Chat2VIS: Multilingual and Customisable Chart Generation using Large Language Models

Journal:	*Transactions on Visualization and Computer Graphics*
Manuscript ID	TVCG-2023-05-0238.R1
Manuscript Type:	Regular
Keywords:	I.2.1.l Natural language interfaces < I.2.1 Applications and Expert Knowledge-Intensive Systems < I.2 Artificial Intelligence <, H.5 Information Interfaces and Representation (HCI) < H Information Technology and Systems, I.6.9.c Information visualization < I.6.9 Visualization < I.6 Simulation, Modeling, and Visualization < I Computing Methodologie, I.2.7.c Language models < I.2.7 Natural Language Processing < I.2 Artificial Intelligence < I Computing Methodologies

Chat2VIS: Multilingual and Customisable Chart Generation using Large Language Models

Paula Maddigan and Teo Susnjak

Abstract—In our data-saturated world, there is a pressing need to harness technology to derive insights. Yet, traditional tools often require significant learning overheads to work with complex charting techniques. Such barriers can hinder those who may benefit from harnessing data for informed decision-making. However, generating data visualisations directly from natural language text (NL2VIS) is an emerging field that addresses this issue. This study showcases Chat2VIS, a state-of-the-art NL2VIS solution for conversational chart generation. This work capitalises on the latest AI technology leveraging large language models (LLMs) such as GPT-3, Codex, and ChatGPT to generate and refine data visualisations conversationally with capabilities that are beyond those demonstrated in previous studies. In addition, this paper presents the novel capability of Chat2VIS to comprehend multilingual natural language requests. Our work is evaluated against two NL2VIS benchmark datasets. In the process, we propose an automated methodology for conducting evaluations which are otherwise performed both manually and infrequently. We contribute findings and recommendations going forward with respect to improving the development of benchmark datasets for NL2VIS in order to enable automated evaluations, and in the process facilitate an acceleration of advancements in this field.

Index Terms—ChatGPT, Codex, GPT-3, end-to-end visualisations from natural language, large language models, natural language interfaces, text-to-visualisation.

1 Introduction

In an era where data has become a valuable commodity, industries are continuing to witness immense growth in its volume. Data visualisations offer an effective and compelling approach to communicating insights from this resource to facilitate better-informed business decisions. The ability to articulate visualisation requests through natural language (NL) text and intuitive interfaces is fast gaining traction [1]. It is an enticing objective to generate suitable charts without the need to acquire programming skills and undergo arduous learning curves associated with visualisation tools [2]. Therefore, in our quest to democratise access to these data visualisation tools and make them more user-friendly, the emerging field of Natural Language to Visualisation (NL2VIS) is poised to transform the way we interact with, and understand data [3].

Despite the existence of the NL2VIS field for two decades, the challenge remains in devising end-to-end systems that can perform multiple tasks like interpreting complex user intents with their inherent ambiguities, automatically selecting appropriate visualisation types, and transforming parsed instructions into visual outputs. Recently, the surge in interest in large language models (LLMs) is driving research to develop NL2VIS with end-to-end capabilities that leverage this advanced AI technology.Since these language models are trained on a large amount of both language texts and code repositories, they exhibit a high level of skill in language semantics and code scripting, which makes them ideal candidates for solving this difficult problem. Therefore, this study investigates the capabilities of OpenAI's GPT-3, Codex, and ChatGPT models in advancing end-to-end data visualisation systems and reports their accuracies against benchmark datasets.

One remarkable feature of the conversational capabilities of advanced LLMs is their unique ability to maintain a coherent dialogue and build on prior exchanges. This conversational capability for iteratively customising a chart has also not been adequately resolved in the field of NL2VIS and represents an existing need [1]. This study showcases how this can indeed be addressed via LLMs, transcending the capabilities of prior NL2VIS architectures in literature.

Recent literature [1] also highlights the lack of multilingual support in existing NL2VIS systems whose capabilities normally only encompass English. Meanwhile, advanced LLMs are inherently capable of multilingual comprehension due to their expansive training data, which encompasses a multitude of languages from various text sources. This extensive linguistic diversity in their training corpus enables LLMs to decode text from multiple languages, albeit, the performance profiles of LLMs across different languages can also be uneven due to the varying representation of languages in their training data. To that end, this research probes the proficiency of the proposed system to generate visualisations from queries originating from several languages which have different levels of representation in the training corpus of the LLMs.

Contribution

The contribution of this study is fourfold. This work advances the field of NL2VIS by presenting novel features

• *P. Maddigan and T. Susnjak are with the School of Mathematical and Computational Sciences, Massey University, Auckland, New Zealand.*

E-mail: paula.maddigan@gmail.com, t.susnjak@massey.ac.nz

Manuscript received May 12, 2023

within the LLM-based Chat2VIS framework. The first is the demonstration of the customisable capabilities of the Chat2VIS framework to incorporate iterative refinements to queries to adjust the generated charts and its aesthetics. We show that the range of customisation options exceeds solely refining a predefined set of chart components as demonstrated by previous NL2VIS approaches, but are instead much broader. This work addressed a second gap in literature by demonstrating the capacity of Chat2VIS to comprehend multilingual NL texts across seven diverse languages and to generate data visualisations with customisations, establishing it as a truly global tool.

Thirdly, we tackle a pressing issue in the current state of NL2VIS: the lack of robust benchmarking tools. The evaluation of NL2VIS systems is often marred by the inherent subjectivity of human judgement, and the manual evaluation approaches are laborious, requiring teams of assessors. In response, this work develops an automated approach for conducting a quantitative analysis of NL2VIS performance which is a first step towards a comprehensive solution. We present the results of our evaluation of Chat2VIS against two benchmarks, highlighting the challenges of developing structured methodologies and measurable baseline standards for NL2VIS. This contribution is valuable as the establishment of effective benchmarks can expedite advancements in this domain, especially if evaluations can be automated. Our study underscores the importance of ensuring that benchmarks fulfill the quality characteristics of *reproducibility*, *fairness*, and *verifiability* [4]. As more benchmark datasets begin to emerge in this domain, we emphasise the need for them to incorporate a well-defined measurement methodology, outlining the process to implement the standard, collect measurements, and evaluate the results [4]. Finally, the developed software artefact has not only been made available to the public via an online portal, but the source code has also been released for researchers to further develop [1].

2 Related Work

Early NL2VIS systems were built on symbolic-based NLP approaches, relying on heuristic algorithms [5], rule-based architectures, and probabilistic grammar-based methods for translating NL queries. Some of the earliest attempts can be traced to 2001 with initial prototype relying only on well-structured queries called InfoStill [6]. Although each subsequent technique displayed incrementally improving accuracies, they also required increasing amounts of computational resources for modest performance improvements. Systems such as Articulate [7], DataTone [8], Eviza [9], and Deep-Eye [10] all used varying symbolic NLP methodologies in translating NL to data visualisations. However notable approaches like NL4DV [11] and FlowSense [12] employed NLTK [13], NER, and Stanford CoreNLP [14] semantic parsers to improve accuracy. Readers are directed to two recent surveys ([1] and [15]) on NL2VIS which delve deeper into the evolution of the field.

Recent advancements in NL2VIS have focused on deep-learning models to achieve greater levels of adaptability, robustness and flexibility compared to achievable performances of previous approaches [16]. Systems such as ADVISor [17] are supported by BERT [18], a large language transformer-based model where the rendered visualisation styles are predetermined based on a defined mapping rule.

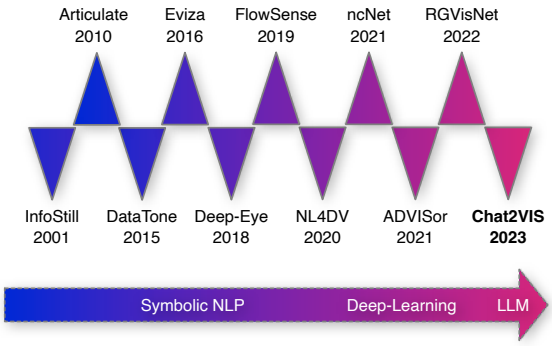

Fig. 1. NL2VIS timeline illustrating the evolution of NL2VIS systems.

An alternative transformer-based approach, ncNet [3], is a machine learning model trained using the nvBench [19] dataset. The model accepts an optional chart template in addition to the requested NL query to guide chart styling of the rendered visualisation. The system has recently been expanded to include speech-to-visualisation capabilities [20].

Furthermore, the hybrid approach of RGVisNet [21] initially retrieves the most relevant visualisation query from a large-scale visualisation codebase. It then revises it via a GNN-based deep-learning model, and subsequently generates the visualisation.

The evolution of NL2VIS systems are illustrated in Fig. 1, depicting the transition from symbolic NLP towards deep learning approaches. With the evolution, increasingly the end-to-end capabilities feature with newer frameworks which integrate multiple components of a NL2VIS pipeline, including natural language understanding, information extraction, visualisation (code) generation into a unified solution. Unlike fragmented approaches in the literature, end-to-end solutions automate the entire process, eliminating the need for multiple components, thus making them more efficient. The latest state-of-the-art artefact, Chat2VIS [22], presents the first NL2VIS NLI to generate data visualisations via LLMs that has the ability to support end-to-end processes. It addresses the next generation of NL2VIS architecture, simplifying the NL2VIS pipeline by offloading language understanding, chart selection and reasoning in the presence of ambiguity, as well as code generation to a single system. The underlying structure provides flexibility and robustness around free-form and complex visualisation requests while decisions pertaining to suitable chart selection and aesthetics are delegated to the LLMs. The architecture underpinning Chat2VIS is exceptionally flexible and decidedly diverse enough to further refine charting elements using NL without additional enhancements to the NL2VIS architecture. This is the first study to address this gap evident in earlier systems. In addition, unseen in previous approaches, this work demonstrates the art of fulfilling multilingual requests with ease, omitting the need for additional prompting, further architectural manipulations,

1. The source code for the Chat2VIS is available at https://github.com/frog-land/Chat2VIS_Streamlit

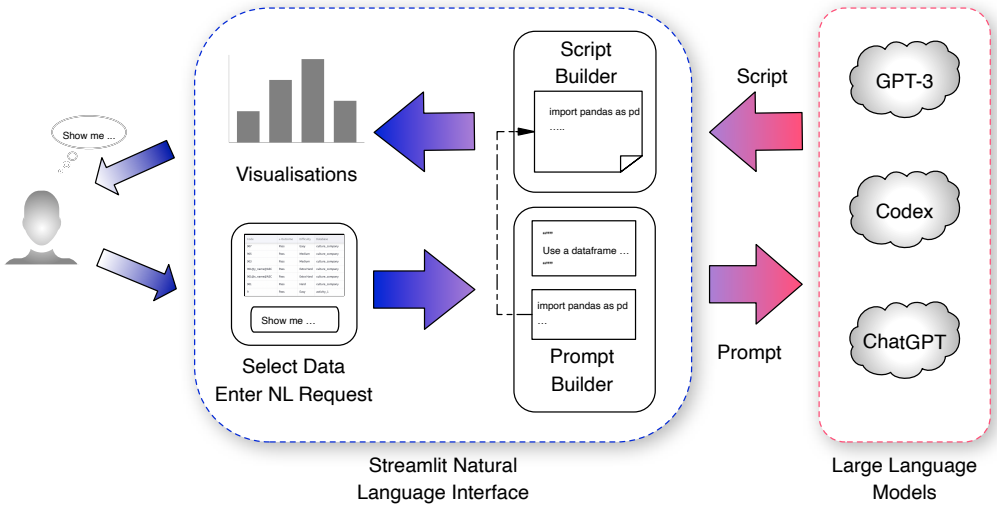

Fig. 2. The Chat2VIS architecture translating NL text into data visualisations via large language models.

or model retraining.

With the sparse existence of NL2VIS benchmarks, we seek to evaluate Chat2VIS against the only two baselines nvBench [19] and the NLV utterance corpus [23] identified in the existing literature. Evaluations [21] [23] against these benchmarks for current NL2VIS approaches provide a degree of comparison for this study. To that end, our analysis contributes to the gap distinctly evident in the literature regarding NL2VIS benchmarking.

3 NL2VIS Architecture

Chat2VIS generates data visualisations using free-form NL text. With an interface utilising OpenAI's state-of-the-art LLMs, it demonstrates unique decision-making skills to autonomously select chart types and plot elements.

3.1 Large Language Models

Chat2VIS is based on the Davinci family of models, currently some of the most capable and advanced model set available within the OpenAI suite. It employs GPT-3 model *"text-davinci-003"*, Codex model *"code-davinci-002"*, and contrasts results with the latest state-of-the-art ChatGPT model *"gpt-3.5-turbo"*.

Using OpenAIs text completion endpoint API to access Codex and GPT-3 models, it retains default parameters with the exception of adjustments to the following:

1) Temperature is a hyperparameter controlling output randomness in LLMs. A lower temperature (closer to zero) makes the model's outputs more deterministic and consistent, ideal for tasks like code generation. For this reason, we set the temperature to zero;
2) Evading excessively verbose scripts by setting the max_tokens parameter to 500 — an ample limit for this study; and
3) Requesting a stop parameter of *"plt.show()"*. This will cease generation upon plot rendering syntax - avoiding the LLMs presenting alternative scripts.

3.2 Chat2VIS

The Chat2VIS application[2] has been developed using Streamlit[3], an open-source Python library that enables developers to create interactive web applications rapidly. The adoption of Streamlit offers an effective means to encapsulate several components of the NL2VIS process, including user interface design, prompt engineering, LLM connectivity, and the subsequent generation and rendering of visualisations from the received scripts. As shown in Fig. 2, the architecture of Chat2VIS is designed with an interface (refer to Fig. 3) that allows users to interact with the application by entering a NL request. The request is specific to a selected dataset that the user wants to visualise. Upon receipt of the user's NL request, Chat2VIS begins the process of engineering the prompt. This process involves combining the NL request with a standardised prompt template. This template is common across all LLMs and is designed to include information about the data types present in the chosen dataset. This engineered prompt is then submitted to the selected LLM. The LLM returns a result that contains the Python code component, which is extracted by Chat2VIS. This Python code represents the visualisation instructions derived from the user's NL request. The application then executes this Python code within its environment and renders the result.

Fig. 4 illustrates the inner workings of the end-to-end capabilities of Chat2VIS (which does not rely on generating intermediate representations in the form of JSON like some systems), while a more concrete example of the prompt structure is seen in Appendix A in Fig. 12. The architecture is discussed by way of an example dataset created from the results of our benchmarking evaluation in Section 5.4. The process is described as follows:

1) Fig. 4(a) shows a sample of the dataset together with the query *"Plot the outcome."*, chosen for its ambiguity that lacks the explicit instruction of **how** to

2. https://chat2vis.streamlit.app/
3. https://streamlit.io/

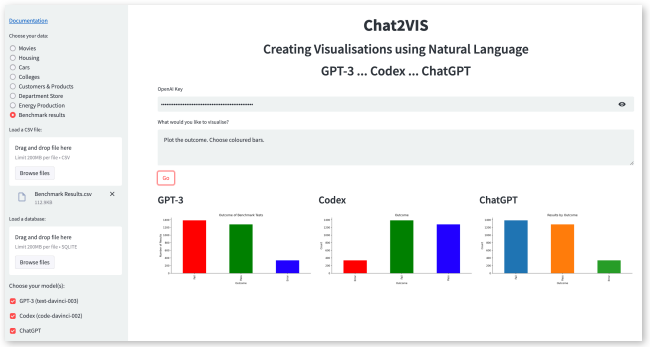

Fig. 3. Chat2VIS Interface

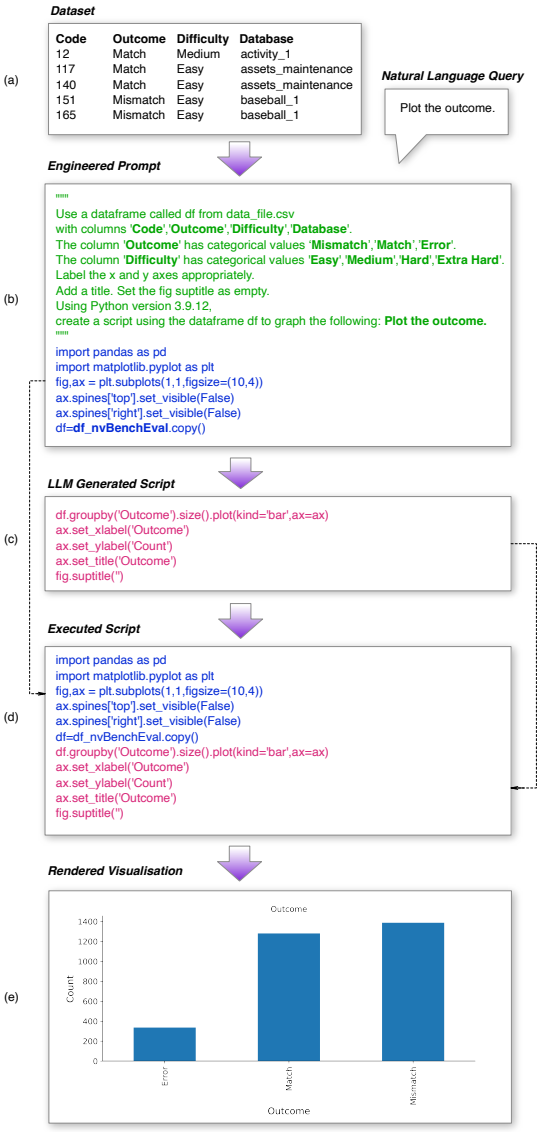

Fig. 4. Illustrated example of the process to convert a NL query into a data visualisation.

derive the desired values and **which** visualisation chart type to use.

2) The engineered prompt in Fig. 4(b) comprises of two parts: (1) a Python docstring description, encapsulated with triple double quotes (green) where columns names together with their data types are defined, including plotting instructions and the Python version to be used; (2) a Python code section providing a starting point for the requested script (blue). The **bolded** type highlights variable substitution values specific to this example.
3) Fig. 4(c) shows the Python script (red) returned by a selected LLM which forms the continuation of the code section within the engineered prompt. The returned script demonstrates both the advanced reasoning and the appropriate chart selection capabilities of the LLMs.
4) In Fig. 4(d), shows the script that is executed to generate the visualisation comprising the combination of the initial Python code (blue) from Fig. 4(a) and the script returned by the LLM (red).
5) The newly-created script is executed to render the requested visualisation, as pictured in Fig. 4(e).

It should be noted that the given prompting architecture has strong privacy-preserving properties. In most cases, only the column names and their data types are sent to the LLM. This mitigates ethical and data privacy risks associated with providing the models with sensitive data. Only in cases where a categorical data type column has less than 20 distinct values, its values are enumerated in the prompt and sent to the LLM; however, even this can be easily resolved by hashing and anonymising those values prior to sending them.

4 Methodology

This study explored (1) the refinement of plot aesthetics using NL queries, (2) the flexibility of Chat2VIS in comprehending multilingual requests, and (3) a quantitative evaluation of Chat2VIS against two benchmark datasets developed in previous studies. All figures in this paper are rendered through Chat2VIS to showcase its abilities.

4.1 Chart Refinements and Multilingual Requests

Previous work [22] confirmed the unique decision-making skills of the LLMs to autonomously select suitable chart types and plot aesthetics. Here, we demonstrate the system's efficacy with iterative refinements to the input query for nominating a specific chart type, enhancing plot elements, and changing styles.

Language choices were driven by the need to ensure semantically coherent requests were formulated, thus languages were prioritised based on the the research team's expertise and fluency. Also, the aim was to achieve a balanced representation of both high-resource languages, such as German and French, and lower-resource languages, like Croatian and especially te reo Māori. High-resource languages have abundant training data available, whereas low-resource languages have limited training data against which the LLMs would have been trained. This deliberate selection allowed the exploration of the performance of LLMs across languages with varying levels of training data exposure.

