Lagos, C. I., Herrera, R. F., Mac Cawley, A. & Alarcón, L.F (2024). An AI Copilot for Make-Ready Planning in the Last Planner System. In D. B. Costa, F. Drevland, & L. Florez-Perez (Eds.), *Proceedings of the 32nd Annual Conference of the International Group for Lean Construction* (IGLC32) (pp. 365–376). doi.org/10.24928/2024/0155

AN AI COPILOT FOR MAKE-READY PLANNING IN THE LAST PLANNER SYSTEM

Camilo I. Lagos[1], Rodrigo F. Herrera [2], Alejandro Mac Cawley[3] and Luis F. Alarcón[4]

ABSTRACT

Many challenges in partial Last Planner System implementations can be attributed to the underutilization of Make-Ready Planning, although other factors also play a role. Failing to identify constraints in time to prevent Reasons for Noncompliance (RNCs) decreases short and long-term performance. Reducing the complexity of identifying, registering, and managing constraints systematically was found as a critical improvement opportunity. This research proposes the use of an artificial intelligence (AI) recommender system to facilitate constraint identification and RNC prevention. The system employs Large Language Model (LLM) embeddings to represent new task descriptions and find the most similar previously seen tasks. Subsequently, it fetches the set of constraints and RNCs belonging to these past tasks, represented in the embedded system, and uses it to produce three prioritized recommendations. Finally, the selected recommendations are categorized using Machine Learning Classification. The model was able to provide three sound recommendations for 69% of tasks and yielded a 60% relative improvement compared to a rule-based frequent pattern probabilistic system. The results pose three benefits for LPS practitioners: Reducing the effort needed to identify and register constraints, alerting probable RNCs needing to be prevented, and enriching data registration, allowing it to be used in future knowledge management.

KEYWORDS

Last Planner System, artificial intelligence, large language models, constraints, reasons for noncompliance.

INTRODUCTION

The Last Planner System (LPS) systematizes the transition from long-term Master Planning to mid-term Lookahead Planning and subsequent Short-term Planning through the Make-Ready Process (Kim, 2019). This process, known as the Make-Ready Process, involves breaking down upcoming activities into manageable work packages typically planned over a variable period that can extend beyond the often-cited four-to-six weeks, especially when dealing with complex materials or detailed design information from engineers, then screened to identify and remove constraints, before selecting constraint-free tasks to commit in the short-term plan and controlling the short-term compliance in search of Reasons for Noncompliance (RNCs)(Ballard & Tommelein, 2016). The process of identifying, managing, and removing constraints to

[1] PhD Candidate, School of Engineering, Pontificia Universidad Católica de Chile, CONXAI Technologies GmbH, colagos@uc.cl, orcid.org/0000-0002-0648-0039

[2] Assistant Professor, School of Civil Engineering, Pontificia Universidad Católica de Valparaíso, rodrigo.herrera@pucv.cl orcid.org/0000-0001-5186-3154

[3] Associate Professor, Dept. of Industrial Engineering, Pontifica Universidad Católica de Chile, amac@ing.puc.cl, orcid.org/0000-0002-4848-4732

[4] Professor, Dept. of Construction Engineering and Management, Pontifica Universidad Católica de Chile, lalarcon@ing.puc.cl, orcid.org/0000-0002-9277-2272

produce a workable inventory of constraint-free tasks is called make-ready planning (MRP)(Jang & Kim, 2007) (Ballard & Howell, 2003).

Systematic MRP allows to effectively select constraint-free tasks for the short-term plan and carry out short-term commitments efficiently (Javanmardi et al., 2020). Effective constraint removal, captured by the Percent of Constraints Removed (PCR), has a direct positive correlation with sustained short-term compliance, measured by the mean and standard deviation of the Percent Plan Complete (PPC) (Lagos & Alarcón, 2021a). Nevertheless, MRP is one of the least mature components in LPS implementations (Samad et al., 2017) and transversal studies have found that over 70% of constraints are identified less than two weeks before task execution (Pérez et al., 2022) and 55% of constraints are removed later than required (Bellaver et al., 2022). Despite the growing interest in applying AI to project management, there remains a significant gap in its integration with established methodologies like the Last Planner System. Particularly, there is a lack of research on how AI can be effectively implemented to predict and manage task constraints in real time, a gap our study seeks to fill by developing an AI copilot designed for Make-Ready Planning. The primary objective of this study is to develop and validate an AI-based recommender system designed to enhance the Make-Ready Planning process within the Last Planner System by systematically identifying potential constraints and preventing Reasons for Noncompliance (RNCs). The use of past knowledge is powered by an Artificial Intelligence algorithm fitted to a dataset of 1,000 tasks retrieved from 30 projects, the 3,300 constraints identified during MRP, and 4,100 RNCs impacting execution. The algorithm uses the relationships found between the tasks, constraints, and RNC categories and descriptions to find common patterns. These associations allow to predict potential constraints when prompted. Its implementation in LPS support systems can facilitate practitioners to identify and register constraints systematically and more efficiently.

