Energy and AI 18 (2024) 100431

Contents lists available at ScienceDirect

Energy and AI

journal homepage: www.sciencedirect.com/journal/energy-and-ai

Supporting energy policy research with large language models: A case study in wind energy siting ordinances

Grant Buster *, Pavlo Pinchuk, Jacob Barrons, Ryan McKeever, Aaron Levine, Anthony Lopez

National Renewable Energy Laboratory, 15013 Denver W Pkwy, Golden, CO 80401, USA

HIGHLIGHTS

- Introduced an automated method using large language models to extract renewable energy siting ordinances from legal documents.
- Achieved an accuracy rate of 85 % to 90 % in ordinance information extraction using a decision tree algorithm powered by large language models.
- Significantly reduced the manual labor required to maintain an up-to-date energy siting ordinance database.
- Potential to automate similar large-scale policy research across the energy sector.

ARTICLE INFO

Keywords:
Energy policy
Energy analysis
Siting ordinances
Large language models (LLMs)
Natural language processing (NLP)

ABSTRACT

The recent growth in renewable energy development in the United States has been accompanied by a simultaneous surge in renewable energy siting ordinances. These zoning laws play a critical role in dictating the placement of wind and solar resources that are critical for achieving low-carbon energy futures. In this context, efficient access to and management of siting ordinance data becomes imperative. The National Renewable Energy Laboratory (NREL) recently introduced a public wind and solar siting database to fill this need. This paper presents a method for harnessing Large Language Models (LLMs) to automate the extraction of these siting ordinances from legal documents, enabling this database to maintain accurate up-to-date information in the rapidly changing energy policy landscape. A novel contribution of this research is the integration of a decision tree framework with LLMs. Our results show that this approach is 85 to 90 % accurate with outputs that can be used directly in downstream quantitative modeling. We discuss opportunities to use this work to support similar large-scale policy research in the energy sector. By unlocking new efficiencies in the extraction and analysis of legal documents using LLMs, this study enables a path forward for automated large-scale energy policy research.

1. Introduction

The energy system is rapidly changing as are the markets and policies that govern it. In recent years, the United States has seen a rise in wind and solar renewable energy generation [1,2]. In response, electricity markets have introduced new market structures and local governments have developed new regulations to accommodate these novel generators [3–5]. From 2018 to 2022, the electricity generation from utility-scale wind and solar sources has increased by 60 and 125 percent, respectively [2]. Over the same period, the number of local zoning ordinances restricting the development of wind energy systems has also increased substantially [5,6]. These policies directly affect the quantity of available, buildable land for renewables and have important implications for our future energy system [5].

These changes are happening at the same time the United States is setting ambitious goals for clean energy deployment in an effort to tackle the climate crisis [7]. Several scientific institutions have recently charted technical paths to achieve such decarbonization goals while identifying key opportunities and challenges [8,9]. However, without accurate data on current local energy policies, these analyses may represent idealized energy transitions without a realistic representation of the local legal challenges to deployment of renewable energy assets. Studies have performed initial assessments of the effect of renewable energy siting regulations on energy systems at the national scale,

Abbreviations: GPT, generative pre-trained transformer; LLM, large language model; LWET, large wind energy turbine; MWET, medium wind energy turbine; NLP, natural language processing; NREL, national renewable energy laboratory; PDF, portable document format; RAG, retrieval-augmented generation.

* Corresponding author.
E-mail address: Grant.Buster@nrel.gov (G. Buster).

https://doi.org/10.1016/j.egyai.2024.100431

Available online 9 October 2024

showing a strong relationship between restrictive siting policies and renewable deployments in a least-cost economic model [10]. Similar developments are possible in numerous other legal, economic, and political domains that surround the energy industry.

In the context of these changes, efficient access to and management of policy data becomes crucial. Historically, the process of analyzing energy policy has only been possible through significant manual labor. For example, the United States Wind and Solar Siting Regulation and Zoning Ordinance Databases recently introduced by the National Renewable Energy Laboratory (NREL) [5,11,12] required approximately 1500 human labor hours to collect siting regulation data from local zoning codes in the United States. This significant labor requirement makes it prohibitively expensive to continually maintain accurate data in these dynamic times. However, technologies in machine learning, natural language processing (NLP), and semantic search have also seen transformative developments in recent years, promising to enable the automation of these tasks.

The advent of Large Language Models (LLMs) such as GPT-4 has changed our understanding of what is possible in automated reading comprehension and text processing [13,14]. Applications of LLMs are being explored in medicine, law, education, finance, engineering, and media [15]. Accordingly, there are numerous possibilities for LLMs to expedite the retrieval and processing of energy policy data from legal documents. We see this as an ideal testing ground for real-world applications of LLMs because of the substantial reading comprehension requirements and the otherwise prohibitive cost of maintaining up-to-date information on the current state of energy policies across the nation.

In this work, we demonstrate how LLMs can be used to retrieve data on renewable energy zoning ordinances from legal documents. We introduce a strategy using decision trees to supplement the LLM fast reasoning skills with subject matter expertise and symbolic logic that leads to improved accuracy when compared to human effort or alternative LLM prompting strategies. Finally, we discuss future applications of the open-source software used here to enable similar large-scale data retrieval efforts elsewhere.

The structure of this article is as follows: Section 2 introduces the experimental methods along with the LLM prompting strategies. Section 3 discusses the results of applying LLMs to a set of ordinance documents. Section 4 discusses the potential for future opportunities and limitations of these methods in energy research. Section 5 concludes the article.

