
Playing by the Book: Towards Agent-based Narrative Understanding through Role-playing and Simulation

Ronen Tamari*

The Hebrew University of Jerusalem
ronent@cs.huji.ac.il

Hiroyuki Shindo

Nara Institute of Technology / RIKEN-AIP
shindo@is.naist.jp

Dafna Shahaf

The Hebrew University of Jerusalem
dshahaf@cs.huji.ac.il

Yuji Matsumoto

Nara Institute of Technology / RIKEN-AIP
matsu@is.naist.jp

Abstract

Understanding procedural text requires tracking entities, actions and effects as the narrative unfolds (often implicitly). We focus on the challenging real-world problem of structured narrative extraction in the *materials science* domain, where language is highly specialized and suitable annotated data is not publicly available. We propose an approach, Text2Quest, where procedural text is interpreted as instructions for an interactive game. A reinforcement-learning agent completes the game by understanding and executing the procedure correctly, in a text-based simulated lab environment. The framework is intended to be more broadly applicable to other domain-specific and data-scarce settings. We conclude with a discussion of challenges and interesting potential extensions enabled by the agent-based perspective.

1 Introduction

Understanding procedural text such as instructions and recipes is increasingly important, with potential applications ranging from robots, conversational agents and AI assistants to domain-specific extraction of structured knowledge from raw text. It requires the ability to track entities, relations between them, and the changes in their states as they unfold throughout the text. For humans, this common-sense reasoning task is trivial and often sub-conscious, yet it remains a highly challenging task for today's natural language understanding (NLU) algorithms (e.g., [Lucy and Gauthier \[2017\]](#), [Levy et al. \[2017\]](#), [Khot et al. \[2018\]](#)). This ability is also closely related to causal reasoning and language grounding ([Mooney \[2008\]](#)), the ability to connect words and phrases to perception of objects and events in the world. As such, it is likely a key feature of intelligent agents in general.

In procedural text understanding, the objective is typically to take a textual passage describing a sequence of events and extract an *action graph* from it. The action graph represents the core entities, actions and relations in a structured form (see Fig. 1). The growing interest in this problem has seen many recent works, with *text-centric* language modeling focused on syntax and surface labels increasingly complemented by *world-centric* modeling which focuses on underlying entities, actions and state changes ([Bosselut et al. \[2018\]](#)). Most efforts along this latter line of research utilize densely annotated datasets to train deep learning models, which can limit their applicability, particularly for domain-specific settings with technical language and complex semantics. Furthermore, the degree to which language understanding can be achieved without grounding via interaction in real/simulated environments remains an open question ([Choi \[2018\]](#)).

*Work was begun while author was an intern at the RIKEN Center for Advanced Intelligence Project and continued at HUJI.

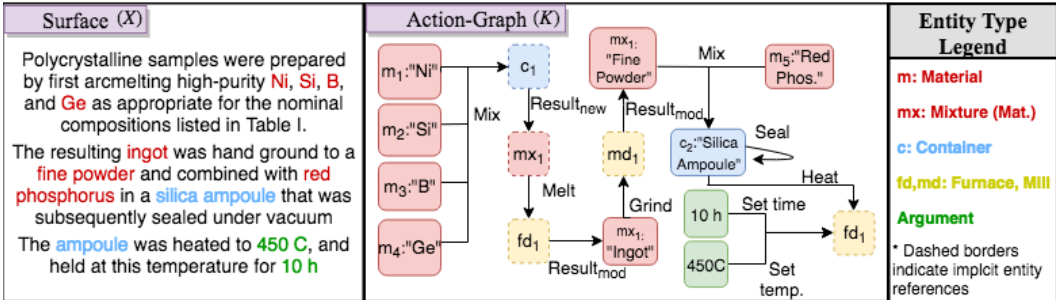


Figure 1: Sample surface text (left) and possible corresponding action-graph (right) for typical material synthesis procedure. Nodes are entities, edges are transition rules affecting them.

Another related line of work focusing on understanding language through simulation is based on Sutton [1990], combining reinforcement learning with world models simulating user interaction (Peng et al. [2018], Su et al. [2018]) for conversational agents. However, the application of these approaches to our task is not straightforward (see Section 2).

