RDF* Graph Database as Interlingua for the TextWorld Challenge

Guntis Barzdins
IMCS
University of Latvia
Riga, Latvia
guntis.barzdins@lumii.lv

Didzis Gosko
IMCS
University of Latvia
Riga, Latvia
didzis.gosko@lumii.lv

Paulis F. Barzdins
IMCS
University of Latvia
Riga, Latvia
paulis.barzdins@lumii.lv

Uldis Lavrinovics
Innovation Labs
LETA
Riga, Latvia
uldis.lavrinovics@leta.lv

Gints Bernans
Innovation Labs
LETA
Riga, Latvia
gints.bernans@leta.lv

Edgars Celms
IMCS
University of Latvia
Riga, Latvia
edgars.celms@lumii.lv

***Abstract*—This paper briefly describes the top-scoring submission to the First TextWorld Problems: A Reinforcement and Language Learning Challenge. To alleviate the partial observability problem, characteristic to the TextWorld games, we split the Agent into two independent components: Observer and Actor, communicating only via the Interlingua of the RDF* graph database. The RDF* graph database serves as the "world model" memory incrementally updated by the Observer via FrameNet informed Natural Language Understanding techniques and is used by the Actor for the efficient exploration and planning of the game Action sequences. We find that the deep-learning approach works best for the Observer component while the Actor policy is better served by backtracking over the set of rules.**

Keywords—reinforcement learning, partial observability, deep learning.

I. INTRODUCTION

The goal of the CoG-2019 First TextWorld Problems: A Reinforcement and Language Learning Challenge (*http://aka.ms/textworld-challenge*) is to build an AI agent that can efficiently play and win simplified text-based games. The agent must navigate and interact within a text environment, i.e. the agent perceives the environment through text and acts in it using text commands (Fig.2). The purpose of the Challenge is to highlight the limitations of existing Natural Language Understanding models when combined with Reinforcement Learning.

In this competition, all the games share a similar theme (cooking in a modern house), similar text commands, and similar entities (i.e. interactable objects within the games). The simplified games are generated using TextWorld [1], an open-source framework that both generates and interfaces with text-based games via OpenAI Gym API. Our submission is currently ranked 1st with 7105.0 points compared to 5258.2 points for the 2nd place (the competition is still open till June 30, 2019, and max possible score is 9940.0). Our code will be made available at *https://github.com/didzis/textworld* upon completion of the competition.

Since the core difficulty within the TextWorld games is the partial observability of the game-space via limited natural language dialogue, our approach focuses on extracting, interpreting and memorizing all bits of information provided by the game dialogue. Instead of training an end-to-end Agent we split the Agent into two independent components: Observer and Actor, communicating only via the Interlingua of the RDF* graph database [2]. RDF* is an extension of the Resource Description Framework (RDF) with statement (edge) level metadata (e.g. in Fig.1 the door leading south from Driveway is annotated with metadata "closed").

Our solution is illustrated in Fig.1. The RDF* graph database serves as the "world model" memory, incrementally built by the task-agnostic Observer via universal FrameNet informed Natural Language Understanding technique [3]. RDF* Interlingua is sufficiently complete to serve as the only input to the Actor for the efficient exploration and planning of the game Action sequences.

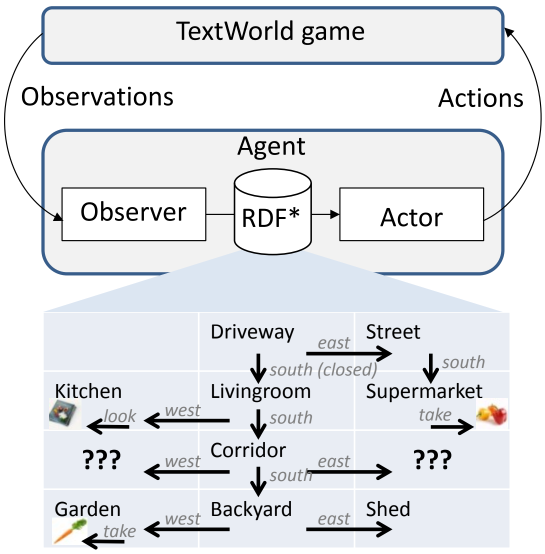

Fig. 1. Reinforcement Learning Agent with visualised incomplete "world model" of interconnected rooms and items in them; model is stored in RDF* Interlingua which splits the Observer and Actor components.

We find that the deep-learning approach works best for the Observer component (Section II) while the optimal Actor policy is backtracking over the set of rules (Section III).

II. OBSERVER

The key principle for the Observer is to extract from the game text only the actionable information (features) as TextWorld games also provide rich, but useless comments. Such filtering is made possible by the FrameNet [5]

978-1-7281-1884-0/19/$31.00 ©2019 IEEE

approach. FrameNet parser extracts from the text only the frames relevant to the particular application [7] such as *Self-motion, Residence, Possession* with corresponding frame elements like *Self-mover, Direction, Theme* etc.

