FLEXFINDER: AUTOMATING MEDIUM VOLTAGE GRID CONTRACTS WITH LANGUAGE MODELS

Jesse J Heckman[1]†, Florian De Koning[1]†, Luc CB Nies[1*], Jochen L Cremer[2]

[1] Research Centre for Digital Technologies, Alliander N.V., Arnhem, The Netherlands

[2] TU Delft AI Energy Lab, Delft University of Technology, Delft, Netherlands

† These authors contributed equally to this work

[*] Corresponding author: Luc.Nies@alliander.com

GRID CONGESTION, FLEXIBILITY CONTRACTS, LARGE LANGUAGE MODELS, MULTI AGENTIC APPROACH

Abstract

The Netherlands' transition to renewable energy is straining the power grid, necessitating innovative solutions to mitigate congestion while awaiting grid extension. Non-firm Connection Transmission Agreements (CTAs), or flexibility contracts, offer a promising solution by enabling variable or on-demand capacity limits. However, determining suitable contracts for customers and ensuring grid compatibility is complex, requiring expert involvement and hindering large-scale implementation. To address this, we propose FlexFinder, a multi-agent system utilising Large Language Models to simplify contract identification and negotiation. FlexFinder interacts with customers, infers their needs, and suggests tailored contracts. It evaluates grid feasibility using power system tools, automating a traditionally expert-intensive process. Simulations across 17 customer roles demonstrated FlexFinder's effectiveness, with GPT-4.0 achieving an accuracy of 83.3% compared to 65.5% for GPT-3.5. Bayesian analysis confirmed the system's robustness across variations in verbosity and spelling errors. FlexFinder facilitates efficient, scalable negotiations, ensuring optimal grid utilisation and accelerating renewable energy adoption.

1 Introduction

The transition toward renewable energy redefines the energy system. Power generation shifts from centralised energy resources such as coal or gas to distributed energy resources such as solar and wind. The electrification of transport, and both domestic and industrial heating increases demand significantly. In The Netherlands, the electrification of the energy system is considered the main path towards a net-zero energy system. This requires a major extension of the power grid. The Dutch grid has to roughly double in size to facilitate this transition. Yet, extending the grid is a time-intensive process, and it is already operating at capacity, leading to congestion and the inability to connect new customers or expand the capacity of existing customers. The social damages resulting from congestion are estimated to cost on average €11,656 per MWh (RVO, 2024). The total estimation of economic damages due to congestion ranges between €10 billion and €35 billion (Ven et al., 2024). Given that grid extension takes years, alternative solutions are being explored to free up capacity and connect more customers to the grid.

1.1 Administrative Congestion

The term congestion in the power grid context often refers to insufficient capacity to meet demand on either generation or consumption. However, it is important to distinguish between physical congestion and administrative congestion. Each customer has a Connection and Transport Agreement (CTA) with their grid operator. This CTA contracts the maximum capacity a customer can use for both consumption and generation. If each customer would utilise their full capacity as contracted in their CTA, then the grid is called "physically congested". Any more load on the grid may cause overloading with damages and outages as a result. Therefore, to ensure a grid's reliability, the sum of all CTAs in a grid should be lower than the grid's capacity including the safety margin. Regulations from the *Autoriteit Consument en Markt* (ACM) prohibit new CTAs from causing the total grid usage to exceed its capacity. If this situation occurs, there is administrative congestion. However, it is rare for all customers to simultaneously utilise their maximum contracted capacity. So despite there being administrative congestion, there is often still some headroom before physical congestion would occur. Properly utilising this headroom is key in mitigating the effects of congestion.

1.2 Congestion Mitigation Measures in Netherlands

To mitigate congestion in the Netherlands, a new type of contract has been introduced: the non-firm Connection Transmission Agreement (CTA), commonly known as a

flexibility contract. In such a contract, the maximum available capacity is no longer constant. At the time of writing two options are available: time-bound contracts in which the available capacity can differ each timeslot, and an on-demand capacity limiting contract in which the operator can request a customer to lower their consumption or generation for which they will be reimbursed (Figure 1).

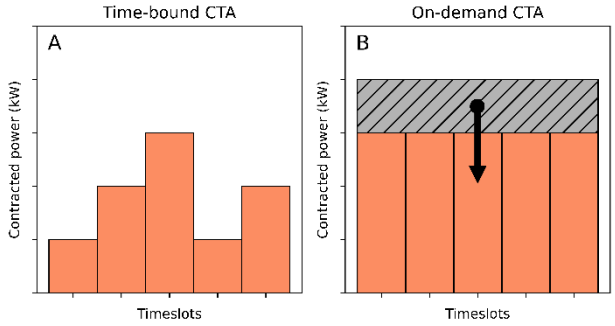

Figure 1 Two options for non-form CTAs. In (A) the contracted power differs each timeslot. In (B) operators can limit consumption on-demand.

