KNEXUS RESEARCH CORPORATION

Workflow Learning

Based on RNN

Lefan Zhang, Shenglan Qian
2018 June - August

TOC

Project Timeline

Preparation

Understood the problem and read related papers.

Goal prediction

Implemented the first model to predict the goal.

Pattern recognition

Designed algorithm to learn patterns from the raw dataset.

Evaluation

Designed several evaluation methods, helping to optimize the model.

Live demo

Built a web based live demo to show how the system works

Future work

Preprocessing data, considering parameters, testing on other datasets

Understanding the problems

01 Actions in a workflow may have certain relationship

Ex. Open The Door -> Close The Door

02 People are doing a sequence of actions to reach certain goal

Ex. Place Cones -> Clean Hazard -> Pickup Cones | Clean Road Hazard

03 Some goals can be generate by sub goals, sub goals are combinations of actions

Ex. (Front end) -> (Back end) -> (Database) | Develop a software

Overview

The purpose of this workflow learning is to provide better user experience while they are creating a new workflow. When the user type in a few actions, our system will predict what goal the user is trying to accomplish, and our system will automatically provide some advices on what the rest of actions that could take in place to accomplish this goal. User could either directly select what system is recommended or they could at least be remind of what are the key actions that could happen in the workflow.

This workflow learning is consist of three main parts:

1. New Pattern Learning (Pattern Recognition)
2. Goal Prediction (RNN)
3. Future Action Prediction

Project objective

Since we don’t have real user data from the Taskport system, we are using similar plan corpus instead (Linux corpus and Monroe plan corpus).

We hope that we can build a model which have great performance on these sample datasets, and once we got the real data, we can use the same model with little modification.

Data set

Linux corpus

Linux corpus is a collection of Linux sessions collected from real users. As a consequence, it contains mistakes.

Total Session	457
Goal schemas	19
Command Types	43
Command Instances	2799
Avg Commands/Session	6.1

Monroe Plan corpus

Monroe Plan Corpus is a hierarchical plan corpus generated automatically using an AI Planner.

Total Sessions	5000
Goal Schemas	10
Action Schemas	30
Avg Actions/Session	9.5
Subgoal Schemas	28
Avg Subgoal Depth	3.8
Max Subgoal Depth	9

Data set

Linux Corpus

Goal: compress-dirs-by-loc-dir(gang)

(ls(,**dot_0**))
(cd(,bin))
(ls(,**dot_1**))
(cd(,gang))
(cd(,**dotdot_2**))
(tar(,-,,gang))
(tar(,-,,gang))
(cd(,**dotdot_3**))
(ls(,**dot_4**))
(cd(,code))
(ls(,**dot_5**))
(cd(,**dotdot_5**))
(cd(,mail))
(ls(,**dot_7**))
(tar(,-,,gang))

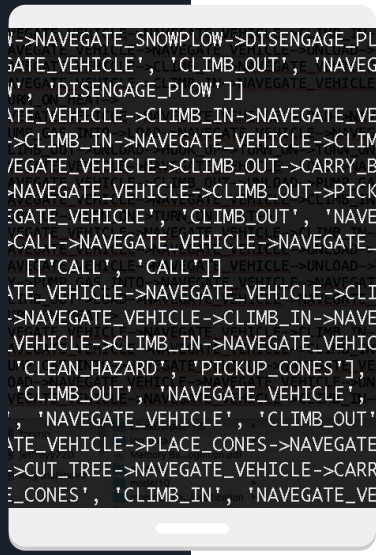
Monroe Plan Corpus

```
((PLOW-ROAD STRONG TEXACO1)
 ((GET-TO PDRIVER2 PITTSFORD-PLAZA)
  ((GET-TO PVAN2 12-CORNERS)
   ((DRIVE-TO PU2 PVAN2 12-CORNERS)
    (!NAVEGATE-VEHICLE PU2 PVAN2 12-CORNERS)))
  ((GET-IN PDRIVER2 PVAN2) (!CLIMB-IN PDRIVER2 PVAN2))
  ((GET-TO PVAN2 PITTSFORD-PLAZA)
   ((DRIVE-TO PU2 PVAN2 PITTSFORD-PLAZA)
    (!NAVEGATE-VEHICLE PU2 PVAN2 PITTSFORD-PLAZA)))
  ((GET-OUT PDRIVER2 PVAN2) (!CLIMB-OUT PDRIVER2 PVAN2)))
 (!NAVEGATE-SNOWPLOW PDRIVER2 PLOW1 STRONG)
 (!ENGAGE-PLOW PDRIVER2 PLOW1)
 (!NAVEGATE-SNOWPLOW PDRIVER2 PLOW1 TEXACO1)
 (!DISENGAGE-PLOW PDRIVER2 PLOW1))
```

Simplify the dataset

Goal: compress

Sequence:
ls->cd->ls->cd->cd->tar->tar->cd->ls->cd->ls->cd->cd->ls->tar



Goal: PLOW_ROAD

Sequence:
NAVEGATE_VEHICLE->CLIMB_IN->NAVEGATE_VEHICLE->CLIMB_OUT->NAVEGATE_SNOWPLOW->ENGAGE_PLOW->NAVEGATE_SNOWPLOW->DISENGAGE_PLOW-><EOS>

Subgoal: GET_TO

Sequence:
NAVEGATE_VEHICLE->CLIMB_IN->NAVEGATE_VEHICLE->CLIMB_OUT-><EOS>

Project Timeline

Preparation

Understood the problem and read related papers.