2. Methods

To explore the capability of LLMs to assist in the retrieval of renewable energy siting data from local ordinance documents, we collect a subset of legal documents from counties represented in the

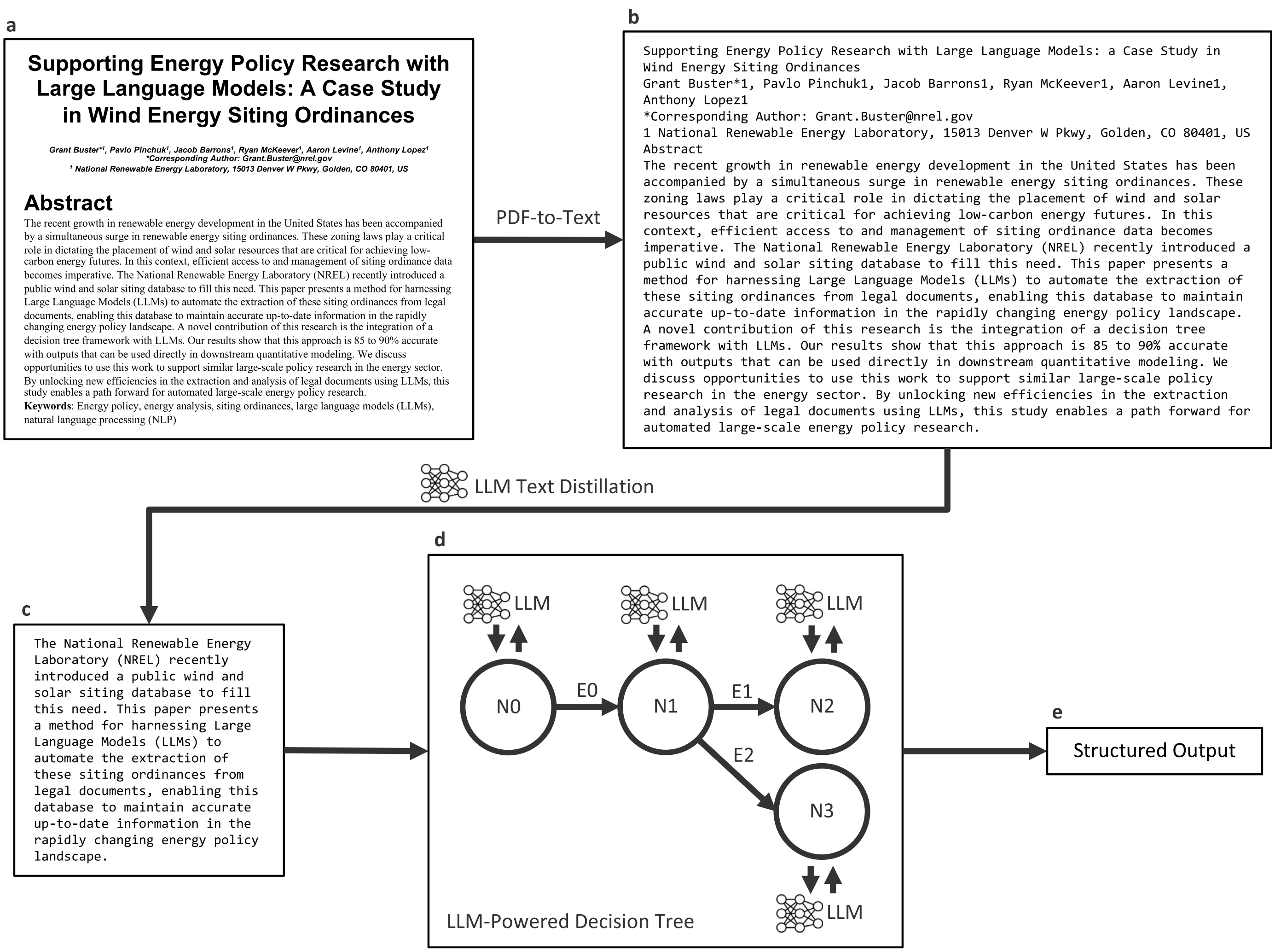

Fig. 1. a–e, Outline of the LLM ordinance data extraction process. a, Source PDF documents (this manuscript used in the figure for illustrative purposes). b, Raw machine-readable text extracted from the PDF. c, Text chunks related to siting ordinances extracted by the LLM from the full document text. d, an illustration of an LLM-powered decision tree framework used to parse structured data from the relevant text. e, Structured data describing siting ordinances output by the decision tree LLM framework.

United States Wind Siting Regulation and Zoning Ordinances Database (hereafter referred to as "the wind ordinance database") [11] and apply LLMs (specifically the Microsoft Azure "2023–03–15-preview" deployment of OpenAI's GPT-4) to extract structured ordinance data from the unstructured text. The document parsing methods are split into three parts, outlined in Fig. 1. First, ordinance documents typically in Portable Document Format (PDF) are converted to text as described in Section 2.1. Second, the ordinance text relevant to wind energy systems is identified and extracted as described in Section 2.1. Lastly, the LLM is used to extract data from the text using a decision tree framework further described in Section 2.2. Alternative prompting strategies that we explored but ultimately decided against are discussed in Section 2.3.

2.1. Document text extraction and distillation

The first part of the methods requires conversion of legal documents from PDF to text. We explored several utilities for PDF-to-text conversion and found that the free software Poppler [16] performed best, especially when considering the ability to convert ordinances in tabular format to space-delimited text.

The second part of the methods requires identifying and extracting only the text relevant to wind energy systems and their corresponding ordinances. Legal documents on zoning can be hundreds of pages long, easily exceeding the context window of most publicly available LLMs, including GPT-4 (4k or 32k tokens at the time of development). Although the context window of publicly available LLMs is rapidly increasing, long-window models can still suffer from low accuracy when requested to extract information from the middle of a large document [17]. As a result, we found that content distillation was an important step to improve the accuracy of these methods.