While these new contracts enable better utilisation of the existing grid, these contracts require assets with more flexibility in electricity consumption and generation. This requirement adds an additional layer of complexity for both the grid operator and the customer.

1.3 Determining Fitting Flexibility Contracts & Requirements

Determining which flexibility contract is appropriate is challenging and time-consuming, as it requires specific domain knowledge of both the customer and the grid. For example, an on-demand limiting contract is typically used for solar farms but not for industry as few industrial processes are flexible enough to reduce load when requested. Time-based contracts might be a better fit for most industries so that production can be planned accordingly. On the customer side, investments in energy management or energy storage systems may be required to be technically equipped to fulfil the contracted duties. On the grid operator's side, assessing the available headroom on the grid for a proposed contract is essential. This task requires experts, as many distribution grids have low observability and a high number of direct customer endpoints. Due to this complexity, skilled grid experts—who are in short supply—are needed to evaluate the availability of headroom using specialised tools that demand domain-specific knowledge.

A key challenge lies in the need for experts to interact with customers individually to determine suitable contracts. With current tools and resources, conducting this process on a per-customer basis is beyond the capacity of grid operators, posing a significant barrier to the large-scale rollout of non-firm CTAs.

1.4 Large Language Models

In recent years, Large Language Models (LLMs) have emerged as transformative tools for addressing a wide range of language-based tasks. Their capabilities have systematically improved with architectural advancements, increased model size, and methods like Retrieval-Augmented Generation (Lewis et al., 2020) and multi-agent systems using LLMs (Wu et al., 2023). Although Marot et al. (2021) introduced a conceptual framework for AI assistants in the energy domain, the integration of LLMs into this industry has so far been modest. Jin et al. (2024) demonstrated a first application with ChatGrid, a retrieval-augmented Q&A system designed to support operators with real-time, domain-specific insights. While this development showcases the adaptability of language models in energy contexts, there remains a significant opportunity to extend their use to other innovations such as negotiating flexibility contracts.

1.5 FlexFinder

The complexity of non-firm CTAs has so far hindered their widespread rollout and adoption. To address this challenge, we propose FlexFinder, a system designed to simplify the process of identifying suitable flexibility contracts for both customers and grid operators. Built on LLMs in an agentic setup (Wu et al., 2023), FlexFinder transforms expert tools traditionally used by grid architects into a format accessible to non-experts. Through natural language interaction, customers provide the necessary information about their requirements, which the system processes to suggest a tailored CTA. Customers can then review and iteratively refine the proposed contract, ensuring a final agreement that aligns with their needs and grid operator constraints.

The remainder of the paper describes the proposed architecture of this system, the prototype which explores these core concepts, and the results of the first experiments.

2 Methodology

FlexFinder addresses the complex and iterative process of implementing non-firm Connection Transmission Agreement (CTA) contracts by utilising a LLM-powered multi agent system. In short, customers can apply for a non-firm CTA through an interactive chatbot. The system infers relevant customer data (i.e. segment and capacity) and executes functions to estimate feasible non-firm CTAs, providing an efficient and user-friendly application process.

2.1 Large Language Models

Large language Models (LLMs) developed by OpenAI, specifically GPT-3.5 and GPT-4.0, were employed in the development of FlexFinder. Temperatures for all LLM were effectively set to 0, restricting their randomness. These models were then accessed via API requests implemented in Python (3.11.8), utilising tools available within Microsoft's Azure ecosystem.

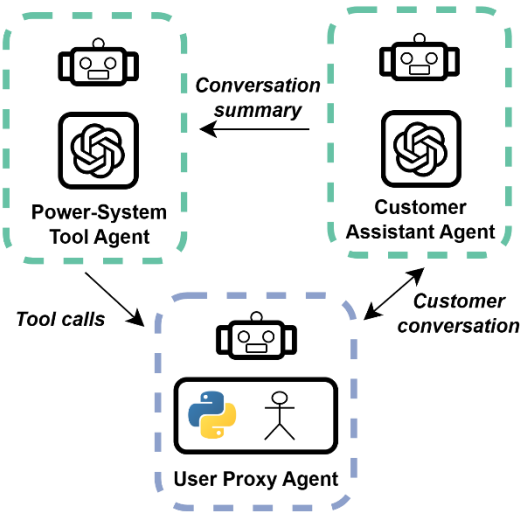

Figure 2 AutoGen implementation in FlexFinder. The User Proxy Agent converses with the Customer Assistant Agent to determine the customer segment and contracted power of the CTA. The conversation is sent to the Power-System Tool Agent that calls the power system analysis functions (Python) with the inferred input from the context.

