Understanding Text Using Agent Based Models

Adrian Mladenic Grobelnik
Artificial Intelligence Laboratory
Jozef Stefan Institute
Ljubljana Slovenia
adrian.m.grobelnik@ijs.si

Marko Grobelnik
Artificial Intelligence Laboratory
Jozef Stefan Institute
Ljubljana Slovenia
marko.grobelnik@ijs.si

Dunja Mladenic
Artificial Intelligence Laboratory
Jozef Stefan Institute
Ljubljana Slovenia
dunja.mladenic@ijs.si

ABSTRACT

The paper proposes a novel approach to text understanding and text generation focusing on short stories. The proposed approach attempts to understand and generate stories by creating an explainable, agent-based world model of the story. The world model is defined through agents, their goals, actions, attributes and relationships between them. We demonstrate our approach on the story of 'Little Red Riding Hood', simulating it as a sequence of 48 actions, involving 7 main agents and 14 goals.

KEYWORDS

Text understanding, agent-based approach, world model, agent-based model

1 Introduction

With recent advancements in deep learning and overall increases in computing power, artificial intelligence systems are now able to make commonsense inferences from simple events, as proposed in research such as COMET [1] and MultiCOMET [2]. While the aforementioned commonsense inferences can be made with a high degree of precision, they lack an explainable and comprehensive structure capable of storing and predicting future events with such inferences. Agent-based models (ABMs), while capable of simulating complex interactions between agents, rarely focus on understanding stories in greater depth. Moreover, they cannot perform commonsense reasoning on agent's goals, actions or attributes. In our research, we draw from existing work on ABMs to create a system capable of understanding short text-based stories, with the potential to incorporate commonsense inferences in the future.

Related work such as 'Automated Storytelling via Causal, Commonsense Plot Ordering' [3] and 'Modeling Protagonist Emotions for Emotion-Aware Storytelling' [4] makes use of COMET to tackle automated story plot generation. As the stories are generated using COMET's commonsense causal inferences, they lack explainability. In our work, we focus on generating explainable stories.

Other related work [5] focuses on story understanding using manually supplied commonsense rules, concept patterns and story text. Our system aims to understand and simulate a story, given the story text, goals and initial attributes of its agents.

The main contributions of this paper are (1) a novel approach to explainable story understanding, (2) a system generating stories given a set of agents with attributes and goals, and (3) implementation of the proposed approach, with publicly available source code [7] allowing users to create and analyze their own stories.

The rest of this paper is organized as follows: Section 2 provides a problem description. Section 3 describes the approach used to tackle the problem. Section 4 demonstrates the functioning of our approach. The paper concludes with discussion and directions for future work in Section 5.

2 Problem Description

The problem we are solving is, given the text of a short story, convert it into a machine understandable and actionable description representing the dynamics of the story being told. Such an actionable description should encode the implicit knowledge assumed by the text in the form of an agent-based world model.

The world model should include enough representational power to fully represent the story. This includes agents, their environment and the relationships between them. The world model should be actionable enough to simulate the dynamics of an input story with all the key elements, and relevant details mentioned in the input text.

As the world model can represent a story given its text, it should also be able to represent and simulate other stories within the world model's constraints.

Some of the key operations the resulting system should support:

1. representation of the story
2. simulation of the story's dynamics
3. question answering about explicit and implicit elements written or assumed within the story
4. creating alternative stories, given their context

3 Approach Description

The general aim of our approach is to provide deep text understanding of the input story. Not all the steps are automatable at this stage. In particular, the biggest challenge is to automatically translate the story text into the knowledge based representation aligned with the world model. We are looking forward to eventually automate all of the steps in the approach.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

Information Society 2021, 4-8 October 2021, Ljubljana, Slovenia

© 2021 Copyright held by the owner/author(s).

