ENTAILMENT OF CAUSAL QUERIES IN NARRATIVES USING ACTION LANGUAGE

Pawan Goyal[1], Laxmidhar Behera[1,2] and T. M. McGinnity[1]

[1]*Intelligent Systems Research Centre, School of Computing and Intelligent Systems, University of Ulster, U.K.*

[2]*Department of Electrical Engineering, Indian Institute of Technology, Kanpur, India*

Keywords: Question answering, Plan recognition.

Abstract: In this paper, Action Language formalism has been used to reason about narratives in a multi agent framework. The actions have been given a semantic frame representation. Hypothetical situations have been dealt using different states for world knowledge and agents' knowledge. A notion of plan recognition has been proposed to answer causal queries. Finally, an algorithm has been proposed for automatically translating a given narrative into the representation and causal query entailment has been shown.

1 INTRODUCTION

A narrative is a course of events about which the reader may not be given the complete knowledge, i.e. some of the knowledge needs to be inferred. Reasoning about the actual cause of events in a narrative is one of the challenging problems in Natural Language Processing (NLP). NLP researchers deal this problem from the point of view of semantics and context, giving rise to a causal relation. There have been attempts for reasoning about causal relations in text by Girju (Girju and Moldovan, 2002) and Bethard (Bethard and Martin, 2008).

A narrative involves multiple agents, where each agent has a set of beliefs, which changes dynamically upon the actions performed by him or other agents. Typically researchers have given various logic formalisms for representing actions, their cause and effects. Much of the work has focused on action languages (Tu et al., 2007) (Baral and Gelfond, 2005) for proper knowledge representation. There have also been attempts using event calculus (Mueller, 2002), situation calculus (Mccarthy and Hayes, 1969), non-monotonic causal logic (Giunchiglia and Lifschitz, 2004) and action temporal logic (Giunchiglia and Lifschitz, 1999). While these approaches are successful in building planning strategies, they are limited to entailing the truth value of fluents[1] at varying situations and only the causal queries captured by the causal model (Baral et al., 1997) can be entailed.

In the knowledge representation of stories, we come across plans made by agents. As we will show, recognizing plans from observations is very essential for certain causal queries. There have been attempts by researchers for plan recognition from narratives (Quilici et al., 1998). Reasoning about causality has also led researchers to look into plan identification (Pearl, 2000). However, in a multi agent framework, we need to reason about a special case of plan, in which the plan made by an agent dictates the actions performed by another agent also. We propose a causal model for this special case. The proposed causal model has been used for entailing causal queries.

The paper has been organized as follows. In section 2, the problem under consideration has been formulated. In section 3, plan recognition and causal query entailment have been discussed using the causal model. In section 4, the mapping to logic programming has been provided. In section 5,the algorithm for translating the narrative into the proposed formalism has been discussed. In section 6, the causal queries being answered by the method and the results have been discussed. Conclusion and future work have been provided in section 7.

[1]In artificial intelligence, a *fluent* is a condition that can change over time and situation.

Goyal P., Behera L. and M. Mcginnity T. (2009).
ENTAILMENT OF CAUSAL QUERIES IN NARRATIVES USING ACTION LANGUAGE.
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, pages 112-118
DOI: 10.5220/0002275601120118

2 PROBLEM DEFINITION

The problem of narrative understanding is being addressed: “To read a narrative and answer the questions at the end of it”. Figure 1 gives an overall description of a narrative understanding system. The input narrative is to be parsed for obtaining the dependency relations. The ‘fact extractor’ extracts the facts as required by the ‘dynamic situation representation’. The ‘word sense disambiguation’ and ‘anaphora resolution’ modules give an unambiguous representation of narrative with distinct objects, agents and verbs. The narrative is translated to the ‘dynamic situation representation’ using the ‘knowledge base’, ‘model for capturing dynamics’ and the ‘inference mechanism’ (which has belief update as its essential part). Finally, a query is translated by the ‘question processor’ and a goal is generated. The prolog program uses the goal to generate the output, which acts as input to the ‘natural language generation’ module, giving the answer in natural language. In this work, the focus is on the ‘Model for capturing dynamics’, ‘Dynamic situation representation’ and the ‘Fact extractor’. The set of queries have been restricted to the causal queries.