A multilingual case study is presented exemplifying both visualisation generation and the adjusting of chart labels using Chat2VIS. The results are assessed visually for accuracy.To achieve a wider coverage of more diverse languages, an additional Case Study using Spanish as well as non-Latin languages, like Mandarin and Japanese is included in the Appendix B.

4.2 Quantitative Evaluation

We conduct a more comprehensive quantitative analysis using the nvBench benchmark[4]. Encompassing 153 databases, 7,274 visualisations, and 7 chart types, it is considered the first public large-scale NL2VIS benchmark [19]. Given the size of the dataset, we propose an automated evaluation strategy. Each example instance comprises of a NL-to-visualisation pair, denoted (NL, VIS). Attributes are stored inside a JSON specification, permitting Vega-Lite chart rendering. Examples are further classified into four categories to denote the difficulty of the query — easy, medium, hard, and extra hard.

In addition, we perform a second evaluation using the NLV Utterance dataset[5] [23], referred to in our study as nlvUtterance. The benchmark covered three databases[6]: movies, cars, and superstore. This benchmark comprises 814 NL queries, with 10 visualisations for each database. Queries were generated from the results of an online study using 102 participants suggesting utterances for the display of each respective chart. Here we use a manual evaluation approach.

4.2.1 Model Selection

We select the Codex *"code-davinci-002"* model to measure results against nvBench. Codex, evolved from GPT-3, was trained on an immense amount of publicly-available GitHub code. It is skilled in more than a dozen programming languages, most notably Python, the underlying programming language of Chat2VIS. Codex is available in Davinci or Cushman models. Among the OpenAI suite of models, the Davinci family is the most capable, and can often perform all tasks of other models using fewer instructions. Cushman, although faster, is less competent than Davinci. In prioritising accuracy over speed, the Davinci model was regarded as the most appropriate choice for this task. Therefore, we deemed Codex *"code-davinci-002"* well-suited for this evaluation.

4.2.2 nvBench Benchmark Evaluation

Determining how to automatically assess the equality between a Chat2VIS chart and its nvBench counterpart is technically challenging. The benchmark specification omits guidance of any evaluation methodology determining what constitutes a match or mismatch. Our attempts to employ image comparison tools proved unreliable due to the complexity involved. Hence we devised a strategy that constructs vectors of the x and y coordinates for each plot, and uses these as a basis for comparative analysis.

Since Chat2VIS is designed to generate charts from a tabular dataset, we removed nvBench instances querying multiple SQL database tables to achieve interoperability. This methodology is consistent with the one used for ncNet [3]. The exclusion criterion relied on identifying the SQL `JOIN` operator inside the VQL mark within the nvBench JSON specification. In addition, we excluded examples containing SQL subqueries within the `WHERE` clause referencing tables distinct from the principal `SELECT` clause, again, in order to preserve compatibility.

Due to the difficulty in automating accurate comparisons across all chart types, we confined our benchmark testing to bar charts which constituted an overwhelming majority of samples. Automating the comparisons against chart types like line, pie and scatter would necessitate devising alternative and tailored comparison mechanisms. nvBench includes up to 5 queries for a given visualisation. In our methodology, we chose the first NL query for inputting into Chat2VIS under the assumption that the first one likely represents a most reasonable expression of intent. Fig. 5 illustrates an example JSON specification[7] for the (VIS, NL) pair *"474@x_name@DESC"*, highlighting areas of interest within the specification discussed in this evaluation approach. The final benchmark test set across 138 databases comprised 3,003 instances, with 812 considered *easy*, 1572 *medium*, 386 *hard*, and 233 *extra hard*.

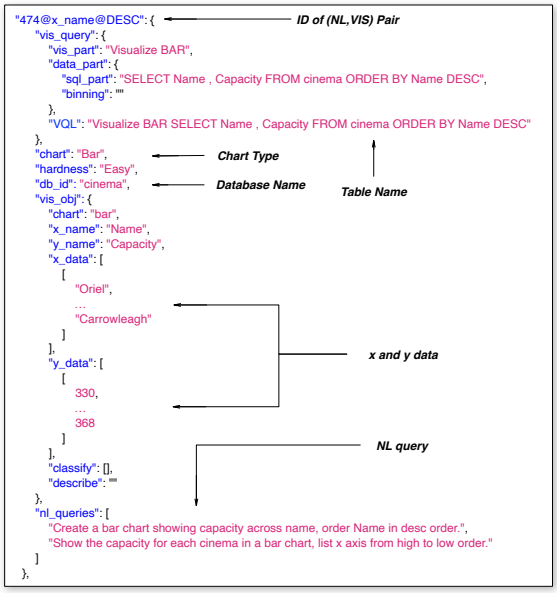

Fig. 5. Example JSON specification from nvBench.

Fig. 6 summarises our proposed automated testing methodology. The steps outlining the methodology are described below:

1) Select a JSON test specification from nvBench.
2) Extract the database name from the db_id field, and the table name specified in the VQL field following the FROM keyword.
3) Import the SQLite database table into a Python Pandas DataFrame structure.
4) Extract the first query from the nl_queries field.

4. https://sites.google.com/view/nvbench
5. https://nlvcorpus.github.io/
6. https://github.com/TsinghuaDatabaseGroup/nvBench/databases.zip
7. https://github.com/TsinghuaDatabaseGroup/nvBench/blob/main/NVBench.json

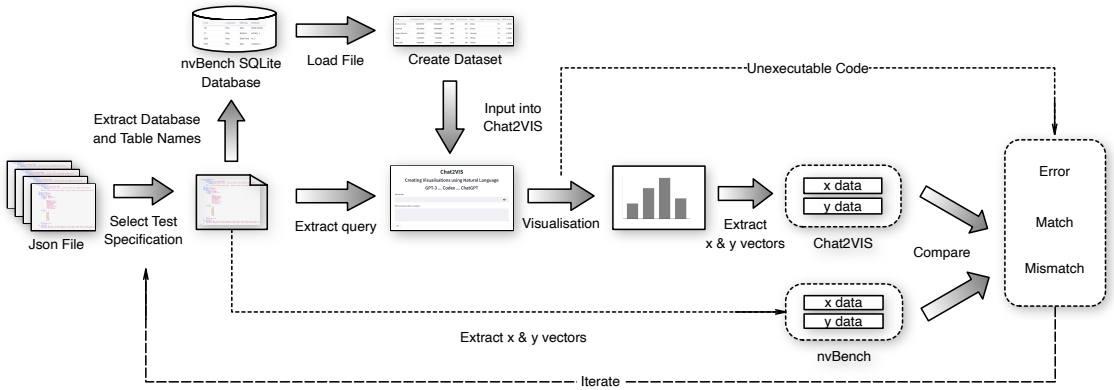

Fig. 6. Overview of nvBench Benchmark Testing.

5) Submit query and dataset to Chat2VIS opting for Codex, and render visualisation.
6) Document the outcome as an `Error` should the code fail to execute.
7) Construct *x* and *y* data coordinate vectors by extracting the Chat2VIS chart elements.
8) Construct *x* and *y* data coordinate vectors using the x_data and y_data fields in the nvBench JSON specification.
9) Apply adjustments to the vectors to address complications impacting a successful comparison:
   - Ensure naming consistency of calendar units, such as recasting Tue, Tues, and tuesday as Tuesday; Sept, Sep september as September.
   - Sort by ascending y values if keywords "sort" and "order" are not specified.
   - Cast integer values to floats, and round all floats to 5dp.
10) Compare Chat2VIS and nvBench vectors. A precise match classifies the outcome as a `Match`, else it is classified as a `Mismatch`, while an `Error` classification indicates that a visualisation failed to be rendered most-frequently attributed to a Python code error.

It should be noted that a `Mismatch` does not necessarily mean that the generated visualisation is incorrect, and may in some instances even be a more effective and appropriate visualisation (see Figure 14 in Appendix C).

4.2.3 nlvUtterance Benchmark Evaluation

A manual visual inspection of the results on the nlvUtterance dataset was used due to its smaller size. All chart types within nlvUtterance were used, namely, bar charts, histograms, line charts, and scatter plots, together with their variations. As outlined in the benchmark description [23]: histograms and single attribute bar charts are used to visualise one categorical or quantitative attribute; bar charts, scatter plots, and line charts for two attributes; and grouped bar charts, stacked bar charts, multi-line charts, coloured scatter, and faceted scatter charts for visualising three or more attributes. 755 of the 814 queries are considered *"singleton"* utterance sets, consisting of a single query request. The remaining 59 instances are considered *"sequential"* utterance sets, containing multiple utterances. After removing erroneous plots without data, the final dataset consisted of 758 queries for testing.

Chat2VIS renders charts for up to 3 models. We employed a three-stage testing methodology. Firstly, the queries are submitted to Codex and the corresponding performance metrics are presented. Secondly, unsuccessful queries are submitted to GPT-3, with the corresponding performance again measured. Finally, any remaining mismatched queries are submitted to ChatGPT. The overall performance statistic for successful matches provides insight into the likelihood that a benchmark result will be generated by at least one LLM.

It is not the intention of this work to compare LLMs *inter se*, but instead contrast the use of LLMs with alternative approaches. Therefore, we do not present benchmark metrics comparing the performance accuracy of Codex, GPT-3 and ChatGPT relative to each other.

5 Results

We demonstrate the conversational ability to refine charts with subsequent requests on the first two case studies, followed by the third case study illustrates multilingual requests. The dataset used for the illustration are results from the nvBench evaluations. Finally, we analyse Chat2VIS' performance against the two benchmarks using the illustrations generated by Chat2VIS.

5.1 Case Study 1: Conversational Chart Refinement

Fig. 7 demonstrates the first conversational refinement of Fig. 4(e) *"Plot the outcome."* to *"Plot the outcome by difficulty."* which illustrates language pragmatics in the clarification of intent. The subsequent refining requests that the chart be rendered *" as a stacked bar plot. Use red for error, light green for match, blue for mismatch."*, followed by a request to refine it with the instruction to *Increase the font size of the axis labels and numbers. Make the title 'Evaluation Results by Difficulty Level' with very large font.*. The figures demonstrate that all LLMs

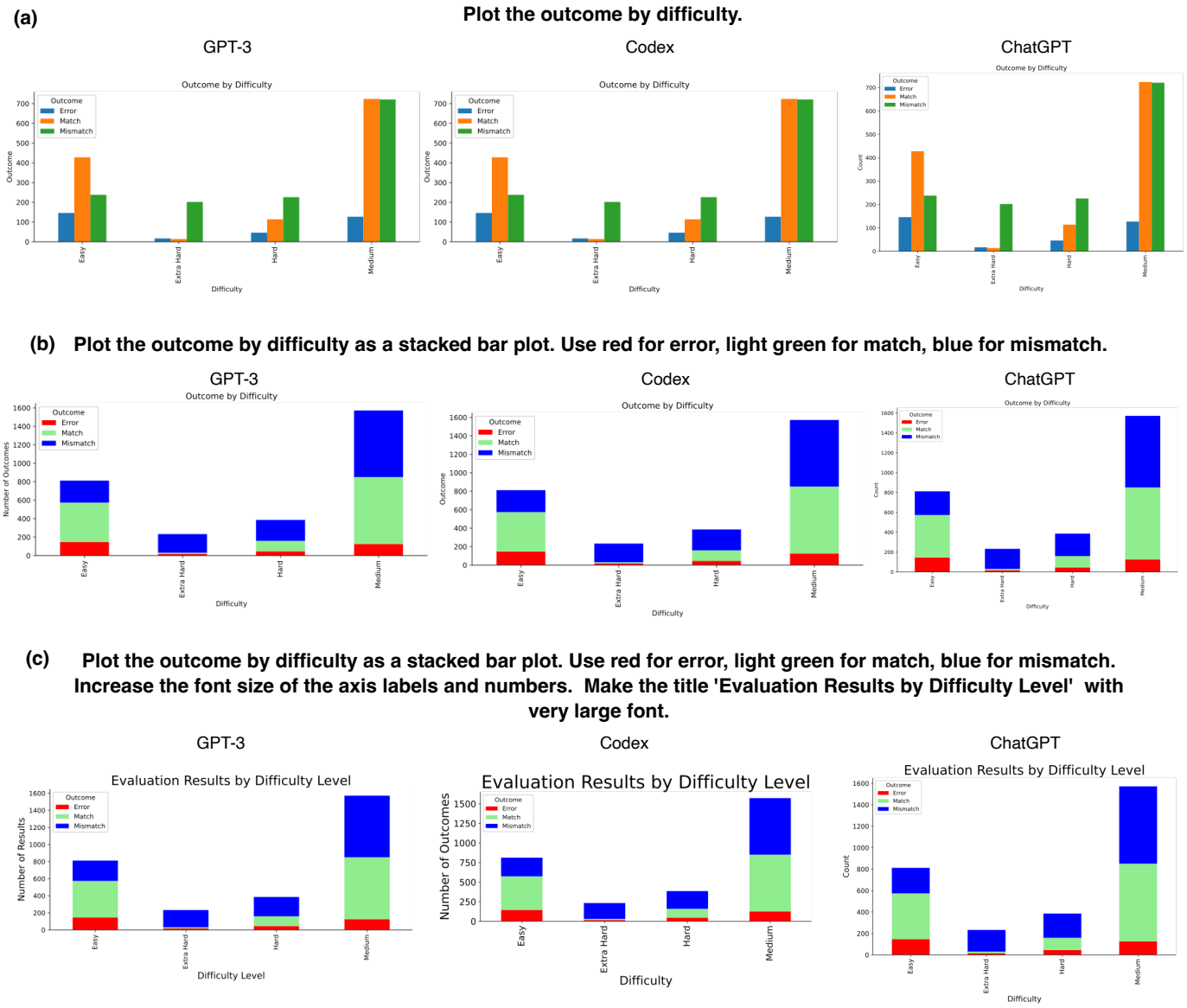

Fig. 7. Case Study 1: Conversational Visualisation Refinements using the nvBench plotting Evaluation Results by Difficulty

interpreted the requests reliably while also demonstrating differences in how they interpreted vague font size requests.

5.2 Case Study 2: Conversational Chart Refinement

Again, the base query *"Plot the outcome."* from the dataset example in Fig. 4(e) serves as a starting point. Illustrations in Fig. 8 demonstrate refining the query with the subsequent instruction *"...using a pie chart. Hide axis labels. Use pastel colours."*. Results show a high accuracy across all LLMs.All LLMs converted the original bar graph into a pie chart successfully. From the perspective of chart layouts, GPT-3 differed in the rendering of the pie graph by separating out the individual slices. GPT-3 and Codex did not follow the instruction to remove the axis label *"Outcome"* from the title, while ChatGPT did and instead reasoned to formulate a meaningful title *"Distribution of Outcome Categories"*. The command to apply pastel colours was followed by both Codex and ChatGPT. Overall, given this example, only ChatGPT followed every refinement request.

5.3 Case Study 3: Multilingual Requests

The capability of the LLMs to comprehend mixed multilingual requests for conversational chart refinement is demonstrated here and depicted in Fig. 9(a), building on the initial prompt *"Plot the outcome."* from Fig. 4(e) to plot results categorised by difficulty (Test 1) in French as *"Regroupez la difficulté par résultat sous forme de graphique à barres.* The refining requests (Test 2) that the *outcome* be plotted along the x-axis also using French *l'axe des x est le résultat."*, translated[8] as *"Group difficulty by outcome as a bar chart. The x axis is the result."*. Subsequently (Test 3), the plot is refined by asking for the title "Benchmark Results" to be altered using Croatian *"Promijenite naslov u 'Rezultati benchmarka'."*. Each LLM correctly rendered the refinements, while retaining both legend and bar labels in English. Meanwhile, GPT-3 and Codex translated both axes' labels into Croatian, while ChatGPT preserved the English labelling.

8. via Google translate https://translate.google.com

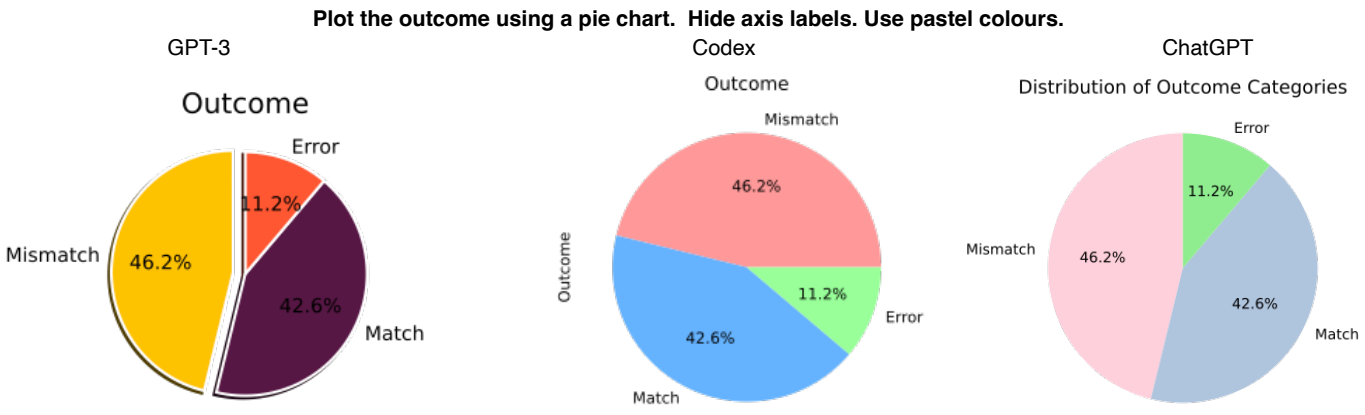

Fig. 8. Case Study 2: Conversational Visualisation Refinements using the nvBench plotting Evaluation Results by Outcome

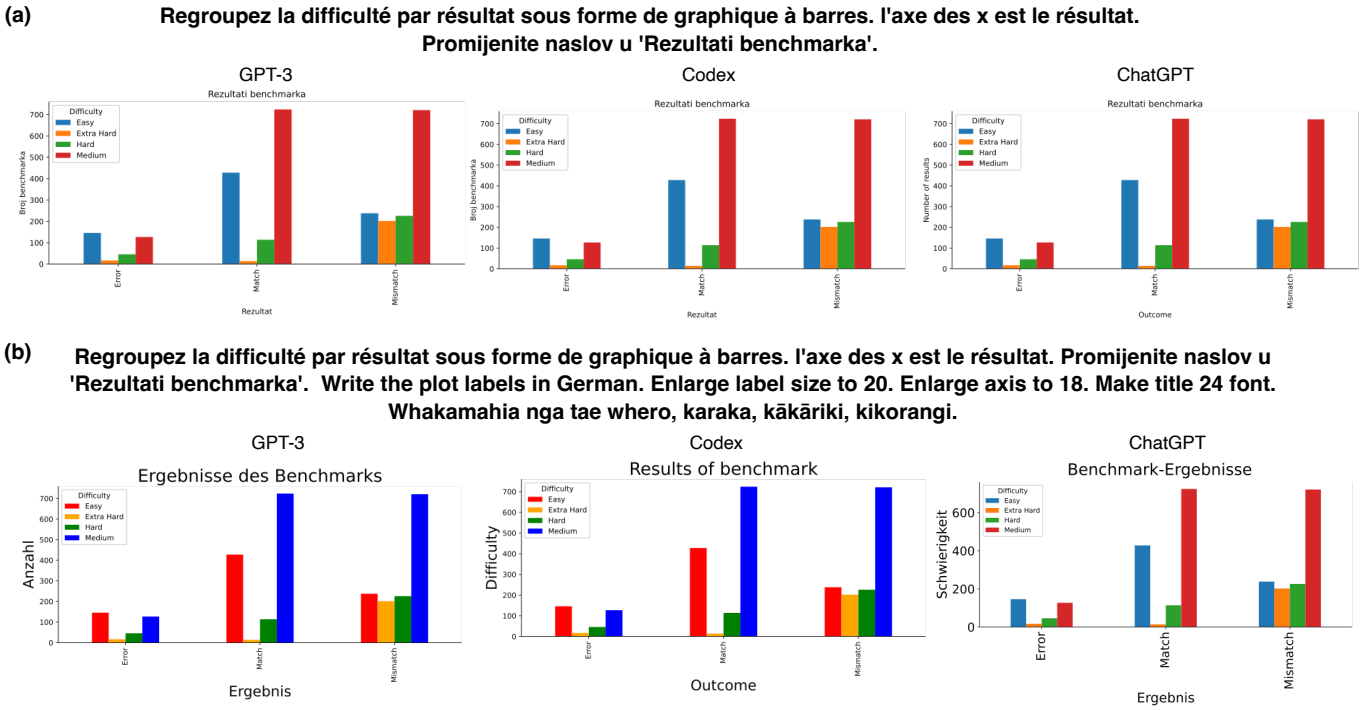

Fig. 9. Case Study 3: Conversational Visualisation Refinements using the nvBench plotting Results Data via Multilingual Requests.

