Generate Believable Causal Plots with User Preferences Using Constrained Monte Carlo Tree Search

Von-Wun Soo and **Chi-Mou Lee** and **Tai-Hsun Chen**
Department of Computer Science National Tsing Hua University, Hsinchu Taiwan
soo@cs.nthu.edu.tw
raymond2866286@yahoo.com.tw
benzpcpc@hotmail.com

Abstract

We construct a large scale of causal knowledge in term of Fabula elements by extracting causal links from existing common sense ontology ConceptNet5. We design a Constrained Monte Carlo Tree Search (cMCTS) algorithm that allows users to specify positive and negative concepts to appear in the generated stories. cMCTS can find a believable causal story plot. We show the merits by experiments and discuss the remedy strategies in cMCTS that may generate incoherent causal plots.

keywords: *Fabula elements, causal story plots, constrained Monte Carlo Tree Search, user preference, believable story generation*

Introduction

Researches on automatically generating believable stories by computers have been conducted many years (Meehan 1977) (Faas 2002) (Swartjes and Theune 2006) (Swartjes 2006) (Terluin 2008) (Kartal, Koenig, and Guy 2013) (Riedl and Young 2010) (Li et al. 2013) (Goudoulakis et al. 2011) (Chang and Soo 2009) (Liu and Singh 2002). We believe for computers to generate a believable story poses at least two main challenges: 1) collecting sufficient amount of common sense knowledge as story content elements and 2) devising a powerful planning and synthesizing method to ensure a coherent story to be found that satisfies believability and interestingness. One story generation school as (Swartjes 2006) proposed Fabula model as the fundamental plot layer of causal knowledge that allows some planning strategies to decide a proper order of events and presents them in natural language at the higher level. However, constructing knowledge base in terms of Fabula is very costly. In (Li et al. 2013), Boyang Li et al. involved human crowdsourcing techniques to collect story elements for such specific domain as bank robbing and automatically construct a causal-like graph on which a story generator can generate an accountable story. On the other hand, when the underlying search space consists of a large scale of causal Fabula elements, a planner synthesizer can become computational intractable. Riedl and Young (Riedl and Young 2010) built a partial order planner to synthesize a story plot so that each action of the characters can be explained by its goal or intention in the story in order to enhance the believability. In (Chang and Soo 2009), Chang and Soo built a planner and a generic mind model so that a sequence of characters actions motivated by the characters mind model can be synthesized. They all encountered the computational problem in generating a long sequence of event plans as the story plot. To cope with the search difficulty, Kartal et. al (Kartal, Koenig, and Guy 2013) adopted Monte Carlo Tree Search (MCTS) to generated believable stories given a bounded computational resource using a large set of believable predicates as common sense knowledge elements. However, it still encounters many difficulties. Firstly, given story conditions to MCTS, the user cannot express his/her preference on a story to be generated. The quality score can only measure in terms of the relative confidence on common sense association as the believability and cannot ensure the believability on event causality in the story plot. It implies two difficulties. Firstly the users may not request MCTS to search a story in a specific direction or with specific context scenario. Secondly, it may still generate ill-form and incoherent causal sequences. In this paper, we propose a design of MCTS called constrained MCTS (cMCTS) to remedy the problems.

Methods

Acquisition of Fabula elements from ConceptNet5

The idea of using common sense to facilitate generating believable stories is not new (Liu and Singh 2002). This paper focuses on the acquisition of Fabulla elements from existing common sense ontology ConceptNet5 for generating believable causal story plots. We extracted various common sense relations in terms of Concept-rel-Concept triples from ConceptNet5 (Liu and Singh 2004) (Speer and Havasi 2012) that can be potential Fabula elements and construct a Fabula knowledge base. Secondly, we classify the concept relation triples into their corresponding Fabula causal elements using a NLP part-of-speech tagger tool. Finally, we search for a sequence of causal links from an initial concept to a goal concept given a desired story length (DSL) based on Fabula model using the constrained Monte Carlo Tree Search algorithm (cMCTS). To extract candidate causal links from