Goal prediction

Implemented the first model to predict the goal.

Pattern recognition

Designed algorithm to learn patterns from the raw dataset.

Evaluation

Designed several evaluation methods, helping to optimize the model.

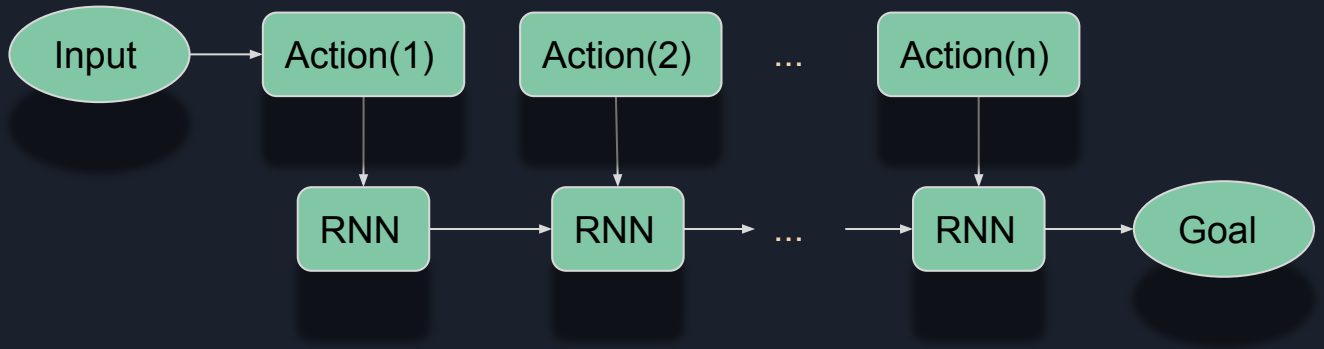
Live demo

Built a web based live demo to show how the system works

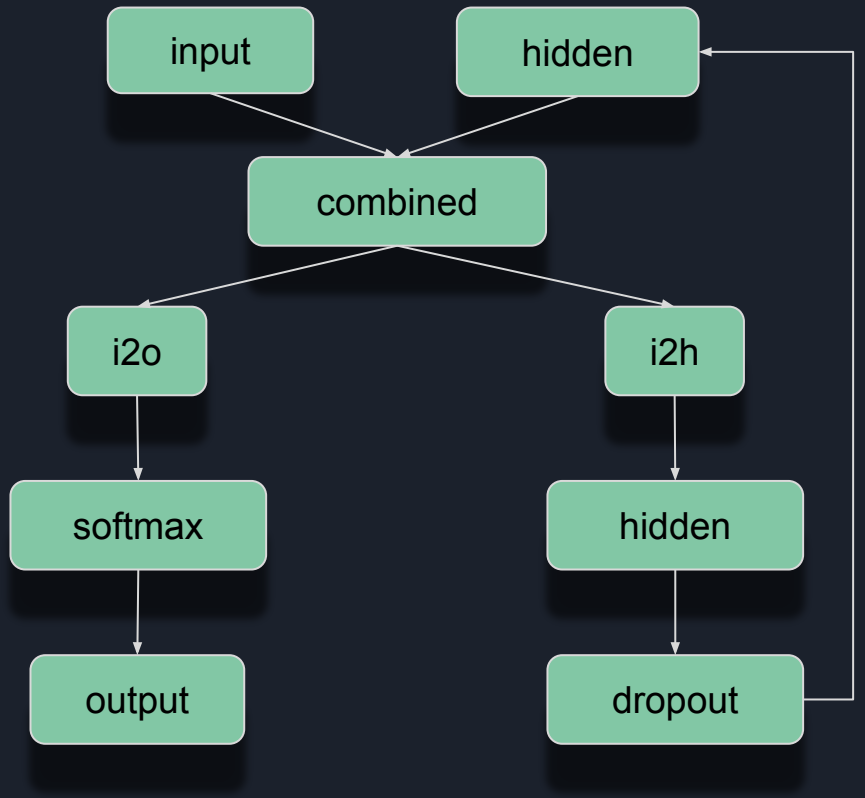
Future work

Preprocessing data, considering parameters, testing on other datasets

Goal prediction using RNN



RNN design



Dropout_prob = 0.2

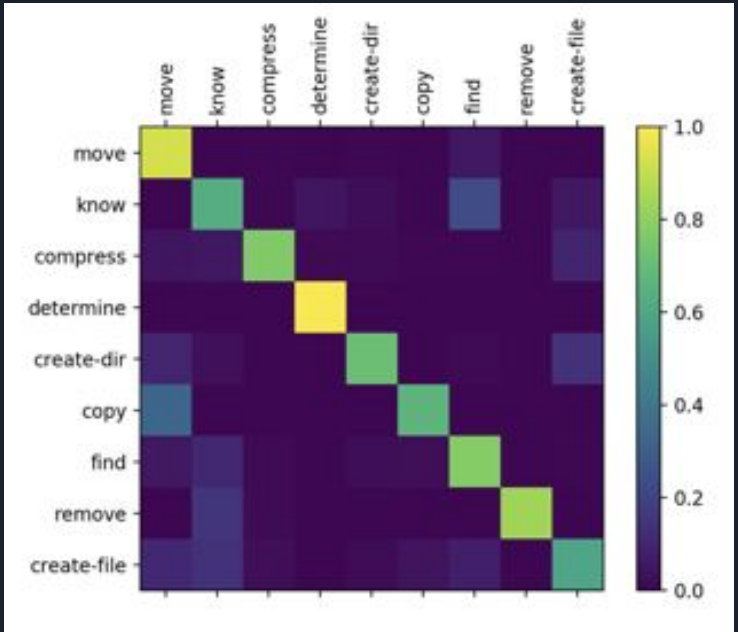
Learning_rate = 1E-4

Loss_function = NLLLoss

Optimizer = RMSprop

Evaluation

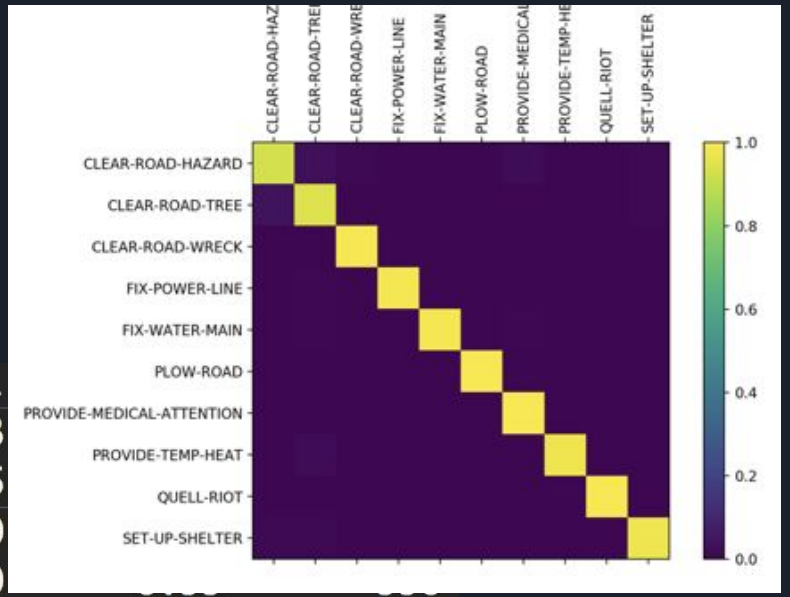
Linux dataset



	precision	recall	f1-score	support
move	0.92	0.77	0.84	170
	+0.25	+0.28	+0.27	
know	0.57	0.67	0.62	30