The subsequent fine tuning can be seen in the next plot in Fig. 9(b) where an instruction (Test 4)is issued in English *...Write the plot labels in German.*, (Test 5) *Enlarge label size to 20.*, (Test 6) *Enlarge axis to 18.*, (Test 7) *Make title 24 font.*.

Following these instructions for refinement, , next, (Test 8) a change in colour ordering of plot bars is issued in New Zealand's te reo Māori (for which there is limited linguistic data available for training LLMs) with the phrase *"Whakamahia nga tae whero, karaka, kākāriki, kikorangi."*, meaning that the colours red, orange, green and blue be used. Following a complex suite of refinement instructions, it can be seen that all 3 LLMs responded to Tests 1 and 2 in French correctly, as well as to Croatian in Test 3. GPT-3 and Codex reasoned one step further without being instructed, and also modified the x axis label to 'Rezultat' in Croatian to match the title, while ChatGPT left this aspect unaltered in English. Test 4 requesting that labels be translated to German was followed by GPT-3 and ChatGPT, while Tests 5 to 7 requesting font and label sizes were only completely followed by ChatGPT. In the final test (Test 8) in te reo Māori, GPT-3 and Codex correctly interpreted the colouring request, with ChatGPT retaining its original colour ordering scheme.

Overall, though none of the LLMs achieved a perfect performance across all tested languages, they nonetheless demonstrated high-fidelity results with arguably ChatGPT outperforming the others on these examples.

5.4 Evaluation against nvBench

When evaluated against nvBench, Chat2VIS demonstrates significant potential. Out of the 3,003 queries tested, 1,280

charts showed an exact match. This 43% "Match" rate, as illustrated in Fig. 4(e), is a notable achievement considering both the narrow and strict conditions used to define a match and the overall complexities involved in data visualisation tasks. Fig. 7 presents summary statistics of the results by difficulty level, further emphasising the effectiveness of the system under various scenarios. Overall only 336 (11%) instances represented charts which could not be rendered, primarily due to erroneous code generated by the LLMs. A detailed examination of both the nature of successful and mismatched instances where our analysis has enabled the identification of certain conditions under which the system performs particularly well is given next.

5.4.1 nvBench Benchmark Matches

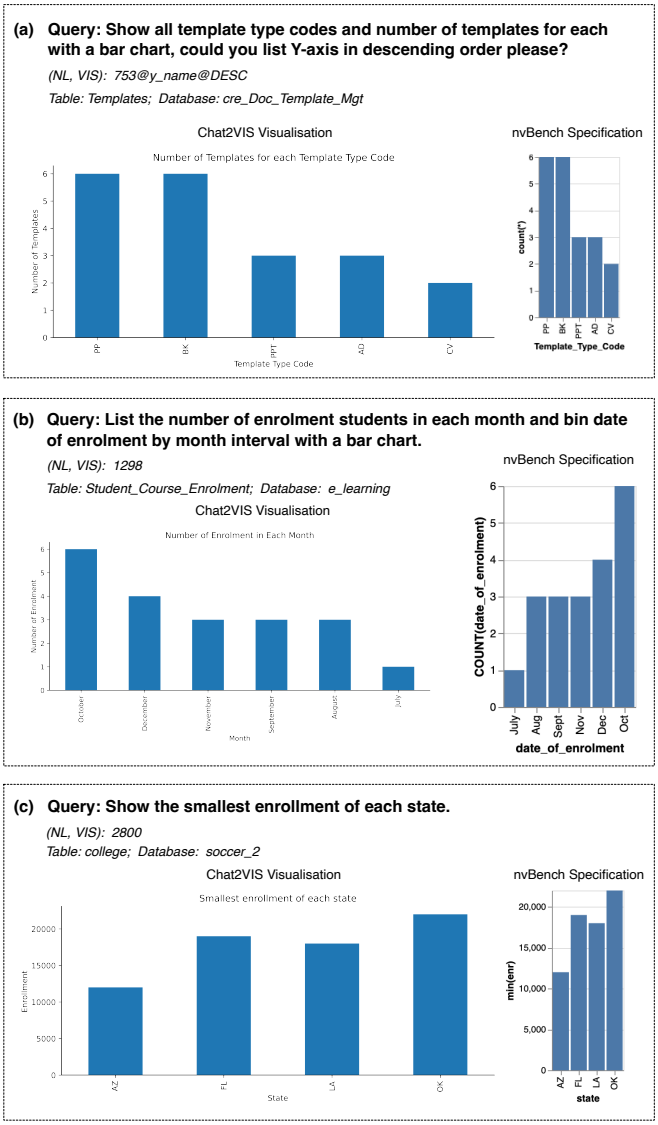

Fig. 10. Examples of Matched Visualisations Between Chat2VIS and nvBench

Chat2VIS excels in producing exact matches to the nvBench benchmark in a number of specific scenarios which are described and contextualised with examples as seen in Fig. 10, Chat2VIS tends to produce matches:

- in instances when the plot type is explicitly specified in the query (exemplified in Fig. 10a)
- in cases when the request for ordering or sorting is explicitly provided within the query, and in instances when all the grouped categories have different numerical values since ties can result in mismatched ordering (Fig. 10a shows a match which would register as a mismatch if 'PP' and 'BK' categories were rendered in a different order to the benchmark)
- in scenarios where the query is specific and is unambiguous (exemplified in Fig.10a and b)
- in instances where the specified plot type in the query aligns with the nature of the requested data (exemplified in Fig. 10a and b)
- in cases when the benchmark plot also accurately represents the information in the database and correctly interprets the query request (Fig. 10a-c)
- in instances when the plotting data did not require visualising non-zero values (Fig. 10a and c)
- in scenarios when the database structure itself correctly represents the data types it holds, and the null values are represented in line with database standards (Fig. 10a - c)
- in cases when grouping information such as 'by Weekday', 'by Month', etc., and when this is explicitly provided in the query (Fig. 10b).

5.4.2 Benchmark Mismatches

Here we summarise broad categories under which mismatches were detected with the benchmark. Many are a result of applying too narrow a definition of a match when devising an automated system which does not have subjective capabilities, while a large percentage of the mismatches can be attributed to deficiencies within the benchmark. We observed instances where the Chat2VIS chart was 'correct' in its rendering, however, errors within the nvBench specification and the inconsistency within the database information resulted in the mismatch classification. We illustrate in Fig. 14(a) a discrepancy in categorical values with variant spellings of *"Andreo"* and *"Lo"* in the nvBench specification contrasted with *"Andreou"* and *"Lou"* persisted in the database. We noted the presence of at least five nvBench specifications omitting the first query. On passing an empty string to Chat2VIS, the LLM demonstrates its effective decision-making to render a visualisation of interest, as shown in 14(b).

In a subset of examples, we found inconsistencies between the query request and the SQL denoted in the nvBench specification. Our findings unearthed instances where the SQL included an `ORDER BY` clause but the NL query omitted instructions to *"order"* or *"sort"* results. Consequently, the Chat2VIS visualisation accurately depicted the data points but was classified as mismatched with that of nvBench's sorted chart. These examples illustrate repeatedly that a mismatch in a strict sense when using automated approaches is not necessarily an incorrect output.

Our methodology utilises only the query component of the specification, disregarding other instructions contained within. Consequently an assortment of examples stored requests for grouping results inside the "binning" mark of the specification, therefore omitting to incorporate the request

as part of the NL query. We illustrate in Fig. 14(c) a common mistake where the Chat2VIS chart summarised by date, and the equivalent nvBench specification grouped by weekday, an instruction solely provided within the "binning" mark.

Further inspection of mismatches showed periodically, without additional stipulations inside the specification, nvBench incorrectly renders only a partial result set, while Chat2VIS visualises the complete set of data. An example is depicted in Fig. 14(d). The reasoning for such behaviour within nvBench is unknown. Additionally, we noted for some instances executing the SQL query from the nvBench specification directly against the database yields conflicting figures to those visualised by nvBench, as shown in Fig. 14(e). Furthermore, we encountered truncating of numeric float types to integer values without additional stipulations inside the specification as shown in Fig 14(f). Consequently causing additional mismatches between nvBench and Chat2VIS results.

5.4.3 Methodology Limitations

Our approach to automated benchmarking against nvBench, in the absence of a standardised methodology, presents some limitations. Occasionally, Chat2VIS generates visually accurate charts but the proposed comparison methodology does not always yield expected results. Differences in the treatment of categories devoid of data (Chat2VIS omits, nvBench assigns zero) lead to mismatches, despite similar visual outcomes, as shown in Fig. 14(g).

The nvBench queries often specify the ordering or sorting of results based on a nominated axis, either in ascending or descending order. As we do not rearrange the x and y vectors if sorting is requested, an exact match is necessary for the charts to be equivalent. However, this approach presents a challenge when multiple x values have identical y values, rendering a correctly ordered, but not identical, chart. We illustrate this assorted ordering in Fig. 14(h) with the majority of bars having value 1.

Occasionally, LLM's unconventional plotting arising from Python's diverse charting techniques, may lead to unexpected outcomes for x and y vectors, resulting in null values at some points of the chart resulting in mismatches while not necessarily being incorrect.

We noted that numeric fields in the nvBench database occasionally contain missing values represented as empty strings instead of NULL values, causing issues with Python's data import and query execution and mismatches. Enhancing the Chat2VIS interface could mitigate this.

Similarly, columns holding numeric data but defined as character type could result in incorrect mathematical computations. Sometimes, older versions of Python syntax generated by the LLM and verbose scripts exceeding token restrictions also led to execution errors.

5.4.4 Ambiguity

Query text ambiguity led to different interpretations and charts by the LLM and nvBench making genuine accuracy assessment challenging. Queries like *"...sort bars in desc order"* or *"...order by the bars from high to low"* were interpreted by Chat2VIS as sorting by y-axis values, whereas nvBench sorted x-axis labels alphabetically, as shown in Fig. 14(i).

Even though bar charts are typically used for comparing group values, we observed instances where they were used to represent two text columns. Fig. 14(j) shows nvBench's approach to this situation, while Chat2VIS inferred a different but more appropriate visualisation, albeit aesthetically unappealing, providing category counts.

5.4.5 Query Misinterpretation

Occasionally, the LLM misinterpreted queries, causing Chat2VIS to generate incorrect visualisations. Date and time values were challenging. While nvBench correctly used only the date component for grouping, Chat2VIS sometimes considered both date and time, leading to individual groupings of date-times.

The nvBench construction methodology often generated similar benchmark examples. Filter requests such as *"...commission is not null or department number does not equal to 40..."* were present in over 70 instances. Chat2VIS incorrectly used the *"and"* operator instead of *"or"* when interpreting this filter, contributing to a disproportionate amount of mismatched results.

We also noted instances where Chat2VIS erroneously self-imposed a limit on the number of returned results and also misunderstood a Chinese language request. These incidents highlight the LLM's occasional misinterpretations, despite its general proficiency in generating Python code from natural language.

5.5 Evaluation against nlvUtterance

Chat2VIS demonstrates robust results against the nlvUtterance benchmark with respect to the strict methodology used. 50% match rate was observed over all chart types when using Codex. This rate improved to 63% when either Codex or GPT-3 produced a matching chart. Meanwhile, the matching rate rose to 72% when at least one of Codex, GPT-3, or ChatGPT generated a matching chart. Fig. 11 presents the evaluation results, categorized by chart type for each stage of testing. Again, it is worth noting that not all mismatches between Chat2VIS and nlvUtterance generated charts imply inaccuracies. The ambiguity within the query and lack of charting specifications often led to alternative yet 'correct' visualisations, which, due to the lack of defined evaluation guidelines and parameters within the benchmark, were sometimes deemed as mismatches in our objective evaluation.

5.5.1 nlvUtterance Benchmark Matches

Chat2VIS exhibited a high degree of matches under several conditions. Firstly, when single-attribute bar plots and scatter plots were utilised, the system generated matches. These types of plots had the highest representation in the dataset, and the system managed to match them at a high rate. Evaluations with Codex, shown in Fig. 11(a), demonstrated that coloured scatter plots and faceted scatter plots had significantly lower matching rates. However, when generated via GPT-3 (Fig. 11(b)) or ChatGPT (Fig. 11(c)), the results were much more favourable. Secondly, the system showed high matching rates with bar charts when generated via the three LLMs.

5.5.2 nlvUtterance Benchmark Mismatches

Nonetheless, our evaluation also revealed instances where the system struggled, particularly with grouped and stacked bar charts, histograms, coloured and faceted scatter charts, and single and multi-line charts. Fig. 15 presents a sample of generated visualisations for the ten chart types based on the movies dataset, enabling comparisons with those presented in prior work [23]. Specific chart types are detailed below with respect to the generation of mismatches.

Bar Charts: A substantial number of grouped bar charts were classified as a mismatch notwithstanding correct charting of the requested data. Considering the query *"average production budget by creative type and content rating"*, the benchmark arranged results grouped by content rating using colour coding to represent the creative type. However, periodically the Chat2VIS counterpart inversely grouped results by creative type with colours representing content rating. Similar issues were observed with mismatches between stacked bar charts. Furthermore, our findings showed instances where benchmark stacked bar charts were presented by Chat2VIS as grouped bar charts, accurately conveying the information, but not in accordance to the benchmark standard. Had the evaluation methodology deemed these visualisations as matches, as indeed they were despite deviating from the benchmark, the accuracy (match) rate would increase significantly. Nonetheless, the absence of methodological instructions within the benchmark failed to provide guidance in such circumstances.

Histograms: We refrained from providing explicit instructions to the LLM on which chart type to render. Consequently a significant number of benchmark histograms were instead plotted as bar charts. Queries such as *"How many orders were placed for each order quantity?"* and *"show me a bar chart of count by order quantity"* did not imply the data should be represented as a histogram, and hence the LLM decided the most appropriate representation of the data was in the form of a bar chart. Furthermore, queries neglected to provide information pertaining to binning size, and consequently the LLM's decision often conflicted with benchmark visualisations.

Scatter Charts: Colour scatter charts use varied colours to represent categories in a single plot. Codex often overlooked this colour coding leading to single-coloured charts. GPT-3 did not share this limitation, but both exhibited a high percentage of mismatches due to incorrect syntax, often incorrectly setting the *"c"* colour parameter value in the Python plotting function. This misstep resulted in erroneous script execution. ChatGPT was more successful, correctly assigning this function parameter. Faceted scatter charts separate categories from coloured scatter plots into distinct charts. Some queries lacked clear instructions for a faceted chart, prompting us to accept single scatter plots categorised by colour. As in the case of coloured scatter charts, Codex often failed to use colour coding to distinguish categories, while GPT-3 and ChatGPT did not have this limitation. However, due to the lack of a benchmark methodology, we accepted alternative charts, which could otherwise have affected the success rate.

Line Charts: The least-represented chart types in the dataset are single and multi-line plots. The most common cause of mismatch was the LLM selecting to render the information as a bar chart. However, although it still accurately presented the requested information when a line chart was not explicitly requested, it was not in accordance with benchmark specifications. In addition, Chat2VIS multi-line plots on occasion inversely rendered the x-axis and line colour categories compared to that of nlvUtterance, hence unsuccessful in meeting benchmark standards. Once more, these decisions of determining if benchmark standards are met significantly impact culminating a successful outcome.

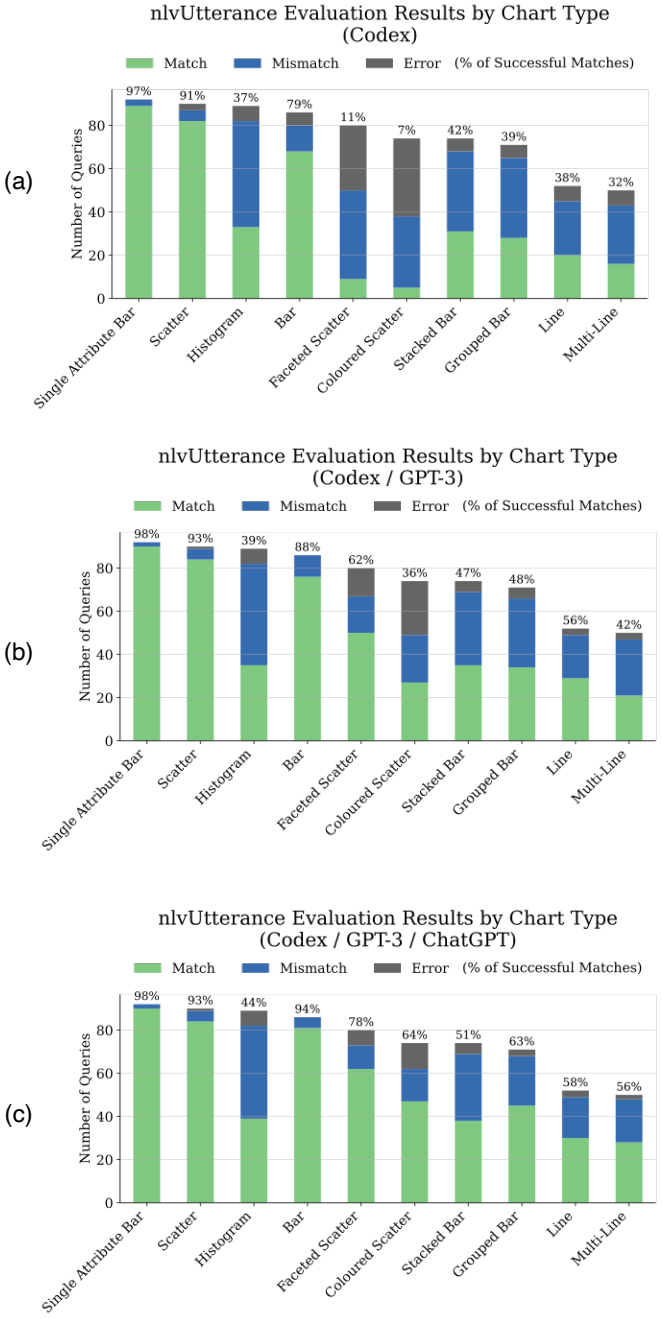

Fig. 11. nlvUtterance Evaluation Results.

5.6 Comparison With Previous Work

Finally, Chat2VIS is contrasted with results from prior studies' on the nvBench benchmark. The comparisons are suggestive but are not exact since our nvBench sample set differs from test set used in previous studies. The overall accuracy of state-of-the-art NL2VIS systems are shown in Table 1 as reported by [21] and contrasted with Chat2VIS, indicating a highly competitive performance of the proposed system.

TABLE 1
nvBench Performance Comparison.