Table 1: Sizes of Fabula causal links acquired from ConceptNet5

Fabula Link types	Number of links
Action to Event	72483
Action to Perception	2492
Event to Event	73331
Event to Perception	2533
Goal to Action	3026
Goal to Goal	197
Internal Element to Action	2780
Internal Element to Goal	66
Internal Element to Internal Element	238
Perception to Internal Element	251

ConceptNet, we adopted 9 types of relations in terms of (A, R, B) tuples between concept A and concept B as follows:

A hasprerequisite B To achieve B before achieve A,

A motivatedbygoal B A To achieve A because of B,

A usedfor B A is used to achieve B,

A capableof B A can be used to achieve B,

A causes B A causes B,

A causesdesire B A cause to desire to achieve B,

A hasfirstsubevent B Achieve A must achieve B first,

A haslastsubevent B Achieve A must achieve B as last step,

A hassubevent B Achieve A must achieve B,

We classified the extracted concepts (e.g. A and B) in these causal links into corresponding Fabula elements based on their semantic and syntactic phrases in the concepts with the aid of java NLP (nature language processing) toolkit (Apache 2010) library. We obtained 2441 perceptions, 85607 actions, 87687 events, 3254 goals, but no outcome elements. We then established the corresponding Fabula causal links based on their Fabula elements concepts in the ConceptNet relations and finally obtained various causal links as shown in Table 1.

The Constrained Monte Carlo Tree Search Scheme

A Monte Carlo tree search (MCTS) algorithm (Browne et al. 2012)(Gelly et al. 2012) conducts iterative execution of four steps as: selection, expansion, simulation, and back propagation to form a search tree and add a new node to a tree at the end of each iteration. MCTS can balance between exploration and exploitation within limited computing resources and thus can find a reasonable quality of solution in a large search space. The Upper Confidence Bounds (UCB) is a scoring parameter that balances between the frequency of visits and the simulation quality of a node as in Eq. 1.

$$UCB = (1-\alpha)S + \alpha\sqrt{\frac{\ln v}{n_v}} \tag{1}$$

where UCB is the evaluation score of selecting a particular child node of a node v, S indicates the exploitation as the quality of the child node while the square root part indicates the exploration, v is the number of visits of the node v while n_v is the number of visits of the selected child node of node v, α weighting parameter balances the exploration and exploitation that can dynamically decrease from 1 to 0 with decrement 1/#MCTS at each stochastic search. We implemented firstly a version of MCTS and allows user specify initial concept and goal concept to generate a story at the desirable length. The quality score S is based on the association weights of the causal links from ConceptNet5 as well as a Gaussian function of the story length. When MCTS reaches the goal concept, it is regarded as a success and will back propagate the quality score S and allocate the search resource according to UCBs of nodes to select the next search direction as in (1). MCTS repeats the stochastic search until it finishes its bounded #MCTS of searches.

Now we wish a story to be generated with certain desirable conditions (positive concepts) but avoid certain undesirable conditions (negative concepts) to appear in the final story. The cMCTS design is motivated by allowing users to specify both hard and soft constraints for the search. The desired story length and the positive and negative concepts are regarded as the soft constraints while the initial and goal states are the hard constraints. The soft constraints can be expressed as a utility function for the cMCTS to find a solution while the hard constraint is for the decision of the success or failure of a search in cMCTS. To design a cMCTS the utility function for evaluating the quality score S of a candidate story is modified. The quality score of a causal story plot in simulation consists of three parts: A) the causal association confidence weight, B) a weight of the desirable story-length C) the degree of satisfying the positive and negative concepts.