LITERATURE RESEARCH

FACTORS LIMITING MRP AND THEIR IMPACTS

While MRP is key in securing a stable flow of tasks into the short-term plan, thus, increasing short-term plan reliability (Kim, 2019), is often one of the least systematized components of LPS (Lagos et al., 2022). Effective MRP requires allocating time and effort from all last planners to identify constraints with subsequent time in advance, registering in a standardized manner, committing their removal to habilitate task execution, and monitoring their status regularly so that the workable inventory can be updated (Bellaver et al., 2022). Prior transversal LPS adoption studies have found that partial LPS implementations fail to systematize the MRP (Hunt & Gonzalez, 2018). These surveys have found that while short-term planning and control is a widely adopted practice, the selection of short-term tasks is carried out without proper use of a workable-task inventory, active monitoring of constraint status, registering constraint identification and removal, or late constraint identification (Salling et al., 2023).

A Danish survey with over 1,600 responses (Salling et al., 2023) showed that 45% of teams did not know "how the project plan looks 1 month from now" and 44% did not know their upcoming tasks one month in advance. A similar study addressed subsequent increments of IT support for MRP in 162 Brazilian projects (Bellaver et al., 2022). The first iteration, using a cloud-based constraint status table in 38 sites, found that most teams failed to identify and manage constraints using the tools provided, despite it automating constraint management indicators. The second added constraint identification surveys and alerts in 92 sites. These were adopted by only half of the projects and only one-third of the constraints included removal commitment dates. The third iteration added automated guides and a standardized flow, tested in 32 projects. The guiding checklists and the requirement to fulfill them before continuing with the Lookahead Plan improved collection significantly ensuring all constraints were committed and reported. While the study observed an average increase of 50% in identification, 55% were identified late and failed to be removed when required for task execution.

On the other hand, a study covering a sample of 71 high-rise building projects employing IT support for LPS across their entire execution (Pérez et al., 2022) found that the constraint removal time of successful projects was 48% lower than in projects that failed to accomplish the scheduled completion. The same study found statistically significant correlations between constraint identification and removal times, the number of constraints identified, the PCR, and PPC. Finally, another study capturing over 24,000 constraints across 3,700 weeks from 69 projects (Lagos & Alarcón, 2021a) found that teams identify them on average 10 running days before execution and take an average of 16 days to remove them. The same study, which clustered the projects into 37 successful and 32 failed ones according to their schedule accomplishment outcome, observed a 47% difference in constraint removal efficiency and 42% in constraint planning efficiency among the clusters. The differences increased by over 150% when excluding projects with mid-schedule-accomplishment outcomes.

IT-SUPPORT SYSTEMS FOR LPS

The support functions of project management software can be categorized into: Assigning overhead rates to tasks; Job sizing adjustments of overheads; Predictive tender modeling; Work structuring and scheduling; Forecasting resources demand; Unit-based reporting; Load balancing; Task completion monitoring and measuring; and Material laydown planning and logistics (Costa et al., 2023). In addition, the key value propositions of software support have been characterized as Providing systems integrations; registering and consolidating information; standardizing workflows; facilitating analysis; automating updates and notifications; and improving communication and information sharing (Stevens & Olayiwola, 2023). LPS software focuses mainly on the support functions comprised of work structuring and scheduling, and task completion and monitoring (Lagos et al., 2019). Based on the literature findings, LPS support software can be characterized as encompassing a workflow of the master, Lookahead, and short-term planning, is immediately available to other users, consolidated over time, and summarized into reports or indicators captured through short-term control (Daniel et al., 2019; Dave et al., 2016; Faloughi et al., 2014). Therefore, forecasting upcoming Master Plan activities through Lookahead Planning and facilitating resource planning and allocation to bring them into the short-term plan (Heigermoser et al., 2019), as well as allowing short-term control cycles to report the accomplishment of the plan and RNCs are fundamental requirements of LPS-supporting software (Sbiti et al., 2021).

A key distinction is the extent to which they integrate constraints into the workflow. While some systems offer detailed means to register constraint identification, categorize, plan, commit, and report their removal (Lagos et al., 2019), others limit the registration of constraints to representing a blocker status in a task (Faloughi et al., 2014). There are also similar differences in the way in which constraints impact the collaboration workflow. More thorough systems incorporate logic to restrict the movement of tasks from the Lookahead Plan to short-term commitment, while others are limited to the collection of constraints and their status almost independently from task planning, commitment, and execution (Warren, 2019). The minimum viable information found across the systems corresponds to the link between a task belonging to a hierarchical and sequential structure in the WBS, its constraints, and the RNCs experienced by it during its execution (Pérez et al., 2022; Warren, 2019). Tasks can be better characterized considering their dependencies to mother-level activities in the WBS, as sequential pre-requisites, while most RNCs could be considered unforeseen constraints, and system categorization functionalities allow the identification of common types of constraints and RNCs associated with those tasks. Finally, depending on the connections captured by the different database architectures employed, the impact of constraints and RNCs on a given task can be estimated using the schedule differences between the planned and actual start and completion of the task (Lagos & Alarcón, 2021b).