System	Accuracy
Seq2Vis [24]	2%
Transformer [25]	3%
ncNet [3]	26%
RGVisNet [21]	**45**%
Chat2VIS	43%

Table 2 summarises our findings using the nlvUtterance benchmark, with those from NL4DV evaluated on the benchmark in previous studies [23]. It should be noted the NL4DV results are based on a 755 instance dataset pertaining to singleton query sets only. In our study we included all query sets, with the exclusion of those pertaining to the two omitted line charts.

TABLE 2
nlvUtterance Performance Comparison.

System	Accuracy
Chat2VIS (Codex)	50%
Chat2VIS (Codex/GPT-3)	63%
Chat2VIS (Codex/GPT-3/ChatGPT)	72%
NL4DV	64%

6 Discussion

The experiments confirm the advanced capability of Chat2VIS to provide a state-of-the-art solution to the NL2VIS problem, exemplifying the conversational ability to refine chart aesthetics and to do so with multilingual instructions.

Furthermore, we provided results of Chat2VIS against two benchmarking datasets and confirmed the state-of-the-art properties of the proposed system. Given the time-consuming and subjective nature of performing these necessary evaluations, we made a contribution towards the development of automated approaches to help researchers and accelerate advancements in this field. Our proposed automation methodology is a first step in realising this goal and has yielded findings that are helpful in advancing the refinement of existing benchmarks and the development of new ones.

While we gratefully applaud the enormous efforts invested by researchers in developing the existing benchmark datasets, we find that there is room for improvement in fulfilling all the necessary quality characteristics like *reproducibility*, *fairness*, and *verifiability* as defined by [4]. The large number of diverse visualisation elements, aesthetics, and chart styles, from a variety of available programming language libraries raises difficulties in generating *reproducible* and measurable outcomes from NL queries. The predominant use of Vega-lite specifications in current benchmarking studies [23] [19] periodically separates important visualisation information from the NL query, limiting the effectiveness of alternative NL2VIS architectures and reducing *fairness*. Inconsistencies exist between test case data and visualisation outcomes thus compromising *verifiability*.

In future, to establish robust benchmarks for NL2VIS, we foresee definitions of a collection of valid visualisations for each NL query accompanied by a comprehensive methodology for chart comparison and evaluation. The evaluation would be independent of charting frameworks.

Study Limitations

Our study investigated the use of LLMs for NL2VIS tasks, employing automated benchmarking. However, some limitations are noteworthy. Technical issues hindered comprehensive testing against nvBench chart types such as line, pie, and scatter, which would require custom comparison mechanisms. We evaluated Chat2VIS against nvBench using only the first equivalent NL query, potentially introducing some bias, though the first NL query was deemed as expressive as any other. While we acknowledge the importance of ethical aspects like reliability, robustness, and possible misuse of LLMs, these fell outside our study's core focus and are the subject of ongoing research. Similarly, an exhaustive analysis of the role of prompt engineering was beyond our scope, despite its potential impact on LLM performance. We formulated a single, generic prompt architecture applicable across various LLMs that solves the research gaps in NL2VIS, but future studies should explore alternative configurations. Our benchmarking process mainly relied on concrete examples, leaving scope for future evaluations to consider a range of language intents and the handling of 'dirty data' and domain-specific terms. This would provide a more realistic evaluation of NL2VIS systems. Lastly, our study focused on a single-step conversion of NL into Python visualisation code. The potential benefits of using structured expressions as an intermediate step in the NL2VIS pipeline were not explored. Such an approach could offer more consistent and accurate outputs and is a promising direction for future research.

7 Conclusion

This seminal study presents the novel features of Chat2VIS for converting natural language into data visualisations in a conversational manner with the ability to refine charts in multiple languages, addressing a previously unsolved research problem.

We demonstrated the capabilities of our system against two benchmarks and have proposed a novel approach for automating the evaluation and comparison of generated visualisations thus contributing towards an additional gap in literature. We explored the challenges in accomplishing this and have identified areas for improvement in the development of benchmark datasets in the field of NL2VIS in order to accelerate the development of future advancements.

REFERENCES

[1] L. Shen, E. Shen, Y. Luo, X. Yang, X. Hu, X. Zhang, Z. Tai, and J. Wang, "Towards natural language interfaces for data visualization: A survey," *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[2] Y. Wang, Z. Hou, L. Shen, T. Wu, J. Wang, H. Huang, H. Zhang, and D. Zhang, "Towards natural language-based visualization authoring," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 1222–1232, 2022.

[3] Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin, "Natural language to visualization by neural machine translation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 217–226, 2021.

[4] S. Kounev, K.-D. Lange, and J. von Kistowski, *Systems Benchmarking: For Scientists and Engineers*. Springer, 2020.

[5] G. Liu, X. Li, J. Wang, M. Sun, and P. Li, "Extracting knowledge from web text with monte carlo tree search," in *Proceedings of The Web Conference 2020*, 2020, pp. 2585–2591.

[6] Cox, Kenneth and Grinter, Rebecca E and Hibino, Stacie L and Jagadeesan, Lalita Jategaonkar and Mantilla, David, "A multi-modal natural language interface to an information visualization environment," *International Journal of Speech Technology*, vol. 4, pp. 297–314, 2001.

[7] Y. Sun, J. Leigh, A. Johnson, and S. Lee, "Articulate: A semi-automated model for translating natural language queries into meaningful visualizations," in *Smart Graphics: 10th International Symposium on Smart Graphics, Banff, Canada, June 24-26, 2010 Proceedings 10*. Springer, 2010, pp. 184–195.

[8] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios, "Datatone: Managing ambiguity in natural language interfaces for data visualization," in *Proceedings of the 28th annual ACM symposium on user interface software & technology*, 2015, pp. 489–500.

[9] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang, "Eviza: A natural language interface for visual analysis," in *Proceedings of the 29th annual symposium on user interface software and technology*, 2016, pp. 365–377.

[10] X. Qin, Y. Luo, N. Tang, and G. Li, "Deepeye: Visualizing your data by keyword search," in *EDBT*, 2018, pp. 441–444.

[11] A. Narechania, A. Srinivasan, and J. Stasko, "Nl4dv: A toolkit for generating analytic specifications for data visualization from natural language queries," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 369–379, 2020.

[12] B. Yu and C. T. Silva, "Flowsense: A natural language interface for visual data exploration within a dataflow system," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 1–11, 2019.

[13] E. Loper and S. Bird, "Nltk: The natural language toolkit," *arXiv preprint cs/0205028*, 2002.

[14] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The monted corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 2014, pp. 55–60.

[15] Q. Wang, Z. Chen, Y. Wang, and H. Qu, "A survey on ml4vis: Applying machine learning advances to data visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 12, pp. 5134–5153, 2022.

[16] H. Voigt, M. Meuschke, K. Lawonn, and S. Zarrieß, "Challenges in designing natural language interfaces for complex visual models," in *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, 2021, pp. 66–73.

[17] C. Liu, Y. Han, R. Jiang, and X. Yuan, "Advisor: Automatic visualization answer for natural-language question on tabular data," in *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*. IEEE, 2021, pp. 11–20.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2018, pp. 4171–4186.

[19] Y. Luo, J. Tang, and G. Li, "nvbench: A large-scale synthesized dataset for cross-domain natural language to visualization task," *arXiv preprint arXiv:2112.12926*, 2021.

[20] J. Tang, Y. Luo, M. Ouzzani, G. Li, and H. Chen, "Sevi: Speech-to-visualization through neural machine translation," in *Proc. of the 2022 International Conference on Management of Data*, 2022, pp. 2353–2356.

[21] Y. Song, X. Zhao, R. C.-W. Wong, and D. Jiang, "Rgvisnet: A hybrid retrieval-generation neural framework towards automatic data visualization generation," in *Proc. of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1646–1655.

[22] P. Maddigan and T. Susnjak, "Chat2vis: Generating data visualisations via natural language using chatgpt, codex and gpt-3 large language models," *IEEE Access*, vol. 11, pp. 45 181–45 193, 2023.

[23] A. Srinivasan, N. Nyapathy, B. Lee, S. M. Drucker, and J. Stasko, "Collecting and characterizing natural language utterances for specifying data visualizations," in *CHI '21: Proc. of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, p. 1–10.

[24] Y. Luo, N. Tang, G. Li, C. Chai, W. Li, and X. Qin, "Synthesizing natural language to visualization (nl2vis) benchmarks from nl2sql benchmarks," in *Proceedings of the 2021 International Conference on Management of Data, SIGMOD Conference 2021, June 20–25, 2021, Virtual Event, China*. ACM, 2021.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, pp. 6000–6010, 2017.

APPENDIX A
PROMPT ENGINEERING

Fig. 12 depicts the structure of the LLM prompt for a given concrete example.

APPENDIX B
MULTILINGUAL EXAMPLES

B.1 Case Study 4: Multilingual Requests

Fig. 13 provides an additional multilingual example of Chat2VIS capabilities, combining Spanish, Japanese and Mandarin to depict the counts of Chat2VIS mismatch performances against various nvBench databases. In Fig. 13 (a), it can be seen that the Spanish NL query to "Plot the number of mismatches per database. Only plot the top 10 databases." was correctly executed by all LLMs but with some different variations in interpreting the ambiguity. Both GPT-3 and GPT-4[9] selected the top 10 databases with respect to the largest number of mismatches, and then depicted the results. Meanwhile, GPT-3.5 first selected the top 10 largest databases and then depicted their respective mismatches from there. In the subsequent refinement of the query in Japanese to "Add the value on top of the bar line.", it can be seen that only GPT-3 correctly responded, while the subsequent request in Mandarin to adjust the x-axis labels as "Print the database names diagonal." was indeed followed by all LLMs. The examples illustrate again the high level of responsiveness of the LLMs to both correctly respond to multilingual requests as well as to plot refinements.

APPENDIX C
BENCHMARK EXAMPLES

Fig. 14 depicts examples of mismatches between Chat2VIS and nvBench together with errors within the benchmark dataset. These errors comprise Fig. 14(a) misspellings, Fig. 14(b) missing NL queries, Fig. 14(c) missing query intents, Fig. 14(d) discordant nvBench visualisations with respect to NL queries, Fig. 14(e) conflicting results, Fig. 14(f) unexpected data type conversions, Fig. 14(g) differences in how zero values are depicted, Fig. 14(h) differences in sorting quantities with equal values, Fig. 14(i) differences in interpreting the sorting intent, Fig. 14(j) differences in handling string handling. These combinations of errors within nvBench, ambiguities and differences in the plotting behaviour of underlying frameworks generated mismatches under the chosen comparison methodology, which would likely have been treated differently under manual and subjective evaluations. Fig. 15 shows a selection of matching Chat2VIS examples with the nlvUtterance dataset.

9. The Codex model has been discontinued and GPT-4 was used instead.

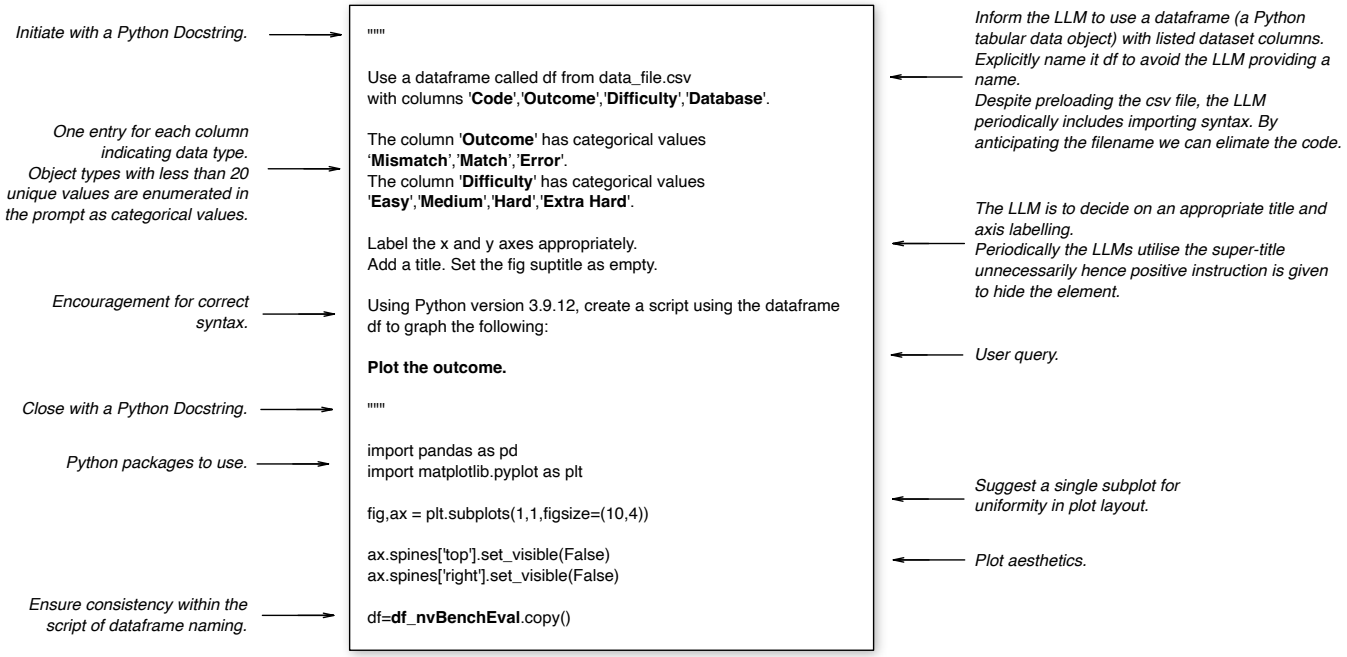

Fig. 12. Explanation of Chat2VIS Prompt

Fig. 13. Case Study 4: Conversational Visualisation Refinements using the nvBench Results Data via Multilingual Requests in Spanish, Japanese and Mandarin.

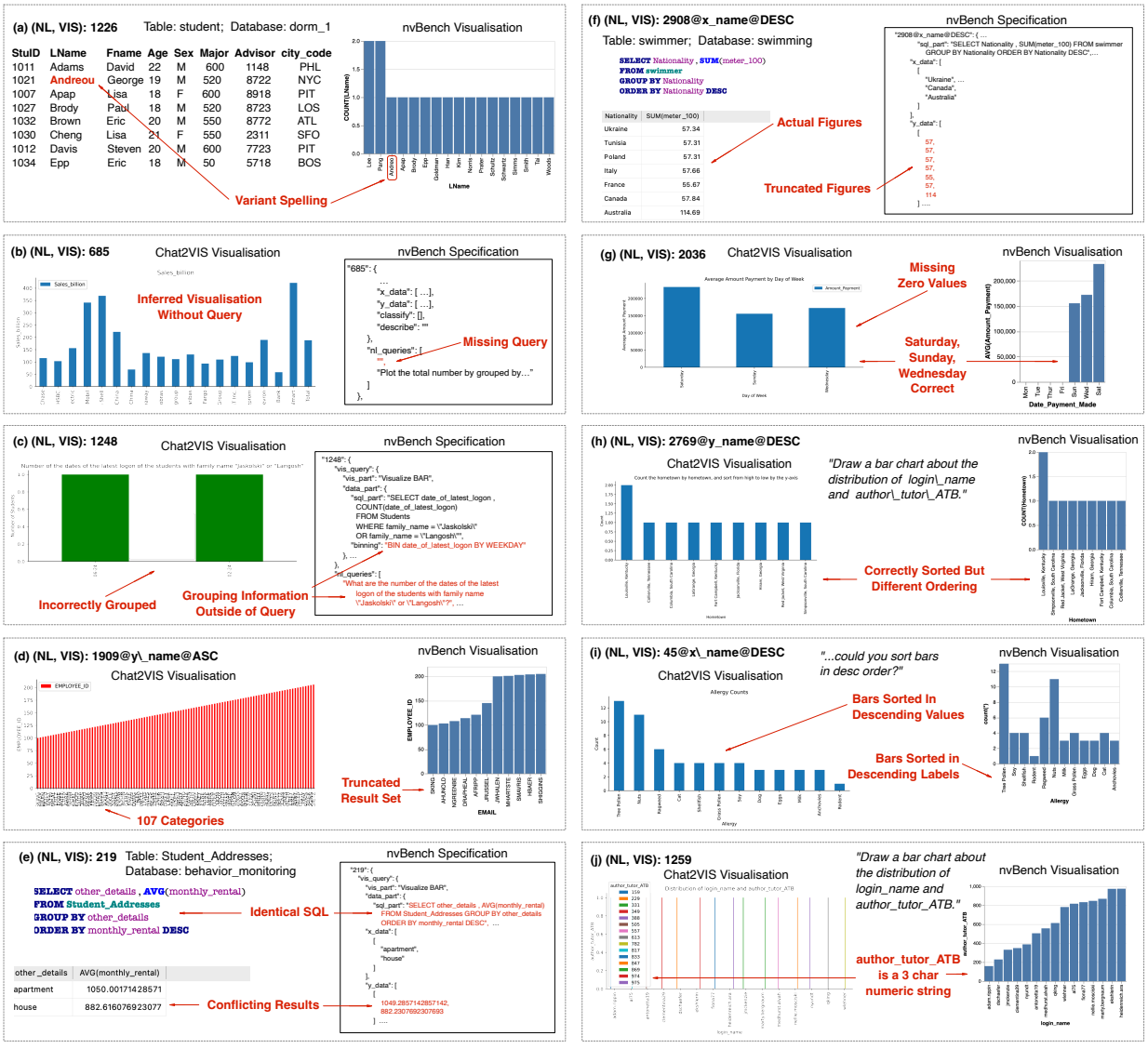

Fig. 14. Examples of Mismatched Visualisations Between Chat2VIS and nvBench

Fig. 15. nlvUtterance Movies Dataset Examples from Chat2VIS

Chat2VIS: Multilingual and Customisable Chart Generation using Large Language Models

Paula Maddigan and Teo Susnjak

Abstract—In our data-saturated world, there is a pressing need to harness technology to derive insights. Yet, traditional tools often require significant learning overheads to work with complex charting techniques. Such barriers can hinder those who may benefit from harnessing data for informed decision-making. However, generating data visualisations directly from natural language text (NL2VIS) is an emerging field that addresses this issue. This study showcases Chat2VIS, a state-of-the-art NL2VIS solution for conversational chart generation. This work capitalises on the latest AI technology leveraging large language models (LLMs) such as GPT-3, Codex, and ChatGPT to generate and refine data visualisations conversationally with capabilities that are beyond those demonstrated in previous studies. In addition, this paper presents the novel capability of Chat2VIS to comprehend multilingual natural language requests. Our work is evaluated against two NL2VIS benchmark datasets. In the process, we propose an automated methodology for conducting evaluations which are otherwise performed both manually and infrequently. We contribute findings and recommendations going forward with respect to improving the development of benchmark datasets for NL2VIS in order to enable automated evaluations, and in the process facilitate an acceleration of advancements in this field.

Index Terms—ChatGPT, Codex, GPT-3, end-to-end visualisations from natural language, large language models, natural language interfaces, text-to-visualisation.

1 Introduction

In an era where data has become a valuable commodity, industries are continuing to witness immense growth in its volume. Data visualisations offer an effective and compelling approach to communicating insights from this resource to facilitate better-informed business decisions. The ability to articulate visualisation requests through natural language (NL) text and intuitive interfaces is fast gaining traction [1]. It is an enticing objective to generate suitable charts without the need to acquire programming skills and undergo arduous learning curves associated with visualisation tools [2]. Therefore, in our quest to democratise access to these data visualisation tools and make them more user-friendly, the emerging field of Natural Language to Visualisation (NL2VIS) is poised to transform the way we interact with, and understand data [3].