In this work we focus on a challenging problem of **action graph extraction from materials-science papers**. These papers often include natural-language descriptions of material synthesis procedures (see example in Figure 1). Extracting action graphs of the synthesis procedures can help in the design and discovery of novel materials (see Mysore et al. [2017]). In this setting, as in many real-world scenarios, no annotated datasets are publicly available.

We propose to frame the problem of procedural text understanding as a text-based reinforcement-learning (text-RL) task. Intuitively, we would like the text to be interpreted by the RL-agent as **instructions for an interactive game** (or “quest”). The agent must complete the game by understanding and executing the instructions correctly. Correct execution would directly yield the desired action graph. To enable this, we propose a domain-specific world-model simulator which generates synthetic quests as well as the appropriate text-based training environment for the agent.

The contributions of this work are as follows: We propose a reformulation of the problem of procedural text comprehension as a text-based reinforcement learning task coupled with an underlying text-based world model simulator for language grounding. We focus on a real-world problem of material synthesis analysis, but the approach is intended to be more generally applicable. We propose a novel approach to data curation and annotation, through use of a **knowledge engine** coupled with **structured text generation** methods. This method is particularly suited for domain-specific settings where manual dense annotation is scarce and costly, and is intended to improve “return of investment” on annotation, both in terms of performance and scalability. We discuss limitations, potential applications and interesting extensions of the framework suggested by the agent-based perspective.

We have developed prototype modules for basic game generation, which are publicly available (<https://github.com/ronentk/TextLabs>) to encourage further research in this direction. While there is a lot more work to be done, structured natural language generation (NLG) and text-RL games are both attracting significant and growing interest, and we expect advances in these areas to directly benefit future versions of the system.

2 Related Work

Action-graph extraction from procedural instructions: Early methods use entity/action segmentation followed by probabilistic connection models (Kidddon et al. [2015], Mysore et al. [2017]). Notably, recent work (Feng et al. [2018]) has applied deep-RL to the problem of extracting action-graphs without assuming a pre-known set of actions. Closer to our approach are Bosselut et al. [2018], Tandon et al. [2018], Das et al. [2018], all of which track entities and world state changes. Bosselut et al. [2018] attempts to predict these using a kind of textual simulation, but without actually using a simulation environment. Das et al. [2018] have recently shown that common-sense knowledge can be learned end-to-end from the data. However, these methods all attempt to learn directly from fine-grained annotated data, and therefore are of limited applicability to tasks where dense annotation is costly and language could be highly technical and complex.

A recent work (Yang et al. [2017]) addresses the task of action-graph extraction using text-based games and gamification to obtain higher quality data, but still yields annotations linear in the amount of annotators, and is limited to short sequences.

Dialogue Agents: This setting bears similarities to action-graph extraction, and has been predominantly approached through RL. Notably, recent work extending the Dyna-Q (DQ) framework (Sutton [1990]) has incorporated a world-model based planning element, which is used to simulate the environment and user experiences for augmentation of training (Peng et al. [2018], Su et al. [2018]). The DQ success is encouraging for our modeling paradigm, as the general framework is similar in spirit to our approach. However, they assume an abundance of annotated data and their texts are interactive, having few long-term dependencies, making them more amenable to step-by-step processing. Their texts are also relatively short and non-technical, making text generation significantly simpler.

Language Grounding and text-RL: Recent years have seen an increased interest in utilizing RL for understanding text. Specifically, text-based games are being used as a sandbox environment to study language grounding and understanding (Zahavy et al. [2018], Narasimhan [2017], Côté et al. [2018]). To date, these works have not applied grounded text-based RL directly to real world problems.

Narrative Analysis: A long researched problem in the field of AI, a full survey is out of scope of this work. Learning-based methods for computational narrative analysis originated in computational literature studies such as Chambers and Jurafsky [2008], Elson et al. [2010].

3 Problem Formulation

While the motivation for this work is action-graph extraction, we frame the problem in more general terms to include any text with a narrative structure.