2.2 Multi-Agentic System

Microsoft's AutoGen (Wu et al., 2023) framework was applied to set up a multi-agentic system with the following agents: (1) the User Proxy Agent, (2) the Customer Assistant Agent, and (3) the Power-System Tool Agent (Figure 2). Each agent had a specific instruction and task.

User Proxy Agent: This agent functioned as a proxy for the customer interacting with the Customer Assistant Agent. In operation, the User Proxy Agent will relay human chat input. Presently, it conveyed simulated human chat input, which was generated experimentally using a large language model (GPT-4). To enhance realism during simulations, the outputs of the User Proxy Agent were deliberately modified to potentially include spelling errors before being forwarded to the Customer Assistant Agent. Additionally, this agent class autonomously executed tool calls suggested by the Power-System Tool Agent and returned the results of the executed code to the Power-System Tool Agent for validation.

Customer Assistant Agent: This agent served as the primary interface with customers, gathering the input data. The agent conversed with the customer until the agent concluded the customer segment and the capacity of the contract. Until reaching this conclusion, the agent asked clarification questions and answered the customer's questions.

Power-System Tool Agent: This agent assessed the physical feasibility of the non-firm CTA. To do so, this agent handled the pre-defined tool calls (Python functions assessing power system tools and the grid data) with the context provided by the Customer Assistant Agent. This agent can reiterate suggesting tool calls if the output is not as expected until a physically feasible and customer-satisfiable contract is determined.

2.3 Bayesian analyses

To test robustness, the effects of two categorical predictors—verbosity and Levenshtein distance (each with three levels or intensities)—on both accuracy and the number of interactions during simulated conversations were estimated. A generalized linear model (GLM) was constructed (Equation 1) and paired with a Gibbs sampler to estimate the posterior distributions of the parameters:

$$(1) \qquad y_i = b_0 + b_1 x_1^{(i)} + b_2 x_2^{(i)}$$

Here, b_0 represents the intercept, while b_1 and b_2 are the coefficients for verbosity and Levenshtein distance, respectively. The variables $x_1^{(i)}$ and $x_2^{(i)}$ correspond to the levels of their respective categorical predictors at observation i. For each prediction, y_i is uniquely estimated using two distinct link functions.

The relationship between the predictors and the posterior distributions of the outcomes was modelled with these link functions: accuracy was derived from a logit link function ($\text{logit}(y_i)$), while the number of interactions was sampled from a normal distribution ($N(y_i,\sigma^2)$). Weakly informative priors, sampled from a normal distribution centred at 0, were applied. All parameters achieved an effective sample size of at least 20,000 samples, ensuring robust estimation.

Note that the study conducted the analysis once per language model, and differences between both GPTs are therefore not directly compared.

3 Case study

3.1 Simulation settings

To evaluate the robustness of this proposed framework, 17 distinct customer roles were defined to act as the User Proxy Agent. Each role represents a description of a specific customer seeking a new grid connection for a certain activity and contracted power. For example, one role represents a baker establishing a new branch with a power requirement of 80 kVA. These roles are grouped into six predefined client segments (Figure 3). Every customer role is simulated by an LLM as the User Proxy Agent.

Additionally, the evaluation considers two key variables related to conversational style: (1) verbosity which is varied by employing three different prompts to simulate different levels of detail in responses (Figure 4B), and (2) the extent of spelling errors introduced in customer responses. The latter is implemented by applying a function to each simulated customer response (at customer conversation in Figure 2), which alters the text based on a Levenshtein distance fraction. For example, a Levenshtein fraction 0.05

indicates that one in 20 characters is modified through insertion, deletion or substitution. The experiment is conducted by applying three levels of Levenshtein fractions: 0 (no modifications), 0.05, and 0.1. Lastly, the input provided by the Customer Assistant Agent to the Power-System Tool Agent was evaluated under two conditions: an LLM-generated summary or the full conversation transcript.

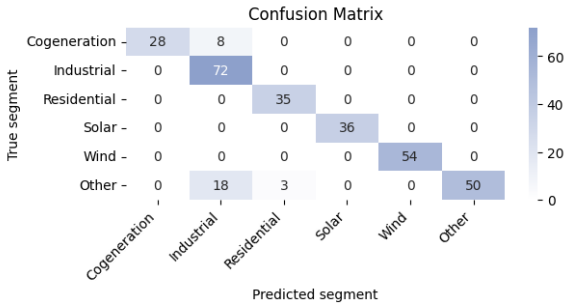

Figure 3 Confusion matrix of segment classification. Example raw classification data of FlexFinder using GPT-4.0.