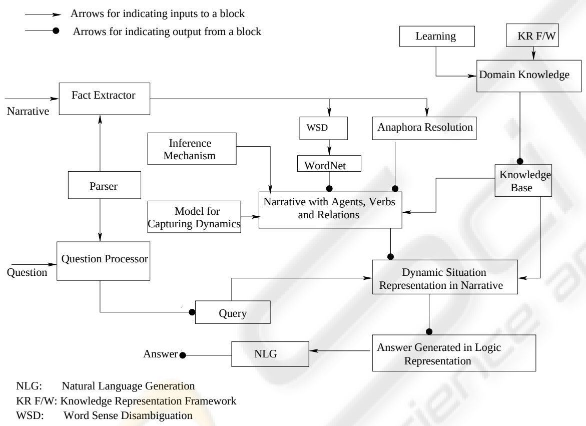

Figure 1: Block Representation of the Narrative Understanding System.

In subsection 2.1, the model used for capturing dynamics has been reviewed. In subsection 2.2, the dynamic situation representation built from a narrative is reviewed.

2.1 Model for Capturing Dynamics

The Action Language $A_C{}^K$ formalism, presented in (Tu et al., 2007) is suitable for representing causal laws and incomplete information. The propositions in $A_C{}^K$ for domain description are as follows:

$$1.executable(a,\psi) \quad 2.causes(a,l,\phi)$$
$$3.if(l,\phi) \quad 4.determines(a,\theta)$$

A proposition of form (1) is called an executability condition. It states that a is executable in any situation in which ψ holds. A proposition (2), called a dynamic causal law, represents a conditional effect of an action. It says that performing a in a situation in which ϕ holds, causes l to hold in the successor situation. A proposition (3), called a static causal law, states that l holds in any situation in which ϕ holds. A knowledge proposition (or $k-proposition$ for short) (4) states that the values of literals in θ, sometimes referred to as sensed-literals, will be known after a is executed (a is the sensing-action).

In our attempt to answer causal queries, we propose to replace ‘executable(a,ψ)’ by ‘isPossible(a,ψ)’. The reason behind this is the following: While dealing with a real world narrative, it is difficult to come up with executability conditions. However, learning methods can be employed to come up with the model, ‘what situations(ψ) caused an agent to perform an action(a)?’. This formalism allows us to answer the queries ‘Why did the agent perform action a?’, given that there is a domain law of the proposed form. Further, we propose to add the pre conditions in sensing actions as well, i.e. replace ‘determines(a,θ)’ with ‘determines(a,θ,ϕ). The reason for the proposed change is the representation of hypothetical situations. Hypothetical situations are caused while a sensing action is performed. However, the agent may like to believe the situation based on certain constraints. ϕ are the constraints (set of fluents). We also propose to add the proposition ‘isGoal(l,ϕ,ag)’, which allows us to select goal for agents ag in various situations. The proposition is needed to recognize that the agent ag has a desire to achieve l and the actions performed by ag may be the plan of the agent, given that ag achieves l (The statement is valid only for successful plans. The work does not consider the unsuccessful plans.).

2.2 Dynamic Situation Representation of Narrative

A narrative can be seen as a pair (D,O) where D is the domain description and O is a set of observations. Observations are to be interpreted with respect to D (D is the set of domain dependent axioms, as enumerated in subsection 2.1. In a narrative involving multiple agents, all the agents may not have complete knowledge about the world. We use an approximate state (combined-state) $S = < s, \sum >$, where s represents the real state of the world, while $\sum$ represents the set of

states[2], the agents are in.
To interpret the facts from D, we need to select the initial situation together with a path, which describes the actual behavior of the system.