Entities, Relations & Rules ($\mathcal{E}, \mathcal{R}, \Lambda$): Assume two vocabularies, of *entities* $\mathcal{E} = \{e_1, \dots, e_N\}$ and *relations* $\mathcal{R} = \{r_1, \dots, r_K\}$. A *fact* f is a grounded predicate of the form $f = (h, r, t)$, $h, t \in \mathcal{E}$, $r \in \mathcal{R}$ (self loops allowed for single-entity facts). We define the set of valid world-states S , where a state $s \in S$ is a set of facts, and validity is decided by a world-model Λ defined using linear logic. Λ is comprised of production rules (or transition rules) over entities and relations governing which new facts can be produced from a given state. In our domain, entities may be devices, materials or arguments such as temperature and pressure. Relations may include properties such as *closed* and *sealed*, or *in(material, container)*. A production rule is a state-affecting action or event such as *grind(material, mill)*, which changes the fact *solid(material)* into *powder(material)*.

Narrative Sketch (K): Intuitively, a *narrative* conveys a consistent chronology of states and the rules which brought about the transitions. Formally, let C be a sequence of valid world-states $\{s_i\}_{i=0}^n$ and the transition rules $\{\lambda_i\}_{i=0}^n$ between them (applying λ_i to s_i results in s_{i+1}): $C = (s_0, \lambda_0, s_1, \lambda_1, \dots, \lambda_{n-1}, s_n)$. Narratives typically focus on salient entities and relations and how they change over time. A *narrative-sketch* K as a sub-sequence of C (where a state s'_i in K may also be a subset of the corresponding state s_i in C). The focus of a sketch is generally on key transitions, with many intermediate states, facts and transitions left implicit. In the specific setting of action-graph extraction, a natural compact representation of K is in graph form, where entities are nodes and edges are the transition rules affecting them. Henceforth we will refer to K as such a graph.

Surface (X): A *surface* is simply a text in natural language with an underlying narrative (roughly a paragraph in our domain). The surface typically follows a roughly topological ordering of K induced by the underlying dependencies. Clearly, multiple surfaces may exist for one narrative sketch.

Learning Task: Our objective is to learn a mapping $\Psi : X \rightarrow K$. X can be seen as a noisy, possibly lossy observation of K ², encoded and “transmitted” by the narrator and “observed” by the audience. We frame the problem as a reinforcement learning task. For an input surface X we learn a game G and its initial state s_0 . The solution of G is the required narrative sketch K . The underlying game can be modelled as a partially observable Markov Decision Process (POMDP) $G = (S, A, T, \Omega, O, R, \gamma)$.

We refer the reader to Côté et al. [2018] for a detailed exposition, and focus here on mapping the RL setting to our setting: S are states, A are actions, T are conditional state transition probabilities derived from production rules Λ . Ω are observations, and O are conditional observations probabilities. The environment is partially observable, with the agent observing the surface X at time $t = 0$.

² K may also be seen as a noisy compression of some ground-truth C , but in the scope of this work for simplicity we assume K itself to be the ground truth.

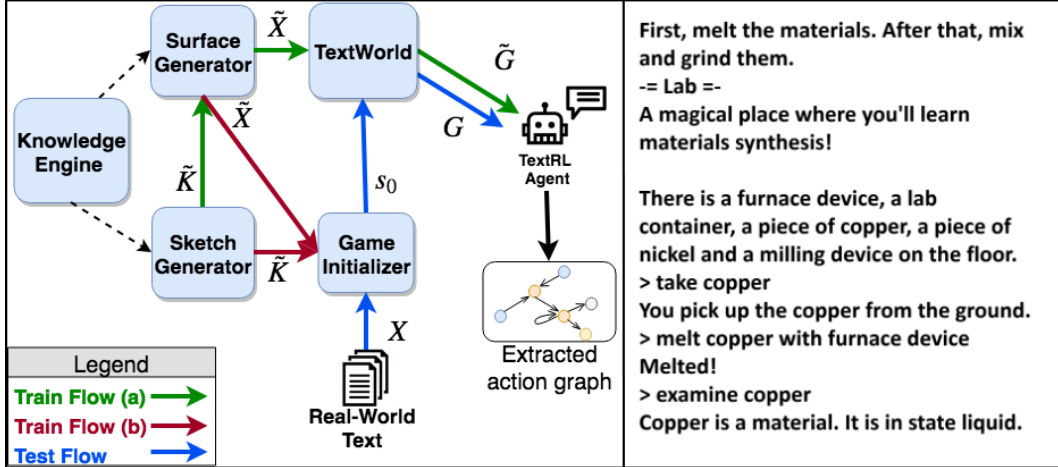


Figure 2: Left: Proposed solution architecture of Text2Quest. (a) Flow for training RL agent on simulated games. (b) Flow for training Game Initializer to extract an initial game state from a real-world surface. Right: Excerpt from an actual “material synthesis quest” generated by our system.