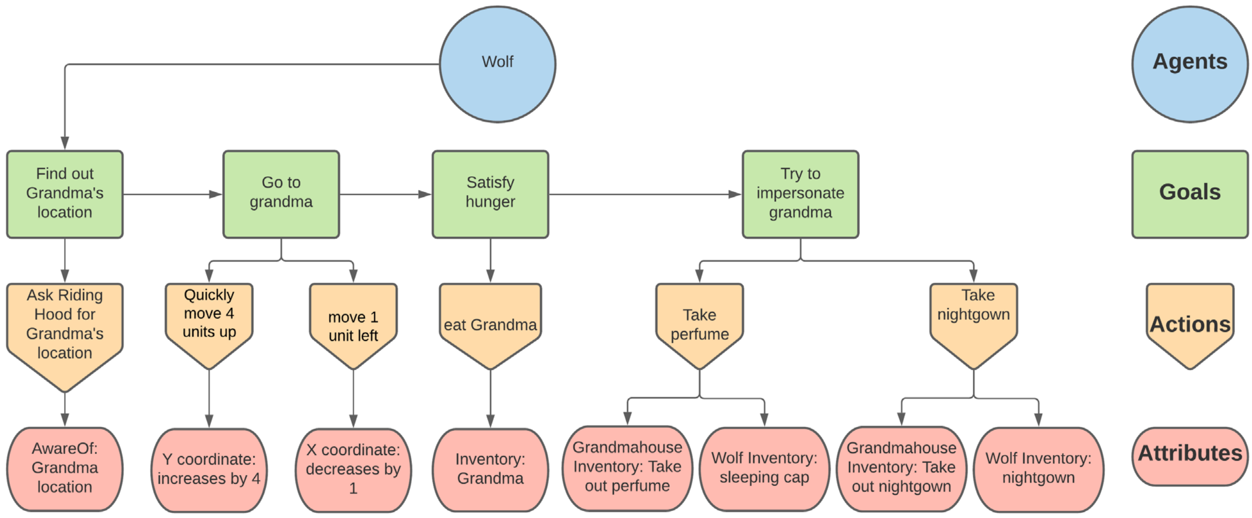

Figure 1: A partial representation of the Wolf agent's goals, actions and attributes.

As a running example of the input story, we selected the popular children's story 'Little Red Riding Hood' [6]. In the first stage, we restructured the original story into 73 simplified sentences where we identified 23 key events involving 7 main agents:

1. Mother
2. Riding Hood
3. Flower Field
4. Butterfly
5. Wolf
6. Grandma
7. Woodsman

Each agent is represented by its goals, actions and attributes (see Figure 1 for an example involving the Wolf). All goals cause actions and all actions change at least one agent's attributes.

As depicted on Figure 2, an agent's goal is defined by a goal state (a set of agents with specific attribute values) and 'pre-goals' (goals that must be completed and act as preconditions for an agent to start working towards the goal).

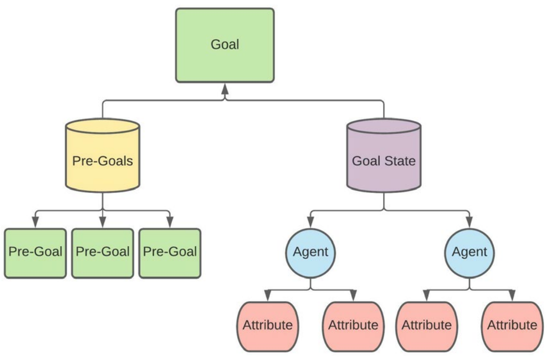

Figure 2: An example representation of a goal

To define actions, we use an action schema proposed as part of 'UCPOP: A Sound, Complete, Partial Order Planner for ADL' [8] where each action consists of a set of parameters, preconditions and effects. We show two example action representations in Figure 3 and Figure 4. The duration of each action corresponds to the passing of one time unit.