	+0.49	-0.33	+0.47	
compress	0.75	0.71	0.73	70
	+0.08	-0.15	-0.02	
determine	0.65	0.59	0.62	120
	-0.32	-0.13	-0.2	
create-dir	0.88	1.00	0.94	80
	-0.12	+0.12	+0.01	
copy	0.80	0.78	0.79	350
	-0.16	+0.27	+0.13	
find	0.60	0.63	0.62	180
	-0.33	+0.24	+0.07	
remove	0.82	0.93	0.87	280
	+0.02	+0.15	+0.08	
create-file	1.00	0.84	0.91	80
	+0.22	-0.04	+0.09	
avg / total	0.79	0.78	0.78	1360
	-0.06	+0.14	+0.09	

Evaluation

Monroe plan dataset



	precision	recall		
CLEAR-ROAD-HAZARD	0.96	0.93		
CLEAR-ROAD-TREE	0.92	0.95		
CLEAR-ROAD-WRECK	0.99	0.99		
FIX-POWER-LINE	1.00	0.99		
FIX-WATER-MAIN	1.00	0.99	1.00	460
PLOW-ROAD	1.00	1.00	1.00	3460
PROVIDE-MEDICAL-ATTENTION	0.99	1.00	0.99	2610
PROVIDE-TEMP-HEAT	1.00	0.98	0.99	1420
QUELL-RIOT	1.00	1.00	1.00	480
SET-UP-SHELTER	0.94	0.98	0.96	440
avg / total	0.99	0.99	0.99	15030