Additional textual descriptive information can easily be incorporated (see Section 6). $R : S \times A \rightarrow \mathbb{R}$ is the reward function. The reward can be derived from a sketch K . In addition to reward upon reaching the final state, there could be intermediate rewards. $\gamma \in [0, 1]$ is the discount factor.

4 Text2Quest: Proposed Solution Architecture

We now describe our solution architecture. Our system consists of six core modules (Figure 2): A knowledge base containing the entity, action and rule vocabularies. This is used by the Surface Generator (NLG) module and Sketch Generator module to generate corresponding pairs (\tilde{X}, \tilde{K}) of synthetic surfaces and their underlying action graphs. These are used during **train time** for two tasks: (a) Simulated games \tilde{G} for training the RL-agent to extract \tilde{K} from \tilde{X} ; (b) Training a Game Initializer module, which extracts an initial game state s_0 from a surface (similar to an entity/relation recognition task). This is used at **test time** to convert a real-world surface to a game instance G for the text-RL agent. Finally, the TextWorld environment (Côté et al. [2018]) is used to run the games and handle interaction with the agent.

Our framework is similar in spirit to the DQ framework (Sutton [1990]). For the sake of exposition, we briefly point out the parallels. The knowledge base and environment correspond to the DQ world model, the simulated synthesis quests correspond to the planning phase, with evaluation being determined by the generated sketch which defines a winning policy. The text-RL agent handles the narrative comprehension policy. In DQ, training consists of alternating real and simulated samples. Our case is different in that real surfaces are not directly learned from, but first must be converted to simulated world instances using the learned Game Initializer module.

We have implemented simple prototypes for our domain-specific Knowledge Base and Sketch Generator. This involved extending the TextWorld knowledge base and underlying game interpreter (written in the Inform7 language) to build a text-based “lab simulator”. In this environment, the player can perform basic common actions on materials such as mixing and melting. For quest generation, we used simple forward chaining and heuristic search strategies to create plausible quests (for example, mixing all materials together). Combining these with a simple rule-based Surface Generator already allows for creating small training game instances (see excerpt in Fig. 2, right). Such games could become part of a **curriculum** towards more difficult material synthesis quests. Future implementations will focus on more realistic text generation and a suitable text-RL agent for actual sketch extraction, from progressively harder surfaces.

5 Discussion

Extracting action graphs from long and complex surfaces is a challenging proposition, requiring understanding implicit references and actions and remembering instructions over multiple steps. We believe that RL can achieve success in this area, particularly through architectures designed for large, partially observable state/action spaces, and training procedures such as curriculum learning (Zahavy et al. [2018], Yang et al. [2017], Mitchell et al. [2018]).

A critical open question in our framework is whether the surface generator will be able to generate surfaces representative enough to allow for generalization to real examples. Current NLG systems are increasingly capable of structured text generation (Trisedya et al. [2018], Cao et al. [2017], Kiddon et al. [2016], Mei et al. [2015]), and though current works produce relatively short surfaces, we believe that **coupling them with the generated sketches** is a promising approach to scaling up to longer sequences while maintaining coherence. Such systems also require supervision in the form of fact-set/text pairs. Fortunately, this is a rather simple annotation task that can be automated to a large degree through weakly-supervised NER/relation-extraction techniques. Paraphrase generation methods could also prove useful, as well as learning discriminators to filter poorly generated surfaces (Su et al. [2018]).

Notwithstanding the challenges, we believe the potential return on investment is worth the effort. Standard crowd-sourced annotation approaches suffer from poor economies of scale (Roth [2017]), where the amount of annotations is linear in the amount of annotators. Furthermore, while traditional wisdom has it that annotation is easier than hand-crafting knowledge bases, we note that in this setting, the gap between the two may be decreasing. In Bosselut et al. [2018] for example, the fine-grained annotation schema is detailed enough to make codifying it into a knowledge base a relatively straightforward task. We believe that in many cases, the core semantics of certain domains may be compactly represented in knowledge bases, and grown organically as more detail is required.