In a narrative, we observe both fluents and actions and the effect of an action may not be explicitly stated. Thus the assumption that 'No action occurs except those needed to explain the facts in the domain descriptions.' (Baral et al., 1997) is not justified and the narrative is inconsistent with respect to the causal interpretation. We, therefore do not discuss causal model in general, but only the causal model for the plans made by the agent.

3 OUR FORMALISM: ENTAILING CAUSAL QUERIES

In our formalism, we discuss the domain representation, observation interpretation and query entailment.

3.1 Domain Description D

The domain is described using the following propositions, discussed in the previous section:

1.$causes(a,l,\phi)$ 2.$if(l,\psi)$
3.$determines(a,\theta,\phi)$ 4.$isGoal(l,\phi,ag)$
5.$isPossible(a,\psi)$

3.2 Observation Interpretation O

Interpreting observation is the central part of the model. It needs to use D to get the causal representation. The algorithm is described in section 5.
The notion of 'desire' for an agent has been used as the basic step in plan recognition. Previously, Baral (Baral and Gelfond, 2005) has used intended actions in planning. Cao (Cao and Pontelli, 2004) has also used desires to allow users to express preferences and priorities in planning. In our formalism, we use the notion as follows: If agent has desire to achieve a certain goal, and the observation shows that he has really achieved the goal, it is most likely a plan and the agent performs actions intending a future step in the plan.
Before getting into the problem, "How do we gather such knowledge", we need to reason about 'how to represent this knowledge', and once represented, how to use it for answering causal queries. In this paper, we discuss only the successful plans. In this plan, the agent may perform actions that (The enumerated examples are taken from the specific story from section 6):
1. Change the knowledge of another agent. (Ex: The rabbit stated that there is another lion, who is challenging the supremacy of the lion.)
2. Are evidences for the constraints satisfying the hypothetical situations (Ex: The rabbit strode along the lion by sunset. The action was performed to satisfy the constraint that if the rabbit met the lion, it must be late.)
3. Any other action, which the agent performs to achieve the goal. (For example, the rabbit lead the lion to the well. The action was performed so that it can show the lion its reflection.)

We give the following definitions for describing the causal model of the plan:

Definition 1. (The Chain of Causal Relations): We say that there exists a chain of causal relations for action a_i and a_{i+1} if the following holds:

1. $occursAt(a_i,S_i).occursAt(a_{i+1},S_{i+1})$.
2. $\exists f, causes(a_i,f,p), p \subseteq S_i, \quad not(holds(f,S_i)), isPossible(a_{i+1},q), f \in q$

For the actions a_i and a_j, $j > i+1$, we say that the chain of causal relations exist if any two actions a_k, a_{k+1} belonging to the sequence of actions $a_i,\ldots,a_j$ satisfy the above two conditions.

Definition 2. (The Plan): Given a narrative and a situation model, the sequence of situations $P = S_S,\ldots,S_E$ is said to be a plan sequence if the following holds true:
1. The agent A has a goal l in situation S_S.
2. $\exists$ a sequence of actions $a_S,\ldots,a_E$ such that: $occursAt(a_S,S_S)\ldots.occursAt(a_E,S_E)$.
3. The agent achieved the goal l in the situation S_{E+1}.
4. For the actions, $a_S,\ldots,a_E$, we have the sequence of actions, where the subject is A given by $act_A = (a_A)_1,\ldots,(a_A)_l$ and there is no chain of causal relations for any two actions $(a_A)_k,(a_A)_{k+1}$ from the action sequence.

Definition 3. (Hypothetical Situation): An agent A is in a hypothetical situation at the situation S_i if the corresponding world state is sw_i, the agent A's state is $(sa_A)_i$ and $\exists f$ s.t. $not(holds(f,sw_i))$ and $holds(f,(sa_A)_i)$.