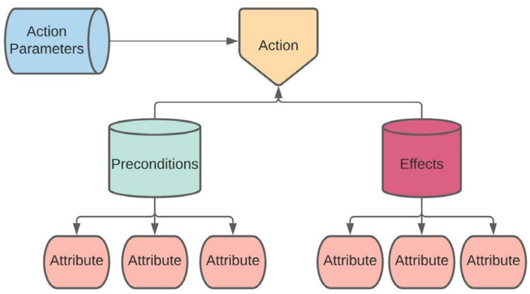

Figure 3: An example representation of an action

```
      action: Eat (?monster, ?victim, ?location)
precondition: knows(?monster, ?victim),
              alive(?monster), alive(?victim),
              ¬eaten( ?victim),¬full(?monster),
              at(?monster, ?location),
              at(?victim, ?location),
              ?monster ≠ ?victim
      effect: eaten(?victim)
              in(?victim, ?monster), full(?monster),
              ¬at(?victim, ?location)
```

Figure 4: An example pseudocode representation of a concrete action, taken from [9]

An attribute is simply defined as any information relating to the agent. For instance, the agent's location, inventory of items and awareness of other agents.

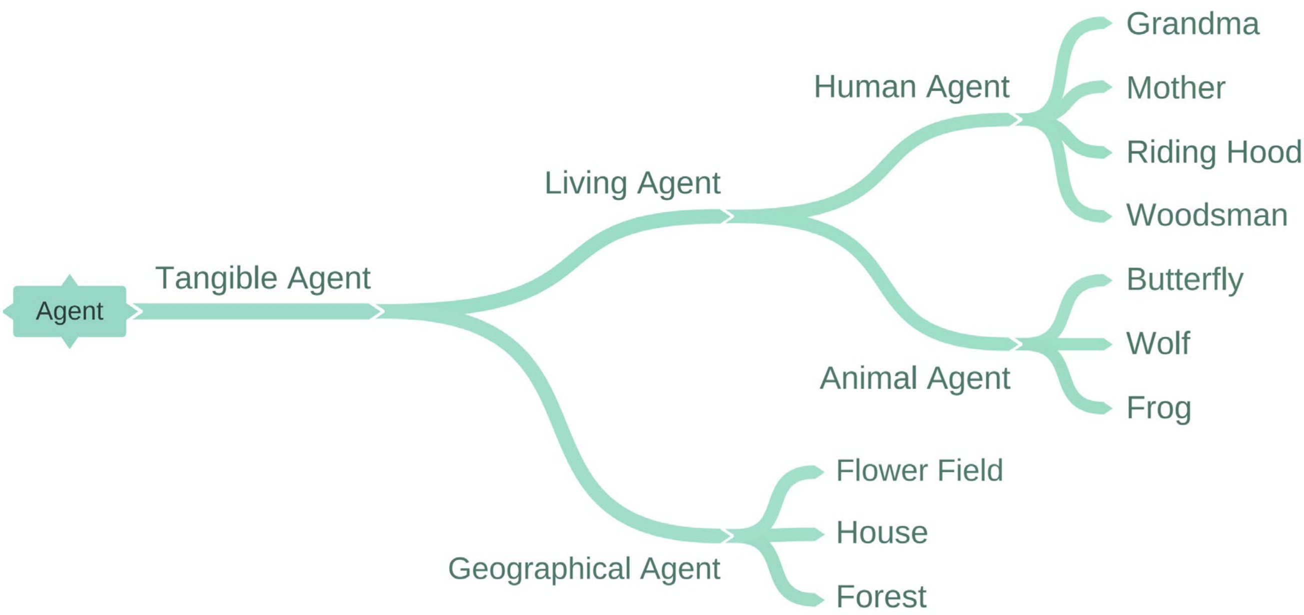

Figure 5: Hierarchy of agents for the Little Red Riding Hood story

The agents are defined through a hierarchy, ensuring consistency across agent goals, actions, attributes and providing a clear overview of the agent types as observed in Figure 5.