6 Conclusions and Future Directions

Recent research suggests the need for complementing text-centric modeling with world-centric understanding by simulation, and a growing sentiment is that a similar need exists in traditional crowd-sourced data curation methods, which suffer from poor scalability. Motivated by these, we have proposed Text2Quest, a novel approach for understanding of natural language procedural instructions using text-based reinforcement learning coupled with a simulator combining symbolic knowledge and learned structured text generation. This enables complementing word-level surface annotations with “knowledge annotation”, which can potentially enable better performance and greater scalability.

We believe that beyond a baseline implementation, the proposed approach opens up multiple interesting new directions for further research. First, many components in our framework can be learned from data. Currently, **vocabularies for entities, relations and rules** \mathcal{E} , \mathcal{R} , Λ are manually created. This could be seen as either an online or primarily unsupervised preliminary step, as in Kim et al. [2017]. Recent work Feng et al. [2018] has explored the possibility of using reinforcement learning to directly identify actions from text. Similarly, we can **learn the environment**. Underlying the current TextWorld environment is the Inform7 interpreter, which has extensive but hard-coded NLU capabilities for understanding the agent commands. This could be enhanced with a learned environment dynamics simulator as in Peng et al. [2018] to take into account language and environment variability.

Deeper reading comprehension capabilities require a combination of multiple “reasoning modes”, including mental simulation for entailment, rote memorization, knowledge fusion from external sources, mathematics, visual perception (e.g., for figures) and more. Recent works show a variety of architectures to cope with the variety of capabilities needed (Luo et al. [2018], Yang and Mitchell [2017], See et al. [2017], Sun et al. [2018]). An agent-based perspective can naturally fit these capabilities under one learning model. We propose endowing the agent with internal **cognitive actions**. Text-based games already offer an *‘examine <entity>’* command. We propose more sophisticated actions, such as external knowledge lookup, memory augmentation (a *memorize* command), or *look* to trigger a visual understanding procedure.

Our framework is intended to be easy to adapt to different domains. A particularly interesting application is **domain-specific neural machine translation**. This is a highly challenging setting as data is scarce, precision can be crucial, and semantics are domain-specialized (Chu and Wang [2018]). Utilizing the underlying graph representation as a kind of intermediate “universal representation” of

entities and relations, multilingual Surface Generators could facilitate structured, domain-specific machine translation.

Finally, we note that both the text-RL problem and the simulator can be gamified. Yang et al. [2017] demonstrated that gamification can be effective in obtaining higher quality data for language grounding.

In conclusion, we hope that the proposed framework will lead to the development of useful systems for real-world narrative understanding, as well as other reading comprehension tasks.

References

- Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. Simulating action dynamics with neural process networks. *Proceedings of the 6th International Conference for Learning Representations*, 2018.
- Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. Faithful to the original: Fact aware neural abstractive summarization. *CoRR*, abs/1711.04434, 2017.
- Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative event chains. *Proceedings of the Association of Computational Linguistics*, 31(14):789–797, 2008. doi: 10.1.1.143.1555.
- Yejin Choi. Why nlu doesn’t generalize to nlg. URL: <https://newgeneralization.github.io/slides/YejinChoi.pdf>, 2018.
- Chenhui Chu and Rui Wang. A survey of domain adaptation for neural machine translation. In *COLING*, 2018.
- Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben A. Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew J. Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. Textworld: A learning environment for text-based games. *CoRR*, abs/1806.11532, 2018.
- Rajarshi Das, Tsendsuren Munkhdalai, Xingdi Yuan, Adam Trischler, and Andrew McCallum. Building Dynamic Knowledge Graphs from Text using Machine Reading Comprehension. pages 1–12, 2018.
- David K Elson, Nicholas Dames, and Kathleen R McKeown. Extracting social networks from literary fiction. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, (July):138–147, 2010.
- Wenfeng Feng, Hankz Hankui Zhuo, and Subbarao Kambhampati. Extracting Action Sequences from Texts Based on Deep Reinforcement Learning. 2018. doi: arXiv:1803.02632v2.
- Tushar Khot, Ashish Sabharwal, and Peter Clark. SCITAIL: A Textual Entailment Dataset from Science Question Answering. *AAAI*, 2018.
- Chloé Kiddon, Ganesa Thandavam Ponnuraj, Luke S. Zettlemoyer, and Yejin Choi. Mise en place: Unsupervised interpretation of instructional recipes. In *EMNLP*, 2015.
- Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. Globally Coherent Text Generation with Neural Checklist Models. *{Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing}*, pages 329–339, 2016. doi: 10.18653/v1/D16-1032.
- Edward Kim, Kevin Huang, Alex Tomala, Sara Matthews, Emma Strubell, Adam Saunders, Andrew McCallum, and Elsa Olivetti. Machine-learned and codified synthesis parameters of oxide materials. *Scientific Data*, 4:170127, 2017. ISSN 2052-4463. doi: 10.1038/sdata.2017.127.
- Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-Shot Relation Extraction via Reading Comprehension. 2017. doi: 10.18653/v1/K17-1034.
- Li Lucy and Jon Gauthier. Are distributional representations ready for the real world? Evaluating word vectors for grounded perceptual meaning. 2017. ISSN 07644469. doi: 10.1016/S0764-4469(00)01284-1.