The association confidence weight A of a causal story plot sequence can be summed up by each relation association weight $score_i$ from ConceptNet5 and then divide it by the total number of relations n in the story and the maximum association weight score Maxscore as in Eq. 2.

$$A = \frac{\sum_1^n score_i}{n}/MaxScore \tag{2}$$

We assume the weight of desired story length B behaves as a Gaussian function of the story length l whose mean μ is the desired story length (DSL) and variance σ is approximately 1/3 μ. This ensures a story length that is the closest to the desired story length to get the highest weight. Thus B can be calculated as Eq. 3.

$$B = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(l-u)^2} \tag{3}$$

The degree of satisfying the positive and negative constraints can be expressed as a constraint satisfaction weight C as in Eq. 4.

$$C = \frac{1 + un + sp - sn}{1 + un + up} \tag{4}$$

where
un is the number of user specified un-desired negative concepts;

Table 2: Redundant concepts cause a cyclic causal sequence

be_sick ⊢ cd → **go_to_doctor** ⊢ pre → **have_physical_exam** ⊢ sub → **drop_your_pant** ⊢ pre → **have_physical_examination** ⊢ c → **visit_doctor** ⊢ pre → **make_sure_you_healthy** ⊢ sub → **visit_to_doctor** ⊢ pre → **have_checkup** ⊢ c → **stress** ⊢ cd → **get_exercise** ⊢ c → **get_healthy**

up is the number of user specified desired positive concepts;
sn is the number of negative concepts in the story;
sp is the number of positive concepts in the story
the number 1 is to prevent the cases when both un and up are both zeros. In this way, when the number of positive concepts is greater than that of negative ones, C will be greater than 1.0, otherwise, C will be less than 1.0.

Since Eqs. 2, 3, 4 are normalized formula, we combine the three scores to obtain the final simulation quality score S as in Eq. 5.

$$S = \rho * A * B * C \tag{5}$$

where ρ is a constant being set at 100 so that story quality score S falls between 0 and 100.

Avoid generating a cyclic causal sequence with redundant concepts

When the generated story has exactly the same concept appear in two places, we can force cMCTS to automatically remove one of them to avoid a redundant and cyclic story to appear. However, when concepts are expressed in different words that have the same or similar meaning (e.g. synonyms), it would be difficult for computers to tell and thus to avoid a redundant story to be generated. Table 2 is an example of a redundant story that is generated by MCTS due to synonyms "go_to_doctor", "visit_doctor", "visit_to_doctor" as well as synonyms "have_physical_exam" and "have_physical_examination". The causal story plot is represented as a sequence causal links in terms of "Concept1 ⊢ rel → Concept2" representation that are selected from common sense Fabula knowledge base. For clarity of representing causal links, in the subsequent representation, we abbrevi- ate different kinds of rels such as c: "causes", pre: "hasprerequisite", sub: "hassubevent", 1st: "hasfirstsubevent", mb: "motivatedbygoal", cd: "causesdesire", u: "usedfor", cap: "capableof", last: "haslastsubevent", respectively.

To remove the redundancy concepts, we adopt ConceptNet5 API of finding similarity between two concepts. We establish a concept redundant set for each concept when a concept has high similarity to others. We show its effects in experimental results.

Generate a story without goal state or initial state

Sometimes users may not wish to specify the goal state or the initial state. In the case of without specifying a goal state, we allow cMCTS to use story length as the sole termination condition to return the quality score. In case of without specifying the initial state, cMCTS simply reverses its search order "from initial state to goal state" to "from goal state to initial state", all the rest of processes in cMCTS remains the same.

Generate a longer story

Theoretically, given sufficient search resources #MCTS, it would be able to yield a story with as arbitrarily long as one desires. However, as #MCTS increases, the computation cost (both time and memory) of MCTS increases too. Therefore we still have limitation to generate a very long story. On the other hand, if #MCTS is given too few, it is unlikely to generate a very long story because the probability that cMCTS can reach a goal node with a very long length will become very low in a huge search space. To obtain a longer sequence, intuitively we may divide a story into many sub pieces and generate each sub piece separately. As starting with an initial state, we can find a sub piece story that reaches certain goal state, and then use the goal state as a new initial state to search for the next sub piece of story. Repeat this process several times and combine sub pieces in sequence we find as a long story as we want.

