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

Answer Set Programming applied to Coreference Resolution and Semantic Similarity

Peter Schüller

Received: date / Accepted: date

Abstract We describe two research projects about solving problems in Computational Linguistics using Answer Set Programming, and we conclude with several lessons learned from these projects.

Answer Set Programming (ASP) [14] is a knowledge representation approach for solving combinatorial problems using logic programming with rules.

OmSieve is a project about Coreference Resolution (CR). CR includes (i) finding those phrases in texts that refer to an entity in the world, e.g., 'he' or 'the sun'; and (ii) grouping these phrases according to the entity they refer to. In OmSieve we applied ASP to the following two tasks.

Automatic consolidation of coreference annotations. We created a Turkish CR corpus [17] by collecting CR annotations from volunteers. Independent annotations of the same document were often inconsistent. Therefore, we collected a large number of annotations with the goal of automatically reconciling them. We used ASP to implement the CaspR[1] tool for combining CR annotations [16] into a single solution that (i) corresponds optimally with the majority of annotators and (ii) obeys linguistic structural constraints. We plan to integrate CaspR into a CR GUI such as MMAX2 [12] or BART [18] for making it accessible for NLP researchers.

P. Schüller
Marmara University, Turkey
Technische Universität Wien, Austria
E-mail: schueller.p@gmail.com

[1] `https://github.com/knowlp/caspr-coreference-tool`

Coreference Resolution Sieve in ASP. The Stanford Sieve [10] predicts CR annotations by performing a sequence of CR decisions that are controlled by rules implemented in Java. We realized these rules and the sequential decision making framework in ASP in order to weaken the sequentiality of the Sieve approach: we wanted conflicts that become apparent in late stages of the decision sequence to be able to undo wrong decisions that were made early in the sequence. In particular we aimed to apply semantic constraints from our previous work [11]. Representing the Sieve in ASP was challenging because many details of the Sieve were only accessible in Java source code. Moreover, the ASP encoding of a single document contained many auxiliary atoms. To fit into a reasonable amount of memory, we reformulated the encoding multiple times. This made reasoning feasible but the encoding became less intuitive. This effort concluded in a MSc thesis, however we did not reach our original goal of increasing the accuracy of the Sieve using semantic knowledge resources.

Inspire is a project where we focus on interpretation of the meaning of given natural language texts.

Interpretation of Language as Abduction. Weighted Abduction [7] is an approach for interpreting language using abduction and backward-chaining. To enable integration of additional semantic reasoning we formalized this approach as a graph optimization problem and represented it in ASP [15].[2] To achieve computational feasibility, we used the Python APIs of the Clingo [6] and WASP [2] ASP systems to generate certain constraints

[2] `https://bitbucket.com/knowlp/asp-fo-abduction`

lazily during combinatorial optimization. Recently, we used this application in an empirical comparison of ASP lazy evaluation methods for constraints [5].

Interpretable Semantic Similarity (iSTS). The iSTS competition [1] requires splitting sentences into phrases (chunking) and creating labeled links between chunks of sentences according to how their meaning is related (e.g., 'puppy' is more specific than 'dog'). We used ASP to implement the existing imperative Java rules of the NeRoSim system [4]. This provided flexibility for adjusting the rules to three subject domains used in the competition. Our system [8] obtained the third place in the competition and was the only one that was not primarily based on machine learning. We also implemented *phrase chunking in ASP* by learning chunking rules using Inductive Logic Programming (ILP). We experimented with several ILP systems and found XHAIL [13] to be the most robust one, however it was not scalable beyond a handful of sentences. Therefore we improved XHAIL: (i) we added a pruning parameter which adjusts the trade-off between the number of examples that can be processed and the level of detail of learning [9], and (ii) we limited solver time and integrated mixed optimization algorithms [3] which provide lower and upper optimality bounds.

The improved XHAIL system is publicly available.[3] The resulting chunker provides an interpretable compact set of rules for chunking, with a performance that is a bit below the state-of-the-art chunkers.