Knowing the goal earlier

Training on partial sequence data

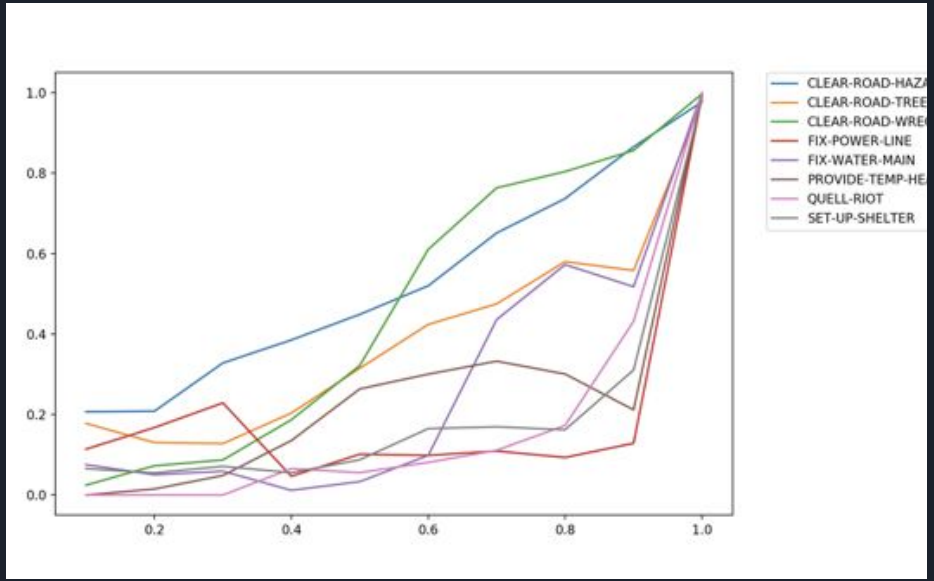


Figure 1: Training on whole sequences

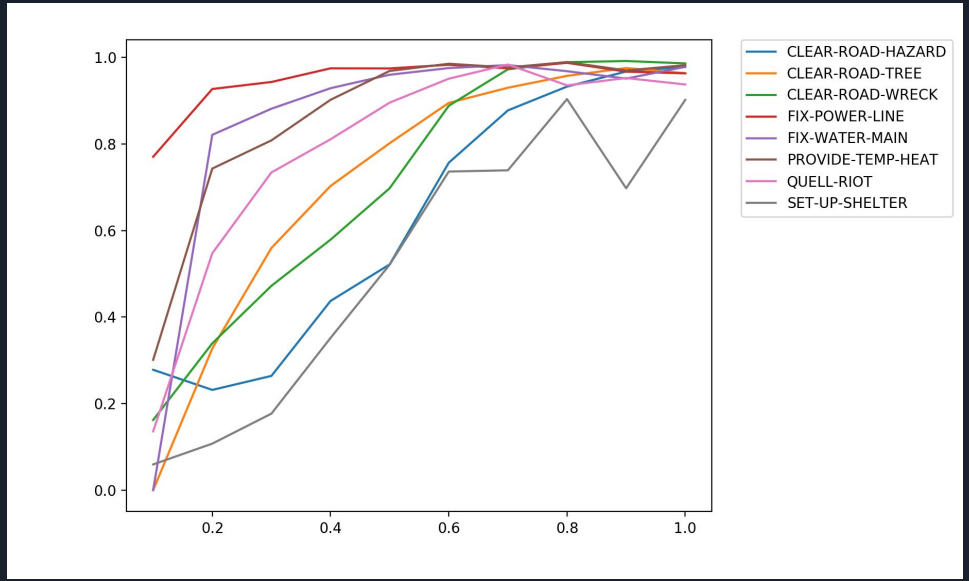
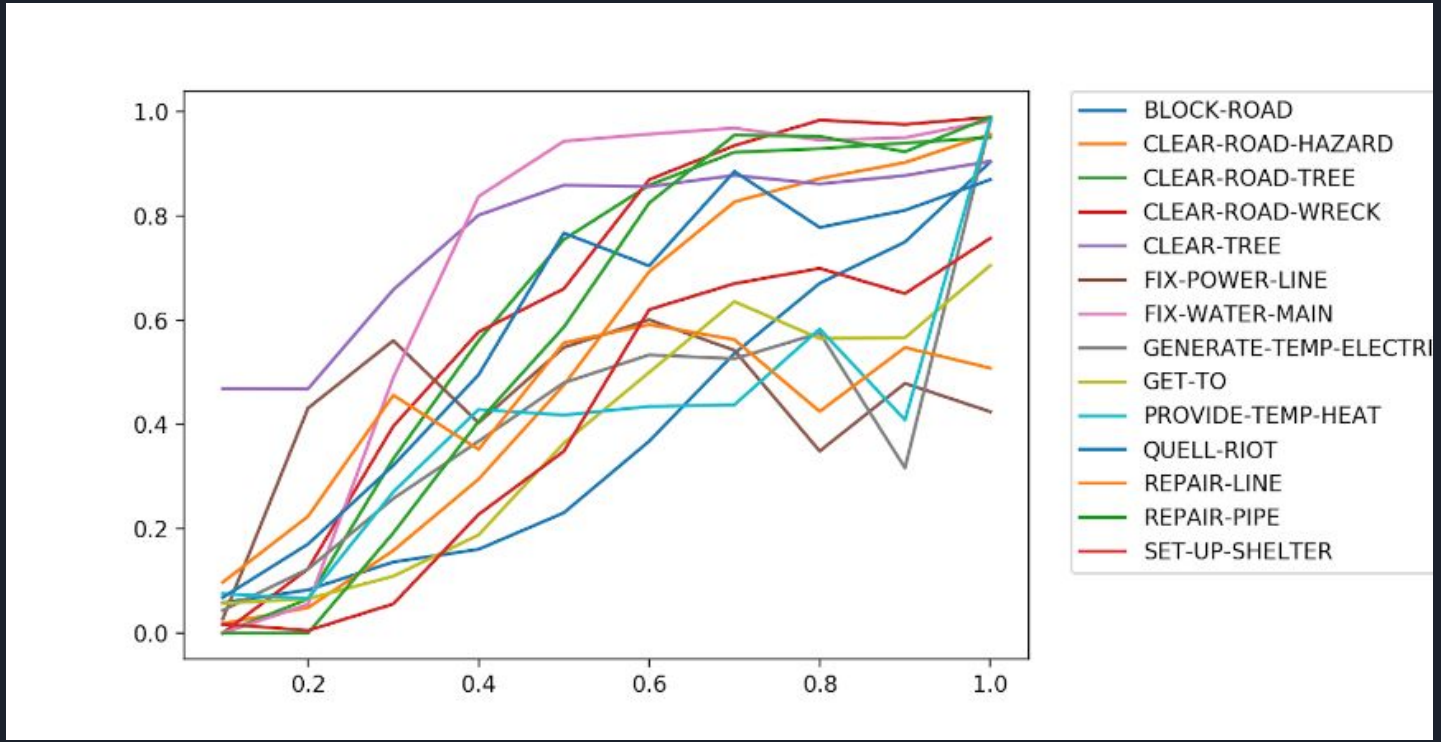


Figure 2: Training on all partial sequences

Adding subgoals



Project Timeline

Preparation

Understood the problem and read related papers.