Definition 4. (Evidence Action): If the plan P entails the sequence of actions act_A, the action $a_i \in act_A$ is said to be evidence action if $\exists S_j$, such that it is a hypothetical situation for agent B and it has the precondition (The hypothetical situation is always caused by

[2] In our formalism, each agent thinks himself to be in a single state and therefore, the number of agents is an upper bound on the number of states in a particular situation. It is possible that some of the agents agree on all the fluents and hence, are in same state.

sensing actions) as the fluent g caused by a_i such that the fluent f is hypothetical, i.e. the following holds:

$$causes(a_i, g, p). \quad determines(a_k, f, g).$$
$$not(holds(f, sw_j)). \quad holds(f, (sa_B)_j).$$

We add the following proposition for these actions in the observation: '$evidenceOf(a_i, f)$.'

Definition 5. (Causal Model for Plan): Given a Plan P of the agent A and the sequence of actions act_A, let us define the following sets: $(act_A)_E$ be the set of evidence actions. $(act_A)_H$ be the set of actions such that the resulting situation is a hypothetical situation for agent $B, B \neq A$. $(act_A)_C = act_A/((act_A)_E \cup (act_A)_H)$. We have the following:
1. If $(act_A)_C = \{a_1, \ldots, a_k\}$, we have the causal model: $causeOf(a_i, a_{i+1}), i \in \{1, \ldots, k-1\}$
2. For an action $a_i \in (act_A)_H$, $causeOf(a_i, f)$, such that if $occursAt(a_i, S_j)$, S_{j+1} is a hypothetical situation for agent $B, B \neq A$ with $not(holds(f, sw_{j+1}))$ and $holds(f, (sa_B)_{j+1})$.
3. For an action $a_i \in (act_A)_E$, we have $evidenceOf(a_i, f)$, where f is defined as above.

3.3 Query Entailment

For the action language formalism, the queries have been limited to the truth values of certain fluent in a certain situation or after a series of actions performed in a situation. Since causal queries are the focus of this work, we are extending the queries being represented. The following categories of causal queries need to be entailed:
1. Fluents caused an action to occur: These queries are encoded in the proposition, proposed in this paper 'isPossible(a, ψ)'. Thus, for the query: 'Why did action a happen?', the answer should be 'due to ψ'.
2. Action caused fluents to hold: These queries are encoded in the already existing dynamic (static) causal law 'causes(a, l, ϕ) (determines(a, θ, ϕ))'. Thus for the query, 'Why does '$l(\theta)$' hold?', the answer should be 'due to 'a'.
3. Actions having causal relations due to planning: These queries are entailed in the causal model for the plan.

It can be easily observed that these queries are entailed in the causal model. We are extending query to handle the chain of causal relations as discussed by Definition 1.

4 MAPPING TO LOGIC PROGRAM

In this section, we describe the logic programming approximation π_D of the domain description D. For actions, we are using semantic frames. Let D be a domain description with the explicit actual path $a_0, \ldots, a_{k-1}$. The logic programming approximation of D will consist of the following rules:

4.1 Domain Dependent Axioms

4.1.1 Description of Actual Path

We have the following axioms for denoting the world states:
$precedes(s_0, s_1). \ldots precedes(s_{k-1}, s_k)$.
$occursAt(a_0, s_0). \ldots occursAt(a_{k-1}, s_{k-1})$.
For agent and world states, we use the notation that in a situation, being referred as s_i, the world is in state w_i and the agents are in state $(sa_1)_i, \ldots, (sa_N)_i$, where N is the number of agents. We have the following axioms:
1. $situationOf(s_i, (sa_j)_i, ag_j)$ refers to the fact that the situation $(sa_j)_i$ is the knowledge of agent ag_j in situation s_i.
2. $allAgree(s_i, (sa_j)_i)$ refers to the fact that all fluents that hold in situation s_i, also hold in situation $(sa_j)_i$.

