming. Fundamenta Informaticae **149**, 159–207 (2016)
44. Schüller, P.: Adjudication of coreference annotations via Answer Set optimization. Journal of Experimental & Theoretical Artificial Intelligence (2018). (**To appear**) arXiv:1802.00033
45. Sharma, A., Vo, N.H., Aditya, S., Baral, C.: Towards addressing the Winograd Schema Challenge - Building and using a semantic parser and a knowledge hunting module. In: IJCAI, pp. 1319–1325 (2015)
46. Sleator, D., Temperley, D.: Parsing English with a Link Grammar. In: International Workshop on Parsing Technologies (1993). arXiv:cmp-lg/9508004
47. Steedman, M.: The syntactic process. MIT Press (2000)
48. Tari, L., Anwar, S., Liang, S., Cai, J., Baral, C.: Discovering drug - drug interactions: a text-mining and reasoning approach based on properties of drug metabolism. Bioinformatics **26**, i547–i553 (2010)
49. Todorova, Y., Gelfond, M.: Toward question answering in travel domains. In: Correct Reasoning, pp. 311–326. Springer (2012)
50. Toivanen, J.M., Järvisalo, M., Toivonen, H.: Harnessing Constraint Programming for Poetry Composition. Intl Conf Computational Creativity pp. 160–167 (2013)
51. Turing, A.: Computing machinery and intelligence. Mind **59**, 433–460 (1950)
52. Vo, N.H., Mitra, A., Baral, C.: The NL2KR platform for building natural language translation systems. In: ACL, pp. 899–908 (2015)
53. Voorhees, E.M.: The TREC-8 Question Answering Track Report. In: TREC, pp. 77–82 (1999)
54. Weston, J., Bordes, A., Chopra, S., Mikolov, T.: Towards AI-complete question answering: A set of prerequisite toy tasks (2015). arXiv:1502.05698