Varying these three parameters over 17 user roles resulted in 306 simulated conversations between the customer and FlexFinder per model. This was done for both GPT-3.5 and GPT-4.0 resulting in a total of 612 simulated conversations.

3.2 Does the tool make errors?

This study assessed how effectively the inference assistant classified contract segments and capacities in simulated negotiation scenarios, and if they were affected by customer verbal expression. Simulations were conducted using both GPT-3.5 and GPT-4.0 to examine result consistency across different language models. Initial results reveal a notable discrepancy in combined classification accuracy, with GPT-4.0 achieving 83.3% compared to 65.5% for GPT-3.5 (Figure 4A). Further analysis suggests this difference is primarily driven by segment inference performance, highlighting the newer language model's stronger semantic understanding of customers.

As verbosity level was embedded in the User Proxy (customer) 'role' instructions, we first examined its influence on the mean number of characters per user proxy response. Results indicate that response length systematically increased with verbosity level across both models (Figure 4B). Notably, the User Proxy was consistently powered by GPT-4.0, meaning that the observed differences in response magnitude between the two language models reflect interactions with the FlexFinder method and its downstream language model.

3.2 Bayesian modelling

3.2.1 Accuracy analysis using a Bernoulli distribution

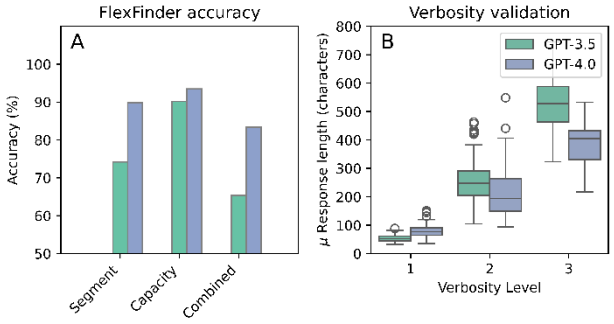

Figure 4 Simulation results. (A) Accuracy comparison of FlexFinder-customer simulations in inferring segment, capacity, and their combination. Bars represent accuracy (%) for each task, with GPT-3.5 shown in green and GPT-4.0 in blue. (B) Box plot illustrating the effect of verbosity level on the mean response length in characters.

A Bayesian generalized linear model (GLM) was applied separately to data generated with GPT-3.5 and GPT-4.0 to evaluate the performance of the proposed FlexFinder method and the effects of verbosity and Levenshtein distance. Posterior predictive checks indicated a good statistical fit for both models (Bayesian p-values: μ=[0.51, 0.44]; σ2=[0.47, 0.49]). For GPT-3.5, the baseline log-odds were estimated at 0.65 (or 65.5%), while for GPT-4.0, the baseline increased to 1.67 (83.3%). Although the analyses did not include the GPT model as an explicit predictor, the higher baseline log-odds observed for GPT-4.0 suggest a substantially improved starting point for inference accuracy, and the importance of language model selection.

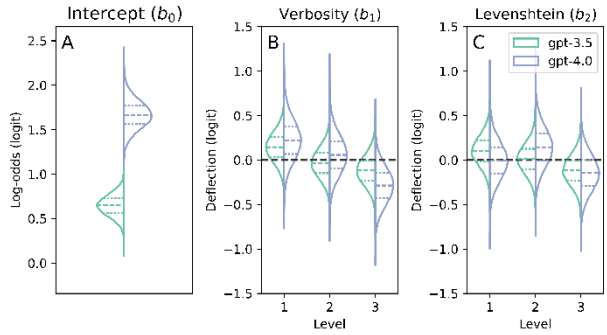

Figure 5 Posterior distributions of the intercept and main effects of verbosity and Levenshtein distance on inference accuracy, derived from a Bayesian generalized linear model (GLM). Violin plots depict normalized posterior densities, with green representing GPT-3.5 and blue representing GPT-4.0. Dashed lines denote the posterior mean, while dotted lines represent standard deviations. (A) Posterior distribution of the intercept, reflecting baseline log-odds. Main effects of (B) verbosity and (C) Levenshtein depict logit defection of each level.

Analysing the main effects showed consistent behaviour across GPT models (Figure 5). Both categorical predictors, verbosity and Levenshtein distance, had minimal impact on inference accuracy. While a slight negative trend for verbosity is visually apparent—accuracy decreases as verbosity increases (Figure 5B)—the distributions are broad and largely overlapping. An element-wise comparison of verbosity levels 1 and 3 provides no evidence to substantiate this trend, as the credible intervals for the difference include

0 (μ=[-0.27, -0.51], 95% CI=[[-0.87, 0.31], [-1.26, 0.29]]). Similarly, little evidence was found to support any effect of Levenshtein distance on inference accuracy.