VALUE OF PAST PROJECT INFORMATION FOR PROACTIVE PLANNING

Standardization, knowledge management, data-driven decision-making, benchmarking, and continuous learning are key Lean Construction practices (Castillo, 2015; Kifokeris, 2021). The inherent characteristics of LPS help promote standardization of workflows, data-driven decision-making, and continuous learning and are enriched with the use of IT support to systematize data collection and automate reports (Lagos et al., 2019). Various scientific contributions have shown the value of benchmarking significant LPS project samples to generate and consolidate knowledge (Kim, 2019; Lagos et al., 2019, 2022; Pérez et al., 2022), but its practical use in the industry remains lacking due to the lack of system integrations with data science and Machine Learning(Gondia et al., 2020; Shehab et al., 2022). Such IT contributions would provide value to practitioners in four ways: Information indexing; alerts and forecasting; prescriptive recommendation; and process automation (Cisterna, Lauble, et al., 2022; Cisterna, Seibel, et al., 2022; Kecman, 2001).

Indexing deals with using patterns and similarities found across data to provide structure. For example, unsupervised clustering algorithms can help differentiate groups of projects with different performances and identify quantitative thresholds to divide or benchmark them (Lagos & Alarcón, 2021a). Also, supervised classification algorithms can be trained to automatically sort information based on patterns learned from past data (Y. R. Wang et al., 2012). Among other uses, they can automatically categorize and/or prioritize constraints and RNCs (Lagos & Alarcón, 2021b). Providing alerts and forecasting is another use of the patterns learned through past data observation. For example, fitting a function to explain the behavior of a repetitively measured variable can help forecast the trends of metrics such as the Schedule Performance Index (SPI). A regression prediction model can allow to estimate of the future SPI employing an array of performance variables already captured by LPS such as the PPC, PCR, and RNCs (Jang & Kim, 2007).

Prescriptive recommendation can also employ predictive regression and classification systems to determine or maximize a future state (Wong, 2004). For example, Machine Learning prescription could recommend the most likely constraints or RNCs to be experienced by a task, given its characteristics, recommend corrective actions given a task and RNC tuple or help identify the tasks from the workable inventory with the highest chance of completion in the short-term plan (Shehab et al., 2022). Finally, process automation is simply the use of indexation, forecasting, and prescription to streamline repetitive processes based on previous patterns, predicting the next step, and providing a recommendation accordingly (Cisterna, Seibel, et al., 2022). Since ML systems can employ advanced algorithms and high computing power, they are more likely to identify underlying patterns invisible to a human practitioner and employ them in their predictions (Cisterna, Lauble, et al., 2022).

APPROACHES TO AID CONSTRAINT IDENTIFICATION AND RNC PREVENTION

The literature research yielded three alternative ML approaches to aid in predicting the most likely constraints and RNCs faced by a project task: Fixed rule-based systems; extrapolation models, and generative models (Cisterna, Seibel, et al., 2022; Hatoum & Nassereddine, 2023; Oprach et al., 2019). Fixed rule-based models employ logical conditions and weights to generate predictions, which remain fixed after the model creation. Consequently, these algorithms cannot extrapolate to unseen dimensions or variables (Feng et al., 2019; Wong, 2004). For example, an expert system trained to classify text based on a predetermined set of keywords cannot adapt to new keywords for making predictions.

Extrapolating and reasoning models introduce a degree of flexibility by extrapolating the "meaning" of a token. In NLP, words and statements are the tokens, represented as vectors elucidating their context in an extensive corpus of keywords (Nie et al., 2020). If an unknown word is encountered, the system leverages other embedded tokens to find similar statements. "Synonyms" of the unknown component allow to extrapolate meaning. The extrapolations can range from embedding exact words to high levels of abstraction through stemming, combining, and transforming (Grohe, 2020; Nie et al., 2020). Their robustness depends directly on the size of the corpus and the number of examples used for training, as the embeddings essentially consist of a set of expert rules and transformations applied to the text.

Generative models enhance prediction flexibility by enabling the "creative" use of embedded similarities to generate new text. Systems like Chat GPT employ transformers, which, in turn, utilize embeddings (Hatoum & Nassereddine, 2023). A transformer functions as an encoder-decoder system, to produce and fetch tokens. A large embedding system transforms these tokens into vector representations, preserving their core "meaning" and use. The decoder, trained on a substantial number of input-output statements, utilizes the embedded representations of tokens to select the best set of output tokens for an appropriate response (N. Wang & Issa, 2023). Finally, the degrees of flexibility gained by moving to more advanced systems pose a trade-off between robustness and precision. Robustness depends, among other factors, on the size of the corpus employed, or in other words, the amount and variety of texts employed to train them. As the amount of the corpus increases, so does the embedded space, and, hence, there is a higher chance of finding similarities with words or statements outside the realm of the problem at hand (Nie et al., 2020). Hence, the precision decreases as the predictions obtained might not correspond to constraints and RNCs commonly experienced by LPS projects.