4.1.2 Boundary Conditions

These conditions are responsible for representing observed fluents at different situations. '$holds(f, s_i)$.' represents that fluent f is observed to hold in situation s_i.

4.1.3 Possible Goals

These propositions are responsible for determining the values of agents at different situations. The proposition '$isGoal(l, p, (sa_j)_K)$.' states that agent a_j has goal l in situation $(sa_j)_K$[3], given the list of fluents p holds in the situation.[4]

4.2 Domain Independent Axioms

These axioms are independent of the domain and need to be included in every narrative, encoded in given

[3]In our formalism, we represented this proposition as $isGoal(l, \phi, ag)$. However, in logic program, we use the situation $(sa_j)_K$ corresponding to the agent ag. It uniquely determines the agent ag.

[4]We are not describing other domain dependent axioms. These can be referred in the paper by Baral (Baral et al., 2000).

formalism. We have used following axioms[5]:

4.2.1 Axioms for Plan Recognition

Firstly, we need an axiom to encode, when the plan is achieved.
$achievedGoal(X,S_B) \; : \; -holds(X,S_B), isGoal(X,Y,S_A)$. Once Goal has been achieved, we can name the sequence of actions performed to be a plan: $isPlan(S_1,S_S) \; : \; -isGoal(X,Y,S_1), achievedGoal(X,S_S)$. We have in addition, the axioms for the definitions given in section 3.

4.2.2 Axioms for Causal Queries

The axioms included are for static and dynamic causal laws, as well as from the causal model of plan.

5 THE ALGORITHM

Given a narrative, the goal is to automatically translate it to the proposed formalism. The algorithm is general for any text but the causal model is specific to the narratives. The basic steps in the algorithm are as follows:
1. Dependency Parse: The dependency parse of the narrative has been obtained using the Stanford dependency parser (MacCartney, 2008).
2. The nouns and verbs are given separate identifiers. The nouns are classified as agents using the criteria that i). It should be identified as a 'living thing' by the Wordnet (Fellbaum, 1998). ii). Its frequency should be more than the given threshold. Finally, the whole narrative is translated in Prolog using the two predicates[6]:
$word(a,X)$: identifier a is used for word X.
$relation(Z,X,Y)$: The dependency relation Z holds in the identifiers X and Y.
3. The domain dependent axioms are entered. These axioms are responsible for the knowledge update of the narrative[7].
4. The algorithm in Figure 2 is used to represent the narrative.

[5] We are not describing the axioms for Inertia rule, effect of actions and initial states. These can be referred in the paper by Baral (Baral et al., 2000).

[6] The identifiers distinguish between nouns, verbs, adverbs and adjectives and therefore, we do not use POS tags.

[7] In this work, we have assumed that the domain dependent axioms are given to us. However, we are working for learning this domain knowledge.

1. Start with the initial situation S_0.
2. The main verb of the sentence is observed. The verb is categorized as a fluent, non-sensing action or sensing action as per the Verbnet.
 i). If the verb represents a fact (fluent f), the observation '$holds(f,s_N)$.' is added, where s_N is the current situation.
 ii). If the verb is a non sensing action, the situation is to be changed using '$occursAt(a,s_N)$.' and '$holds(f,Res(a,s_N))$.', where the later is added using the causal relations of the domain '$causes(a,l,\phi), \phi \subseteq s_N$', $Res(a,s_N)$ is the situation as a result of applying action a in situation s_N.
 iii). If the verb is a sensing action and determines the value of fluent θ using the proposition $determines(a,\theta,\phi)$, the situation of the corresponding agent is changed.
3. The proposition $isGoal(l,\phi,ag)$ is used to get the corresponding goal of the agent ag, if any. If there existsa goal, it is added to the program: '$isGoal(l,\phi,s_N)$'.
4. The static causal laws are used to represent the value of other fluents.
5. The inertia law for the fluents is added: $holds(f,Res(a,s)) : -holds(f,s), not(ab(f,a_s))$
6. If the fluent resembles a goal, already declared by $isGoal(l,\phi,s_i)$, the axiom '$acievedGoal(l,ag,s_N)$' is added. The causal model for the plan is built using the Definition 5.