In order to distill the lengthy legal documents, we first perform an asynchronous semantic search on overlapping chunks of the text, instructing the LLM to "extract text related to the restrictions of wind energy systems" (see the code associated with this paper for full details on LLM prompts). A challenge here is preserving parts of the document that do not explicitly define the ordinances but that set important context such as definition of terms. Generally, however, a 100-page document on all zoning ordinances in a county typically does not have more than a few pages relevant to wind energy systems. This step distills the relevant information into a manageable amount of text that can be used by the LLM. An example of the desired output for a few select counties can be seen in Table 1.

2.2. Structured data extraction with decision trees

The last part of the method is to extract the discrete siting restrictions from the legal text relevant only to wind energy systems. The goal here is to output quantitative data that can be used in analysis such as that by Lopez et al. [11]. For this study, we chose 13 features that wind energy systems are commonly required to consider in siting decisions: structures (participating and non-participating), property lines (participating and non-participating), roads, railroads, transmission lines, bodies of water, noise restrictions, maximum system heights, minimum lot sizes, shadow flicker restrictions, and turbine density restrictions. The siting ordinances relative to the physical features are typically called setbacks (e.g., a turbine must be "set back" a pre-defined distance from these features). For this process to be useful in downstream quantitative analysis, the output must be concise and machine readable. An example query might be "what is the setback from participating residences in Monroe, Wisconsin?" and the output should consist of a numerical value and a categorical definition e.g., the value would be 1.1 and the categorical definition would be "maximum tip height multiplier". In this example, we know the county and the feature (Monroe; participating residences) and we can extract the numeric setback value and how to calculate the final distance value (1.1; a multiple of the maximum tip height). While this may be an intuitive task for a human, the small set of ordinance text in Table 1 shows the potential for significant heterogeneity in how these setbacks are defined in the real world. There are frequently multiple setback values to choose from, multiple subclasses of feature within a broader feature category, multiple classes of turbine based on size and application, and a wide variety of verbiage and abbreviations.

Table 1
Sample text on wind energy siting from ordinance documents. Page count is for the full ordinance document and sampled text is an LLM-selected excerpt related to wind energy siting.

County	Pages	Example of Extracted Wind Ordinance Text
Laramie, WY	223	The center of the base of each wind tower shall be located no less than 1.5 (hub height + rotor diameter) from adjacent unplatted nonparticipating property lines and dedicated public roads.
Ottawa, MI	16	Medium Wind Energy Turbine (MWET) shall also be subject to the following: • Occupied Building Setback: The setback from all occupied buildings on the applicant's parcel shall be a minimum of twenty (20) feet measured from the base of the Tower. Large Wind Energy Turbine (LWET) shall also be subject to the following: • Occupied Building Setback: Each LWET shall be set back from the nearest Occupied Building that is located on the same parcel as the LWET a minimum of two (2) times its Total Height, or one thousand (1000) feet, as measured from the base of the Tower, whichever is greater.
Monroe, WI	26	Occupied community buildings: • The lesser of 1250 feet or 3.1 times the maximum blade tip height. Participating residences: • 1.1 times the maximum blade tip height. Nonparticipating residences: • The lesser of 1250 feet or 3.1 times the maximum blade tip height

The strategy we found to work best for extracting ordinance data from legal documents is an approach using LLMs guided by a decision tree embedded with knowledge of the subject matter. Using a decision tree as a programmatic structure, we can break down our larger goal into many smaller requests (nodes) with logical transitions (edges) between each request. This is illustrated in Fig. 1: nodes N0-N3 interact with the LLM using unique prompts and the conversation history. Edges E0-E2 each have callable functions or additional LLM prompts that would trigger node transitions. Failure of all edge conditions in response to an LLM output at a node results in a null result (e.g., no relevant text) or can be flagged for human review. Leaf nodes N2 and N3 result in structured outputs in json format that can be put into a database. For example, leaf node N2 might return a wind turbine tip height multiplier setback while node N3 would return a fixed distance setback.

The LLM-powered decision tree is essentially a pre-programmed multi-prompt LLM conversation based on subject matter expertise. In practice, this looks like a series of nodes each containing an LLM prompt ("does the text define a setback?") with functions ("is the answer yes?") determining transitions between nodes. We also implement branching logic to handle the various types of ordinances that are recorded in the original wind ordinance database, including branches for ordinances with multiple conditions and instructions on what assumptions to make under this ambiguity (see Fig. 2). The output of this structure can be the setback text in our desired format (including as structured data) as returned by a leaf node (e.g., a node with no edges directed away from it) or an exception raised because no edge transition conditions were met. In the latter case, depending on where in the tree the exception was raised, we can assume there is no setback defined or flag the ordinance for human review. This approach is slower than a single-prompt strategy because it uses multiple chained prompts per document but is similarly limited only by the rate limit imposed by the LLM provider. A description of the decision tree algorithm is documented below in Algorithm 1.

This implementation of an LLM-powered decision tree operates

USER: Is there text in the legal document that describes how close i can site or how far i have to setback wind energy systems to buildings, structures, and/or residences? Please only say "Yes" or "No".

ASSISTANT: Yes

USER: Based on your last message, choose the option that best describes the required setback from buildings, structures, and/or residences:
#1.1 The setback from buildings and structures is a single distance value
#7.1 The setback from buildings and structures has multiple conditions such as a fixed distance and a multiple of the total system height
#0 None of the above options are descriptive of this ordinance related to setbacks from buildings and structures

ASSISTANT: #7.1 The setback from buildings and structures has multiple conditions such as a fixed distance and a multiple of the total system height

USER: Lets assume the supporting tower is 377 feet with blades 279 feet long. So the rotor diameter would be 558 feet and the total system height would be 656 feet. What would the multiple setback distances be? Please show your work.