METHODOLOGY

This research employed the Design Science Research Methodology (DSRM) (Venable et al., 2017). DSRM empirically evaluates a solution artifact to validate it and identify key benefits. The artifact resides in the intersection between the Problem Space and the Solution Space, which contrasts the state-of-the-art in the body of knowledge to the state of practice. The body of knowledge captured by the literature research can be summarized as follows:

- Problem Space: Constraint identifying complexity causes late identification and misses. Registering and managing efforts then decreases MRP effectiveness. IT support systems facilitate capture and future use to a limited extent.
- Solution Space: Tasks constraints and RNCs can be retrieved from IT support. ML can detect underlying relations and use them in similarity analyses. LLM embeddings allow similar tasks, and their constraints, and RNCs can retrieved, embedded, and used to produce a new set of recommended constraints for this new task.

The following constructs were produced employing DSRM:

- Solution concept: An input-output predictor producing a prioritized set of recommended constraints, categorized based on their similarities with previously existing ones.
- Proof of Concept (POC): Using embeddings to represent tasks, constraints, and RNCs; K-Nearest-Neighbours to find similar tasks and fetch their constraints and RNCs. Then, employing their embeddings to provide three prioritized and categorized constraints.
- POC validation scope: AI predictions against a deterministic expert prediction model.

DATA COLLECTION

30 projects were randomly selected from a universe of 110 Chilean high-rise building projects that used the same IT support during their complete execution scope. The sample represented almost 30,000 unique construction tasks, filtered to obtain a rich sample of task-constraint and task-RNC (input-output) tuples, based on the following conditions:

- Tasks are limited to a set of 12 superstructure framing categories.

- Exclude tasks without at least one RNC belonging to an MRP-preventable category.
- Exclude tasks that did not contain at least six constraints and/or preventable RNCs.
- Exclude tuples with inputs or outputs with less than five unique stemmed keywords.

Limiting task selection to those with at least six constraints and/or preventable RNCs, as well as five stemmed keywords was required to ensure sufficient data to train, test, and compare the AI and expert systems. A lower threshold could result in failing to predict at least three new constraints. The criteria produced almost 3,000 tasks and 21,000 tuples. The inputs were represented by the name of the task and its mother activities, up to two levels above and excluding the mother activities representing the building's floor levels. The outputs were represented by the category and the description of the constraint or RNC. To ensure clarity, we define 'Mother activities' as tasks that are hierarchically superior or precede the current task in the project schedule. These are activities that need to be completed or significantly progressed before the current task can commence. For example, in the case of constructing a building, if the current task is 'Pouring concrete for the second-floor slab,' its mother activities might include 'Completion of all first-floor structural worksA random sample of 1,000 unique tasks, linked to 3,300 constraints and 4,100 preventable RNCs was retrieved. The input (task categories) and output (constraints or RNCs) taxonomy are presented in Figure 1.

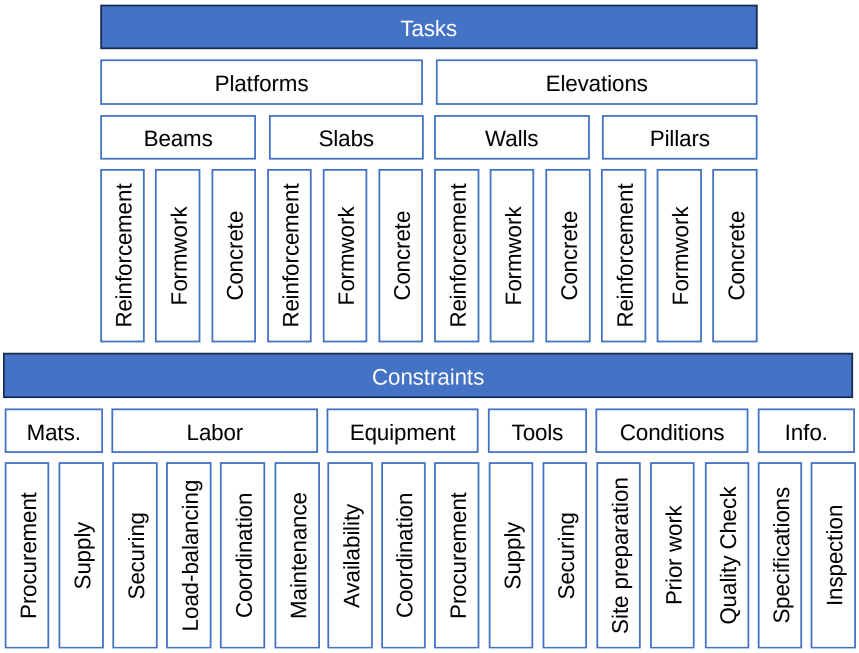

Figure 1. Task and constraint recommendation taxonomy (categories)

AI MODEL

Figure 2 describes the system. The original texts are embedded using the Word2Vec model available in the Python Gensim Library, with default parameters and excluding the use of multiword Ngram embeddings. The vector size was limited to 100 dimensions to avoid overfitting and improve performance. The embedded texts are categorized into input and output statements. A Euclidian distance algorithm computing the sum of squared differences between NumPy arrays is used to measure embedded text similarities for inputs and outputs respectively. A Naive-Bayes text classification algorithm predicts the lowest-level category of any given new task, and the classification results are used to slice the 1,000 task descriptions in the embedded space by removing all who do not belong to the same L2 group as the predicted category. This slicing helps improve performance and avoid unrepresentative similarities in future steps.