Goal prediction

Implemented the first model to predict the goal.

Pattern recognition

Designed algorithm to learn patterns from the raw dataset.

Evaluation

Designed several evaluation methods, helping to optimize the model.

Live demo

Built a web based live demo to show how the system works

Future work

Preprocessing data, considering parameters, testing on other datasets

Learning goals using unlabeled data

- In real world project like taskport, manually labelling training data cost large amount of time and money. As a result, we would like to have a learning and predicting system based on unlabeled dataset.

- Using pattern recognition algorithms can help us dealing with the raw dataset.

New Pattern Learning

- Shares the same idea as Apriori Algorithm
- PrefixSpan (sequential pattern mining):
  a. Input sequence set S and a threshold T
  b. Generate length i prefix (start from 1)
  c. Count number of time the prefix show in sequence set S
  d. Select the prefix that have count larger than T,
  e. Delete all sequence that don’t have those prefix and i= i+1
  f. Recursively do Step b-e, until there are no new prefix generated

New Pattern Learning

- Ex : Threshold = 2
  - [NV - NV - CI - CRH - CO - NV], [NV - CT - NV - CI - CO], [NV - CI - CT -NV],
  - [NV - CI - NV], [CA - CA]
- Length 1 prefix: {NV : 4, CI: 4, CRH: 1, CO: 2, CT: 1, CA:1}
- Remain prefix: {NV, CI, CO}
- Remain Sequence for CI: (Do the same on NV, CO)
  - [... - CRH - CO - NV]
  - [... - CO]
- Length 2 prefix: {CI-CRH : 1, CI-NV: 1, CI-CO:2}
- Remain prefix: {CI-CO}
- Remain Sequence for CI-CO:
  - [... - NV]
- Stop

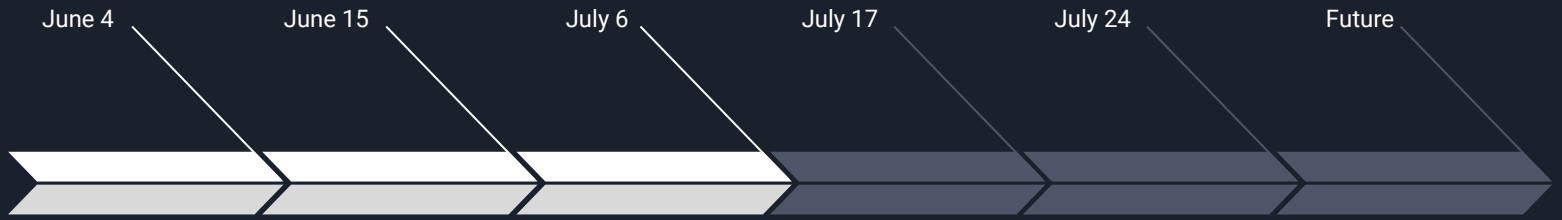
Output Patterns

- Distinct Patterns
  - NV - Place Cones - Clean Hazard - Pickup Cones
  - NV - Climb In - NV - Climb Out - Treat In Hospital
  - NV - Climb In - NV - Climb Out - Cut Tree
  - Load - NV - Unload - NV
  - Hook To Tow Truck - Climb In - NV - Climb Out - Unhook From Tow Truck
- Common Patterns
  - CALL - NV
  - CALL - CALL
  - NV - NV - NV -NV

New Pattern Learning

- Issues 1:
  - There exist similar patterns which only common words show up in different place
    - NV - NV - Climb In - Climb Out - Place Cones - Pickup Cones
    - NV - Climb In - NV - Climb Out - Place Cones - NV - Pickup Cones
  - Solution: Second round pattern recognition
    - Generate new common patterns from old common patterns
    - Get rid of common words
- Issues 2:
  - Definition of common words
  - Solution: Current only check the frequency of each words appears in patterns
    - Total Count / Total Number of words in all patterns
    - Patterns that contain the words / Total number of patterns

Project Timeline



Preparation

Understood the problem and read related papers.

Goal prediction

Implemented the first model to predict the goal.

Pattern recognition

Designed algorithm to learn patterns from the raw dataset.

Evaluation

Designed several evaluation methods, helping to optimize the model.