- Fuli Luo, Tianyu Liu, Zexue He, Qiaolin Xia, Zhifang Sui, and Baobao Chang. Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1402–1411, 2018.
- Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. What to talk about and how? Selective Generation using LSTMs with Coarse-to-Fine Alignment. pages 720–730, 2015. doi: 10.18653/v1/N16-1086.
- Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha P. Talukdar, Bo Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matt Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil A. Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. Never-ending learning. *Commun. ACM*, 61(5):103–115, 2018. doi: 10.1145/3191513.
- Raymond J. Mooney. Learning to connect language and perception. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 1598–1601, 2008.
- Sheshera Mysore, Edward Kim, Emma Strubell, Ao Liu, Haw-Shiuan Chang, Srikrishna Kompella, Kevin Huang, Andrew McCallum, and Elsa Olivetti. Automatically Extracting Action Graphs from Materials Science Synthesis Procedures. nov 2017.
- Karthik Narasimhan. *Grounding natural language with autonomous interaction*. PhD thesis, Massachusetts Institute of Technology, Cambridge, USA, 2017.
- Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, Kam-Fai Wong, and Shang-Yu Su. Deep Dyna-Q: Integrating Planning for Task-Completion Dialogue Policy Learning. 2018.
- Dan Roth. Incidental Supervision: Moving beyond Supervised Learning. *Aaai*, 2017. ISSN 1367-4811. doi: 10.1093/bioinformatics/btq538.
- Abigail See, Peter J. Liu, and Christopher D. Manning. Get To The Point: Summarization with Pointer-Generator Networks. 2017. ISSN 1496-8975. doi: 10.1007/s12630-011-9639-7.
- Shang-Yu Su, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Yun-Nung Chen. Discriminative Deep Dyna-Q: Robust Planning for Dialogue Policy Learning. 2018.
- Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W. Cohen. Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text. 2018.
- Richard S. Sutton. Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming. In *Machine Learning Proceedings 1990*. 1990. ISBN 1558601414. doi: 10.1016/B978-1-55860-141-3.50030-4.
- Niket Tandon, Bhavana Dalvi Mishra, Joel Grus, Wen-tau Yih, Antoine Bosselut, and Peter Clark. Reasoning about Actions and State Changes by Injecting Commonsense Knowledge. (2), 2018.
- Bayu Distiawan Trisedya, Jianzhong Qi, Rui Zhang, and Wei Wang. GTR-LSTM: A Triple Encoder for Sentence Generation from RDF Data. *Proceedings of ACL*, pages 1627–1637, 2018.
- Bishan Yang and Tom Mitchell. Leveraging Knowledge Bases in LSTMs for Improving Machine Reading. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1436–1446, 2017. doi: 10.18653/v1/P17-1132.
- Zhilin Yang, Saizheng Zhang, Jack Urbanek, Will Feng, Alexander H. Miller, Arthur Szlam, Douwe Kiela, and Jason Weston. Mastering the Dungeon: Grounded Language Learning by Mechanical Turker Descent. pages 1–16, 2017.
- Tom Zahavy, Matan Haroush, Nadav Merlis, Daniel J. Mankowitz, and Shie Mannor. Learn what not to learn: Action elimination with deep reinforcement learning. *CoRR*, abs/1809.02121, 2018.