Afterward, the new task description enters the embedded space containing only the filtered tasks according to its predicted category. Subsequently, a K-Nearest-Neighbors algorithm fetches the 50 tasks with the minimum Euclidian distance. The ground truth data is then used to describe the constraint descriptions present in all tuples containing the 50 most similar tasks. The embedded representations of these past constraint descriptions are used to predict the three most likely new constraint recommendations. The recommendations are limited to constructing text descriptions using a maximum of five tokens. Since the constraint categories were included in the descriptions when inputting them into the embedding system, the produced outputs are most likely to contain one of the first-level constraint categories (materials, labor, equipment, tools, conditions, and information), and one of the second level categories (16). The remaining tokens are used to construct the specific description, most likely, providing a verb, noun, and adjective to construct the recommended constraint.

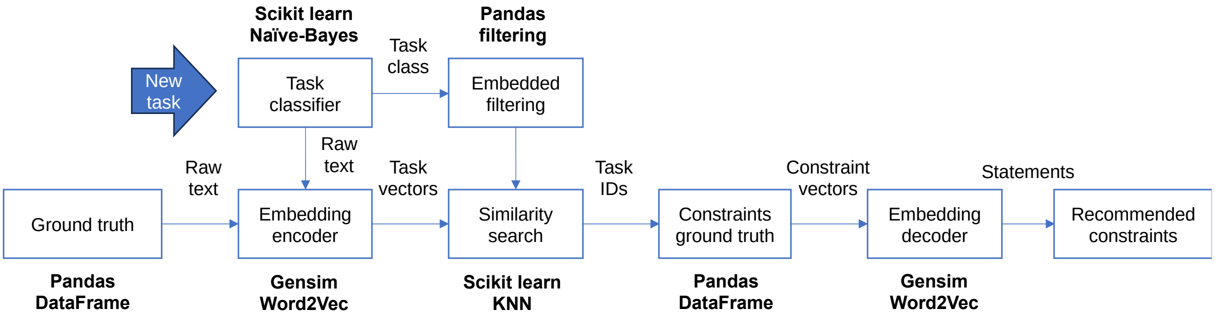

Figure 2. AI model pipeline

MODEL COMPARISON

A rule-based model was employed to benchmark the performance of the recommender system. Since fixed rules cannot be applied to natural language descriptions, the system uses the task-category prediction given by Naive-Bayes as an input to predict constraint categories. The model employs frequent pattern recognition to derive the most-likely class to enter a known set and turns these probabilities into rule-propagation tree using the FP-Growth algorithm. The FP-Growth algorithm, like A-Priori, is an pattern mining algorithm fitted to cases referred to as the supermarket cart problem (Feng et al., 2019). It predicts the most likely N+1 element to be added to a known bundle. The bundle of unique task categories and their linked constraint and RNC categories (ground truth) are passed as association rules. Since tasks contain at least six unique constraints and/or RNCs, the association rules can rank up to six predictions for each task category, the latest being the most frequently observed full set. In this implementation, the FP-Growth predictions were not limited to bundles composed of unique predicted categories, since a task can contain two or more constraints belonging to the same category. Also, the order of the factors in the ground truth bundles did not alter the resulting rules, since the predictions were based on the conditional probability of a new constraint appearing given the frequency of the set's unique components (i.e. constraints and RNCs), structured into a probabilistic frequent patterns tree. As a result, all tasks in the same category received the same ranked sets.

The expert system was implemented employing the FP-Growth model available in the Python MLX tend Library. Both the AI and rule-based systems were trained and tested employing the same 80%-20% sample split. Also, both used the same task category prediction obtained from the Naïve-Bayes algorithm as an input. The cases where the task classification was incorrect were excluded from the subsequent model performance comparison. The outputs produced by both models were contrasted against the ground truth categories of each of the constraints associated with the test tasks to measure the performance. Each model was assigned one point for each predicted constraint present in the ground truth and deducted one point for each prediction that was not present in it.

RESULTS

First, the Naïve-Bayes classification algorithm produced 182 correct task classifications out of the 200-task category ground truth samples employed in the testing set, yielding a recall of 89%. The recall is calculated as the number of correct predictions divided by the test set size. Table 1 presents the prediction recall results. Subsequently, the model comparisons employed a testing subset comprised exclusively of the 178 correctly predicted task categories. It must be noted that FP-Growth results can be directly compared at the category level, but lack a more detailed natural language description, while the AI results give a natural language description but can also not allow the direct identification of a category. In total, 34 out of 600 NLP predictions did not allow for clearly detecting one of the 16 possible constraint categories, 29 of which occurred in the correctly classified tasks. Hence, the maximum theoretical points to achieve with the AI model and FP growth were 505 and 534, respectively.

Table 1. Task categorization results.