Despite the existence of the NL2VIS field for two decades, the challenge remains in devising end-to-end systems that can perform multiple tasks like interpreting complex user intents with their inherent ambiguities, automatically selecting appropriate visualisation types, and transforming parsed instructions into visual outputs. Recently, the surge in interest in large language models (LLMs) is driving research to develop NL2VIS with end-to-end capabilities that leverage this advanced AI technology.Since these language models are trained on a large amount of both language texts and code repositories, they exhibit a high level of skill in language semantics and code scripting, which makes them ideal candidates for solving this difficult problem. Therefore, this study investigates the capabilities of OpenAI's GPT-3, Codex, and ChatGPT models in advancing end-to-end data visualisation systems and reports their accuracies against benchmark datasets.

One remarkable feature of the conversational capabilities of advanced LLMs is their unique ability to maintain a coherent dialogue and build on prior exchanges. This conversational capability for iteratively customising a chart has also not been adequately resolved in the field of NL2VIS and represents an existing need [1]. This study showcases how this can indeed be addressed via LLMs, transcending the capabilities of prior NL2VIS architectures in literature.

Recent literature [1] also highlights the lack of multilingual support in existing NL2VIS systems whose capabilities normally only encompass English. Meanwhile, advanced LLMs are inherently capable of multilingual comprehension due to their expansive training data, which encompasses a multitude of languages from various text sources. This extensive linguistic diversity in their training corpus enables LLMs to decode text from multiple languages, albeit, the performance profiles of LLMs across different languages can also be uneven due to the varying representation of languages in their training data. To that end, this research probes the proficiency of the proposed system to generate visualisations from queries originating from several languages which have different levels of representation in the training corpus of the LLMs.

Contribution

The contribution of this study is fourfold. This work advances the field of NL2VIS by presenting novel features

• *P. Maddigan and T. Susnjak are with the School of Mathematical and Computational Sciences, Massey University, Auckland, New Zealand.*

E-mail: paula.maddigan@gmail.com, t.susnjak@massey.ac.nz

Manuscript received May 12, 2023

within the LLM-based Chat2VIS framework. The first is the demonstration of the customisable capabilities of the Chat2VIS framework to incorporate iterative refinements to queries to adjust the generated charts and its aesthetics. We show that the range of customisation options exceeds solely refining a predefined set of chart components as demonstrated by previous NL2VIS approaches, but are instead much broader. This work addressed a second gap in literature by demonstrating the capacity of Chat2VIS to comprehend multilingual NL texts across seven diverse languages and to generate data visualisations with customisations, establishing it as a truly global tool.

Thirdly, we tackle a pressing issue in the current state of NL2VIS: the lack of robust benchmarking tools. The evaluation of NL2VIS systems is often marred by the inherent subjectivity of human judgement, and the manual evaluation approaches are laborious, requiring teams of assessors. In response, this work develops an automated approach for conducting a quantitative analysis of NL2VIS performance which is a first step towards a comprehensive solution. We present the results of our evaluation of Chat2VIS against two benchmarks, highlighting the challenges of developing structured methodologies and measurable baseline standards for NL2VIS. This contribution is valuable as the establishment of effective benchmarks can expedite advancements in this domain, especially if evaluations can be automated. Our study underscores the importance of ensuring that benchmarks fulfill the quality characteristics of *reproducibility*, *fairness*, and *verifiability* [4]. As more benchmark datasets begin to emerge in this domain, we emphasise the need for them to incorporate a well-defined measurement methodology, outlining the process to implement the standard, collect measurements, and evaluate the results [4]. Finally, the developed software artefact has not only been made available to the public via an online portal, but the source code has also been released for researchers to further develop [1].

2 Related Work

Early NL2VIS systems were built on symbolic-based NLP approaches, relying on heuristic algorithms [5], rule-based architectures, and probabilistic grammar-based methods for translating NL queries. Some of the earliest attempts can be traced to 2001 with initial prototype relying only on well-structured queries called InfoStill [6]. Although each subsequent technique displayed incrementally improving accuracies, they also required increasing amounts of computational resources for modest performance improvements. Systems such as Articulate [7], DataTone [8], Eviza [9], and Deep-Eye [10] all used varying symbolic NLP methodologies in translating NL to data visualisations. However notable approaches like NL4DV [11] and FlowSense [12] employed NLTK [13], NER, and Stanford CoreNLP [14] semantic parsers to improve accuracy. Readers are directed to two recent surveys ([1] and [15]) on NL2VIS which delve deeper into the evolution of the field.

Recent advancements in NL2VIS have focused on deep-learning models to achieve greater levels of adaptability, robustness and flexibility compared to achievable performances of previous approaches [16]. Systems such as ADVISor [17] are supported by BERT [18], a large language transformer-based model where the rendered visualisation styles are predetermined based on a defined mapping rule.

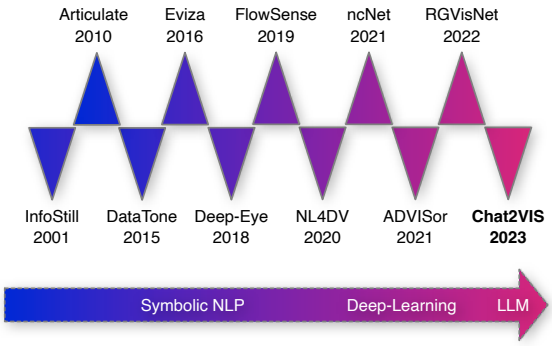

Fig. 1. NL2VIS timeline illustrating the evolution of NL2VIS systems.

An alternative transformer-based approach, ncNet [3], is a machine learning model trained using the nvBench [19] dataset. The model accepts an optional chart template in addition to the requested NL query to guide chart styling of the rendered visualisation. The system has recently been expanded to include speech-to-visualisation capabilities [20].

Furthermore, the hybrid approach of RGVisNet [21] initially retrieves the most relevant visualisation query from a large-scale visualisation codebase. It then revises it via a GNN-based deep-learning model, and subsequently generates the visualisation.

The evolution of NL2VIS systems are illustrated in Fig. 1, depicting the transition from symbolic NLP towards deep learning approaches. With the evolution, increasingly the end-to-end capabilities feature with newer frameworks which integrate multiple components of a NL2VIS pipeline, including natural language understanding, information extraction, visualisation (code) generation into a unified solution. Unlike fragmented approaches in the literature, end-to-end solutions automate the entire process, eliminating the need for multiple components, thus making them more efficient. The latest state-of-the-art artefact, Chat2VIS [22], presents the first NL2VIS NLI to generate data visualisations via LLMs that has the ability to support end-to-end processes. It addresses the next generation of NL2VIS architecture, simplifying the NL2VIS pipeline by offloading language understanding, chart selection and reasoning in the presence of ambiguity, as well as code generation to a single system. The underlying structure provides flexibility and robustness around free-form and complex visualisation requests while decisions pertaining to suitable chart selection and aesthetics are delegated to the LLMs. The architecture underpinning Chat2VIS is exceptionally flexible and decidedly diverse enough to further refine charting elements using NL without additional enhancements to the NL2VIS architecture. This is the first study to address this gap evident in earlier systems. In addition, unseen in previous approaches, this work demonstrates the art of fulfilling multilingual requests with ease, omitting the need for additional prompting, further architectural manipulations,

1. The source code for the Chat2VIS is available at https://github.com/frog-land/Chat2VIS_Streamlit

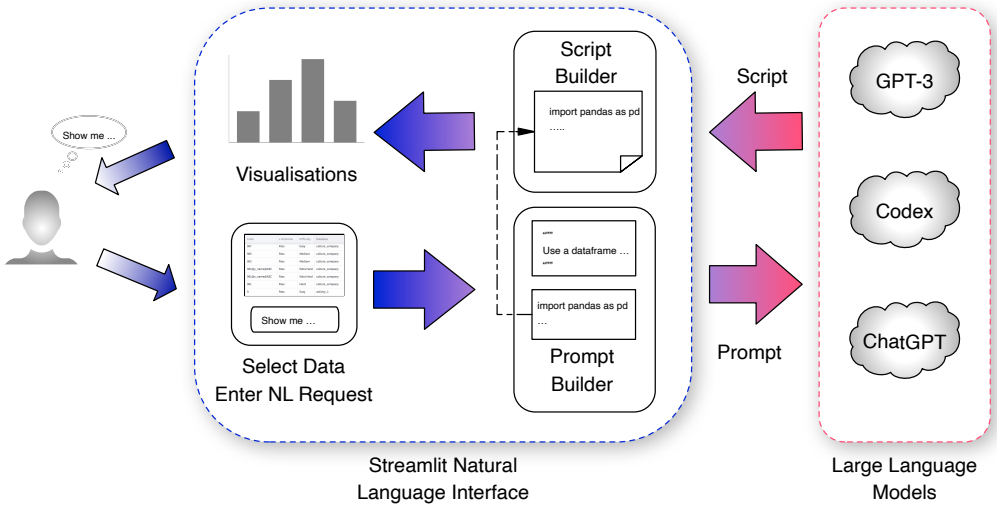

Fig. 2. The Chat2VIS architecture translating NL text into data visualisations via large language models.

or model retraining.

With the sparse existence of NL2VIS benchmarks, we seek to evaluate Chat2VIS against the only two baselines nvBench [19] and the NLV utterance corpus [23] identified in the existing literature. Evaluations [21] [23] against these benchmarks for current NL2VIS approaches provide a degree of comparison for this study. To that end, our analysis contributes to the gap distinctly evident in the literature regarding NL2VIS benchmarking.

3 NL2VIS Architecture

Chat2VIS generates data visualisations using free-form NL text. With an interface utilising OpenAI's state-of-the-art LLMs, it demonstrates unique decision-making skills to autonomously select chart types and plot elements.

3.1 Large Language Models

Chat2VIS is based on the Davinci family of models, currently some of the most capable and advanced model set available within the OpenAI suite. It employs GPT-3 model *"text-davinci-003"*, Codex model *"code-davinci-002"*, and contrasts results with the latest state-of-the-art ChatGPT model *"gpt-3.5-turbo"*.

Using OpenAIs text completion endpoint API to access Codex and GPT-3 models, it retains default parameters with the exception of adjustments to the following:

1) Temperature is a hyperparameter controlling output randomness in LLMs. A lower temperature (closer to zero) makes the model's outputs more deterministic and consistent, ideal for tasks like code generation. For this reason, we set the temperature to zero;
2) Evading excessively verbose scripts by setting the max_tokens parameter to 500 — an ample limit for this study; and
3) Requesting a stop parameter of *"plt.show()"*. This will cease generation upon plot rendering syntax - avoiding the LLMs presenting alternative scripts.

3.2 Chat2VIS

The Chat2VIS application[2] has been developed using Streamlit[3], an open-source Python library that enables developers to create interactive web applications rapidly. The adoption of Streamlit offers an effective means to encapsulate several components of the NL2VIS process, including user interface design, prompt engineering, LLM connectivity, and the subsequent generation and rendering of visualisations from the received scripts. As shown in Fig. 2, the architecture of Chat2VIS is designed with an interface (refer to Fig. 3) that allows users to interact with the application by entering a NL request. The request is specific to a selected dataset that the user wants to visualise. Upon receipt of the user's NL request, Chat2VIS begins the process of engineering the prompt. This process involves combining the NL request with a standardised prompt template. This template is common across all LLMs and is designed to include information about the data types present in the chosen dataset. This engineered prompt is then submitted to the selected LLM. The LLM returns a result that contains the Python code component, which is extracted by Chat2VIS. This Python code represents the visualisation instructions derived from the user's NL request. The application then executes this Python code within its environment and renders the result.

Fig. 4 illustrates the inner workings of the end-to-end capabilities of Chat2VIS (which does not rely on generating intermediate representations in the form of JSON like some systems), while a more concrete example of the prompt structure is seen in Appendix A in Fig. 12. The architecture is discussed by way of an example dataset created from the results of our benchmarking evaluation in Section 5.4. The process is described as follows:

1) Fig. 4(a) shows a sample of the dataset together with the query *"Plot the outcome."*, chosen for its ambiguity that lacks the explicit instruction of **how** to

2. https://chat2vis.streamlit.app/
3. https://streamlit.io/

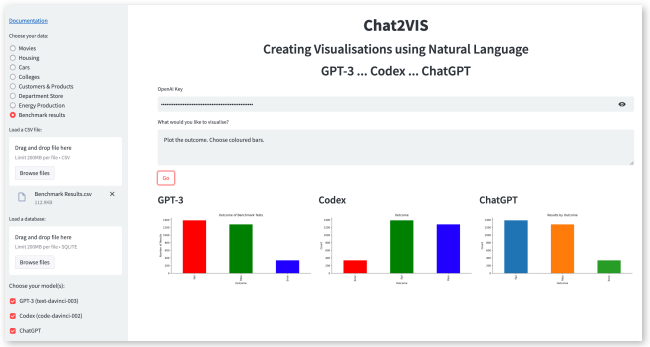

Fig. 3. Chat2VIS Interface

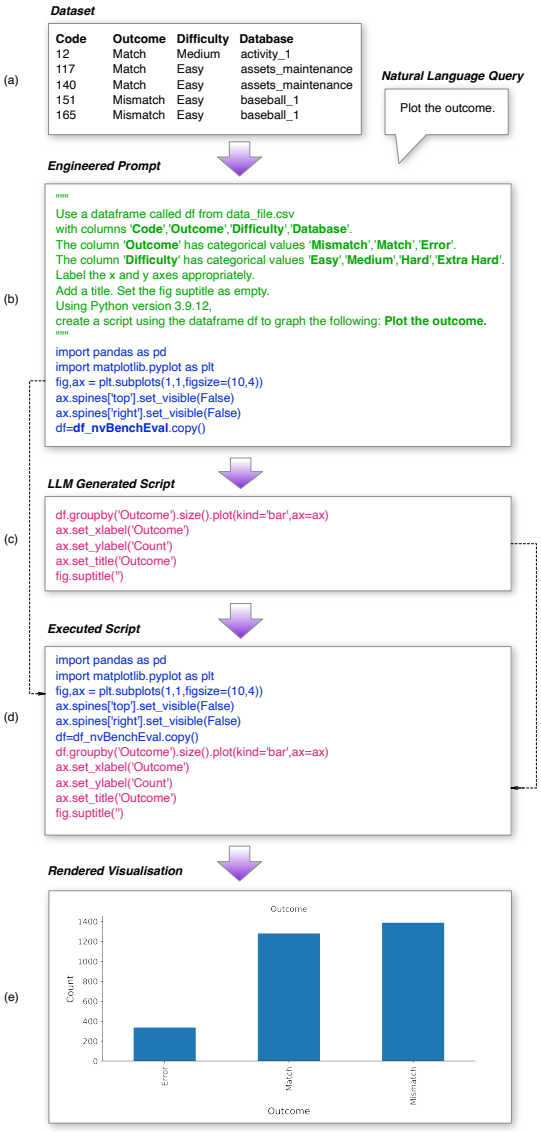

Fig. 4. Illustrated example of the process to convert a NL query into a data visualisation.

derive the desired values and **which** visualisation chart type to use.

2) The engineered prompt in Fig. 4(b) comprises of two parts: (1) a Python docstring description, encapsulated with triple double quotes (green) where columns names together with their data types are defined, including plotting instructions and the Python version to be used; (2) a Python code section providing a starting point for the requested script (blue). The **bolded** type highlights variable substitution values specific to this example.
3) Fig. 4(c) shows the Python script (red) returned by a selected LLM which forms the continuation of the code section within the engineered prompt. The returned script demonstrates both the advanced reasoning and the appropriate chart selection capabilities of the LLMs.
4) In Fig. 4(d), shows the script that is executed to generate the visualisation comprising the combination of the initial Python code (blue) from Fig. 4(a) and the script returned by the LLM (red).
5) The newly-created script is executed to render the requested visualisation, as pictured in Fig. 4(e).

It should be noted that the given prompting architecture has strong privacy-preserving properties. In most cases, only the column names and their data types are sent to the LLM. This mitigates ethical and data privacy risks associated with providing the models with sensitive data. Only in cases where a categorical data type column has less than 20 distinct values, its values are enumerated in the prompt and sent to the LLM; however, even this can be easily resolved by hashing and anonymising those values prior to sending them.

4 Methodology

This study explored (1) the refinement of plot aesthetics using NL queries, (2) the flexibility of Chat2VIS in comprehending multilingual requests, and (3) a quantitative evaluation of Chat2VIS against two benchmark datasets developed in previous studies. All figures in this paper are rendered through Chat2VIS to showcase its abilities.

4.1 Chart Refinements and Multilingual Requests

Previous work [22] confirmed the unique decision-making skills of the LLMs to autonomously select suitable chart types and plot aesthetics. Here, we demonstrate the system's efficacy with iterative refinements to the input query for nominating a specific chart type, enhancing plot elements, and changing styles.

Language choices were driven by the need to ensure semantically coherent requests were formulated, thus languages were prioritised based on the the research team's expertise and fluency. Also, the aim was to achieve a balanced representation of both high-resource languages, such as German and French, and lower-resource languages, like Croatian and especially te reo Māori. High-resource languages have abundant training data available, whereas low-resource languages have limited training data against which the LLMs would have been trained. This deliberate selection allowed the exploration of the performance of LLMs across languages with varying levels of training data exposure.

A multilingual case study is presented exemplifying both visualisation generation and the adjusting of chart labels using Chat2VIS. The results are assessed visually for accuracy.To achieve a wider coverage of more diverse languages, an additional Case Study using Spanish as well as non-Latin languages, like Mandarin and Japanese is included in the Appendix B.

4.2 Quantitative Evaluation

We conduct a more comprehensive quantitative analysis using the nvBench benchmark[4]. Encompassing 153 databases, 7,274 visualisations, and 7 chart types, it is considered the first public large-scale NL2VIS benchmark [19]. Given the size of the dataset, we propose an automated evaluation strategy. Each example instance comprises of a NL-to-visualisation pair, denoted (NL, VIS). Attributes are stored inside a JSON specification, permitting Vega-Lite chart rendering. Examples are further classified into four categories to denote the difficulty of the query — easy, medium, hard, and extra hard.

In addition, we perform a second evaluation using the NLV Utterance dataset[5] [23], referred to in our study as nlvUtterance. The benchmark covered three databases[6]: movies, cars, and superstore. This benchmark comprises 814 NL queries, with 10 visualisations for each database. Queries were generated from the results of an online study using 102 participants suggesting utterances for the display of each respective chart. Here we use a manual evaluation approach.

4.2.1 Model Selection

We select the Codex *"code-davinci-002"* model to measure results against nvBench. Codex, evolved from GPT-3, was trained on an immense amount of publicly-available GitHub code. It is skilled in more than a dozen programming languages, most notably Python, the underlying programming language of Chat2VIS. Codex is available in Davinci or Cushman models. Among the OpenAI suite of models, the Davinci family is the most capable, and can often perform all tasks of other models using fewer instructions. Cushman, although faster, is less competent than Davinci. In prioritising accuracy over speed, the Davinci model was regarded as the most appropriate choice for this task. Therefore, we deemed Codex *"code-davinci-002"* well-suited for this evaluation.

4.2.2 nvBench Benchmark Evaluation

Determining how to automatically assess the equality between a Chat2VIS chart and its nvBench counterpart is technically challenging. The benchmark specification omits guidance of any evaluation methodology determining what constitutes a match or mismatch. Our attempts to employ image comparison tools proved unreliable due to the complexity involved. Hence we devised a strategy that constructs vectors of the x and y coordinates for each plot, and uses these as a basis for comparative analysis.

Since Chat2VIS is designed to generate charts from a tabular dataset, we removed nvBench instances querying multiple SQL database tables to achieve interoperability. This methodology is consistent with the one used for ncNet [3]. The exclusion criterion relied on identifying the SQL `JOIN` operator inside the VQL mark within the nvBench JSON specification. In addition, we excluded examples containing SQL subqueries within the `WHERE` clause referencing tables distinct from the principal `SELECT` clause, again, in order to preserve compatibility.