Live demo

Built a web based live demo to show how the system works

Future work

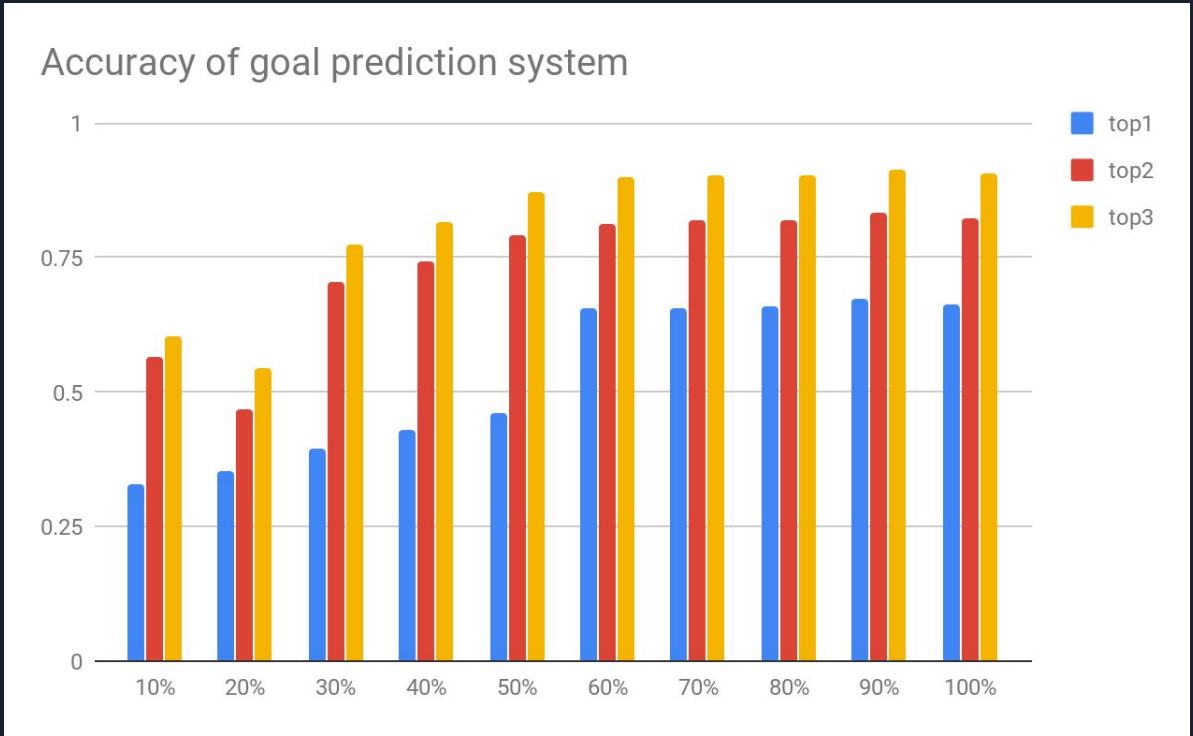
Preprocessing data, considering parameters, testing on other datasets

Future Actions Prediction

- Combination of New Pattern Learning System with Goal Prediction System
- Training Process:
  - New Pattern Learning System generate high frequency patterns
  - Divide the training data into different goals according to what pattern are contained in the data (assuming that one distinct pattern is one goal)
  - Feed the training data to the Goal Prediction System, and train a RNN model which could provide high accuracy predictions.
- User Using Process:
  - User type in a few actions
  - System predict what goal the user are trying to accomplish (return top N goals if necessary)
  - System predict future actions that could happen according to previous predict goals
  - Return the predict goal, future actions sequences and the recommended next one step action