Throughout the story simulation of ‘Little Red Riding Hood’ 3 key agents jointly had 14 goals, causing them to perform a total of 48 actions composed of 12 unique action types.

We propose a simple textual description of each performed action, stating why the agent executed the action and which other agents were involved. See Figure 8 for an example.

At the highest conceptual level, we randomly select an agent and simulate all of its possible next actions. We then select the action that brings the agent closest to all it’s currently active goals, and execute this action. We repeat this until there are no more agents with active goals in our world model, as depicted in Figure 6.

```
Until All Active Goals Not Complete
  For Random Agent
    For All Possible Actions
      Simulate Action
      If Action brings Agent closest to its Goals
        Set this as new Best Action
    Execute the Best Action
```

Figure 6: High level pseudocode of the simulation within the world model

4 Approach Demonstration

```
*******************|grandmahouse,grandma|************forest13|******forest14,wolf
************forest1|*******************|************forest5|***********forest12
****forest4,woodsman|*******************|************forest8|***********forest10
************forest3|*******************|************forest7|***********forest11
************forest2|*******************|***forest6,butterfly|************forest9
******flower field 1|******flower field 2|******flower field 3|*******************
*******************|*******************|*******************|*******************
*******************|*******************|*******************|*******************
*******************|******rhhouse,mother|*******************|*********riding hood
```

Figure 7: Initial state of the agents’ locations within the world model; each X, Y slot includes a list of agents at that location

We first initialize the world model to an initial setting similar to that of ‘Little Red Riding Hood’, illustrated in Figure 7. For instance, agents ‘forest4’ and ‘woodsman’ are in the same location, 1 unit above agent ‘forest3’. The model is initialized with the agents, their initial attributes with values and their goals in the story. Once initialized, we can run the model and see the agents interact with each other within their environment. For an example, see Figure 9.

One could divide the story into the following 5 main segments:

1. Riding Hood discusses visiting Grandma with Mother **(6 actions)**
2. Riding Hood meets Wolf and goes to Grandma **(23 actions)**
3. Wolf eats Grandma and tries to impersonate her; Riding Hood arrives at GrandmaHouse and cries for help **(6 actions)**
4. Woodsman saves Grandma and takes Wolf away, Riding Hood gifts Grandma **(13 actions)**

As an example, in the third story segment the actions occur in the following order:

1. Wolf eats Grandma to satisfy hunger.
2. Wolf took perfume from GrandmaHouse’s inventory to try impersonating Grandma.
3. Wolf took nightgown from GrandmaHouse’s inventory to try impersonating Grandma.
4. Wolf took sleeping cap from GrandmaHouse’s inventory to try impersonating Grandma.
5. Riding Hood moved 1 unit up to visit Grandma.
6. Riding Hood cried for help to get help.

The system is able to automatically generate the textual description of the story simulation over time, as depicted in Figure 8.

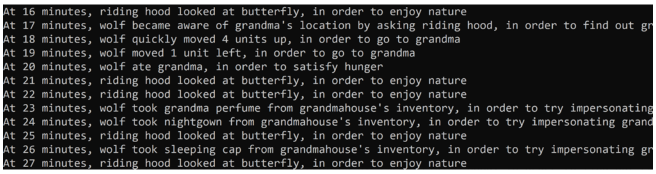

Figure 8: A part of an example story, generated by the system

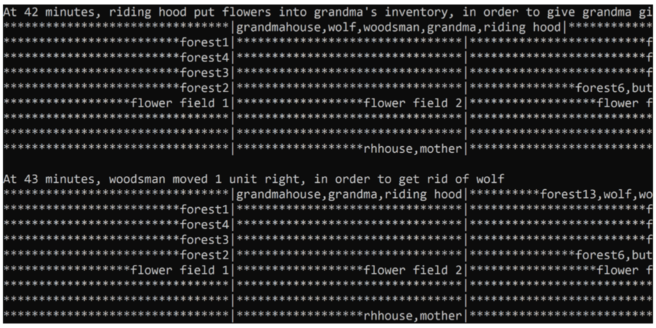

Figure 9: Screenshot of two subsequent agent location configurations on the map: (1) after Riding Hood gives Grandma flowers and (2) after Woodsman carries away Wolf

One of the more conceptually complex parts of the story was Riding Hood asking Mother for permission to visit Grandma. This required the creation of a new attribute for human agents to describe their opinions of other agents' goals.