ASSISTANT: Based on the information provided, we can calculate the setback distances for the different types of buildings and structures:
1. Occupied community buildings: The lesser of 1,250 feet or 3.1 times the maximum blade tip height 3.1 * 656 feet = 2,033.6 feet. The lesser value is 1,250 feet.
2. Participating residences: 1.1 times the maximum blade tip height 1.1 * 656 feet = 721.6 feet
3. Nonparticipating residences: The lesser of 1,250 feet or 3.1 times the maximum blade tip height 3.1 * 656 feet = 2,033.6 feet. The lesser value is 1,250 feet.

So, the multiple setback distances are:
1,250 feet for occupied community buildings
721.6 feet for participating residences
1,250 feet for nonparticipating residences

USER: What is the final setback value? If the ordinance states which of the multiple distances should be chosen, use that guidance. Otherwise, choose the largest setback value.

ASSISTANT: The ordinance does not provide specific guidance on choosing between the multiple distances. Therefore, we will choose the largest setback value. The final setback value is 1,250 feet.

USER: State the final setback like this: "The setback is XXX (units)"

ASSISTANT: The setback is 1,250 feet.

Fig. 2. Example of decision tree path for a complex setback rule from the Monroe, Wisconsin Code of Ordinances. Note that for illustrative purposes we only abridged messages from the full conversation that occurred during the experiment.

Algorithm 1
Decision tree pseudocode.

```
Input: body of text from the document in question t, target setback feature f
Function DecisionTree.run(t, f):
    node = DecisionTree.InitialNode
    while is_not_leaf_node(node):
        prompt = Prompter(t, f, node)
        response = LLM(prompt)
        Edges = GetEdges(node)
        for Edge_i in Edges:
            condition = Edge_i(response)
            if condition:
                node = get_next_node(Edge_i)
                break
            if not condition:
                raise Exception(prompt, node, response)
    return response
```

similar to the traditional decision tree algorithm in the machine-learning literature [18] but uses LLMs to process text at nodes instead of logical rules. This implementation can also be described as a rudimentary implementation of neuro-symbolic artificial intelligence [19], combining the powerful text-processing capabilities of the neural network with the ability for a subject matter expert to craft a logical structure for how a human would extract relevant information from a document.

The decision tree strategy is also similar to the tree-of-thought approach [20,21] but has some fundamental differences. Most crucially, in our data retrieval task we do not rely on self-evaluation of progress by an LLM or a rule-based algorithm. Instead, we rely on human-developed instruction and logic to guide the edge transitions and raise exceptions in the case of unrecognized ordinance types. Also, in our use case we define only a directed acyclic graph, whereas the tree-of-thought approach allows for backtracking through the graph. The software used here can accommodate cyclic graphs, but the acyclic assumption is convenient for this use case and prevents infinite loops.

It is also worth noting that the software we use to implement the decision tree is sufficiently generic to allow for any callable function to serve as a node or edge property. More complex problems could easily implement complex logical functions, neural networks, or nested graphs to processes edge conditions.

The software that implements the decision tree architecture is open-source and can be found in the Code and Data Availability section.

2.3. Alternative prompting strategies

We also ran experiments with a Retrieval-Augmented Generation (RAG) framework [22,23] and single-prompt data retrievals instead of the multi-prompt decision tree. We found that both strategies were ineffective for retrieving accurate structured information from a larger body of legal text. The heterogeneity of ordinance verbiage and setback values in tabular structures rendered vector comparison in a RAG framework unreliable. The complexity of the retrieval query also proved to be too challenging for a single large prompt, especially when the LLM was faced with ambiguous conflicting information such as with regulations on both small and large wind energy systems or with regulations that required comparative logic between multiple values. Both experiments were also unable to provide output in consistently machine-readable quantitative values paired with discrete categories for the ordinance type (the ability of decision trees to do this is discussed later in Section 3). In theory, we could augment single-prompt approaches with additional strategies such as providing the LLM with an example of successful ordinance extraction to guide its thought process [24] but given the diversity of possible ordinance examples we decided this was not tractable. Ultimately, we did not run these strategies on the full test set of ordinance documents because of poor performance in development with the training set of documents.

3. Results

We apply the LLM process illustrated in Fig. 1 to the wind ordinance documents in two sets: 18 documents in the “training” set and 85 documents in the “test” set. Each document here represents ordinances for a single county with known restrictions on wind energy systems. Although the LLM framework is not trained through the direct observation of numerical data like a traditional quantitative regression model might be, we find a similar training and test experimental setup useful to ensure extensibility of our approach. Here, the documents in the training set are used to develop the LLM approach, which includes development of the software for text cleaning and the LLM prompting strategies. The software and prompting strategies are then frozen and run against the test set documents. The test set documents are a “ground truth” dataset compiled by two researchers not involved in the development of the LLM strategies. The two researchers reviewed all 85 documents separately, resolved any discrepancies, and produced the most accurate test dataset possible. Comparable performance of the LLM strategies in both the training and test sets supports the extensibility of these methods, especially when considering the heterogeneity of ordinance text in the training and test documents.

Tables 2 and 3 summarize the results of the analysis on the training and test sets, respectively. Between the two sets of documents, the decision tree LLM strategy was found to have an accuracy score of 85–90 %, a precision score of 91–96 %, and a recall score of 75–81 %. The accuracy, precision, and recall metrics are defined here as:

Table 2
Results of ordinance retrieval from 18 training documents with 13 ordinance types (234 total ordinance values). Overall accuracy, precision, and recall were 86 %, 96 %, and 75 % respectively.

	LLM Found Ordinance		
		True	False
Ordinance Exists	True	61 (26 %) – Correct 11 (5 %) – Incorrect	20 (9 %)
	False	2 (1 %)	140 (60 %)

Table 3
Results of ordinance retrieval from 85 test documents with 13 ordinance types (1105 total ordinance values). Overall accuracy, precision, and recall were 90 %, 91 %, and 81 % respectively.