Due to the difficulty in automating accurate comparisons across all chart types, we confined our benchmark testing to bar charts which constituted an overwhelming majority of samples. Automating the comparisons against chart types like line, pie and scatter would necessitate devising alternative and tailored comparison mechanisms. nvBench includes up to 5 queries for a given visualisation. In our methodology, we chose the first NL query for inputting into Chat2VIS under the assumption that the first one likely represents a most reasonable expression of intent. Fig. 5 illustrates an example JSON specification[7] for the (VIS, NL) pair *"474@x_name@DESC"*, highlighting areas of interest within the specification discussed in this evaluation approach. The final benchmark test set across 138 databases comprised 3,003 instances, with 812 considered *easy*, 1572 *medium*, 386 *hard*, and 233 *extra hard*.

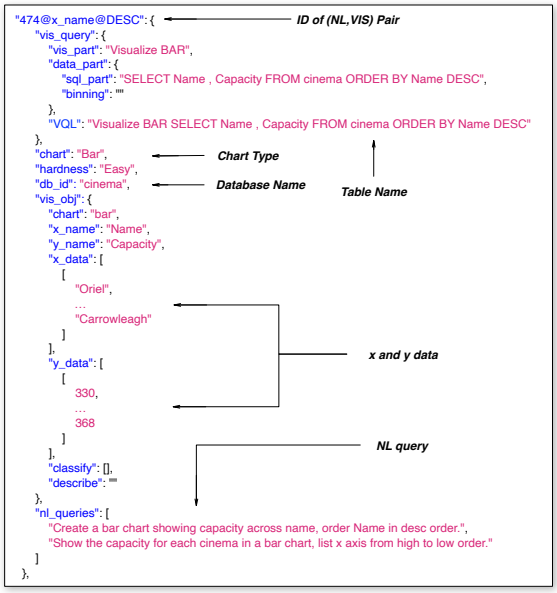

Fig. 5. Example JSON specification from nvBench.

Fig. 6 summarises our proposed automated testing methodology. The steps outlining the methodology are described below:

1) Select a JSON test specification from nvBench.
2) Extract the database name from the db_id field, and the table name specified in the VQL field following the FROM keyword.
3) Import the SQLite database table into a Python Pandas DataFrame structure.
4) Extract the first query from the nl_queries field.

4. https://sites.google.com/view/nvbench
5. https://nlvcorpus.github.io/
6. https://github.com/TsinghuaDatabaseGroup/nvBench/databases.zip
7. https://github.com/TsinghuaDatabaseGroup/nvBench/blob/main/NVBench.json

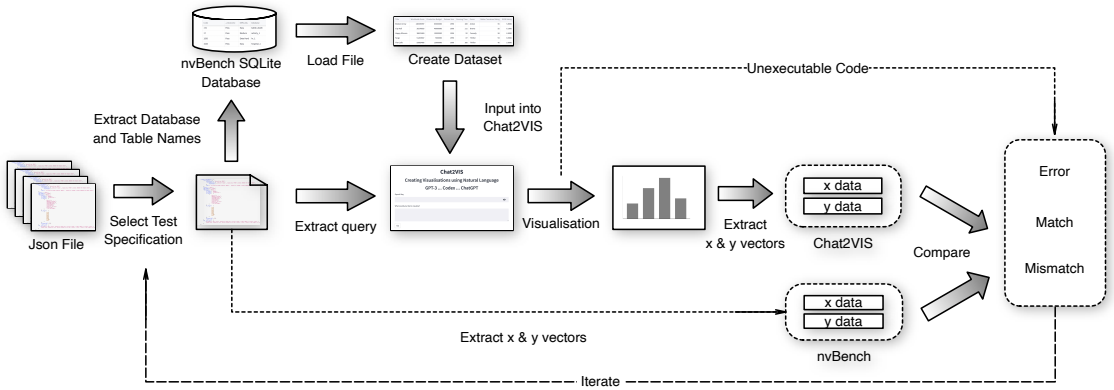

Fig. 6. Overview of nvBench Benchmark Testing.

5) Submit query and dataset to Chat2VIS opting for Codex, and render visualisation.
6) Document the outcome as an `Error` should the code fail to execute.
7) Construct x and y data coordinate vectors by extracting the Chat2VIS chart elements.
8) Construct x and y data coordinate vectors using the x_data and y_data fields in the nvBench JSON specification.
9) Apply adjustments to the vectors to address complications impacting a successful comparison:
   - Ensure naming consistency of calendar units, such as recasting Tue, Tues, and tuesday as Tuesday; Sept, Sep september as September.
   - Sort by ascending y values if keywords "sort" and "order" are not specified.
   - Cast integer values to floats, and round all floats to 5dp.
10) Compare Chat2VIS and nvBench vectors. A precise match classifies the outcome as a `Match`, else it is classified as a `Mismatch`, while an `Error` classification indicates that a visualisation failed to be rendered most-frequently attributed to a Python code error.

It should be noted that a `Mismatch` does not necessarily mean that the generated visualisation is incorrect, and may in some instances even be a more effective and appropriate visualisation (see Figure 14 in Appendix C).

4.2.3 nlvUtterance Benchmark Evaluation

A manual visual inspection of the results on the nlvUtterance dataset was used due to its smaller size. All chart types within nlvUtterance were used, namely, bar charts, histograms, line charts, and scatter plots, together with their variations. As outlined in the benchmark description [23]: histograms and single attribute bar charts are used to visualise one categorical or quantitative attribute; bar charts, scatter plots, and line charts for two attributes; and grouped bar charts, stacked bar charts, multi-line charts, coloured scatter, and faceted scatter charts for visualising three or more attributes. 755 of the 814 queries are considered *"singleton"* utterance sets, consisting of a single query request. The remaining 59 instances are considered *"sequential"* utterance sets, containing multiple utterances. After removing erroneous plots without data, the final dataset consisted of 758 queries for testing.

Chat2VIS renders charts for up to 3 models. We employed a three-stage testing methodology. Firstly, the queries are submitted to Codex and the corresponding performance metrics are presented. Secondly, unsuccessful queries are submitted to GPT-3, with the corresponding performance again measured. Finally, any remaining mismatched queries are submitted to ChatGPT. The overall performance statistic for successful matches provides insight into the likelihood that a benchmark result will be generated by at least one LLM.

It is not the intention of this work to compare LLMs *inter se*, but instead contrast the use of LLMs with alternative approaches. Therefore, we do not present benchmark metrics comparing the performance accuracy of Codex, GPT-3 and ChatGPT relative to each other.

5 Results

We demonstrate the conversational ability to refine charts with subsequent requests on the first two case studies, followed by the third case study illustrates multilingual requests. The dataset used for the illustration are results from the nvBench evaluations. Finally, we analyse Chat2VIS' performance against the two benchmarks using the illustrations generated by Chat2VIS.

5.1 Case Study 1: Conversational Chart Refinement

Fig. 7 demonstrates the first conversational refinement of Fig. 4(e) *"Plot the outcome."* to *"Plot the outcome by difficulty."* which illustrates language pragmatics in the clarification of intent. The subsequent refining requests that the chart be rendered *" as a stacked bar plot. Use red for error, light green for match, blue for mismatch."*, followed by a request to refine it with the instruction to *Increase the font size of the axis labels and numbers. Make the title 'Evaluation Results by Difficulty Level' with very large font.*. The figures demonstrate that all LLMs

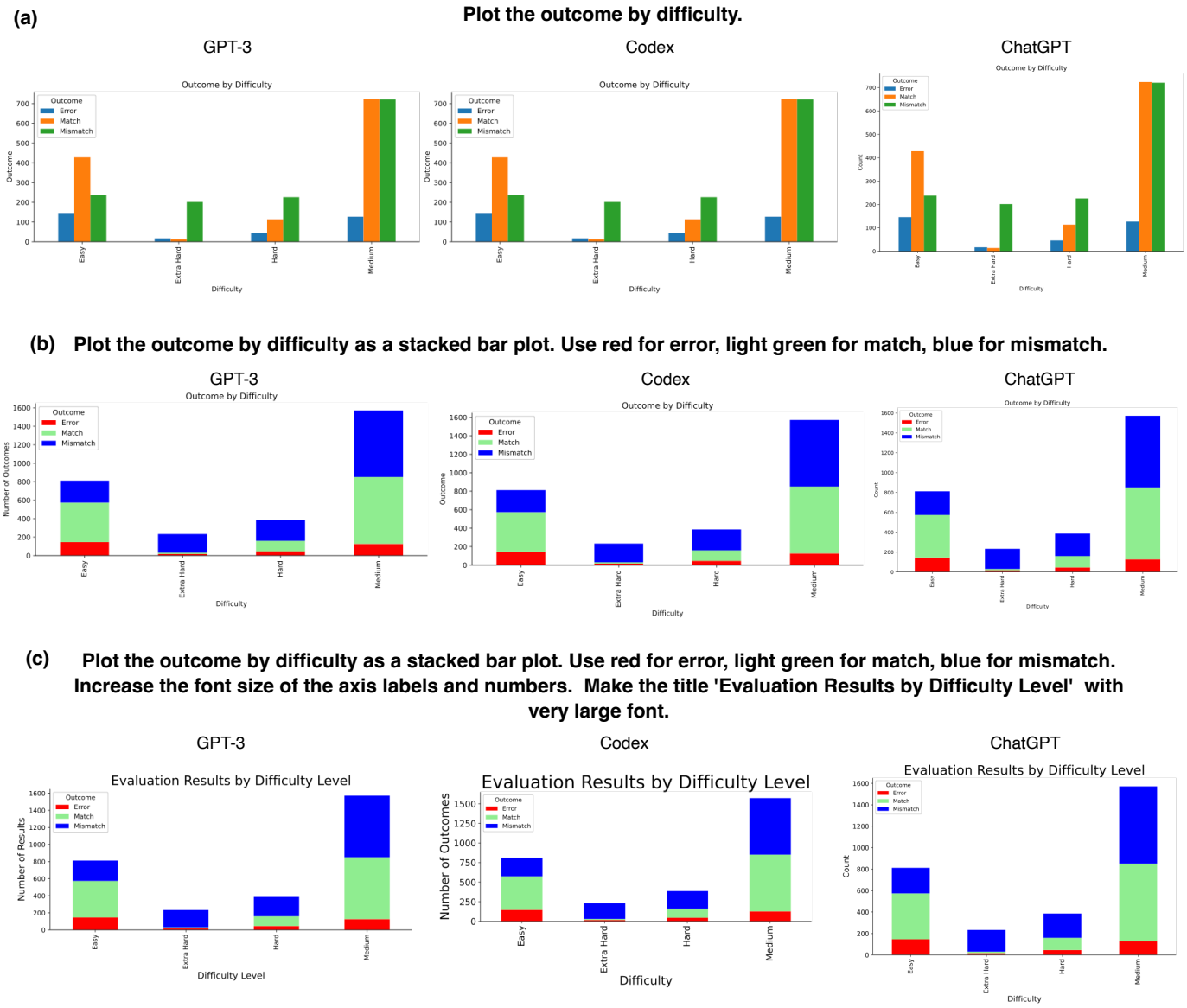

Fig. 7. Case Study 1: Conversational Visualisation Refinements using the nvBench plotting Evaluation Results by Difficulty

interpreted the requests reliably while also demonstrating differences in how they interpreted vague font size requests.

5.2 Case Study 2: Conversational Chart Refinement

Again, the base query *"Plot the outcome."* from the dataset example in Fig. 4(e) serves as a starting point. Illustrations in Fig. 8 demonstrate refining the query with the subsequent instruction *"...using a pie chart. Hide axis labels. Use pastel colours."*. Results show a high accuracy across all LLMs. All LLMs converted the original bar graph into a pie chart successfully. From the perspective of chart layouts, GPT-3 differed in the rendering of the pie graph by separating out the individual slices. GPT-3 and Codex did not follow the instruction to remove the axis label *"Outcome"* from the title, while ChatGPT did and instead reasoned to formulate a meaningful title *"Distribution of Outcome Categories"*. The command to apply pastel colours was followed by both Codex and ChatGPT. Overall, given this example, only ChatGPT followed every refinement request.

5.3 Case Study 3: Multilingual Requests

The capability of the LLMs to comprehend mixed multilingual requests for conversational chart refinement is demonstrated here and depicted in Fig. 9(a), building on the initial prompt *"Plot the outcome."* from Fig. 4(e) to plot results categorised by difficulty (Test 1) in French as *"Regroupez la difficulté par résultat sous forme de graphique à barres.* The refining requests (Test 2) that the *outcome* be plotted along the x-axis also using French *l'axe des x est le résultat."*, translated[8] as *"Group difficulty by outcome as a bar chart. The x axis is the result."*. Subsequently (Test 3), the plot is refined by asking for the title "Benchmark Results" to be altered using Croatian *"Promijenite naslov u 'Rezultati benchmarka'."*. Each LLM correctly rendered the refinements, while retaining both legend and bar labels in English. Meanwhile, GPT-3 and Codex translated both axes' labels into Croatian, while ChatGPT preserved the English labelling.

8. via Google translate https://translate.google.com

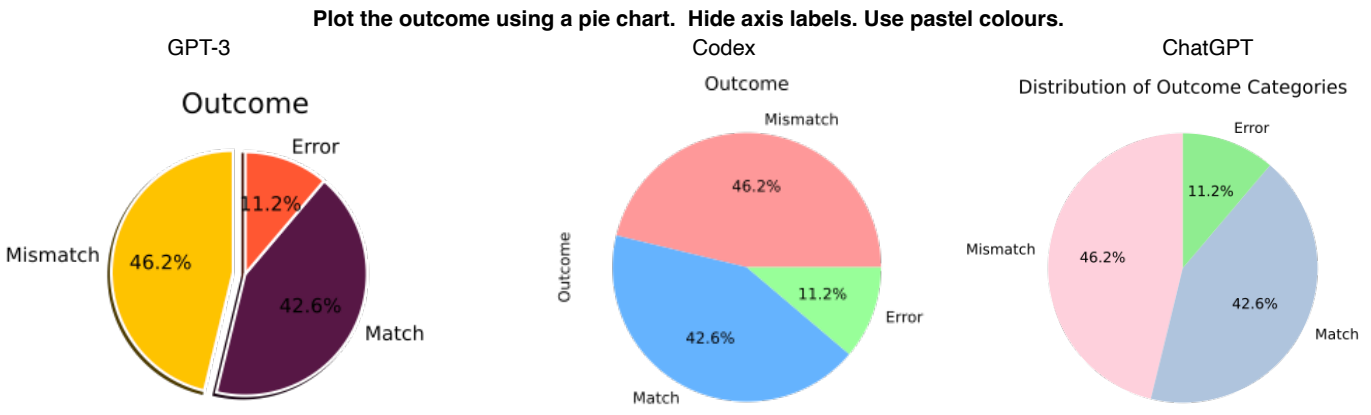

Fig. 8. Case Study 2: Conversational Visualisation Refinements using the nvBench plotting Evaluation Results by Outcome

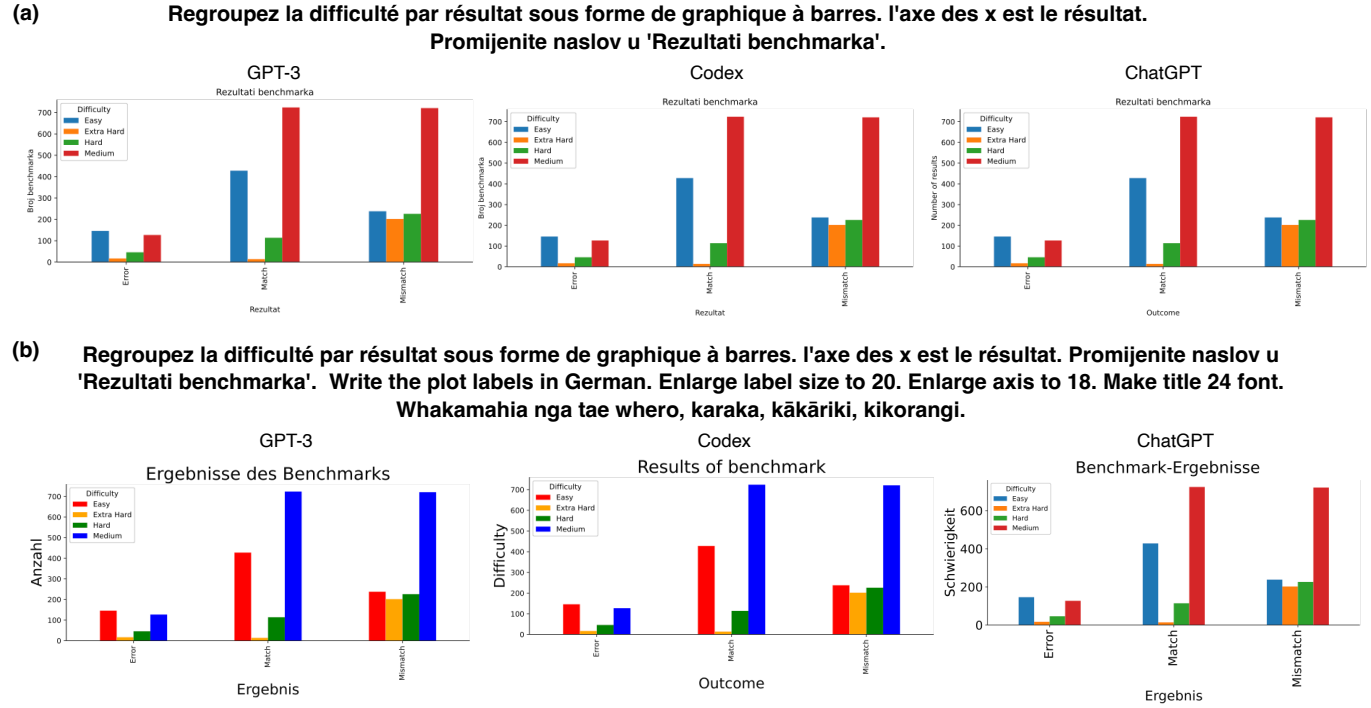

Fig. 9. Case Study 3: Conversational Visualisation Refinements using the nvBench plotting Results Data via Multilingual Requests.

The subsequent fine tuning can be seen in the next plot in Fig. 9(b) where an instruction (Test 4)is issued in English *...Write the plot labels in German.*, (Test 5) *Enlarge label size to 20.*, (Test 6) *Enlarge axis to 18.*, (Test 7) *Make title 24 font.*.

Following these instructions for refinement, , next, (Test 8) a change in colour ordering of plot bars is issued in New Zealand's te reo Māori (for which there is limited linguistic data available for training LLMs) with the phrase *"Whakamahia nga tae whero, karaka, kākāriki, kikorangi."*, meaning that the colours red, orange, green and blue be used. Following a complex suite of refinement instructions, it can be seen that all 3 LLMs responded to Tests 1 and 2 in French correctly, as well as to Croatian in Test 3. GPT-3 and Codex reasoned one step further without being instructed, and also modified the x axis label to 'Rezultat' in Croatian to match the title, while ChatGPT left this aspect unaltered in English. Test 4 requesting that labels be translated to German was followed by GPT-3 and ChatGPT, while Tests 5 to 7 requesting font and label sizes were only completely followed by ChatGPT. In the final test (Test 8) in te reo Māori, GPT-3 and Codex correctly interpreted the colouring request, with ChatGPT retaining its original colour ordering scheme.

Overall, though none of the LLMs achieved a perfect performance across all tested languages, they nonetheless demonstrated high-fidelity results with arguably ChatGPT outperforming the others on these examples.

