ATPT: Automate Typhoon Contingency Plan Generation from Text

Yifeng Zeng, Zhangrui Yao, Yinghui Pan, Wanqing Chen, Junxin Zhou, Junhan Chen, Biyang Ma, Zhong Ming

Northumbria University NEWCASTLE

UNIVERSITAS AMOIENSIS

SHENZHEN UNIVERSITY 1983

Abstract

Artificial intelligence (AI) planning models play an important role in decision support systems for disaster management e.g. typhoon contingency plan development. However, constructing an AI planning model always requires significant amount of manual effort, which becomes a bottleneck to emergency response in a time-critical situation. In this demonstration, we present a framework of automating a domain model of planning domain definition language from natural language input through deep learning techniques. We implement this framework in a typhoon response system and demonstrate automatic generation of typhoon contingency plan from official typhoon plan documents.

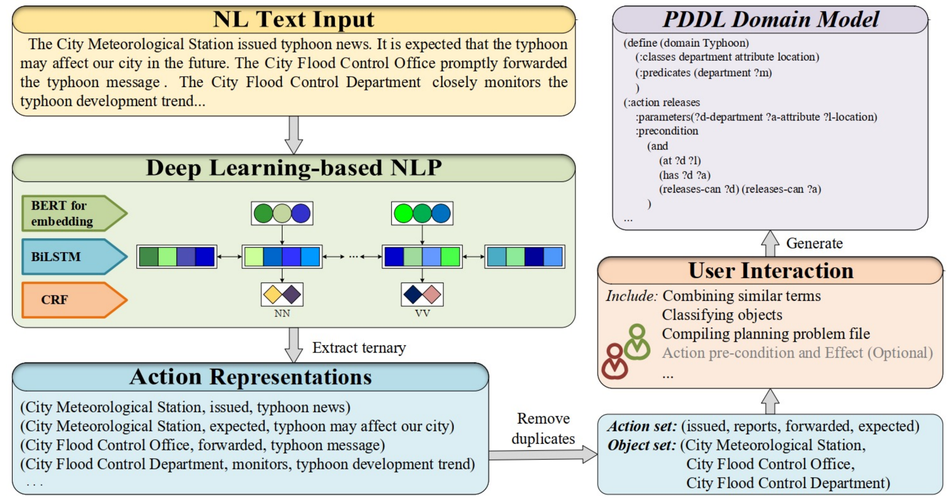
Research Problem

Typhoon response systems demand time-critical decision making, which expects a planning model to be implemented in an emergency mode. In particular, many inputs to build an AI planning model in order to generate a situation-aware typhoon contingency plan are presented in the form of text e.g. typhoon response manuals, a situational report or even a collection of messages from social media platforms. This needs an intelligent agent that is able to read natural language inputs and build an AI planning model accordingly. It challenges a fundamental research issue of automating an AI planning model, most of which focuses on learning the planning model from available data.

From Text To Plan Models

We present the end-to-end PDDL planning model development Framework.

- Receive an entire paragraph of text and convert the action extraction into a sequence labelling problem.
- Resort to BERT to process the inputs and use a BiLSTM- CRF model to learn an action representation with a triplet (subject, predicate, object) from the processed text.
- Implement a user interaction component where users have chance to refine the learned triplets. Then the refined triplets become the inputs to learning PDDL domain models. In addition, a problem file is specified in the user interaction and is compiled together with the domain model to build a PDDL model.
- Develop a typhoon contingency plan based on the learned PDDL model.



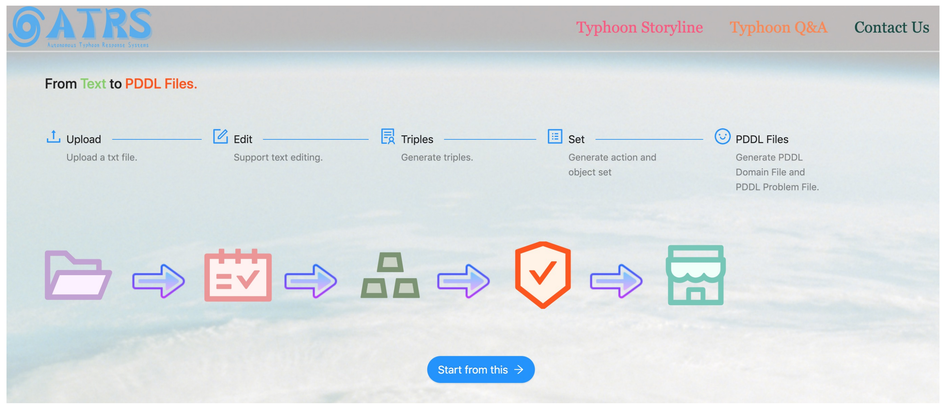
Conclusion

Our new learning framework makes a further step of widening the type of inputs for learning the PDDL model when a text document is provided to a typhoon response system. We have implemented it to generate a proper typhoon contingency plan in the autonomous typhoon response system. This work contributes into an automatic generation of planning models for intelligent agent planning and scheduling in intelligent systems. Learning PDDL domain models still faces many challenges and user interaction is needed to remedy the learning process in order to generate a precise PDDL model. We are continuously improving the PDDL model learning and providing more friendly user interactions in ATRS.

Autonomous Typhoon Response System

We implement a web-based autonomous typhoon response system (ATRS) that has the main functionalities of generating a typhoon contingency plan, plotting typhoon tracks and maintaining a knowledge graph of typhoon response systems.

The system first asks for a text document to be uploaded and then processes the document to retrieve the corresponding triplets using the deep learning framework. The identified triplets are presented to users who could add more fields into the list. Subsequently, the users can upload a problem file that joins with the learned domain file to compose the final PDDL plan model. The model is to be downloaded and run by some well developed PDDL solvers, which provides a typhoon contingency plan to the users.



In addition, we provide the comparison between our techniques and the latest domain model learning methods that is used for the purpose of generating a storyline. In the current format, our new framework has better performance than others, and can deal with general text inputs.