	LLM Found Ordinance		
		True	False
Ordinance Exists	True	325 (29 %) – Correct 34 (3 %) – Incorrect	42 (4 %)
	False	34 (3 %)	670 (61 %)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN + IP} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

where TP are the true positive results (correct values in the top left quadrant of Tables 2 and 3), TN are the true negative results (values in the bottom right quadrant of Tables 2 and 3), FP are the false positive results (values in the bottom left quadrant of Tables 2 and 3), FN are the false negative results (values in the top right quadrant of Tables 2 and 3), and IP are the incorrect positive results (incorrect values in the top left quadrant of Tables 2 and 3). Note that we use the precision and recall definitions common to binary classification problems, but in the definition of the accuracy metric we divide by the total number of samples including IP which is more typical of a multi-class classification problem [25].

Additionally, the decision tree outputs are structured in machine-readable format and can be used directly in downstream quantitative analysis. The comparable accuracy between the training and test documents supports the extensibility of this strategy to a larger set of ordinance documents.

These results are comparable to small-scale (around 20 counties) internal validation showing that the original human-developed wind ordinance database [11] is also between 85 % and 90 % accurate. This result is understandable when considering the immense number of human labor hours required for the initial development of the wind ordinance database (1500 h), the repetitive nature of the work required, and the repeated identification of fatigue as a common driver in human errors [26]. In fact, even in the development of the 85-document test set, ambiguity and fatigue led to disagreement between the two researchers tasked with development of the "ground truth" dataset. This was resolved upon lengthy discussion and deliberation between the two researchers, but it highlights the difficulty of this problem. An example of the decision tree strategy extracting the structure setback data from the Monroe, Wisconsin ordinance document is shown in Fig. 2. This example is highlighted because it requires some of the most complex mathematical and logical reasoning found in ordinance documents. The ordinance states different setback requirements from three types of structures, with some types of structures having multiple options to calculate and compare. The LLM shows its work while calculating all possible effective setback distances and is able to select an appropriate final value based on the guidance provided. In contrast, experiments with a single-prompt LLM containing the same guidance returned “the setback is 3.1 times the maximum blade tip height” without showing its work. This is a reasonable output, but it is incorrect given the provided guidance.

An additional benefit of the decision tree method not represented in the accuracy metrics is the ability for the decision to act as a classification model in addition to a data extraction utility. One node of the decision tree requests the LLM to categorize the ordinance based on a set of archetypes from the original ordinance database (e.g., tip-height multiplier, fixed distance value, etc.…). The result is that the type of setback can be easily determined from the decision tree history and can be output in a machine-readable format. This enables the scalability of this method to the full wind ordinance database.

A common concern on using LLMs for automation is the stochasticity of the models. LLMs have several hyperparameters that attempt to control the variability of the outputs, such as the sampling temperature [23,27]. During the generation step, the LLM selects the next output token from several likely options. A temperature of 0 corresponds to greedy sampling where only the most likely next token will be selected. A larger temperature causes the model to select the output token from a probability distribution of the several most likely next tokens. Lower temperatures are therefore considered more deterministic and better for focused high-accuracy applications, although in practice this does not eliminate all stochasticity from the LLM outputs. To explore how the implicit stochasticity impacts the reliability of the ordinance data retrieval, we select five counties from our "test" set and run the LLM-powered ordinance data retrieval 10 times for each county with the temperature set to 0 (note that we used a temperature of 0 for all the results in this work). We find that in the structured output dataset, we see an average 4.8 % variability rate. That is, when we collect the 13 targeted ordinance types for a given county 10 times (a total of 130 collected ordinance values), we see on average 6 values deviate from the most common value for each ordinance type. However, this can vary from county-to-county, with Beadle, South Dakota exhibiting a 10.8 % variability rate, while Palo Alto Iowa exhibited a 0.0 % variability rate. It may be possible to further reduce this rate by running the model several times and selecting the most common response, or by attempting to reduce the number of prompt interactions in the decision tree to give the LLM less opportunities for variability. Also, this is very likely a topic of research in the private industry, and future LLMs may exhibit less variability.

Given that the LLM-powered ordinance data extraction does not provide perfect accuracy, and that it exhibits variability in output even with the same input ordinance document, it is crucial to understand how the model might fail and what the consequences might be. Below, we further describe the three failure modes we observed in the training and test results.

The first failure mode is a false positive where the LLM identifies a regulation that is not present. In our initial tests we found that the LLM can "hallucinate" fictional regulations. This was primarily during experiments using the LLM to reformat massive bodies of text. The LLM appeared to perform poorly when requested to output a large body of text that was a highly faithful reproduction of the input prompt. This problem was mostly eliminated by explicitly instructing the LLM to not add additional text, by only using the LLM to extract small chunks of relevant text from the larger document, and by implementing a heuristic N-Gram similarity check [28] to further prevent large-scale hallucinations. In the final experiments, this failure mode occurred between 1 and 3 % and appeared to be primarily due to the LLM mistaking a real regulation as applying to an erroneous feature (e.g., assuming that a setback applied to structures and property lines even though it only applied to structures).

The second failure mode is a false negative where the model fails to retrieve an ordinance that was present in a document. A major cause of this error type was when the PDF conversion process outputted poorly formatted text. The LLM also sometimes failed to recognize text as relevant to the question, although this was rare. In training, this was the most common error mode occurring about 9 % of the time, almost double each of the other two failure modes. However, when applying the LLM to the larger test set, the false negative rate decreased to 4 % which was only slightly more common than the other two failure modes. We observed more false negatives in the training set versus the test set primarily due to errors in Ottawa, Michigan and Lawrence, South Dakota. The LLM failed to identify eight ordinances from these two counties, nearly half of the false positives from the 18-county training set. It is not immediately clear why these documents were uniquely difficult for the LLM, as the ordinance values successfully made it through the text extraction and distillation. The LLM may have incorrectly distinguished the difference between participating and non-participating structure and property lines, or it could simply be due to the stochastic nature of the LLM that these ordinances were difficult to parse. Based on the results from the test set, it appears that these counties may be outliers and that in a larger set these false negatives are rare. Nevertheless, this serves to illustrate that parsing documents using LLM's is an intrinsically stochastic process that may result in errors that are difficult to attribute.