5.4 Evaluation against nvBench

When evaluated against nvBench, Chat2VIS demonstrates significant potential. Out of the 3,003 queries tested, 1,280

charts showed an exact match. This 43% "Match" rate, as illustrated in Fig. 4(e), is a notable achievement considering both the narrow and strict conditions used to define a match and the overall complexities involved in data visualisation tasks. Fig. 7 presents summary statistics of the results by difficulty level, further emphasising the effectiveness of the system under various scenarios. Overall only 336 (11%) instances represented charts which could not be rendered, primarily due to erroneous code generated by the LLMs. A detailed examination of both the nature of successful and mismatched instances where our analysis has enabled the identification of certain conditions under which the system performs particularly well is given next.

5.4.1 nvBench Benchmark Matches

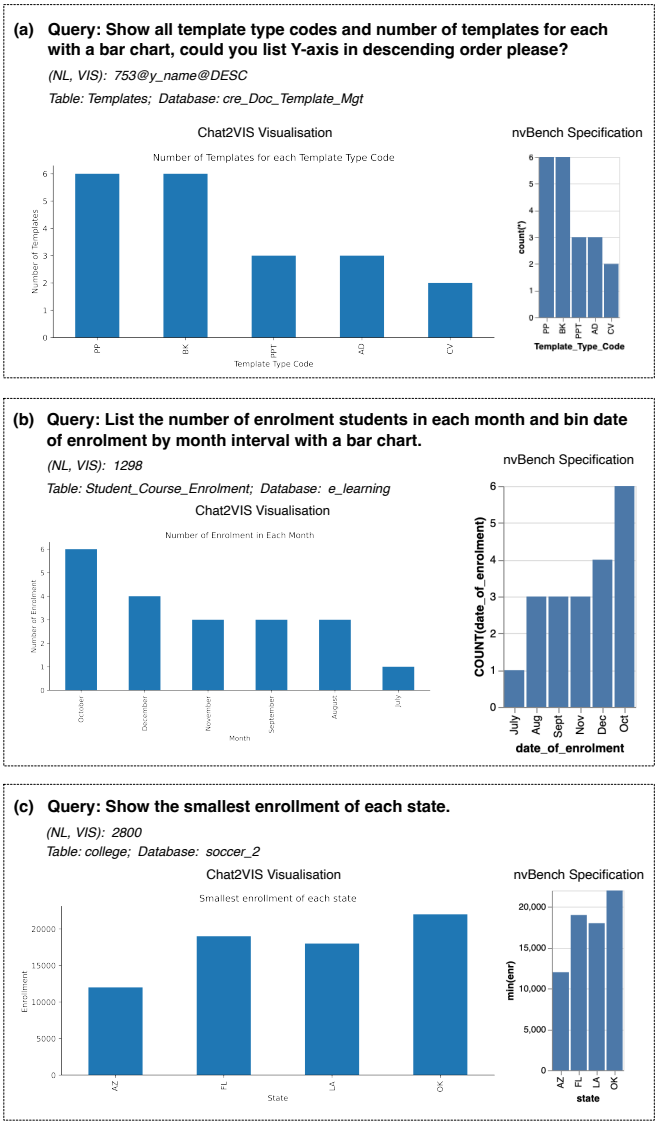

Fig. 10. Examples of Matched Visualisations Between Chat2VIS and nvBench

Chat2VIS excels in producing exact matches to the nvBench benchmark in a number of specific scenarios which are described and contextualised with examples as seen in Fig. 10, Chat2VIS tends to produce matches:

- in instances when the plot type is explicitly specified in the query (exemplified in Fig. 10a)
- in cases when the request for ordering or sorting is explicitly provided within the query, and in instances when all the grouped categories have different numerical values since ties can result in mismatched ordering (Fig. 10a shows a match which would register as a mismatch if 'PP' and 'BK' categories were rendered in a different order to the benchmark)
- in scenarios where the query is specific and is unambiguous (exemplified in Fig.10a and b)
- in instances where the specified plot type in the query aligns with the nature of the requested data (exemplified in Fig. 10a and b)
- in cases when the benchmark plot also accurately represents the information in the database and correctly interprets the query request (Fig. 10a-c)
- in instances when the plotting data did not require visualising non-zero values (Fig. 10a and c)
- in scenarios when the database structure itself correctly represents the data types it holds, and the null values are represented in line with database standards (Fig. 10a - c)
- in cases when grouping information such as 'by Weekday', 'by Month', etc., and when this is explicitly provided in the query (Fig. 10b).

5.4.2 Benchmark Mismatches

Here we summarise broad categories under which mismatches were detected with the benchmark. Many are a result of applying too narrow a definition of a match when devising an automated system which does not have subjective capabilities, while a large percentage of the mismatches can be attributed to deficiencies within the benchmark. We observed instances where the Chat2VIS chart was 'correct' in its rendering, however, errors within the nvBench specification and the inconsistency within the database information resulted in the mismatch classification. We illustrate in Fig. 14(a) a discrepancy in categorical values with variant spellings of *"Andreo"* and *"Lo"* in the nvBench specification contrasted with *"Andreou"* and *"Lou"* persisted in the database. We noted the presence of at least five nvBench specifications omitting the first query. On passing an empty string to Chat2VIS, the LLM demonstrates its effective decision-making to render a visualisation of interest, as shown in 14(b).

In a subset of examples, we found inconsistencies between the query request and the SQL denoted in the nvBench specification. Our findings unearthed instances where the SQL included an `ORDER BY` clause but the NL query omitted instructions to *"order"* or *"sort"* results. Consequently, the Chat2VIS visualisation accurately depicted the data points but was classified as mismatched with that of nvBench's sorted chart. These examples illustrate repeatedly that a mismatch in a strict sense when using automated approaches is not necessarily an incorrect output.

Our methodology utilises only the query component of the specification, disregarding other instructions contained within. Consequently an assortment of examples stored requests for grouping results inside the "binning" mark of the specification, therefore omitting to incorporate the request

as part of the NL query. We illustrate in Fig. 14(c) a common mistake where the Chat2VIS chart summarised by date, and the equivalent nvBench specification grouped by weekday, an instruction solely provided within the "binning" mark.

Further inspection of mismatches showed periodically, without additional stipulations inside the specification, nvBench incorrectly renders only a partial result set, while Chat2VIS visualises the complete set of data. An example is depicted in Fig. 14(d). The reasoning for such behaviour within nvBench is unknown. Additionally, we noted for some instances executing the SQL query from the nvBench specification directly against the database yields conflicting figures to those visualised by nvBench, as shown in Fig. 14(e). Furthermore, we encountered truncating of numeric float types to integer values without additional stipulations inside the specification as shown in Fig 14(f). Consequently causing additional mismatches between nvBench and Chat2VIS results.

5.4.3 Methodology Limitations

Our approach to automated benchmarking against nvBench, in the absence of a standardised methodology, presents some limitations. Occasionally, Chat2VIS generates visually accurate charts but the proposed comparison methodology does not always yield expected results. Differences in the treatment of categories devoid of data (Chat2VIS omits, nvBench assigns zero) lead to mismatches, despite similar visual outcomes, as shown in Fig. 14(g).

The nvBench queries often specify the ordering or sorting of results based on a nominated axis, either in ascending or descending order. As we do not rearrange the x and y vectors if sorting is requested, an exact match is necessary for the charts to be equivalent. However, this approach presents a challenge when multiple x values have identical y values, rendering a correctly ordered, but not identical, chart. We illustrate this assorted ordering in Fig. 14(h) with the majority of bars having value 1.

Occasionally, LLM's unconventional plotting arising from Python's diverse charting techniques, may lead to unexpected outcomes for x and y vectors, resulting in null values at some points of the chart resulting in mismatches while not necessarily being incorrect.

We noted that numeric fields in the nvBench database occasionally contain missing values represented as empty strings instead of NULL values, causing issues with Python's data import and query execution and mismatches. Enhancing the Chat2VIS interface could mitigate this.

Similarly, columns holding numeric data but defined as character type could result in incorrect mathematical computations. Sometimes, older versions of Python syntax generated by the LLM and verbose scripts exceeding token restrictions also led to execution errors.

5.4.4 Ambiguity

Query text ambiguity led to different interpretations and charts by the LLM and nvBench making genuine accuracy assessment challenging. Queries like *"...sort bars in desc order"* or *"...order by the bars from high to low"* were interpreted by Chat2VIS as sorting by y-axis values, whereas nvBench sorted x-axis labels alphabetically, as shown in Fig. 14(i).

Even though bar charts are typically used for comparing group values, we observed instances where they were used to represent two text columns. Fig. 14(j) shows nvBench's approach to this situation, while Chat2VIS inferred a different but more appropriate visualisation, albeit aesthetically unappealing, providing category counts.

5.4.5 Query Misinterpretation

Occasionally, the LLM misinterpreted queries, causing Chat2VIS to generate incorrect visualisations. Date and time values were challenging. While nvBench correctly used only the date component for grouping, Chat2VIS sometimes considered both date and time, leading to individual groupings of date-times.

The nvBench construction methodology often generated similar benchmark examples. Filter requests such as *"...commission is not null or department number does not equal to 40..."* were present in over 70 instances. Chat2VIS incorrectly used the *"and"* operator instead of *"or"* when interpreting this filter, contributing to a disproportionate amount of mismatched results.

We also noted instances where Chat2VIS erroneously self-imposed a limit on the number of returned results and also misunderstood a Chinese language request. These incidents highlight the LLM's occasional misinterpretations, despite its general proficiency in generating Python code from natural language.

5.5 Evaluation against nlvUtterance

Chat2VIS demonstrates robust results against the nlvUtterance benchmark with respect to the strict methodology used. 50% match rate was observed over all chart types when using Codex. This rate improved to 63% when either Codex or GPT-3 produced a matching chart. Meanwhile, the matching rate rose to 72% when at least one of Codex, GPT-3, or ChatGPT generated a matching chart. Fig. 11 presents the evaluation results, categorized by chart type for each stage of testing. Again, it is worth noting that not all mismatches between Chat2VIS and nlvUtterance generated charts imply inaccuracies. The ambiguity within the query and lack of charting specifications often led to alternative yet 'correct' visualisations, which, due to the lack of defined evaluation guidelines and parameters within the benchmark, were sometimes deemed as mismatches in our objective evaluation.

5.5.1 nlvUtterance Benchmark Matches

Chat2VIS exhibited a high degree of matches under several conditions. Firstly, when single-attribute bar plots and scatter plots were utilised, the system generated matches. These types of plots had the highest representation in the dataset, and the system managed to match them at a high rate. Evaluations with Codex, shown in Fig. 11(a), demonstrated that coloured scatter plots and faceted scatter plots had significantly lower matching rates. However, when generated via GPT-3 (Fig. 11(b)) or ChatGPT (Fig. 11(c)), the results were much more favourable. Secondly, the system showed high matching rates with bar charts when generated via the three LLMs.

5.5.2 nlvUtterance Benchmark Mismatches

Nonetheless, our evaluation also revealed instances where the system struggled, particularly with grouped and stacked bar charts, histograms, coloured and faceted scatter charts, and single and multi-line charts. Fig. 15 presents a sample of generated visualisations for the ten chart types based on the movies dataset, enabling comparisons with those presented in prior work [23]. Specific chart types are detailed below with respect to the generation of mismatches.

Bar Charts: A substantial number of grouped bar charts were classified as a mismatch notwithstanding correct charting of the requested data. Considering the query *"average production budget by creative type and content rating"*, the benchmark arranged results grouped by content rating using colour coding to represent the creative type. However, periodically the Chat2VIS counterpart inversely grouped results by creative type with colours representing content rating. Similar issues were observed with mismatches between stacked bar charts. Furthermore, our findings showed instances where benchmark stacked bar charts were presented by Chat2VIS as grouped bar charts, accurately conveying the information, but not in accordance to the benchmark standard. Had the evaluation methodology deemed these visualisations as matches, as indeed they were despite deviating from the benchmark, the accuracy (match) rate would increase significantly. Nonetheless, the absence of methodological instructions within the benchmark failed to provide guidance in such circumstances.

Histograms: We refrained from providing explicit instructions to the LLM on which chart type to render. Consequently a significant number of benchmark histograms were instead plotted as bar charts. Queries such as *"How many orders were placed for each order quantity?"* and *"show me a bar chart of count by order quantity"* did not imply the data should be represented as a histogram, and hence the LLM decided the most appropriate representation of the data was in the form of a bar chart. Furthermore, queries neglected to provide information pertaining to binning size, and consequently the LLM's decision often conflicted with benchmark visualisations.

Scatter Charts: Colour scatter charts use varied colours to represent categories in a single plot. Codex often overlooked this colour coding leading to single-coloured charts. GPT-3 did not share this limitation, but both exhibited a high percentage of mismatches due to incorrect syntax, often incorrectly setting the *"c"* colour parameter value in the Python plotting function. This misstep resulted in erroneous script execution. ChatGPT was more successful, correctly assigning this function parameter. Faceted scatter charts separate categories from coloured scatter plots into distinct charts. Some queries lacked clear instructions for a faceted chart, prompting us to accept single scatter plots categorised by colour. As in the case of coloured scatter charts, Codex often failed to use colour coding to distinguish categories, while GPT-3 and ChatGPT did not have this limitation. However, due to the lack of a benchmark methodology, we accepted alternative charts, which could otherwise have affected the success rate.

Line Charts: The least-represented chart types in the dataset are single and multi-line plots. The most common cause of mismatch was the LLM selecting to render the information as a bar chart. However, although it still accurately presented the requested information when a line chart was not explicitly requested, it was not in accordance with benchmark specifications. In addition, Chat2VIS multi-line plots on occasion inversely rendered the x-axis and line colour categories compared to that of nlvUtterance, hence unsuccessful in meeting benchmark standards. Once more, these decisions of determining if benchmark standards are met significantly impact culminating a successful outcome.

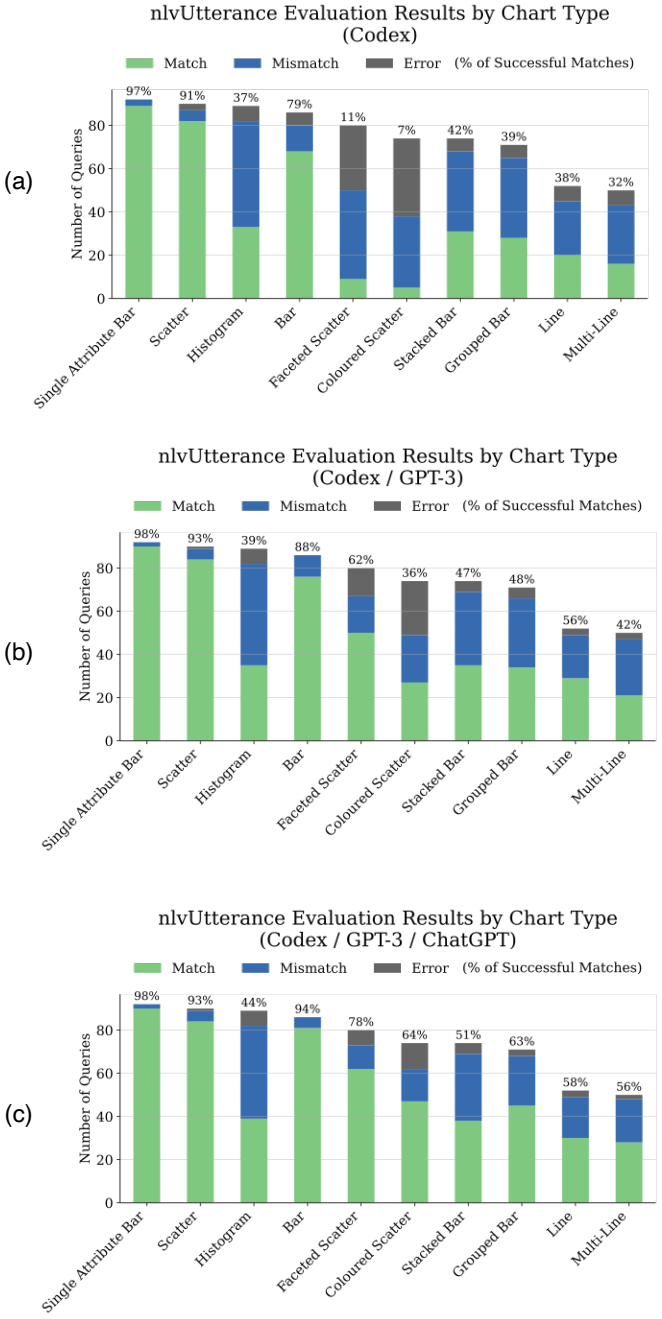

Fig. 11. nlvUtterance Evaluation Results.

5.6 Comparison With Previous Work

Finally, Chat2VIS is contrasted with results from prior studies' on the nvBench benchmark. The comparisons are suggestive but are not exact since our nvBench sample set differs from test set used in previous studies. The overall accuracy of state-of-the-art NL2VIS systems are shown in Table 1 as reported by [21] and contrasted with Chat2VIS, indicating a highly competitive performance of the proposed system.

TABLE 1
nvBench Performance Comparison.

System	Accuracy
Seq2Vis [24]	2%
Transformer [25]	3%
ncNet [3]	26%
RGVisNet [21]	**45%**
Chat2VIS	43%

Table 2 summarises our findings using the nlvUtterance benchmark, with those from NL4DV evaluated on the benchmark in previous studies [23]. It should be noted the NL4DV results are based on a 755 instance dataset pertaining to singleton query sets only. In our study we included all query sets, with the exclusion of those pertaining to the two omitted line charts.

TABLE 2
nlvUtterance Performance Comparison.

System	Accuracy
Chat2VIS (Codex)	50%
Chat2VIS (Codex/GPT-3)	63%
Chat2VIS (Codex/GPT-3/ChatGPT)	72%
NL4DV	64%

6 Discussion

The experiments confirm the advanced capability of Chat2VIS to provide a state-of-the-art solution to the NL2VIS problem, exemplifying the conversational ability to refine chart aesthetics and to do so with multilingual instructions.

Furthermore, we provided results of Chat2VIS against two benchmarking datasets and confirmed the state-of-the-art properties of the proposed system. Given the time-consuming and subjective nature of performing these necessary evaluations, we made a contribution towards the development of automated approaches to help researchers and accelerate advancements in this field. Our proposed automation methodology is a first step in realising this goal and has yielded findings that are helpful in advancing the refinement of existing benchmarks and the development of new ones.

While we gratefully applaud the enormous efforts invested by researchers in developing the existing benchmark datasets, we find that there is room for improvement in fulfilling all the necessary quality characteristics like *reproducibility*, *fairness*, and *verifiability* as defined by [4]. The large number of diverse visualisation elements, aesthetics, and chart styles, from a variety of available programming language libraries raises difficulties in generating *reproducible* and measurable outcomes from NL queries. The predominant use of Vega-lite specifications in current benchmarking studies [23] [19] periodically separates important visualisation information from the NL query, limiting the effectiveness of alternative NL2VIS architectures and reducing *fairness*. Inconsistencies exist between test case data and visualisation outcomes thus compromising *verifiability*.

In future, to establish robust benchmarks for NL2VIS, we foresee definitions of a collection of valid visualisations for each NL query accompanied by a comprehensive methodology for chart comparison and evaluation. The evaluation would be independent of charting frameworks.

Study Limitations

Our study investigated the use of LLMs for NL2VIS tasks, employing automated benchmarking. However, some limitations are noteworthy. Technical issues hindered comprehensive testing against nvBench chart types such as line, pie, and scatter, which would require custom comparison mechanisms. We evaluated Chat2VIS against nvBench using only the first equivalent NL query, potentially introducing some bias, though the first NL query was deemed as expressive as any other. While we acknowledge the importance of ethical aspects like reliability, robustness, and possible misuse of LLMs, these fell outside our study's core focus and are the subject of ongoing research. Similarly, an exhaustive analysis of the role of prompt engineering was beyond our scope, despite its potential impact on LLM performance. We formulated a single, generic prompt architecture applicable across various LLMs that solves the research gaps in NL2VIS, but future studies should explore alternative configurations. Our benchmarking process mainly relied on concrete examples, leaving scope for future evaluations to consider a range of language intents and the handling of 'dirty data' and domain-specific terms. This would provide a more realistic evaluation of NL2VIS systems. Lastly, our study focused on a single-step conversion of NL into Python visualisation code. The potential benefits of using structured expressions as an intermediate step in the NL2VIS pipeline were not explored. Such an approach could offer more consistent and accurate outputs and is a promising direction for future research.