Figure 2: The algorithm for the dynamic situation representation of the narrative.

6 RESULTS AND DISCUSSIONS

The algorithm discussed in the previous section has been implemented on a set of stories. An example of one such story has been given in Figure 3. After the narrative is automatically translated into the 'dynamic situation representation' using the algorithm, proposed in section 5, we had 18 different situations. The domain dependent axioms were given to the system. We have followed the Verbnet syntax (Schuler, 2005), so that the axioms are more general and applicable to any text. Below are two examples of these axioms:

Once a ferocious lion lived in the forest. The lion was greedy. It started killing animals in the forest indiscriminately. The animals gathered. They decided to approach the lion. They had an agreement with the lion that one animal of each species will volunteer to be eaten by the lion everyday. So every day it was the turn of one of the animals. In the end came the rabbits' turn. The rabbits chose an old rabbit among them. The rabbit was wise. It took long time to go to the lion. the lion got impatient on not seeing any animal come by. It swore to kill all animals the next day. The rabbit then strode along to the lion by sunset. The lion was angry at it. But the wise rabbit was calm. the rabbit told the lion that an angry lion attacked the rabbit on the way. Somehow it escaped to reach safely, the rabbit said. The rabbit said that the other lion was challenging the supremacy of his Lordship the lion. The lion was naturally very enraged and asked the rabbit to take him to the location of the other lion. The wise rabbit agreed and led the lion towards a deep well filled with water. Then the rabbit showed the lion his reflection in the water of the well. The lion was furious and started growling. Naturally the image in the water, the other lion, was equally angry. Then the lion jumped into the water at the other lion to attack it, and so lost its life in the well. Thus the wise rabbit saved the forest and its inhabitants from the proud lion.

Figure 3: The algorithm for the dynamic situation representation of the narrative.

```
========axiom 1=========
VERB:live
THEME:nsubj->INHABITANT
THEME:prep_in->LOCATION
ab(live,lost):-nsubj(lost,INHABITANT),
dobj(lost,life).
========axiom 2=============
VERB:kill
THEME:nsubj->KILLER
THEME:dobj->PREY
isGoal(PREY,not(kill)):-aspect(kill,start).
```

Axiom 1: If the fluent 'live' holds in this situation, it will not hold (i.e. abnormality) when the subject of 'live' loses its life. The fluent 'live' holds in the narrative at initial situation. The abnormality condition is added by axiom 1 to entail 'not(live)', when the action 'the lion lost its life' occurs in the narrative.

Axiom 2: If the action 'kill' occurs at this situation with aspect of 'start', the PREY (which is a theme role) has a goal to stop the killing 'not (kill)'. The action 'kill' occurs in the initial situation in the story. The axiom generates the goal 'not (kill)' for the 'animals'.

As can be seen, we have used theme roles for the domain dependent axioms (belief update rules).
The system answers factual queries[8] using pattern matching and the synonyms information. We used the predicate '$causeOf(X,Y)$' to reason about various actions and fluents. We show below examples of some causal queries[9]:
1. Why did the animals decide to approach the lion?
A: To have an agreement.
2. Why did the lion start killing animals indiscriminately in the forest?
A: Because the lion was greedy.
3. Why was the lion getting impatient?
A: Because it did not see any animal coming.
These queries are easily answered using the causal laws, sensing actions and possible actions axioms. The set of queries answered due to the causal model of Plan are as follows:
4. Why did the rabbit show the reflection of the lion?
A: It wanted the lion to assume that there is another lion.
The answer is obtained since the action '$show \in (act_A)_H$', where A refers to the rabbit.
5. Why did the rabbit stride to the lion by sunset?
A: As an evidence to show that there was another lion. The fact that 'the rabbit came late' is a precondition (or constraint) for the lion to believe that there was another lion. Thus '$stride \in (act_A)_E$'.

