Software Engineering for AI

1. Introduction

My research focuses on applying machine learning (ML) methods to automated planning. Automated planning addresses the challenge of solving sequential decision-making problems by finding a sequence of actions—referred to as a *plan*—that leads from an initial state to a goal state.

I work with classical planning, which assumes a fully observable, deterministic environment with a set of states and deterministic actions. A common approach to classical planning is heuristic search, where a heuristic function estimates the distance from a given state to the goal. The task of a planner is to generate a plan, given an initial state, goal condition, and a set of applicable actions.

Over the past 25 years, the majority of state-of-the-art planners have relied on state-space search guided by domain-independent heuristics. In classical planning, a domain is an environment from which multiple tasks with different initial and goal states can be generated. A major challenge in modern planning systems is scalability. As task size grows, planners struggle to solve them due to combinatorial explosion, also known as state-space explosion. When heuristic estimates are inaccurate, solving large tasks within limited time and memory becomes increasingly difficult. Although heuristic functions are often computationally efficient, they can be complex and may not generalize well across all tasks. Their domain independence, while beneficial for broad applicability, means that they often ignore useful domain-general properties. For example, if there are two tasks with different initial states but the same goal state, domain-independent heuristics typically do not exploit any possible similarities between tasks.

In my research, I aim to overcome these limitations by using ML to capture domain-general properties of planning tasks. Given a specific planning domain, we train a model on a set of representative instances to learn generalizable knowledge about the domain. The learned model is then used to tackle much larger tasks than those seen during training. Our goal is that this domain-level knowledge enables effective generalization and improved scalability in planning systems.

2. Lecture principles

The first principle is *AI testing*. In my research, there are two stages: learning and planning. A model that performs well during the learning stage does not necessarily achieve good results in the planning stage. We typically optimize the learning model using a metric such as F1 score. However, we have observed that this metric alone is often insufficient to properly assess generalization from a planning perspective. Instead, we analyze how the model makes predictions for different types of states. For example, we visualize and identify which states are most challenging for our models. During the lecture, I considered how running our models on small training instances with slightly different initial states but the same goal states, and vice versa, could help us observe how these perturbations affect performance. This approach could provide insight into the robustness of the model.

The second concept is *addressing opacity*. Currently, I work with decision tree models and features from logical formalisms. In theory, both the model type and the feature language should contribute significantly to making the model more transparent (white box). In practice, however, we observe that the trees are very deep, and the features are long and complex. An additional challenge is that our ML algorithms should ideally scale and perform better than existing symbolic methods, which are already highly explainable and interpretable. Although the lecture did not mention specific methods applicable to my scenario, it reinforced the importance of this aspect. This factor is especially crucial if the model is to be deployed in production. While our current algorithms are not black boxes (they are somewhere between grey and white boxes), I believe that future work could lead to even more transparency.

3. Guest-Lecture Principles

Requrements Engineering. At first glance, it is difficult to see how this discipline could be applied in pure research settings. However, the most valuable lessons I have taken from it are the importance of structure, organization, and iteration. While these concepts may seem idealistic or even restrictive to those who value a purely exploration-driven approach to science, the reality—especially during a PhD—is that we must publish papers within limited time frames. This makes it essential to organize everything from daily routines to problem formulation.

Although the initial hypothesis often comes from a supervisor, it can change significantly or even become irrelevant as the research progresses. I would emphasize *iteration* as a critical aspect here: the ability to periodically step back and assess the overall direction of the project by evaluating contributions, novelty, and impact. In other words, we need to understand how our initial expectations align with the results we obtain, and decide whether to change direction, adjust the scope, or focus more deeply on specific results.

This process can be relatively straightforward when writing a single paper, but it becomes much more challenging over the course of a PhD, where one carefully develops ideas from one paper to the next while minimizing disconnects between them.