The last failure mode occurred when the LLM found an ordinance, but it was not the correct ordinance for the request. As an example, many ordinance documents specify regulations for different types of residences and for different sizes of turbines, which can sometimes confuse the LLM especially when the text formatting is not intuitive. Anecdotally, there were also several cases where researchers did not initially agree on the correct interpretation of the ordinance text, so clearly this failure mode is not limited solely to LLMs.

In all failure modes, the frequency of such errors is relatively small, but persistent. We posit that an LLM-driven approach to collecting large amounts of structured data from unstructured text will never be guaranteed to be perfectly accurate given the heterogeneity of legal text and the tendency of LLMs to make simple mistakes. However, we also want to note that a human-driven approach will have similar errors and based on the limited internal review described above it seems that the LLM and human error rates are comparable. Energy developers should still work with licensed attorneys to ensure their projects adhere to local ordinances. This is discussed further in the next section.

The impact of these LLM errors on downstream energy analysis can be estimated based on previous studies, which show that more severe wind energy siting restrictions reduce wind deployment in a least cost electricity system [10]. In this context, false negatives in ordinance detection would artificially increase the expected wind deployment while false positives would artificially decrease the expected wind deployment. However, when looking at the magnitude of cause and effect reported by Mai et al. [10], we see that the siting regimes in question have large bounds: the Open Access siting regime has 6.7x greater technical potential available than the Limited Access regime. So, while errors in ordinance detection will affect the estimated technical potential, errors at rates reported in Table 3 are minimal compared to the uncertainties in future siting regimes being considered. This is, of course, only one potential application of the siting ordinances data, and other applications should perform their own assessments of whether the error rates in Table 3 are consequential.

4. Discussion

The results of this paper support the use of LLMs in the continued maintenance of national-scale wind siting ordinances. However, the software used in this work may also enable similar national-scale data retrieval for analysis of utility rate structures [29], challenges in transmission siting [30], and other impacts of state-to-state variation in socio-political context [31]. The following paragraphs discuss the applicability and limitations of such strategies with respect to the lessons learned from the work presented here.

First, the challenge presented by maintaining a national database of wind siting ordinances is an ideal task for automation by LLMs because it is based on reading comprehension that has historically required significant human effort and is prone to errors from human fatigue. Although LLMs have developed rudimentary data analysis skills, their primary value proposition is still rooted in text-based reading comprehension. That is why we focus on the application of LLMs to policy data, which is typically stored in unstructured text. Still, even though the results presented here are promising, a significant remaining challenge is handling heterogeneity and unexpected ordinance structures in new documents. The decision tree framework implemented here relies on a pre-programmed prompt tree developed by subject matter experts for the wind ordinances in the 18 document “training” set. We found that the documents in the training set were sufficiently diverse to enable model to successfully parse the 85 document “test” set, but it is a near certainty that there will be never-before-seen ordinance definitions in a nation-wide survey that would present unique challenges to the decision tree. Adaptive prompting strategies to handle unforeseen challenges may be the focus of future research. Alternatively, the decision tree can be configured to flag these unique regulations for human review while still reducing the majority of the document review burden.

Applications of LLMs are discussed here as tools for policy research and not as certified legal counsel. LLM outputs are best used in applications that are not safety-critical and have no risk asymmetry. That is, even if we engineer a process in which LLMs can be highly accurate, we should always ensure that inaccurate LLM responses will not result in severe real-world consequences. Users of the ordinance database should take steps to minimize the impacts of inaccurate values, and developers designing a wind plant should still consult the local zoning codes with help from the appropriate legal counsel. The value of such national-scale data comes from the ability to perform comparative analysis across regions, to maintain an understanding of national trends in renewable energy policy, to identify opportunities for effective future renewable energy policies, and to act as an initial screening tool for development opportunities. The distributed nature of the problem is important to highlight here: a single incorrect ordinance resulting from an inaccurate LLM response (or from human effort) will not invalidate the larger research and analysis effort. Indeed, based on the results presented here, we should expect the LLM-driven process to result in occasional errors. However, based on a limited investigation of the wind ordinance database by Lopez et al. [11], we assert that the LLM approach has similar accuracy to a manual human-driven collection process. This should not be surprising given the significant human fatigue resulting from the repetitive reading task that can lead to errors [26].

When considering the efficiency benefits of the LLM-driven process, we can compare the efforts outlined in this paper to the initial manual effort by Lopez et al. [11], which we have previously stated required 1500 labor hours to collect and would require a similar effort to manually update. The work presented here required approximately 500 labor hours, including time for experimentation, software development, validation, and the writing of this manuscript. We estimate that scaling this work to all counties in the United States would only require 10–20 % of the original development effort (50–100 h) along with approximately $6000 in LLM service costs (calculated from Azure GPT-4 rates as of April 2024). A similar number of labor hours (50–100 h) would be required to repeat the national assessment for subsequent updates. Note that this only includes the document processing. The automated retrieval of such documents from the internet is still in development and could double this estimate. In summary, although the initial LLM framework development was considerable, we estimate that it would require much less human labor for a regularly updated and maintained process. This follows the general motivation for automation: the initial software development can be costly and much more expensive than a small manual effort, but there is typically a point at which automation becomes much more cost effective for a sufficiently large and repetitive task.