	Platforms: Beams			Platforms: Slabs			Elevations: Walls			Elevations: Pillars		
	R	**F**	**C**	**R**	**F**	**C**	**R**	**F**	**C**	**R**	**F**	**C**
G.T.	11	9	10	22	28	23	20	24	25	11	9	8
Pred.	8	8	7	20	26	21	20	23	21	10	8	6
Rec.		77%			92%			93%			86%	

FP growth provided 395 correct recommendations and predicted 139 constraints that were not present in the ground truth data, yielding a net performance of 256 points or 43% predictive capability. On the other hand, the AI model achieved a score of 415 points, with 460 correct predictions, 45 incorrect predictions, and 29 predictions that could not be categorized. Overall, considering that 89% of the task descriptions were misclassified, the AI model showed a consolidated capability to foresee and recommend effective real constraints in 69% of cases when prompted only with the task description and 78% if aided with a human-verified category. Table 2 summarizes this analysis. Furthermore, part of the AI predictions that were not present in the ground truth data could also be explained by the lack of proper registration by the users of the IT support system employed for the data collection.

DISCUSSION

POC LIMITATIONS

This POC was carried out employing 7.400 tuples from 1,000 unique tasks. The data corresponded exclusively to superstructure framing activities and a taxonomy of 16 constraints captured by the single IT-support system in Chilean high-rise building projects, employing Chilean-Spanish dialect and construction terminology when registering information. A hybrid ML architecture was employed instead of state-of-the-art LLMs. Also, the models employed in the POC architecture were not finetuned to further optimize prediction performance, since the results already signaled its benefits over traditional fixed-rule systems and manual identification. Finally, the model was fitted and evaluated using an artificially enriched sample produced by the selection of tasks linked to at least six constraints or preventable RNCs. Enriching the sample was deemed necessary to facilitate prediction outcomes and the existence of constraint-free tasks could be caused simply by the lack of proper registration in the IT support systems. Nevertheless, the POC model should be expanded with a larger corpus to include variance in the tuples and trim unnecessary predictions when a task exhibits a low constraint probability or a reduced set of potential constraints and preventable RNCs.

Table 2. Model performance results.

	Cat. tasks	Constraint preds.	Cat. predictions	Correct predictions	Incorrect predictions	Score
Ground truth	200	1320	1320	600 (aim)	0	600
Expert System	178	534	534	395	139	256 (43%)
AI Model	178	534	505	460	45	415 (69%)

CONTRIBUTIONS

This research shows that even a basic AI pipeline can provide domain-accurate constraint recommendations on 70% of cases based on past project information. This means that 70% of constraints that should be identified during the MRP could now be automatically registered as soon as a task enters the Lookahead Plan. Also, an LPS practitioner employing the IT support systems could quickly validate or modify the predicted task categories and recommended constraints, improving the percentage of use cases covered. Hence, it is estimated that up to 90% of the constraint identification workflow could be streamlined, significantly decreasing the time and effort needed to kickstart the MRP and giving Last Planners a wider time scope to then commit and remove those constraints. Having access to better-registered information facilitates the use of knowledge management for other data-driven decision tasks (Franz et al., 2022), such as improving MRP performance and reducing constraint removal times. Finally, the same concept can be applied to repetitive decisions such as corrective action implementation, RNC registration, and selection of tasks from the workable inventory. The study significantly contributes to the integration of AI in the Last Planner System by providing a validated AI-driven approach to Make-Ready Planning. This contribution not only enhances theoretical understanding but also offers practical tools that can improve efficiency and predictability in construction projects. The implications of these contributions are profound, potentially enabling project managers to reduce delays and better manage resources. While we acknowledge certain limitations in our current model, these should be viewed as starting points for further refinement. The opportunities for future research, such as integrating more advanced algorithms, ultimately aim to build upon the solid foundation this work provides.

OPPORTUNITIES

This POC was carried out employing only 7.400 tuples from 1,000 unique tasks, while the data captured by the single IT support system employed in this study comprises over 290 projects, with over 100,000 unique tasks, 220,000 reasons for noncompliance, and 130,000 constraints. Also, several other LPS support systems have collected similarly sized structured datasets. The results can be further improved by finetuning the existing architecture. Particular examples of finetuning opportunities include (1) the number of neighbors employed by KNN, which was fixed to 50 tasks, (2) the dimension of the embedding vectors, which was limited to 100, (3) the number of tokens employed to produce the outputs, currently fixed to five, (4) adding additional relative value to the tokens describing the constraint category predictions, and (5) employing alternative task classification algorithms instead of NB. State-of-the-art models allow for better embeddings fitted to significantly larger LLM corpora. Open-source LLMs like Large Language Model Meta AI (LLaMA) also offer finetuning and retraining opportunities to better capture the specific LPS task management corpus. Equivalent models like Google's BARD and Bidirectional Encoder Representations from Transformers (BERT) offer end-to-end question-and-answering pipelines, which already incorporate reasoning and generative technologies. State-of-the-art AI research is constantly producing new open-source models and architectures for a wide array of domain-specific challenges, contexts, and corpora.