Behavioral AI Engineering. The second topic concerned human behavioral aspects in the modern AI landscape. To some extent, we are experiencing similar changes not only at the industry level but also in academia. When Google releases a paper like *AlphaEvolve*, it naturally raises questions about how scientific research should be conducted today. The process of scientific discovery is becoming increasingly automated. In our own workflow, we aim to automate tedious tasks, but now even our primary research activities are becoming candidates for automation. While some of this may still seem utopian or dystopian, there is certainly substance to these concerns. As always, these are problems that are easier to identify than to solve.

My main takeaway is the importance of increasing our awareness and understanding of these systems. As someone whose interest in mathematics has grown significantly in recent years, I believe it is essential that research continues to thoroughly analyze these complex systems. Otherwise, we risk creating massive (synthetic) neural networks that may ultimately be beyond our comprehension.

4. Data Scientists versus Software Engineers

I agree that there is a clear difference between two disciplines. As mentioned in the book, in traditional software engineering, the engineer knows exactly what function needs to be implemented, and this function can often be hard-coded. The input to this function is also well understood and carefully studied during problem formulation. Additionally, unit tests are designed to continuously verify the correctness of the implementation.

How is a function implemented in ML? In this case, we do not know the exact form of the function in advance. The primary goal of ML is to learn a function from a set of data points. We may have a general idea of what the function should do, but its precise form is unknown. Essentially, the engineer has to write software for a function whose form is undetermined. This *undetermined* aspect highlights the role and value of the data scientist. The nature of the data, the process of learning a function (or a model) from it, its testing and verification are all integral components that are not clearly defined in traditional software engineering. All of this now requires a significant amount of mathematics, which would typically be abstracted away in traditional software engineering.

From my limited experience exploring the job market before starting my PhD, it seemed that most companies were searching for *unicorns*—people capable of designing robust data pipelines, building strong models, and possessing MLOps skills to deploy and continuously monitor those models. At least, that was my impression at the time. Why is this the case? First, it is cost-effective for companies. Second, it is a low-risk investment, as companies (before the LLM boom) were cautious about adopting ML. Finally, they hope these unicorns can eventually build strong teams and have a comprehensive understanding of all critical aspects of ML engineering. I do not have much information about the current job market, but I believe this pattern may still persist. One example is the AI engineer role, which I have encountered several times.

If we set aside the profit-driven motives of businesses and consider the nature of these roles more deeply, the *T-shaped* approach mentioned in the book offers a good balance between two extremes. Both disciplines can learn from each other with the focus on mutual understanding and complementarity rather than replacement.

5. Paper analysis

5.1 Paper 1: Exploring Hyperparameter Usage and Tuning in Machine Learning Research

The core idea of the paper is to examine how hyperparameter tuning procedures and values are reported in modern ML research, and to highlight their importance. The paper shows that, in commonly used ML frameworks (such as scikit-learn, PyTorch, and TensorFlow), only a few hyperparameters are typically tuned by researchers. Furthermore, it points out that many papers do not report the values used or the tuning procedures applied. Finally, it notes that most papers that do report their tuning procedures rely on rather conservative techniques, such as random search, grid search or manual tuning, despite the availability of more advanced methods.

In my research, I perform hyperparameter tuning to obtain models with good generalization performance. Beyond generalization, I also have to consider how the learned models will be used during the planning phase. In particular, the inference time of the model is directly affected by the chosen hyperparameter values. For example, if a learned tree is very deep or has a high number of leaf nodes, it will have many learned rules to evaluate, which can be prohibitive since planning is usually performed within a limited time frame.

As the paper emphasizes, hyperparameter tuning is an integral but sometimes overlooked aspect of ML systems. As I have also observed in my own research, many production-oriented ML projects can benefit greatly from a thorough analysis of hyperparameter values and their tuning. In such projects, both generalization and inference time are critical properties. One key finding from the paper is the need to move beyond conservative tuning strategies. The community could benefit from adopting well-established and proven tuning frameworks. At the same time, it is worth noting that existing ML frameworks already offer a wide range of tuning strategies, and carefully applying these can already improve the learning pipeline. Another important issue is the significant computational resources required to tune complex models, such as neural networks. Even in my own work, tuning just a few hyperparameters with a large range of values can be very time-consuming. Therefore, I can understand the tendency to rely on default values, even when tuning large models.