Automatically translate the causal links to quasi-English sentences

In order to facilitate readability of a generated story plot, a simple English sentence generator is designed to convert the causal sequence into English sentences. We call it quasi-English sentence generation, because we did not intend to get a syntactic-error free natural language generator but just a semantic interpretable one at this point. We build sentence templates for each relation type in ConceptNet as shown in Table 3. Each template is semantic equivalent, so it is selected randomly each for a specific relation type. Since in ConceptNet, all verbs tend to be the present tense, we converted into their past tense using JWKTL (Meyer and Gurevych 2012). The generated causal plot is expressed in terms of a sequence of causal Fabula elements between concepts c_i's as: $c_1 \vdash rel_1 \rightarrow c_2$, $c_2 \vdash rel_2 \rightarrow c_3$, $c_3 \vdash rel_3 \rightarrow c_4$, ..., $c_{f-1} \vdash rel_{f-1} \rightarrow c_f$,. To avoid redundancy, we generate only a sentence for every pair of concepts. Namely, generate a sentence only for $c_1 \vdash rel_1 \rightarrow c_2$ and $c_3 \vdash rel_3 \rightarrow c_4$, etc. If there is a last causal for odd number of causal elements we generate with "and + (he/she) + c_f".

In the subsequent experiments, we all assume the character P is Jason and the G is male so "he", "his", "him" are used in corresponding pronominal cases in generating quasi-natural language sentences.

Experimental Results

We ran cMCTS under different experimental conditions using a PC at Intel Core i5 CPU whose speed is 2.9 GHz to demonstrate the performance of cMCTS in generating causal story plots.

Experiments with user preferences expressed as positive and negative concepts

Suppose the user wishes to generate a story to describe a character who prefers sports but hates learning. We can allow the user to specify constraints as positive and negative

Table 3: Sentence templates to generate quasi-natural language sentences (P: character agent, G: gender of the character he/she; c_1 and c_2 are concepts in $c_1 \vdash$ rel $\rightarrow c_2$ link)

Relation	Sentence templates
Hasprerequisite	P needed to c_1 to c_2. After P c_1, G c_2. P c_1, then G c_2.
Motivatedbygoal	In order to c_1, P would c_2. P wanted to c_1, so G c_2. Since P would like to c_1, G c_2.
Usedfor	P wanted to c_1, so G c_2. Since P would like to c_1, G c_2. In order to c_1, P would c_2.
Capableof	P c_1 to c_2. P c_1 in order to c_2. P wanted to c_1, so G c_2.
Causes	When P c_1, G would c_2. P c_1, so G c_2. P c_1, therefore G would c_2.
Causesdesire	When P c_1, G would like to c_2. P c_1, so G would desire to c_2. P c_1, therefore G would want to c_2.
Hasfirstsubevent hasSubevent haslastsubevent	When P c_1, G would like to c_2. While P c_1, G would c_2. As P c_1, G would c_2.

concepts for cMCTS to generate a story and see how the story will be generated to tell the character who may be doing from "wake up" to "go home". Assume user has specified initial concept and goal concept as well as the positive and negative concepts in the first row in Table 4 where DSL is the desired story length specified, FSL is the final story length generated by cMCTS, Score is the quality score of the generated story, #MCTS is the number bound of the stochastic search in cMCTS. A causal story plot with score 31.12 was generated in the second row and its corresponding quasi-English output is shown in the third row.

As we could see that both "jog" and "walk" concepts appear in the story, while the learning related concepts "study", "read", and "learn" do not. Thus, the story tends to concentrate more toward the characters preference on sports. To compare the number of positive and negative concepts appeared against that without positive and negative concepts, we ran an experiment in the unspecified goal stories and chose top 500 highest score stories. We then count how many specified positive and negative concepts appear in the stories respectively. Table 5 shows that the frequency for positive constraint concepts to appear is 345/500 (39 "exercise"; 296 "play_baseball"; 10 "swim") while that for the negative concepts to appear is 1/500 (only 1 "read") if we specified the constraints. On the other hand if we did not specify the constraints, the frequency of positive concepts to appear is 51/500 while that of the negative ones is 28/500. This indicates cMCTS has effectively selected more positive concepts while filtered out more negative ones as we specified.