The most complex action implemented was "cry for help". This involved the creation of a new goal "respond to cry for help" for all human agents within a certain radius of the agent crying for help, provided they were conscious and able to respond.

The story ends when Riding Hood gives Grandma the flowers she picked and the basket Mother gave her, and Woodsman carries the Wolf "deep into the forest where he wouldn't bother people any longer" [6].

The system was implemented in about 3,000 lines of C++ code, available on GitHub [7].

5 Discussion

In our research we expanded on and adapted existing work on agent-based models, providing an alternate approach to text understanding and generation involving short stories. As a proof of concept, we applied our approach on the children's story of 'Little Red Riding Hood', describing it through a series of 48 highly explainable actions involving 7 main agents.

Adapting the system to another story using our source code is relatively easy, provided the action and attribute types of the agents in the story are similar to those in the 'Little Red Riding Hood'. If the story requires the implementation of new actions or attributes, this can be done by extending the class structure in C++ using already implemented actions and attributes as examples.

In our future work we intend to integrate commonsense inferences, such as those from MultiCOMET into our model to further the system's degree of textual understanding. Our system could also benefit from the addition of dynamic and simultaneous goals that change based on the agent's environment. Another possible future line of work is to use our approach in other domains to describe more complex phenomena, such as real-world events or geopolitics. Lastly, a user evaluation of our system's performance on a variety of stories and scenarios could provide further insight into the efficacy of our approach.

ACKNOWLEDGMENTS

The research described in this paper was supported by the Slovenian research agency under the project J2-1736 Causalify and co-financed by the Republic of Slovenia and the European Union under the European Regional Development Fund. The operation is carried out under the Operational Programme for the Implementation of the EU Cohesion Policy 2014–2020.

REFERENCES

[1] Bosselut, A.; Rashkin, H.; Sap, M.; Malaviya, C.; Celikyilmaz, A.; and Choi, Y. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In ACL, 4762–4779.

[2] Adrian Mladenic Grobelnik, Marko Grobelnik, Dunja Mladenic. 2020. MultiCOMET – Multilingual Commonsense Description. In *Proceedings of the 23rd international multiconference information society, pages 37-40*

[3] Prithviraj Ammanabrolu, Wesley Cheung, William Broniec, and Mark O Riedl. 2021. Automated storytelling via causal, commonsense plot ordering. In Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI).

[4] Faeze Brahman and Snigdha Chaturvedi. 2020. Modeling protagonist emotions for emotion-aware storytelling. In Proceedings of EMNLP, pages 5277– 5294.

[5] Patrick Henry Winston. The genesis story understanding and story telling system: A 21st century step toward artificial intelligence. 2014. Technical report, Center for Brains, Minds and Machines (CBMM).

[6] Little Red Riding Hood by Leanne Guenther. https://www.dltk-teach.com/RHYMES/littlered/story.htm. Accessed 16.09.2021.

[7] Understanding Text Using Agent Based Models GitHub. https://github.com/AMGrobelnik/Understanding-Text-Using-Agent-Based-Models . Accessed 16.09.2021.

[8] Penberthy, J., & Weld, D. 1992. UCPOP: a sound, complete, partial-order planner for ADL. In Proceedings of KR'92, pp. 103–114, Los Altos, CA. Kaufmann.

[9] An Introduction to AI Story Generation. https://thegradient.pub/an-introduction-to-ai-story-generation/. Accessed 16.09.2021.