Legacy FrameNet parsers [6] merely identify the sentence fragments corresponding to the Frame target and its frame elements, regardless of the surface forms (e.g. noun or pronoun) and syntactic constructs used to convey them. This is insufficient for gameplay, as we need to precisely identify the referenced objects to build a "world model" for further use by the Actor in planning and generating the action sequences. Therefore we train our own FrameNet like parser[1] from the gameplay traces illustrated in Fig.2 using seq2seq neural AMR parsing technique described in [8] which is also applicable to the FrameNet parsing.

Fig. 2. Example of the TextWorld gameplay trace.

Examples of TextWorld specific frames extracted from the gameplay trace in Fig. 2 are *„refrigerator in kitchen", „pantry in kitchen", „counter in kitchen", „red_pepper on counter", „cookbook on counter", „knife on counter", „stove in kitchen", „kitchen door south + closed"*. The last frame is an example of a n-ary relation requiring the RDF* graph database, as it links three frame elements: *kitchen, south, closed*.

It shall be noted that Observer does not perform any exploration on its own – it is a completely passive component which merely observes the game feedback, while the Actor component performs exploration, planning and decision making to issue actions.

III. ACTOR

For the Actor, we experimented with Deep Reinforcement learning techniques [4] for finding the shortest path in the grid-world, but abandoned them because they cannot handle Prolog-style depth-first backtracking which is essential for efficient first-person game room-search – a key task in the TextWorld games. To our knowledge there are no DNN techniques capable of exploration via backtracking (a promising future research topic).

The TextWorld competition envisioned handicapping for skipping some backtracking tasks like finding the Cookbook and obtaining the Recipe, but this incurred uncompetitive penalties. Instead, we settled on the rule-based Actor implementing the Objective from Fig.2 elaborated manually (ideally such elaboration had to be learned) into the sequence of tasks:

1. Find kitchen (room-search with backtracking),
2. Look up Recipe in the Cookbook,
3. Drop all inventory in the kitchen
4. Find missing ingredients and knife (room-search with backtracking) and bring them to the kitchen,
5. Cut the ingredients with knife,
6. For grilling find BBQ (room-search with backtracking),
7. For roasting and frying use oven and stove in the kitchen,
8. Prepare and eat the meal.

The Actor accomplishes above tasks by observing the environment solely via the RDF* graph database. Whenever the Actor issues an action changing the environment state, the Observer understands the game feedback, acknowledging the successful or unsuccessful action, and changes the RDF* database accordingly.

IV. CONCLUSION

Using the RDF* graph database as an Interlingua for Natural Language Understanding is a promising approach not only in the TextWorld games but also in other domains like News article understanding and visualization [7]. It also clearly identifies the lack of depth-first backtracking support as the key deficiency of the current deep reinforcement learning approaches.

ACKNOWLEDGMENT

The research was supported by ERDF project 1.1.1.1/18/A/045 at IMCS, University of Latvia. The research leading to these results has also received funding from the research project "Competence Centre of Information and Communication Technologies" of EU Structural funds, contract No. 1.2.1.1/18/A/003 signed between IT Competence Centre and Central Finance and Contracting Agency, Research No. 2.4 "Platform for the semantically structured information extraction from the massive Latvian news archive".

REFERENCES

[1] M.A. Côté, A.K. Xingdi, Y.B. Kybartas, T. Barnes, E. Fine, J. Moore, M. Hausknecht, L.E. Asri, M. Adada, W. Tay, A Trischler. "TextWorld: A Learning Environment for Text-based Games. Computer Games", Workshop at ICML/IJCAI, 2018.

[2] O. Hartig. "Reconciliation of RDF⋆ and Property Graphs". arXiv:1409.3288v2, 2014.

[3] N. Gruzitis and G. Barzdins.. "Polysemy in controlled natural language texts". In Proceedings of the 2009 conference on Controlled natural language (CNL'09), Norbert E. Fuchs (Ed.). Springer-Verlag, Berlin, Heidelberg, 102-120. 2009.

[4] A. Tamar, Y. Wu, G. Thomas, S. Levine, and P. Abbeel. "Value iteration networks". In Advances in Neural Information Processing Systems, pp. 2154-2162. 2016.

[5] C.J. Fillmore, C.R. Johnson, and M.R.L. Petruck. "Background to FrameNet". International Journal of Lexicography, 16, pp.235-250. 2003.

[6] D. Das, D. Chen, A.F.T. Martins, N. Schneider, N.A. Smith."Frame-Semantic Parsing", Computational Linguistics, 40(1), pp.9--56. 2014.

[7] G. Barzdins, D. Gosko, L. Rituma, and P. Paikens. "Using C5.0 and Exhaustive Search for Boosting Frame-Semantic Parsing Accuracy". In: Proceedings of the 9th Language Resources and Evaluation Conference (LREC), pp. 4476-4482. 2014.

[8] G.Barzdins, D. Gosko. "Riga: Impact of Smatch Extensions and Character-Level Neural Translation on AMR Parsing Accuracy". In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), pp. 1143–1147. 2016.

[1] Our current submission uses a hand-crafted Frame feature extractor developed by the fellow competitor Justin D. Harris