Table 4: cMCTS allows the user to specify positive and negative concepts as soft constraints

Initial: wake_up Goal: go_home Positive concepts: exercise, jog, walk Negative concepts: study, read, learn #MCTS : 1000000 DSL: 12	Score: 31.12 FSL: 10 Run Time: 3 min 25 s
wake_up $\vdash$ pre $\rightarrow$ **eat_breakfast** $\vdash$ last$\rightarrow$ **brush_your_tooth** $\vdash$ pre $\rightarrow$ **go_out** $\vdash$ pre $\rightarrow$ **go_for_walk** $\vdash$ c $\rightarrow$ **be_outdoors** $\vdash$ cd $\rightarrow$ **play_frisbee** $\vdash$ mb $\rightarrow$ **go_to_park** $\vdash$ pre $\rightarrow$ **go_jog** $\vdash$ last $\rightarrow$ **go_home**	
Jason woke up, then he ate breakfast. After Jason brushed his tooth, he went out. When Jason went for walk, he would be outdoors. In order to play frisbee, Jason would go to park. When Jason went jog, he would like to go home.	

Table 5: Effectiveness of expressing positive and negative concepts

Initial: wake_up Positive concepts: exercise, play_baseball,swim Negative concepts:study, read, learn #MCTS: 1000000 DSL: 12 Run Time: With / without : 2min 41sec / 2min 38sec	# of positive concepts appeared in the generated stories	# of negative concepts appeared in the generated stories
With Constraints	39+296+10 =345	0+1+0 = 1
Without Constraints	7+43+1= 51	0+18+10 = 28

Experiments to avoid generation of redundant concepts

This set of experiments shows how to avoid redundant and cyclic stories by using similarity score among concepts of ConceptNet API with threshold set at 0.9. In Table 6 we show a story found that has redundant concepts go_to_doctor, visit_doctor, and find_doctor that are similar concepts and cause redundancy of a story. In this case, we did not specify positive and negative concepts.

After cMCTS dealing with concept similarity, the result is shown in Table 7.

As we could see that, the redundancy of visiting doctor is removed. However, we also notice that the FSL = 6 which is far from DSL=10 indicating that in current knowledge base, visiting doctor is the only way to reach getting healthy. Therefore removing redundancy concept nodes, cMCTS simply cannot find a longer path to reach the goal concept.

Experiments without specifying the goal concept or initial concept

Table 8 shows the result that the user does not specify a goal concept. It comes up with a story plot after the initial state

Table 6: A scenario with redundancy in a story plot due to concept similarity

Initial: be_sick Goal: get_healthy #MCTS : 1000000 DSL: 10	Score: 30.75 FSL: 11 Run Time: 2 min 34 sec
be_sick ⊢ cd → **go_to_doctor** ⊢ pre → **have_physical_examination** ⊢ mb → **find_doctor** ⊢ pre → **make_sure_you_healthy** ⊢ mb → **visit_doctor** ⊢ pre → **lose_weight** ⊢ last → **work_out** ⊢ mb → **go_to_gym** ⊢ pre → **exercise** ⊢ c → **get_healthy**	
Jason wanted to make sure him healthy, so he visited doctor. While Jason losing weight, he would work out. After Jason went to gym, he exercised. And he got healthy.	

Table 7: Dealing with redundancy by removing the concept similarity

Initial: be_sick Goal: get_healthy #MCTS : 1000000 DSL: 10	Score: 14.42 FSL: 6 Run Time: 2 min 6 sec
be_sick ⊢ cd → **have_checkup** ⊢ mb → **go_to_doctor** ⊢ pre→ **have_physical_examination** ⊢ pre → **do_some_exercise** ⊢ c → **get_healthy**	
Jason was sick, so he would desire to have checkup. Jason went to doctor, then he had physical examination. When Jason did some exercise, he would get healthy.	

wake up and positive and negative concepts specified by the user. And it leads to the person who went to swim as the exercise context specified by the user.