While the first three questions can be answered using the causal model, the questions 4 and 5 need deep understanding, which is accomplished in our formalism using plan recognition, hypothetical actions and evidence actions. It is clear that the category of questions is limited by the chosen model.

[8]These queries include the WH-questions and the decision questions

[9]The answers generated were in Prolog form. However, different scripts are used to generate the natural language answers, depending upon the semantics of the query.

7 CONCLUSIONS AND FUTURE WORK

The work focuses on the problem of coming up with a theoretical formalism to answer causal queries in a real world narrative. The main contribution of the work is to use the plan recognition for reasoning about the cause. However, the notion of causality (Pearl, 2000) has to be incorporated in the formalism to reason about actual cause. Future work will demonstrate the system to answer counterfactual queries. Another important aspect for future work will be to translate the queries into Prolog representation to generate goals and use the answers given by Prolog to generate natural language answers.

To make a fully automated system, which can answer causal queries, substantial additional research is needed. The probabilistic extension of the model is required to handle the incomplete domain knowledge and uncertainty. The belief update model has to be built based on the Verbnet (Schuler, 2005) and Framenet (Baker and Sato, 2003) representation. Semantic entailment will need to be used but the soundness and completeness of the representation needs to be investigated.

REFERENCES

Baker, C. F. and Sato, H. (2003). The framenet data and software. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 161–164, Morristown, NJ, USA. Association for Computational Linguistics.

Baral, C. and Gelfond, M. (2005). Reasoning about intended actions. In *AAAI*, pages 689–694.

Baral, C., Gelfond, M., and Provetti, A. (1997). Representing actions: Laws, observations and hypothesis. *Journal of Logic Programming*, 31:31–1.

Baral, C., Mcilraith, S., and Son, T. C. (2000). Formulating diagnostic problem solving using an action language with narratives and sensing. In *KR 2000*, pages 311–322.

Bethard, S. and Martin, J. H. (2008). Learning semantic links from a corpus of parallel temporal and causal relations. In *ACL-08: HLT, short papers*, pages 177–180.

Cao, T. and Pontelli, S. E. (2004). Planning with preferences using logic programming. In *Proc. LPNMR04*, pages 247–260. Springer.

Fellbaum, C. (1998). *WordNet: An Electornic Lexical Database*. Bradford Books.

Girju, R. and Moldovan, D. (2002). Text mining for causal relations. In *Proceedings of the FLAIRS Conference*, pages 360–364.

Giunchiglia, E. and Lifschitz, J. L. V. (2004). Nonmonotonic causal theories. *Artificial Intelligence*, 153:2004.

Giunchiglia, E. and Lifschitz, V. (1999). Action languages, temporal action logics and the situation calculus. In *Working Notes of the IJCAI'99 Workshop on Nonmonotonic Reasoning, Action, and Change*.

MacCartney, B. (2008). The stanford parser version 1.6.

Mccarthy, J. and Hayes, P. J. (1969). Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence*, pages 463–502. Edinburgh University Press.

Mueller, E. T. (2002). Understanding script-based stories using commonsense reasoning. *Cognitive Systems Research*, 5:2004.

Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

Quilici, A., Yang, Q., and Woods, S. (1998). Applying plan recognition algorithms to program understanding. *Journal of Automated Software Engineering*, 5:372.

Schuler, K. K. (2005). *Verbnet: a broad-coverage, comprehensive verb lexicon*. PhD thesis, Philadelphia, PA, USA. Supervisor-Palmer, Martha S.

Tu, P. H., Son, T. C., and Baral, C. (2007). Reasoning and planning with sensing actions, incomplete information, and static causal laws using answer set programming. *Theory Pract. Log. Program.*, 7(4):377–450.