The ideas presented in the paper reinforce the importance of hyperparameter tuning in my research. There are three main aspects from the paper that I could apply to my own work. First, I should revisit the default values I have left unchanged and reassess their impact. For example, I recently observed a significant performance improvement by adjusting a single hyperparameter. Second, I should focus on more thorough reporting of tuning strategies, chosen values and their impact. Reporting the effect on generalization is especially important, as readers can be interested in how performance varies across different hyperparameter combinations. Finally, it is worthwhile to explore more advanced tuning strategies. While established frameworks already provide many options, investing time in analyzing state-of-the-art techniques could be beneficial in the long term.

5.2 Paper 2: Investigating Issues that Lead to Code Technical Debt in Machine Learning Systems

Similar to the first paper, this paper also focuses on raising awareness about existing problems in ML systems. The paper analyzes the origins of technical debt (TD) in modern ML pipelines. The extensive study reveals that phases such as *data collection* and *data pre-processing* contribute significantly to TD. In real-world use cases, data collection should be thoroughly organized to avoid incomplete or biased datasets. While some incompleteness in real-world data is inevitable, it is still crucial to document the reasons for missing data and account for them in later modeling phases. Data pre-processing is identified as the most significant contributor to TD in ML systems. Even the misuse or neglect of simple strategies like rescaling and rebalancing is a commonly occurring problem.

From my perspective, data collection and model evaluation are particularly challenging aspects. I work with toy domains that are not fully representative of real-world data. For example, one such domain involves tasks with multiple trucks, packages and target locations arranged in a grid. The goal for each task is to generate a plan—a sequence of actions—to transport packages to their target locations. The number of possible tasks in this domain is infinite. For instance, there could be *n* trucks, *m* packages, and grids of various *(x, y)* dimensions. This raises the question of how to generate data and, specifically, how many tasks with which properties are needed to achieve generalization.

Model evaluation is also challenging. Ideally, the generalization of learned models should translate to generalization in the planning phase, but this is not always the case. In our work, we try to determine whether new evaluation metrics, in addition to standard ones, are needed to better assess our models.

Unfortunately, the authors of the paper focus primarily on identifying issues rather than proposing solutions. This is understandable, given that many lines of research are dedicated to different phases of the ML pipeline. One key lesson from this paper for large-scale AI projects is not only to be aware of TD but also to be proactive in implementing strategies to mitigate it. From my experience, domain expertise is critical for understanding how to approach each stage. Even though I work with toy planning domains that do not fully replicate real-world scenarios, I am in constant communication with my supervisors to better understand the properties of these tasks and how to approach them in the learning phase. This ongoing dialogue should ultimately contribute to better data pre-processing and model evaluation strategies.

As mentioned, I am currently focused on understanding how data collection should be implemented in my research to build a robust learning model. The paper also highlights the importance of model evaluation, an area I have studied extensively. Feature selection, as part of data pre-processing, is also identified as a highly relevant issue, which I frequently encounter in my own research. However, it is easy to feel overwhelmed by the sheer number of factors to consider. I believe that maintaining an incremental and patient approach to addressing these challenges is essential to avoid the trap of trying to solve everything at once.

6. Research Ethics & Synthesis Reflection

I tried to find papers that were relevant to my research. Initially, I looked at the page listing all accepted papers and discovered that there was also a theme-based grouping. This provided me with more context and a better understanding of the conference.

I am not knowledgeable enough about the overall topic to identify potentially misleading titles. However, as with many AI-related conferences, this one also placed significant emphasis on large language models.

As the lecturer mentioned in the final minutes of the presentation, the purpose of this exercise is for us to reflect on and demonstrate what we have gained from the course. Since this field of research is different from my own, I saw this as an opportunity to broaden my perspective and learn about other fields. My interest in drawing connections between different fields also influenced how I approached and tried to understand discussed issues.