Table 9 shows a result that the user does not specify an initial concept. It can come up with a story to explain why a person cried by setting the goal concept cry.

Experiments of generating a longer story length

Intuitively, generating longer stories will be more likely to come up with intriguing and interesting stories. But how long a story can cMCT generate given current causal knowledge base size? To answer the question, we show simulation results in Figure I by increasing DSL from 10 to 50 using initial concept = "go_to_school"; goal concept = "go_home"; #MCTS = 3000000. We observed that the average story length tended to decrease at around DSL= 40. This implies that with bounded stochastic search resource and knowledge base size, cMCTS can still have limitation to come up with arbitrarily long story, even the user specifies a very long DSL.

We showed a long story with length = 30 that is generated by cMCTS with initial concept as "wake_up" and goal concept "go_home".

The long story seems to be believable in causality except the link to went to "dmv (division of mobile vehicle)". There are many factors that incoherence can arise that are to be discussed in the discussion section.

Table 8: A scenario without specifying goal concept

Initial: wake_up Positive concepts: exercise, jog, walk Negative concepts: study, read, learn #MCTS : 1000000 DSL: 10	Score: 22.07 FSL: 8 Run Time: 2 min 41 s
wake_up ⊢ pre → **eat_breakfast** ⊢ c → **be_late_for_work** ⊢ c → **eat_quickly** ⊢ c → **eat_too_much** ⊢ c → **get_fat** ⊢ cd→ **exercise** ⊢ mb → **go_for_swim**	
Jason needed to wake up to eat breakfast. Jason was late for work, so he ate quickly. Jason ate too much, therefore he would get fat. In order to exercise, Jason would go for swim.	

Table 9: A scenario without specifying initial concept

Goal: cry Positive concepts: exercise, play_baseball, swim Negative concepts: study, read, learn #MCTS : 1000000 DSL: 10	Score: 13.18 FSL: 7 Run Time: 1 min 11 s
need_exercise ⊢ mb → **play_ball** ⊢ mb → **go_to_park** ⊢ pre → **play_baseball** ⊢ c → **break_arm** ⊢ c → **be_in_pain** ⊢ cd → **cry**	
Jason wanted to need exercise, so he played ball. Jason went to park to play baseball. When Jason broke arm, he would be in pain. And he cried.	

Discussion

When we ran the experiments by allowing users to specify only the initial or goal concepts, we found an interesting phenomenon. It is that the FSL of the generated story tends to be shorter when either the initial or goal concept is missing. In our current implementation, it turns out the FSLs of generated stories were usually not longer than 15. The rationale behind it, in one aspect is that current knowledge base has limited amount of causal knowledge, as in Table 1, the number of causal Fabula elements that we extracted from ConceptNet5 are relatively few in Goal-to-Goal, Internal Element-to-Goal, Internal Element-to-Internal Element, and Perception-to-Internal Element. The lack of these elements may cause a bottle-net not only in generating a long story, but also in generating the mental motivation of a character in a story. In another aspect, which is more important, is that without specifying the initial concept (backward search) or goal concept (forward search), cMCTS simply does not have a precise termination node to guide the search. In these cases, cMCTS can only rely on the length quality and the constraint satisfaction quality as the guide of stochastic search. However, these two types of utility scores tend to provide less guidance in the search direction, it ends up that cMCTS could not effectively allocate its bounded search resources (#MCTS) on better quality stories and thus it results in shorter FSL due to its more evenly exploration at the huge search space.