Evaluation

Goal prediction system trained on unlabeled dataset

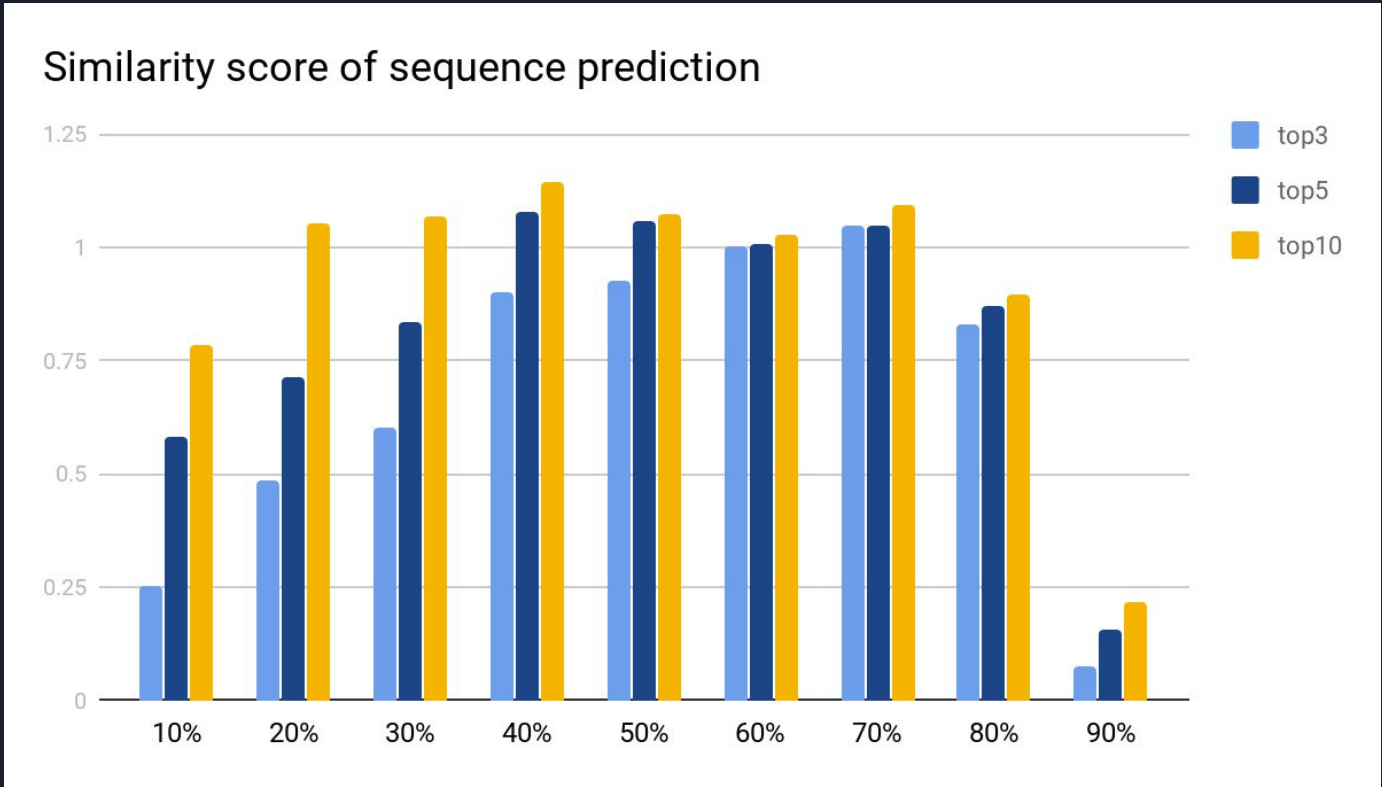


Ave goals a sequence belongs to: 2.016

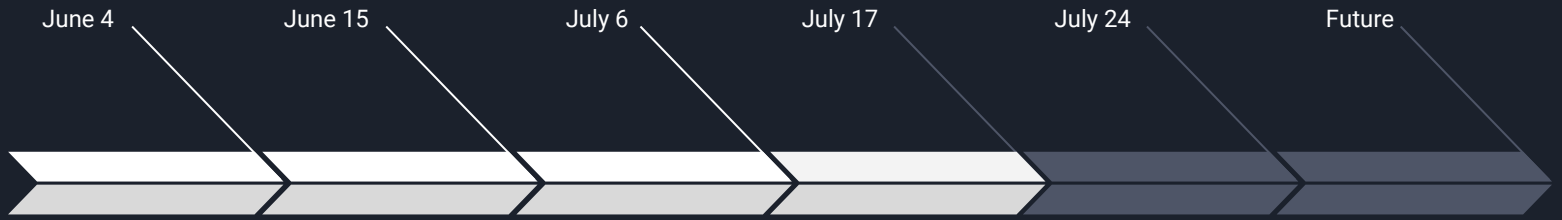
Accuracy = Num(correct goals predicted) / Num(total correct goal it belongs to)

Evaluation

Future Sequence Prediction



Project Timeline



Preparation

Understood the problem and read related papers.

Goal prediction

Implemented the first model to predict the goal.

Pattern recognition

Designed algorithm to learn patterns from the raw dataset.

Evaluation

Designed several evaluation methods, helping to optimize the model.

Live demo

Built a web based live demo to show how the system works

Future work

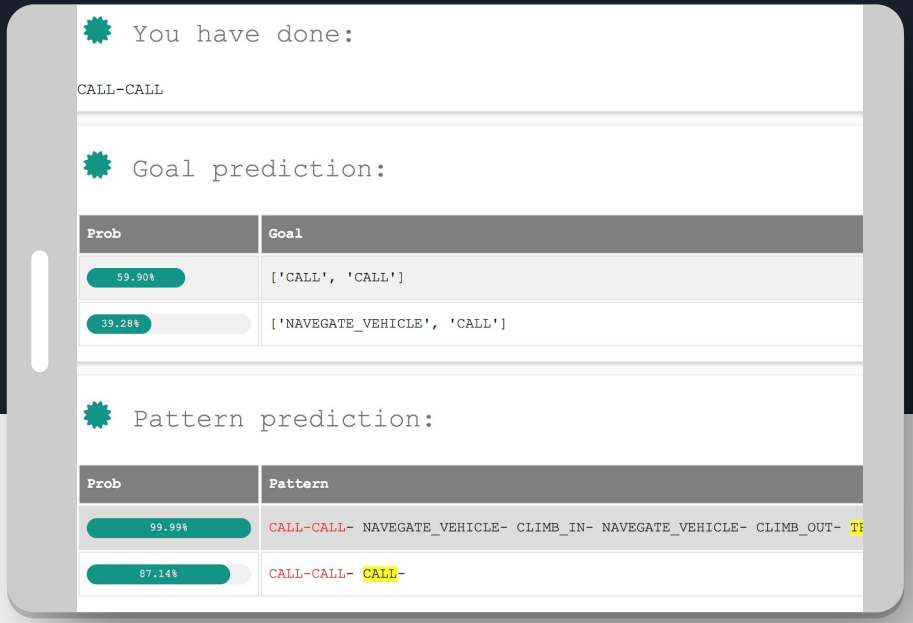
Preprocessing data, considering parameters, testing on other datasets

Introducing: Live demo

Showcase how our model works

This is an interactive demo, and it can:

1. taking an user input
2. processing the data
3. making predictions



- Actions done
- Key actions

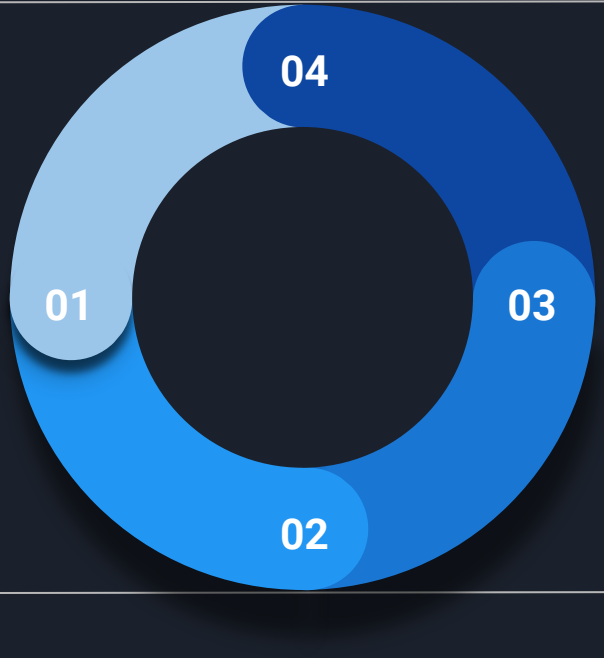
Live demo Cycle diagram

User input

Take an user input, sending it to server.