Regarding data privacy, it is worth noting that the application of LLMs to zoning ordinance documents presents no data privacy or intellectual property challenges. All policies analyzed in this document are publicly available and contain no sensitive information. It is still somewhat unclear how text submitted in LLM prompts may be used by LLM providers, and policies vary from provider to provider (e.g., OpenAI and Google may have different policies on data privacy with respect to their LLMs). Applying LLMs to sensitive information is possible with some technology providers, but such applications add additional challenges to the application of this new technology.

Lastly, we believe that great care should be taken in the application of LLMs to any real-world challenge. We describe our experimental setup in Section 2 for testing the LLM strategies on documents that are hidden from the primary developer. We plan to implement a sampled validation procedure when working with the full set of ordinance documents and recommend similar validation exercises be performed when applying LLMs to new problems. These models are black boxes with little transparency behind their decision-making processes even when asked to show their work. Applying an LLM as an analysis tool to a pre-existing body of text reduces the likelihood that the model will generate fictional responses (e.g., “hallucinations”), but it is still possible. In this work, we found that we could greatly reduce the risk of fictional responses by stating that the model was only allowed to return text from the original legal document, by reducing the amount of text we expected the model to output, and by always explicitly providing an option for the model to provide a negative answer (e.g., “there is no such ordinance in the provided text”).

5. Conclusion

The effort required to collect local policy data at a national-scale is considerable and many institutions do not have sufficient resources to successfully execute such work, even if the results would be of significant value to the energy industry and research community. The challenge of ensuring accurate information in a constantly changing industry presents an additional burden, which may incur similar costs but yield lesser benefits compared to initial endeavors.

In this work, we present strategies for the automated extraction of wind turbine setback data from ordinance documents. We show that an LLM guided by a decision tree framework can reliably extract siting ordinances from legal documents while requiring little human supervision or review. The results are not perfectly accurate and the LLMs do make mistakes when extracting ordinance data. However, based on a limited investigation of previous ordinance data, we assert that the LLM-driven approach performs with similar accuracy to a manual collection process. Accordingly, this work is not appropriate for applications with zero tolerance for errors. The results are intended primarily for research purposes and are no substitute for certified legal counsel when designing and developing a real-world renewable energy project. The true value proposition of this work is to enable high-quality policy analysis in a rapidly-evolving sector at an unprecedented scale. At a minimum, the methods presented here appear to be a viable path forward for maintaining up-to-date information on national wind siting ordinances with fair accuracy.

Not explored here is the possibility for LLMs to enable retrieval of such documents from the internet via automated search. These methods are still in development and the expanding offerings from the private sector are also rapidly improving what is possible on this front. However, we are optimistic that LLMs can assist with the automation of a significant portion of this task, especially given the strong baseline of known ordinances in the original U.S. Wind and Solar Siting Regulation and Zoning Ordinances Databases [11,12].

The software and data used in this work are open-source and available at no cost in the associated github repository https://github.com/NREL/elm version 0.0.5 [32]. The ordinance documents used in this report and the results from the experiments are available here: https://github.com/NREL/elm/tree/main/examples/ordinance_gpt. The results in this work can be replicated by downloading the ordinance documents from the included link and running the “parse_pdf.py” example code. Some variability is to be expected even with the same LLM model deployment as described in Section 3. Details of the decision tree structure used to extract the ordinance values can be found in the following module: https://github.com/NREL/elm/blob/main/elm/ords/extraction/graphs.py.

Note that GPT-4 application programming interface (API) was used in the development of this work. Specifically, the Microsoft Azure "2023-03-15-preview" deployment of GPT-4 was used. At the submission of this paper, the GPT-4 API is not freely available to the public and requires a paid subscription with OpenAI or Microsoft Azure.

Disclosures

During the preparation of this work the authors used GPT-4 as a central piece of the experimental methods. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

CRediT authorship contribution statement

Grant Buster: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Pavlo Pinchuk:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jacob Barrons:** Validation, Data curation. **Ryan McKeever:** Validation, Data curation. **Aaron Levine:** Writing – review & editing, Supervision, Project administration. **Anthony Lopez:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Grant Buster reports financial support was provided by US Department of Energy Wind Energy Technologies Office. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank Keith Benes, Patrick Gilman, Dan Bilello, Meghan Mooney, and Taylor Curtis for their thoughtful reviews of an earlier draft.

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36–08GO28308. Funding provided by the DOE Wind Energy Technologies Office (WETO), and internal research funds at the National Renewable Energy Laboratory. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

Data availability

The software and data used in this work are open-source and available at no cost in the associated github repository https://github.com/NREL/elm version 0.0.5 [32]. The ordinance documents used in this report and the results from the experiments are available here: https://github.com/NREL/elm/tree/main/examples/ordinance_gpt.
The results in this work can be replicated by downloading the ordinance documents from the included link and running the "parse_pdf.py" example code. Some variability is to be expected even with the same LLM model deployment as described in Section 3. Details of the decision tree structure used to extract the ordinance values can be found in the following module: https://github.com/NREL/elm/blob/main/elm/ords/extraction/graphs.py.

Note that GPT-4 application programming interface (API) was used in the development of this work. Specifically, the Microsoft Azure "2023-03-15-preview" deployment of GPT-4 was used. At the submission of this paper, the GPT-4 API is not freely available to the public and requires a paid subscription with OpenAI or Microsoft Azure.