Lessons learned. From these projects, we conclude that (i) there is a need for tools that assist the optimization of ASP encodings, (ii) integrating ASP project results into software of other communities is necessary for making an interdisciplinary impact with ASP, and (iii) there is a lot of potential in learning ASP rules from data, which yields reasonable prediction accuracy and has the benefit of providing explanations for decisions.

Acknowledgements I am grateful to my collaborators and to my students. OmSieve and Inspire have been supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under grants 114E430 and 114E777.
This is a pre-print of an article published in KI - Künstliche Intelligenz. The final authenticated version is available online at: `https://dx.doi.org/10.1007/s13218-018-0539-7` .

References

1. Agirre, E., Gonzalez-Agirre, A., Lopez-Gazpio, I., Maritxalar, M., Rigau, G., Uria, L.: SemEval-2016 Task 2: Interpretable Semantic Textual Similarity. In: SemEval, pp. 512–524 (2016)
2. Alviano, M., Dodaro, C., Faber, W., Leone, N., Ricca, F.: WASP: A native ASP solver based on constraint learning. In: LPNMR, pp. 54–66 (2013)
3. Alviano, M., Dodaro, C., Marques-Silva, J., Ricca, F.: Optimum stable model search: algorithms and implementation. J Logic Comput (2015)
4. Banjade, R., Niraula, N.B., Maharjan, N., Rus, V., Stefanescu, D., Lintean, M., Gautam, D.: NeRoSim: A System for Measuring and Interpreting Semantic Textual Similarity. In: SemEval, pp. 164–171 (2015)
5. Cuteri, B., Dodaro, C., Ricca, F., Schüller, P.: Constraints, Lazy Constraints, or Propagators in ASP Solving: An Empirical Analysis. Theor Pract Log Prog **17**, 780–799 (2017)
6. Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., Wanko, P.: Theory Solving made easy with Clingo 5. In: ICLP TC. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016)
7. Hobbs, J.R., Stickel, M., Martin, P., Edwards, D.: Interpretation as abduction. Artif Intell **63**(1-2), 69–142 (1993)
8. 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)
9. Kazmi, M., Schüller, P., Saygn, Y.: Improving Scalability of Inductive Logic Programming via Pruning and Best-Effort Optimisation. Expert Syst Appl **87**, 291–303 (2017)
10. Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., Dan Jurafsky: Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules. Comput Linguist **39**(4), 885–916 (2013)
11. Lierler, Y., Schüller, P.: Towards a Tight Integration of Syntactic Parsing with Semantic Disambiguation by means of Declarative Programming. In: IWCS, pp. 383–389 (2013)
12. Müller, C., Strube, M.: Multi-level annotation of linguistic data with MMAX2. In: Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods, pp. 197–214. Peter Lang (2006)
13. Ray, O.: Nonmonotonic abductive inductive learning. J Appl Logic **7**, 329–340 (2009)
14. Schaub, T., Woltran, S.: Answer set programming unleashed! Künstliche Intelligenz. (2018). This issue.
15. Schüller, P.: Modeling Variations of First-Order Horn Abduction in Answer Set Programming. Fundam Inform **149**, 159–207 (2016)
16. Schüller, P.: Adjudication of Coreference Annotations via Answer Set Optimization. J Exp Theor Artif Intell (2018). In press, arXiv:1802.00033
17. Schüller, P., Cingilli, K., Tunçer, F., Sürmeli, B.G., Pekel, A., Karatay, A.H., Karakas, H.E.: Marmara Turkish Coreference Corpus and Coreference Resolution Baseline. Tech. rep., Marmara University (2017). arXiv:1706.01863
18. Versley, Y., Ponzetto, S.P., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yang, X., Moschitti, A.: BART: A Modular Toolkit for Coreference Resolution. In: ACL, pp. 9–12 (2008)

[3] `https://github.com/knowlp/XHAIL`