Parsing to model

Parsing the input string to a vector, feeding it to the model.



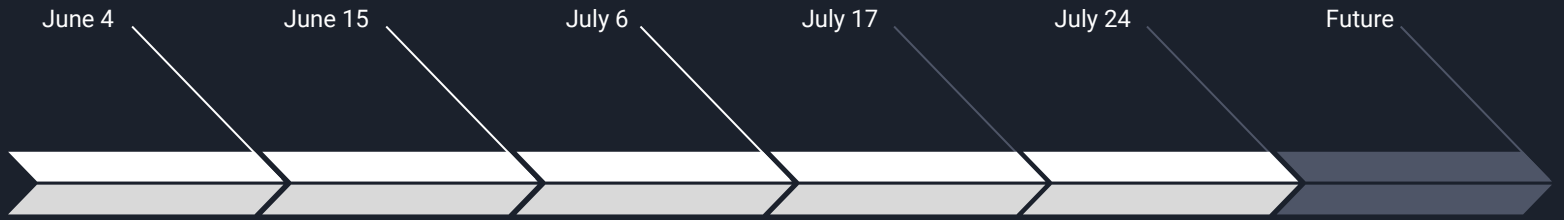
Make prediction

Making Goal, pattern, action prediction, getting in to next round.

Processing

Processing the data, calculate confidence and choose the optimal result.

Project Timeline



Preparation

Understood the problem and read related papers.

Goal prediction

Implemented the first model to predict the goal.

Pattern recognition

Designed algorithm to learn patterns from the raw dataset.

Evaluation

Designed several evaluation methods, helping to optimize the model.

Live demo

Built a web based live demo to show how the system works

Future work

Preprocessing data, considering parameters, testing on other datasets

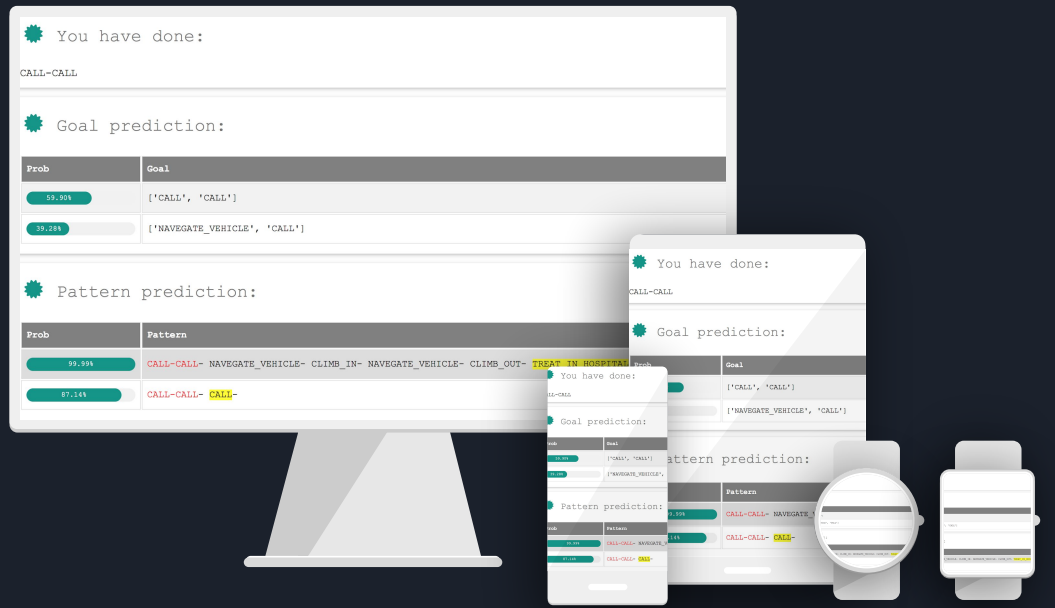
Future Work

- Pre-process on raw data before doing the pattern learning
- Test on Monroe dataset when noise data is added
- Test on Linux dataset
- Modify the systems to take parameters into account, while doing the Pattern Learning and Goal Prediction

References

- Krishna, Bala. "PrefixSpan: Mining Sequential Patterns by Prefix-Projected Pattern." International Journal of Computer Science & Engineering Survey 2.4(2012).
- Blaylock, Nate, and J. Allen. "Fast hierarchical goal schema recognition." National Conference on Artificial Intelligence AAAI Press, 2006:796-801.
- Tecuci, Dan, and Bruce W. Porter. "Memory Based Goal Schema Recognition." FLAIRS Conference. 2009.
- Nau, Dana S., et al. "SHOP2: An HTN planning system." Journal of artificial intelligence research 20 (2003): 379-404.
- Bagozzi, Richard P., and Susan K. Kimmel. "A comparison of leading theories for the prediction of goal-directed behaviours." British Journal of Social Psychology 34.4 (1995): 437-461.

Thank you!


You have done:
CALL-CALL
Goal prediction:
Prob
Goal
59.90%
['CALL', 'CALL']
39.28%
['NAVEGATE_VEHICLE', 'CALL']
Pattern prediction:
Prob
Pattern
99.99%
CALL-CALL- NAVEGATE_VEHICLE- CLIMB_IN- NAVEGATE_VEHICLE- CLIMB_OUT- TREAT_IN_HOSPITAL-
87.14%
CALL-CALL- CALL-