CONCLUSIONS

This research assessed the use of AI to identify and register constraints early during Lookahead Planning and facilitate Make-Ready Planning. LPS adoption studies found that inefficient MRP impacts short- and long-term performance. Practitioners often fail to identify constraints in time, and they take longer than planned to be removed, 55% of times past planned execution. The hypothesis that task-constraint relationships found on past projects can be used to recommend new ones for a task entering the Lookahead Plan was validated via the Design Science Research Methodology. A Proof-of-Concept AI model was developed using 1,000 tasks connected to 7,400 constraints and preventable RNCs. LLM embeddings helped find similarities among them and use tokens to produce three ranked recommendations. Its performance was compared against an expert recommender based on the FP-Growth algorithm. While FP growth showed a 43% net rate of correct recommendations, the AI model achieved 69%. This rate increases to 78% if the input task is classified beforehand. The model can be finetuned to improve results using a larger sample and the POC can be replaced by state-of-the-art end-to-end question and answering LLM models. The resulting system acts as a copilot for LPS practitioners, helping them kickstart MRP early and more effectively. The same approach can be employed to recommend corrective actions, select tasks to enter short-term plans, or register RNCs. Further research could explore the integration of more advanced machine learning algorithms to improve predictive accuracy and user interaction. Additionally, extending this research to other phases of project management or different industries could provide insights into the broader applicability of AI tools. Long-term studies focusing on the sustained impacts of AI integration in actual project environments would also be invaluable to assess its long-term benefits and challenges.

ACKNOWLEDGMENTS

This study was financed thanks to ANID project FONDECYT Regular N°1210769 and ANID project Iniciación N°11. The authors would like to thank the Production Management Centre GEPUC from Pontificia Universidad Católica de Chile, GEPRO, and CONXAI Technologies GmbH for facilitating this study.

REFERENCES

Ballard, G., & Howell, G. (2003). An update on last planner. 11th Annual Conference of the International Group for Lean Construction, 1–10.

Ballard, G., & Tommelein, I. (2016). Current Process Benchmark for the Last Planner System. Lean Construction Journal, 13(1), 57–89.

Bellaver, G. B., Santos, D. O. R. S., Etges, B. M., Santos, P. H. J., & Mota, W. de S. (2022). Implementing Lookahead Planning and Digital Tools to Enable Scalability and Set of Information in a Multi-Site Lean Implementation. 30th Annual Conference of the International Group for Lean Construction, IGLC 2022, 750–761.

Castillo, T. (2015). Analyzing the Interrelation Between Management Practices , Organizational Characteristics and Performance Indicators for Construction Companies. Proceedings for the 23th Annual Conference of the International Group for Lean Construction, 691–700.

Cisterna, D., Lauble, S., Haghsheno, S., & Wolber, J. (2022). Synergies Between Lean Construction and Artificial Intelligence: AI Driven Continuous Improvement Process. 30th Annual Conference of the International Group for Lean Construction (IGLC), 199–210.

Cisterna, D., Seibel, S., Oprach, S., & Haghsheno, S. (2022). Artificial Intelligence for the Construction Industry - A Statistical Descriptive Analysis of Drivers and Barriers (pp. 283–295). https://doi.org/10.1007/978-3-030-87687-6_27

Costa, S., Carvalho, M. S., Pimentel, C., & Duarte, C. (2023). A Systematic Literature Review and Conceptual Framework of Construction Industrialization. Journal of Construction Engineering and Management, 149(2).

Daniel, E., Pasquire, C., & Dickens, G. (2019). Development of Approach to Support Construction Stakeholders in Implementation of the Last Planner System. Journal of Management in Engineering, 35(5), 04019018.

Dave, B., Seppänen, O., & Modrich, R. (. (2016). Modeling Information Flows Between Last Planner and Location Based Management System. www.iglc.net

Faloughi, M., Bechara, W., Chamoun, J., & Hamzeh, F. (2014). SimpLean: An Effective Tool for Optimizing Construction Workflow. Proceedings for the 22nd Annual Conference of the International Group for Lean Construction, 281–292.

Feng, W., Zhu, Q., Zhuang, J., & Yu, S. (2019). An expert recommendation algorithm based on Pearson correlation coefficient and FP-growth. Cluster Computing, 22(S3), 7401–7412. https://doi.org/10.1007/s10586-017-1576-y

Franz, B., Leicht, R., El Asmar, M., & Molenaar, K. (2022). Methodological Consistency for Quantitative Analysis and Reporting in Project Delivery System Performance Research. Journal of Construction Engineering and Management, 148(9).

Gondia, A., Siam, A., El-Dakhakhni, W., & Nassar, A. H. (2020). Machine Learning Algorithms for Construction Projects Delay Risk Prediction. Journal of Construction Engineering and Management, 146(1).

Grohe, M. (2020). word2vec, node2vec, graph2vec, X2vec: Towards a Theory of Vector Embeddings of Structured Data. Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, 1–16.