We apply concept similarity checking to remove redundant concepts automatically that may lead to cyclic incoherent story plots. However, in subtle scenarios when a re-

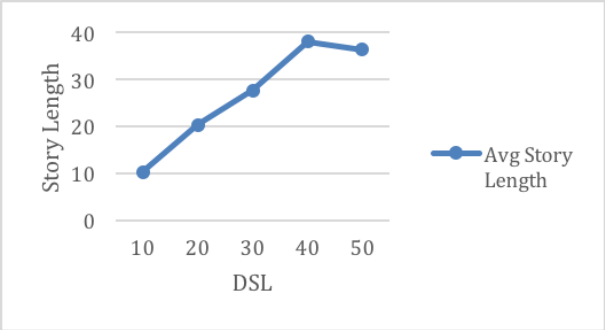

Figure 1: Average story lengths generated vs. desired story lengths

dundant behavior is indeed needed in some story plot, this has to be treated separately. Although cMCTS seems to generate many causal stories that are believable to users, we still encountered a difficulty of ensuring global causal coherency with respect to human intuition and common sense. In the long story we generated in Table 10 actually, "wait_on_line" and "stand_in_queue" should be similar concepts that wasnt detected and filtered and thus causes cyclic and incoherent causality. In another example, in Table 11, "Drinking_coffee" may lead to "go_to_bathroom", while "go_to_bathroom" maybe a prerequisite of "sleep_at_night", both are locally coherent. But combining them together into a causal sequence as a story plot, it becomes that "drinking coffee" causes a subsequent action for "sleep_at_night" which seems to be weir against the daily life experience of most people.

When a concept has multiple meanings (polysemy), it may lead to incoherent causal links too. In Table 12, it shows incoherent causal links before and after the "sign_paper" concept. It is due to that "sign_paper" means differently in different contexts such as "loan_contract" or "divorce_agreement". It is incoherent in that buying a house can cause one to sign a paper but that is different from the purpose of getting divorce.

To remove such risk of trapping global incoherence, it needs to keep all conditions in local causal links and inspect their compatibility and applicability at every subsequent causal links. The computational cost will become as intractable as the traditional planning techniques in dealing with a large scale of domain problems. Therefore how to cope with the global incoherence arising from local coherence can be in general a hard problem to deal with, especially applying in a non-specific domain.

Conclusion

We extracted Fabula elements from common sense ontology ConceptNet5 based on the Fabula causal model and established a stochastic search framework using cMCTS algorithms to generate a Fabula causal plot. cMCTS can find a believable story plot with a long sequence of causal events from a large search space within reasonable time. It allows users to specify initial state and goal state as the hard constraints and the desired story length, the positive (desirable) and negative (undesirable) concepts as the soft constraints. In this way, it tends to find believable stories in the desirable context and in avoid of the undesirable context for the users. It sheds some light to the possibility of allowing logical combinations of constraints at high level as long as it does not incur too much computational cost. In the future cMCTS design, we could also allow users to specify both positive and negative concepts as hard constraints. In that case, cMCTS simply has to embed these additional constraints as the success or failure criteria as the goal concept in the current stochastic search. Our research goal would also look into higher levels of narrative and discourse strategies that can integrate with the cMCTS to generate an interesting story.

Table 10: A long story with FSL=30

Initial: wake_up Goal: go_home #MCTS : 2000000 DSL: 30	Score: 32.72 FSL: 30 Run Time: 8 min, 30 sec
wake_up ⊢ cd → **begin_work** ⊢ c → **stress** ⊢ c → **fatigue** ⊢ cd → **rest** ⊢ c → **be_late_to_work** ⊢ c → **get_fired** ⊢ pre → **go_on_vacation** ⊢ last → **pack** ⊢ pre → **take_trip** ⊢ mb → **save_money** ⊢ pre → **pay_by_credit_card** ⊢ mb → **go_to_bank** ⊢ pre → **wait_on_line** ⊢ mb → **go_to_dmv** ⊢ cd → **stand_in_queue** ⊢ mb → **wear_comfortable_shoe** ⊢ pre → **go_to_mall** ⊢ c → **buy_some_shoe** ⊢ pre → **take_walk** ⊢ c → **go_outside** ⊢ pre → **go_for_run** ⊢ mb → **put_on_your_sneaker** ⊢ pre → **walk_around_lake** ⊢ c → **pleasure** ⊢ mb → **have_fun** ⊢ mb → **go_somewhere** ⊢ mb → **ride_bike** ⊢ c → **travel** ⊢ last → **go_home**	
Jason woke up, so he would desire to begin work. Jason stressed, so he fatigue. When Jason rested, he would be late to work. After Jason got fired, he went on vacation. After Jason packed, he took trip. Jason saved money, then he paid by credit card. Jason went to bank to wait on line. Jason went to dmv, therefore he would want to stand in queue. Jason wear comfortable shoe to go to mall. Jason buy some shoe to take walk. Jason went outside, then he went for run. Jason put on his sneaker to walk around lake. Since Jason would like to pleasure, he had fun. Jason wanted to go somewhere, so he rode bike. When Jason traveled, he would go home.	