References

[1] U.S. Energy Information Administration, *EIA-860 Annual Electric Generator Report*, 2022. Available at: https://www.eia.gov/electricity/data/eia860/.
[2] U.S. Energy Information Administration, *Electric Power Monthly*, February 2023, preliminary data for 2022. Available at: https://www.eia.gov/electricity/monthly/.
[3] Outka U. Renewable energy siting for the critical decade. Kansas Law Review, Inc 2021. Available at SSRN: https://ssrn.com/abstract=3816920.
[4] Ela E, Wang C, Moorty S, Ragsdale K, O'Sullivan J, Rothleder M, et al. Electricity markets and renewables: a survey of potential design changes and their consequences. IEEE Power Energy Mag 2017;15:70–82. https://doi.org/10.1109/MPE.2017.2730827.
[5] Lopez A, Cole W, Sergi B, Levine A, Carey J, Mangan C, et al. Impact of siting ordinances on land availability for wind and solar development. Nat Energy 2023: 1–10. https://doi.org/10.1038/s41560-023-01319-3.
[6] Eisenson M. Opposition to renewable energy facilities in the United States: may 2023 edition. Sabin Center for Climate Change Law 2023. Available at: https://scholarship.law.columbia.edu/sabin_climate_change/200/.
[7] Biden J., Executive order #14008: "executive order on tackling the climate crisis at home and abroad". January 27th, 2021, Fed Regist, 86 FR 7619. Available at: https://www.federalregister.gov/d/2021-02177.
[8] Larson E, Greig C, Jenkins J, Mayfield E, Pascale A, Zhang C, et al. Net zero america: potential pathways, infrastructure, and impacts, final report. Princeton, NJ: Princeton University; 2021.
[9] Denholm P, Brown P, Cole W, Mai T, Sergi B, Brown M, et al. Examining supply-side options to achieve 100% clean electricity by 2035. Golden, CO (United States): National Renewable Energy Lab. (NREL); 2022. https://doi.org/10.2172/1885591.
[10] Mai T, Lopez A, Mowers M, Lantz E. Interactions of wind energy project siting, wind resource potential, and the evolution of the U.S. power system. Energy 2021; 223:119998.
[11] Lopez A., Levine A., Carey J., Mangan C. "U.S. Wind siting regulation and zoning ordinances". OpenEI 2022. Web. 10.25984/1873866.
[12] Lopez A., Levine A., Carey J., Mangan C. "U.S. Solar siting regulation and zoning ordinances". OpenEI 2022. Web. 10.25984/1873867.
[13] OpenAI. GPT-4 Technical Report 2023. 10.48550/arXiv.2303.08774.
[14] Bubeck S., Chandrasekaran V., Eldan R., Gehrke J., Horvitz E., Kamar E., et al. Sparks of artificial general intelligence: early experiments with GPT-4 2023. 10.48550/arXiv.2303.12712.
[15] Hadi M.U., Tashi Q.A., Qureshi R., Shah A., Muneer A., Irfan M., et al. Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects 2023. 10.36227/techrxiv.23589741.v3.
[16] Poppler n.d. https://poppler.freedesktop.org/(accessed October 27, 2023).
[17] Liu, N.F. et al. Lost in the middle: how language models use long contexts. Preprint at http://arxiv.org/abs/2307.03172 (2023).
[18] Navada A, Ansari AN, Patil S, Sonkamble BA. Overview of use of decision tree algorithms in machine learning. In: Proceedings of the 2011 IEEE control and system graduate research colloquium; 2011. p. 37–42. https://doi.org/10.1109/ICSGRC.2011.5991826.
[19] Garcez A d'A, Lamb LC. Neurosymbolic AI: the 3rd wave. Artif Intell Rev 2023;56: 12387–406. https://doi.org/10.1007/s10462-023-10448-w.
[20] Long J. Large language model guided tree-of-thought 2023. 10.48550/arXiv.2305.08291.
[21] Yao S., Yu D., Zhao J., Shafran I., Griffiths T.L., Cao Y., et al. Tree of thoughts: deliberate problem solving with large language models 2023. 2024 10.48550/arXiv.2305.10601.
[22] Lewis P, et al. Retrieval-Augmented generation for knowledge-intensive NLP tasks. Advances in neural information processing systems, 33. Curran Associates, Inc.; 2020. p. 9459–74.
[23] Majumder S, Dong L, Doudi F, Cai Y, Tian C, Kalathil D, et al. Exploring the capabilities and limitations of large language models in the electric energy sector. Joule 2024;8:1544–9. https://doi.org/10.1016/j.joule.2024.05.009.
[24] Zhang Z., Zhang A., Li M., Smola A. Automatic chain of thought prompting in large language models 2022. 10.48550/arXiv.2210.03493.
[25] Berrar D. Performance Measures for Binary Classification. In: Ranganathan S, Gribskov M, Nakai K, Schönbach C, editors. Encyclopedia of bioinformatics and computational biology. Oxford: Academic Press; 2019. p. 546–60. https://doi.org/10.1016/B978-0-12-809633-8.20351-8.
[26] Folkard S, Lombardi DA. Modeling the impact of the components of long work hours on injuries and "accidents. Am J Ind Med 2006;49:953–63. https://doi.org/10.1002/ajim.20307.
[27] Renze M., Guven E. The effect of sampling temperature on problem solving in large language models 2024. 10.48550/arXiv.2402.05201.

[28] Kondrak G. N-gram similarity and distance. In: Consens M, Navarro G, editors. String processing and information retrieval. Berlin, Heidelberg: Springer; 2005. p. 115–26. https://doi.org/10.1007/11575832_13.

[29] Ong S, Denholm P, Doris E. Impacts of commercial electric utility rate structure elements on the economics of photovoltaic systems. Golden, CO (United States): National Renewable Energy Lab. (NREL); 2010. https://doi.org/10.2172/983405.

[30] Vajjhala SP, Fischbeck PS. Quantifying siting difficulty: a case study of US transmission line siting. Energy Policy 2007;35:650–71. https://doi.org/10.1016/j.enpol.2005.12.026.

[31] Wilson EJ, Stephens JC. Wind deployment in the United States: states, resources, policy, and discourse. Environ Sci Technol 2009;43:9063–70. https://doi.org/10.1021/es900802s.

[32] Pinchuk P., Buster G., Podgorny S. Energy language model (ELM) version 0.0.5. 2024 10.5281/zenodo.12710491.