Hatoum, M. B., & Nassereddine, H. (2023). Unleashing the Power of Chatgpt for Lean Construction: An Early Outlook. 208–219. https://doi.org/10.24928/2023/0243

Heigermoser, D., García de Soto, B., Abbott, E. L. S., & Chua, D. K. H. (2019). BIM-based Last Planner System tool for improving construction project management. Automation in Construction, 104, 246–254.

Hunt, R. J., & Gonzalez, V. A. (2018). Innovation in the New Zealand Construction Industry – Diffusion of the Last Planner System. 422–431.

Jang, J. W., & Kim, Y. W. (2007). Use of Percent of Constraints Removal to Measure the Make-Ready Process. July, 529–538.

Javanmardi, A., Abbasian-Hosseini, S. A., Liu, M., & Hsiang, S. M. (2020). Improving Effectiveness of Constraints Removal in Construction Planning Meetings: Information-Theoretic Approach. Journal of Construction Engineering and Management, 146(4). https://doi.org/10.1061/(ASCE)CO.1943-7862.0001790

Kecman, V. (2001). Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models. MIT Press.

Kifokeris, D. (2021). Variants of Swedish Lean Construction Practices Reported in Research: Systematic Literature Review and Critical Analysis. Journal of Construction Engineering and Management, 147(7).

Kim, Y. W. (2019). The impact of make-ready process on project cost performance in heavy civil construction projects. Production Planning and Control, 30(13), 1064–1071.

Lagos, C. I., & Alarcón, L. F. (2021a). Assessing the Relationship between Constraint Management and Schedule Performance in Chilean and Colombian Construction Projects. Journal of Management in Engineering, 37(5).

Lagos, C. I., & Alarcón, L. F. (2021b). Composition and Impact of Reasons for Noncompletion in Construction Projects. IGLC 2021 - 29th Annual Conference of the International Group for Lean Construction - Lean Construction in Crisis Times: Responding to the Post-Pandemic AEC Industry Challenges, 817–826.

Lagos, C. I., Herrera, R. F., & Alarcón, L. F. (2019). Assessing the Impacts of an IT LPS Support System on Schedule Accomplishment in Construction Projects. Journal of Construction Engineering and Management, 145(10), 04019055.

Lagos, C. I., Herrera, R. F., Muñoz, J., & Alarcón, L. F. (2022). Influence of Last Planner® System Adoption Level on Project Management and Communication. 30th Annual Conference of the International Group for Lean Construction (IGLC), 211–222.

Nie, L., Li, Y., Feng, F., Song, X., Wang, M., & Wang, Y. (2020). Large-Scale Question Tagging via Joint Question-Topic Embedding Learning. ACM Transactions on Information Systems, 38(2), 1–23. https://doi.org/10.1145/3380954

Oprach, S., Steuer, D., Krichbaum, V., & Haghsheno, S. (2019). Smart Data - Dealing With Task Complexity in Construction Scheduling. 347–358.

Pérez, D., Lagos, C., & Fernando Alarcón, L. (2022). Key Last Planner System Metrics to Assess Project Performance in High-Rise Building and Industrial Construction Projects. Journal of Construction Engineering and Management, 148(1).

Salling, S. T., Pérez, C. T., & Wandahl, S. (2023). Perception of Project Management Among Construction Workers: A Survey in Denmark. Proceedings of the 31st Annual Conference of the International Group for Lean Construction (IGLC31), 882–893.

Samad, G. El, Hamzeh, F., & Emdanat, S. (2017). Last Planner System – The Need for New Metrics. Proceedings of the 25th Annual Conference of the International Group for Lean Construction (IGLC), July, Volume II, 637–644.

Sbiti, M., Beddiar, K., Beladjine, D., Perrault, R., & Mazari, B. (2021). Toward BIM and LPS Data Integration for Lean Site Project Management: A State-of-the-Art Review and Recommendations. Buildings, 11(5), 196.

Shehab, L., Salhab, D., Pourrahimian, E., Noueihed, K., Lucko, G., & Hamzeh, F. R. (2022). Machine Learning Framework to Predict Last Planner System Performance Metrics. Construction Research Congress 2022, 292–301.

Stevens, M., & Olayiwola, K. (2023). Examining the Gap Between Construction Software Modules and Lean Contractor Practices. Proceedings of the 31st Annual Conference of the International Group for Lean Construction (IGLC31), 13–23.

Venable, J. R., Pries-Heje, J., & Baskerville, R. L. (2017). Choosing a Design Science Research Methodology. Australasian Conference on Information Systems - ACIS 2017, 112–123.

Wang, N., & Issa, R. (2023, January). Transfer Learning-Based Question Generation for Building a Construction Safety Chatbot. Computing in Civil Engineering 2023.

Wang, Y. R., Yu, C. Y., & Chan, H. H. (2012). Predicting construction cost and schedule success using artificial neural networks ensemble and support vector machines classification models. International Journal of Project Management, 30(4), 470–478.

Warren, C. J. (2019). Collaborative Scheduling Methods: The Most Collaborative and Software to Support.

Wong, C. H. (2004). Contractor Performance Prediction Model for the United Kingdom Construction Contractor: Study of Logistic Regression Approach. Journal of Construction Engineering and Management, 130(5), 691–698.