Table 11: A causal plot may lead to global incoherent result

wake_up ⊢ 1st → **brush_your_tooth** ⊢ pre → **begin_work** ⊢ sub → **drink_coffee** ⊢ c → **go_to_bathroom** ⊢ pre → **sleep_at_night** ⊢ 1st → **close_your_eye** ⊢ pre → **go_to_bed**

Table 12: polysemy concepts may lead to a global incoherent result

get_marry ⊢ mb → **make_money** ⊢ mb → **buy_house** ⊢ last → **sign_paper** ⊢ pre → **get_divorce**

References

Apache. 2010. Java nlp toolkit. *https://opennlp.apache.org/*.

Browne, C.; Powley, E.; Whitehouse, D.; Lucas, S.; Cowling, P. I.; Rohlfshagen, P.; Tavener, S.; Perez, D.; Samoth-

rakis, S.; and Colton, S. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games* 4(1).

Chang, P., and Soo, V.-W. 2009. Planning-based narrative generation in simulated game universes. *IEEE Transactions on Computational Intelligence and AI in Games* 1(3).

Faas, S. 2002. Virtual storyteller: An approach to computational storytelling. master's thesis. *University of Twente*.

Gelly, S.; Kocsis, L.; Schoenauer, M.; Sebag, M.; Silver, D.; Szepesvári, C.; and Teytaud, O. 2012. The grand challenge of computer go: Monte carlo tree search and extensions. *Communications of the ACM* 55(3):106–113.

Goudoulakis, E.; Rhalibi, A. E.; Merabti, M.; and Taleb-Bendiab, A. 2011. Evaluation of planning algorithms for digital interactive storytelling. *PGNet conference*.

Kartal, B.; Koenig, J.; and Guy, S. J. 2013. Generating believable stories in large domains. *Intelligent Narrative Technologies: AIIDE Workshop*.

Li, B.; Lee-Urban, S.; Johnston, G.; and Riedl, M. O. 2013. Story generation with crowdsourced plot graphs. *Proceedings of the 27th AAAI Conference on Artificial Intelligence*.

Liu, H., and Singh, P. 2002. Makebelieve: Using commonsense knowledge to generate stories. *In Proceedings of the Eighteenth National Conference on Artificial Intelligence, AAAI*.

Liu, H., and Singh, P. 2004. Conceptnet - a practical common sense reasoning tool-kit. *BT Technology Journal* 22(4):211– 226.

Meehan, J. R. 1977. Tale-spin, an interactive program that writes stories. 91–98.

Meyer, C. M., and Gurevych, I. 2012. *Wiktionary: A new rival for expert-built lexicons? Exploring the possibilities of collaborative lexicography*. Electronic Lexicography. Oxford University Press. chapter 13, 259–291.

Riedl, M. O., and Young, R. M. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research 39* 217–268.

Speer, R., and Havasi, C. 2012. Representing general relational knowledge in conceptnet 5. *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*.

Swartjes, I., and Theune, M. 2006. A fabula model for emergent narrative. *Third International Conference on Technologies for Interactive Digital Storytelling and Entertainment* 49–60.

Swartjes, I. 2006. The plot thickens: bringing meaning and structure into automated story generation. *Master's thesis, University of Twente*.

Terluin, D. 2008. From fabula to fabulous - using discourse structure relations to separate paragraphs in automatically generated stories.