Learning High Level Planning From Text

Nate Kushman

S.R.K. Branavan, Tao Lei, Regina Barzilay
Goal: Show that planning can be improved by utilizing precondition information in text

**Precondition/Effects Relationships**

**Castles are built with magic bricks**

**Classical Planning:**

\[
\text{have} \left[ \text{magic bricks} \right] \rightarrow \text{have} \left[ \text{castle} \right]
\]

**NLP: Linguistic Relation**

\[
\text{Castles are built with} \quad \text{magic bricks}
\]
A pickaxe, which is used to harvest stone, can be made from wood.

Preconditions

- Wood
- Pickaxe

Plan

- Move to location: <3,3>
- Harvest: wood
- Retrieve: harvested wood
- Setup crafting table
- Place on crafting table: wood
- Craft: pickaxe
- Retrieve: pickaxe
- Move to location: <1,2>
- Pickup tool: pickaxe
- Harvest: stone with: pickaxe
- Retrieve: stone

Challenge: Preconditions from text cannot map directly to planning action preconditions
Classical Planning’s Problem:
Exponential heuristic search

Traditional Solution:
Analyze domain to induce subgoals

Text:
A **pickaxe**, which is used to harvest **stone**, can be made from **wood**.

Precondition Relations:

- pickaxe \(\rightarrow\) stone
- wood \(\rightarrow\) pickaxe

Key Idea: Map text precondition information to subgoals
Key Departures

Utilize domain specific information in text to induce subgoals
Jonsson and Barto, 2005; Wolfe and Barto, 2005; Mehta et al., 2008; Barry et al., 2011

*looked only at domain, did not utilize text*

Learn from only environment feedback
Girju and Moldovan, 2002; Chang and Choi, 2006; Blanco et al., 2008; Beamer and Girju, 2009; Do et al., 2011; Kwiakowski, 2012

*Learns from supervised data, does not utilize environment feedback*

Utilize text providing abstract domain relationships (not goal specific)
Oates, 2001; Siskind, 2001; Yu and Ballard, 2004; Fleischman and Roy, 2005; Mooney, 2008; Branavan et al., 2009; Liang et al., 2009; Vogel and Jurafsky, 2010; Branavan et al., 2009; Branavan et al. 2010; Vogel and Jurafsky, 2010; Branavan et al., 2011

*Focused on grounding words to objects, does not ground relations*
Hybrid Model

- **Text**: Log-linear model → Precondition relations
- **Planning target goal**: Log-linear model → Sub-goal sequence
- **Learn model parameters from planning feedback**

- **Model precondition descriptions**: Log-linear model
- **Model object relations in world, and ground preconditions**: Log-linear model → Low-level planner → Plan
Modeling the World

- State is represented by a set of predicates
  
  \[
  \text{current\_location}(1,2) = \text{TRUE} \quad \text{current\_tool}(\text{pickaxe}) = \text{TRUE}
  \]

- Actions represented by preconditions and effects

  \[
  \text{Preconditions:} \quad \text{tree\_at}(1,2) = \text{TRUE} \quad \text{current\_location}(1,2) = \text{TRUE}
  \]

  \[
  \text{Effect:} \quad \text{tree\_at}(1,2) \rightarrow \text{FALSE} \quad \text{have}(\text{wood}) \rightarrow \text{TRUE}
  \]

Goals and subgoals are represented as predicates
Cooked fish is obtained when raw fish is cooked in a furnace.

**Goal Independent**
Model Part 2: Predict Subgoal Sequence

Given Goal State

- Model as a Markov process
- Explicitly model preconditions observed via planner
Policy Functions

Model Part 1: Predict Precondition Relations from text

\[ p(x_i \rightarrow x_j | \vec{w}_k, q_k; \theta_c) \propto e^{\theta_c \cdot \phi_c(x_i, x_j, \vec{w}_k, q_k)} \]

Prediction per pair: Manual groundings, \( x \) 
Sentence, \( w \), dependency parse, \( q \)

Model Part 2: Predict Subgoal Sequence

\[ p(x_t | x_{t-1}, s^g_0, s^g_f, C; \theta_x) \propto e^{\theta_x \cdot \phi_x(C, x_t, x_{t-1}, s^g_0, s^g_f)} \]

Markov Assumption 
Relations from text, \( C \) 
Relations between predicates, \( x \)
Learn Parameters Using Feedback from the Planner

Model Parameters $\theta$

Model

Sub-goal sequence

Low-level planner

Parameter Updates

Planner succeeds or fails on each step

Reinforcement Learning Algorithm (Policy Gradient)
Parameter Updates: Relation Prediction

Separate update for each relation

Negative update for all unnecessary preconditions
Parameter Updates: Subgoal Sequence Prediction

Start → String → Fishing pole → Raw fish → Cooked fish

Start → Go to market → Buy fish → Raw fish

One update for the whole sequence
Updates

Model Part 1: Precondition Relation Prediction

\[ \Delta \theta_c \leftarrow \alpha_c r \begin{bmatrix} \phi_c(\cdot) - \mathbb{E}[\phi_c(\cdot)] \end{bmatrix} \]

Success/failure of one subgoal pair

standard log-linear gradient

Model Part 2: Subgoal Sequence Prediction

\[ \Delta \theta_x \leftarrow \alpha_x r \sum_{t} \begin{bmatrix} \phi_x(\cdot) - \mathbb{E}[\phi_x(\cdot)] \end{bmatrix} \]

Success or failure of entire sequence

Sum over all subgoal pairs
Experimental Domain

World:

*Minecraft virtual world*

Documents:

*User authored wiki articles*

Text Statistics:

-Sentences: 242
-Vocabulary: 979

Planning task Statistics:

-Tasks: 98
-Avg. plan length: 35
-Min. Branching Factor: 8

*Pickaxes*

*Pickaxes* are one of the most commonly used *tools* in the game, being required to mine all *ores* and many other types of blocks. Different qualities of pickaxe are required to successfully
Models compared

Unmodified Low-level Planner
  *Fast-Forward – standard baseline in classical planning*
  *No induced subgoals*

No Text
  *Second half of model given no relations from text*

All Text
  *Generate all connections with grounded phrase in same sentence*
  *Second-half of model with this set of connections*

Full Model
  *As described so far*

Manual Text Connections
  *Manually annotate all connections implied by the text Use second half of model with the manual connections*
Results

Low-level Planner: 40.80%

Full Model: 80.2%

% of tasks completed successfully
Results

Low-level Planner: 40.80%
No Text: 69.4%
Full Model: 80.2%

% of tasks completed successfully
Results

- **Low-level Planner**: 40.8%
- **No Text**: 69.4%
- **All Text**: 75.5%
- **Full Model**: 80.2%
- **Manual Text**: 84.5%

Very close to upper bound
Results: Tasks Longer Than 35 Actions

% of tasks completed successfully

- **Low-level Planner**: 14%
- **No Text**: 31%
- **All Text**: 48%
- **Full Model**: 59%
- **Manual Text**: 64%

Almost twice the performance of No Text
Results: Text Analysis

![Graph showing the comparison between Model F-score and SVM F-score across learning iterations. The F-score for the model increases rapidly and then plateaus around 0.65, while the SVM F-score remains constant at 0.7.](image-url)
Conclusion

- Our method can learn to ground textual descriptions of precondition relations

- Precondition relationship information can improve performance on complex planning tasks

*Code and data available at:*

http://groups.csail.mit.edu/rbg/code/planning/