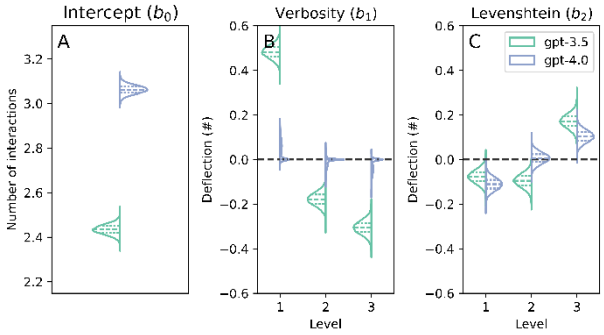

Figure 6 Posterior distributions of the intercept and main effects of verbosity and Levenshtein distance on inference efficiency, derived from a Bayesian generalized linear model (GLM). Violin plots depict normalized posterior densities, with green representing GPT-3.5 and blue representing GPT-4.0. Dashed lines denote the posterior mean, while dotted lines represent standard deviations. (A) Depicts the posterior distribution of the intercept with (B+C) depicting the deflection by verbosity and Levenshtein.

Together these results indicate that the FlexFinder method performs robustly across both models, with a potential advantage in baseline accuracy when using GPT-4.0. Furthermore, the findings suggest that FlexFinder remains unbiased toward linguistic variations such as verbosity or spelling errors.

3.2.1 Efficiency analysis using a normal distribution

A similar GLM to predict the number of interactions (Figure 6; Bayesian p-values: μ=[0.46, 0.49]; σ2=[0.49, 0.42]), was conducted to study the effect of verbosity and Levenshtein on the number of interactions between the User Proxy and FlexFinder. Simulations with GPT-4.0 resulted in longer interactions, with a mean of 3.06 messages compared to a baseline of 2.43 for GPT-3.5. Notably, different patterns in main effects emerged across the two language models. For GPT-3.5, there is strong evidence that increasing verbosity reduces the number of interactions, supported by the narrow distributions and large effect size. However, no such effect was observed for GPT-4.0.

The main effect of Levenshtein distance showed greater consistency across both language models, albeit with a modest effect size. A positive trend was confirmed through element-wise comparison of levels 1 and 3, as the credible intervals for the subtracted distributions did not span 0 (GPT-3.5: 95% CI=[0.14, 0.36]; GPT-4.0: 95% CI=[0.11, 0.31]).

4 Conclusion

This work presents one of the first conceptual applications of LLMs within the energy sector, demonstrating how expert tools can be made more accessible. Customers are enabled to iteratively negotiate their own flexibility to get access to the grid. This work addressed a specific bottleneck in the customer-interaction and lays the foundation for applying agentic frameworks across multiple tools. Initial simulation results highlight the robustness of the system to variations in user language and open the possibility to real-world experimentation. Future work will focus on refining the system for deployment, integrating it into operational grid management to effectively reduce congestion in grids through flexible contracts.

5 Acknowledgements

We extend our gratitude to Archana Ranganathan for her significant contributions to the early stages of this project.

6 References

Netherlands Enterprise Agency (RVO) and Ministerie van Economische Zaken en Klimaat (2024) Maatschappelijke kostprijs van netcongestie, https://www.rijksoverheid.nl/documenten/rapporten/2024/06/25/studie-maatschappelijke-kosten-netcongestie, accessed 17-01-2025

Venema, T., van Swieten, T., van den Boogaard, S., Bieze, R., & Middelbos, M. (2024) Haal de kink uit de kabel - Zes interventies om de congestie op het Nederlandse elektriciteitsnet versneld te verlichten, Boston Consulting Group, https://www.bcg.com/publications/2024/netherlands-haal-de-kink-uit-de-kabel, accessed 17-01-2025

Marot, A., Kelly, A., Naglic, M., Barbesant, V., Cremer, J., Stefanov, A., & Viebahn, J. (2022). Perspectives on future power system control centers for energy transition. *Journal of Modern Power Systems and Clean Energy*, *10*(2), 328-344.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, *33*, 9459-9474.

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., ... & Wang, C. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Jin, S., & Abhyankar, S. (2024, October). ChatGrid: Power Grid Visualization Empowered by a Large Language Model. In *2024 IEEE Workshop on Energy Data Visualization (EnergyVis)* (pp. 12-17). IEEE.