7 Conclusion

This seminal study presents the novel features of Chat2VIS for converting natural language into data visualisations in a conversational manner with the ability to refine charts in multiple languages, addressing a previously unsolved research problem.

We demonstrated the capabilities of our system against two benchmarks and have proposed a novel approach for automating the evaluation and comparison of generated visualisations thus contributing towards an additional gap in literature. We explored the challenges in accomplishing this and have identified areas for improvement in the development of benchmark datasets in the field of NL2VIS in order to accelerate the development of future advancements.

REFERENCES

[1] L. Shen, E. Shen, Y. Luo, X. Yang, X. Hu, X. Zhang, Z. Tai, and J. Wang, "Towards natural language interfaces for data visualization: A survey," *IEEE Transactions on Visualization and Computer Graphics*, 2022.
[2] Y. Wang, Z. Hou, L. Shen, T. Wu, J. Wang, H. Huang, H. Zhang, and D. Zhang, "Towards natural language-based visualization authoring," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 1222–1232, 2022.
[3] Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin, "Natural language to visualization by neural machine translation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 217–226, 2021.
[4] S. Kounev, K.-D. Lange, and J. von Kistowski, *Systems Benchmarking: For Scientists and Engineers*. Springer, 2020.
[5] G. Liu, X. Li, J. Wang, M. Sun, and P. Li, "Extracting knowledge from web text with monte carlo tree search," in *Proceedings of The Web Conference 2020*, 2020, pp. 2585–2591.
[6] Cox, Kenneth and Grinter, Rebecca E and Hibino, Stacie L and Jagadeesan, Lalita Jategaonkar and Mantilla, David, "A multi-modal natural language interface to an information visualization environment," *International Journal of Speech Technology*, vol. 4, pp. 297–314, 2001.
[7] Y. Sun, J. Leigh, A. Johnson, and S. Lee, "Articulate: A semi-automated model for translating natural language queries into meaningful visualizations," in *Smart Graphics: 10th International Symposium on Smart Graphics, Banff, Canada, June 24-26, 2010 Proceedings 10*. Springer, 2010, pp. 184–195.
[8] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios, "Datatone: Managing ambiguity in natural language interfaces for data visualization," in *Proceedings of the 28th annual ACM symposium on user interface software & technology*, 2015, pp. 489–500.
[9] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang, "Eviza: A natural language interface for visual analysis," in *Proceedings of the 29th annual symposium on user interface software and technology*, 2016, pp. 365–377.
[10] X. Qin, Y. Luo, N. Tang, and G. Li, "Deepeye: Visualizing your data by keyword search," in *EDBT*, 2018, pp. 441–444.
[11] A. Narechania, A. Srinivasan, and J. Stasko, "Nl4dv: A toolkit for generating analytic specifications for data visualization from natural language queries," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 369–379, 2020.
[12] B. Yu and C. T. Silva, "Flowsense: A natural language interface for visual data exploration within a dataflow system," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 1–11, 2019.
[13] E. Loper and S. Bird, "Nltk: The natural language toolkit," *arXiv preprint cs/0205028*, 2002.
[14] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The monted corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 2014, pp. 55–60.
[15] Q. Wang, Z. Chen, Y. Wang, and H. Qu, "A survey on ml4vis: Applying machine learning advances to data visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 12, pp. 5134–5153, 2022.
[16] H. Voigt, M. Meuschke, K. Lawonn, and S. Zarrieß, "Challenges in designing natural language interfaces for complex visual models," in *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, 2021, pp. 66–73.
[17] C. Liu, Y. Han, R. Jiang, and X. Yuan, "Advisor: Automatic visualization answer for natural-language question on tabular data," in *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*. IEEE, 2021, pp. 11–20.
[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2018, pp. 4171–4186.
[19] Y. Luo, J. Tang, and G. Li, "nvbench: A large-scale synthesized dataset for cross-domain natural language to visualization task," *arXiv preprint arXiv:2112.12926*, 2021.
[20] J. Tang, Y. Luo, M. Ouzzani, G. Li, and H. Chen, "Sevi: Speech-to-visualization through neural machine translation," in *Proc. of the 2022 International Conference on Management of Data*, 2022, pp. 2353–2356.
[21] Y. Song, X. Zhao, R. C.-W. Wong, and D. Jiang, "Rgvisnet: A hybrid retrieval-generation neural framework towards automatic data visualization generation," in *Proc. of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1646–1655.
[22] P. Maddigan and T. Susnjak, "Chat2vis: Generating data visualisations via natural language using chatgpt, codex and gpt-3 large language models," *IEEE Access*, vol. 11, pp. 45 181–45 193, 2023.
[23] A. Srinivasan, N. Nyapathy, B. Lee, S. M. Drucker, and J. Stasko, "Collecting and characterizing natural language utterances for specifying data visualizations," in *CHI '21: Proc. of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, p. 1–10.
[24] Y. Luo, N. Tang, G. Li, C. Chai, W. Li, and X. Qin, "Synthesizing natural language to visualization (nl2vis) benchmarks from nl2sql benchmarks," in *Proceedings of the 2021 International Conference on Management of Data, SIGMOD Conference 2021, June 20–25, 2021, Virtual Event, China*. ACM, 2021.
[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, pp. 6000–6010, 2017.

APPENDIX A
PROMPT ENGINEERING

Fig. 12 depicts the structure of the LLM prompt for a given concrete example.

APPENDIX B
MULTILINGUAL EXAMPLES

B.1 Case Study 4: Multilingual Requests

Fig. 13 provides an additional multilingual example of Chat2VIS capabilities, combining Spanish, Japanese and Mandarin to depict the counts of Chat2VIS mismatch performances against various nvBench databases. In Fig. 13 (a), it can be seen that the Spanish NL query to "Plot the number of mismatches per database. Only plot the top 10 databases." was correctly executed by all LLMs but with some different variations in interpreting the ambiguity. Both GPT-3 and GPT-4[9] selected the top 10 databases with respect to the largest number of mismatches, and then depicted the results. Meanwhile, GPT-3.5 first selected the top 10 largest databases and then depicted their respective mismatches from there. In the subsequent refinement of the query in Japanese to "Add the value on top of the bar line.", it can be seen that only GPT-3 correctly responded, while the subsequent request in Mandarin to adjust the x-axis labels as "Print the database names diagonal." was indeed followed by all LLMs. The examples illustrate again the high level of responsiveness of the LLMs to both correctly respond to multilingual requests as well as to plot refinements.

APPENDIX C
BENCHMARK EXAMPLES

Fig. 14 depicts examples of mismatches between Chat2VIS and nvBench together with errors within the benchmark dataset. These errors comprise Fig. 14(a) misspellings, Fig. 14(b) missing NL queries, Fig. 14(c) missing query intents, Fig. 14(d) discordant nvBench visualisations with respect to NL queries, Fig. 14(e) conflicting results, Fig. 14(f) unexpected data type conversions, Fig. 14(g) differences in

9. The Codex model has been discontinued and GPT-4 was used instead.

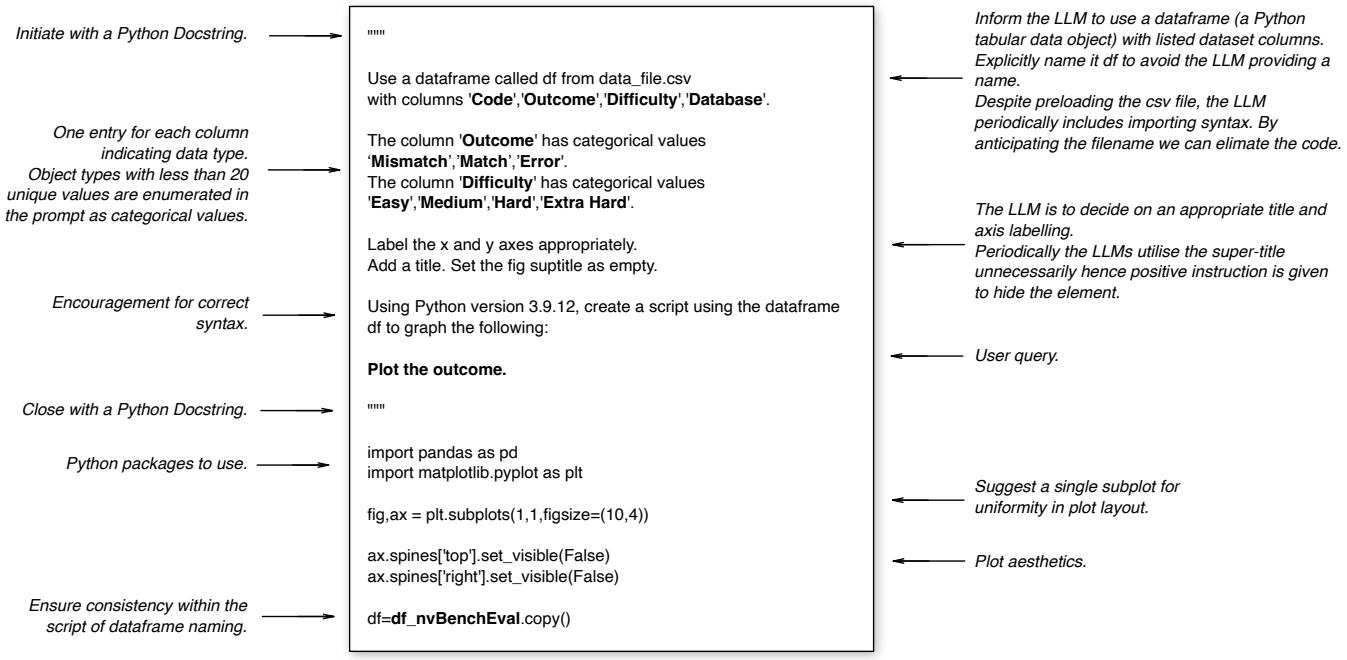

Fig. 12. Explanation of Chat2VIS Prompt

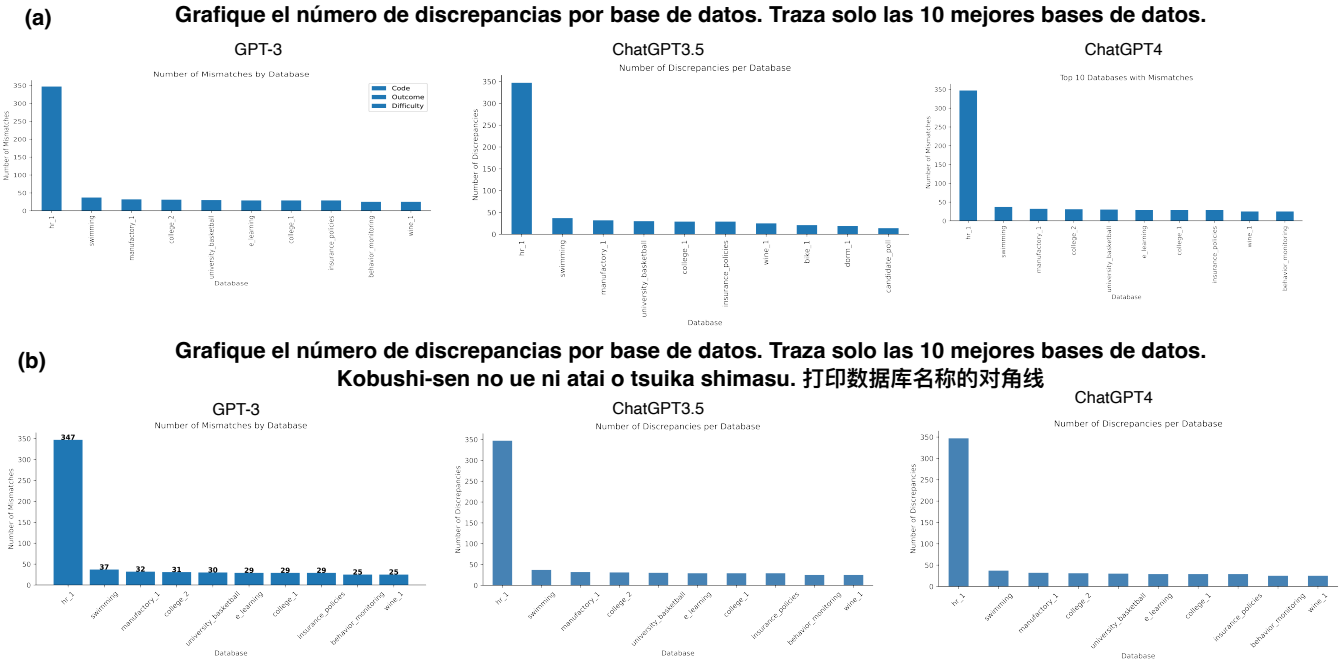

Fig. 13. Case Study 4: Conversational Visualisation Refinements using the nvBench Results Data via Multilingual Requests in Spanish, Japanese and Mandarin.

how zero values are depicted, Fig. 14(h) differences in sorting quantities with equal values, Fig. 14(i) differences in interpreting the sorting intent, Fig. 14(j) differences in handling string handling. These combinations of errors within nvBench, ambiguities and differences in the plotting behaviour of underlying frameworks generated mismatches under the chosen comparison methodology, which would likely have been treated differently under manual and subjective evaluations. Fig. 15 shows a selection of matching Chat2VIS examples with the nlvUtterance dataset.

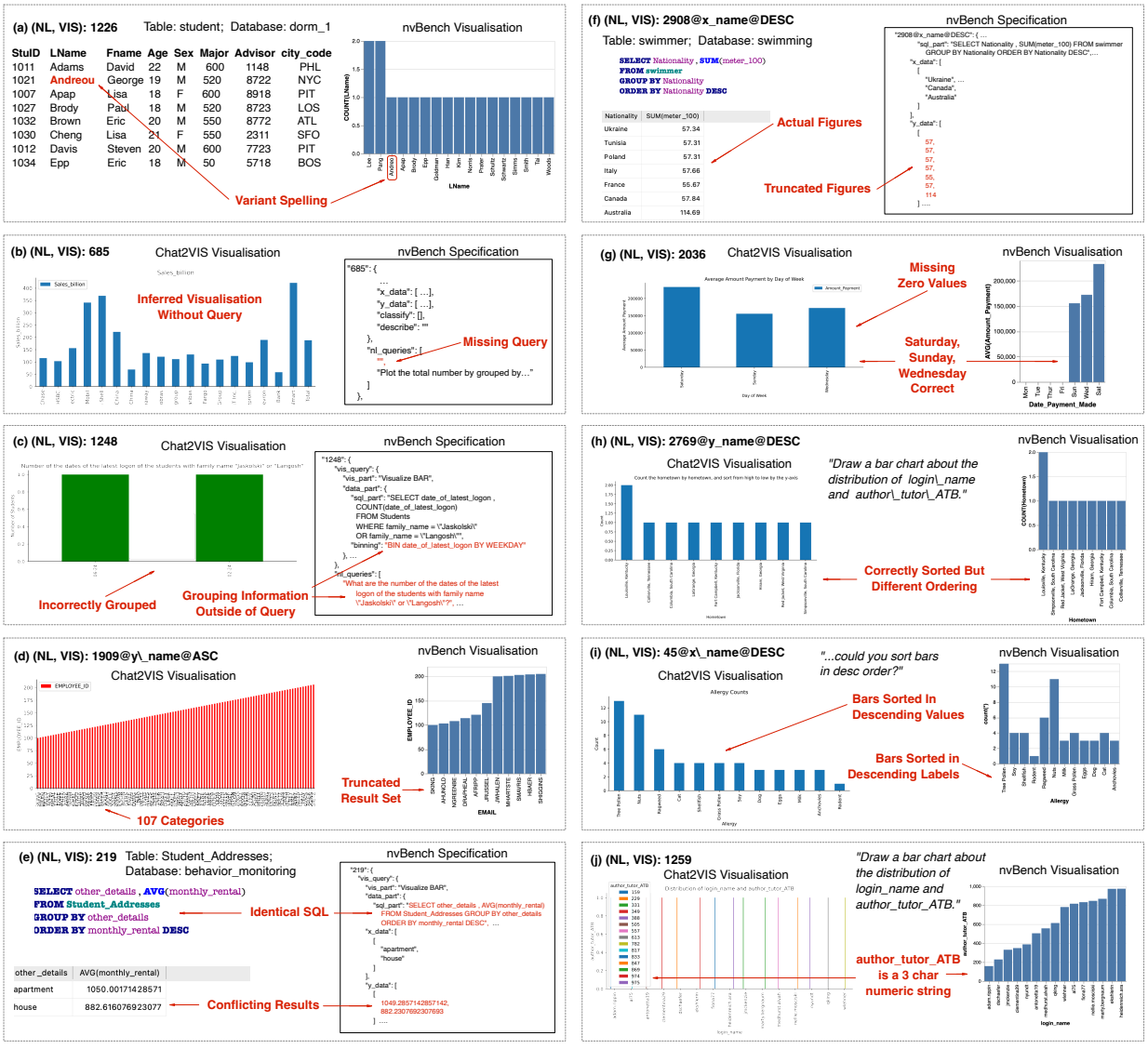

Fig. 14. Examples of Mismatched Visualisations Between Chat2VIS and nvBench

Fig. 15. nlvUtterance Movies Dataset Examples from Chat2VIS

Manuscript ID: TVCG-2023-05-0238
Title: Chat2VIS: Fine-Tuned Data Visualisations using Natural Language with Multilingual Capabilities via Large Language Models
Journal: IEEE Transactions on Visualization and Computer Graphics

Due date: 09/10/2023

Dear Prof. Han-Wei Shen

We thank you for giving us the opportunity to further improve the manuscript and for the constructive comments that the reviewers have provided. The manuscript has now been substantially reworked following the feedback. We have addressed each comment below and given a detailed description of our response. Further, all revisions have been highlighted in the manuscript.

We also appreciate your summary and collation of the reviewers' comments. Below, we provide a response to each of the key categories you have identified for us to address:

1. I**mprove the benchmarking experiments (all reviews); suggestions include adding more examples (especially more complex examples such as those with ambiguity or language pragmatics), adding explanation around the examples, and strengthening the rationale for experimental decisions**:
    a. We have provided additional experimental examples which include some more complex, specifically multilingual case studies which now include Mandarin, Japanese and Spanish (space permitting as more could have been done)
    b. We have revised and expanded our exposition of experimental results
    c. We have strengthened the discussion of our rationale for experimental decisions
2. **Address numerous missing details around the system architecture and implementation (R2, R3)**:
    a. More implementation details have been included and a new figure has been created with a clear example
    b. All our software code has now also been made available through GitHub so that readers can examine the implementation details themselves and replicate the study too
3. **Clearer statement of contribution and novelty (R2, R3)**:
    a. The entire contribution section has been rewritten. The four key contributions of the study have been explicitly stated. The introduction has also been written in order to highlight the gaps in the literature that our work is addressing and for which we provide a solution.
4. **Extend related work and explain how chat2vis improves on state of the art approaches (R1, R3)**:

a. We have expanded both the related work section and the introduction to highlight the fact that prior state-of-the-art NL2VIS were unable to handle conversational customisation of charts, they did not have multilingual capabilities and did not have the ability to integrate language understanding, reasoning in the midst of ambiguity and visualisation code translation in a single end-to-end system.
b. We have carefully re-examined the latest literature. There are no other significant papers which we have overlooked. We have included one older paper.
c. Since our work was not a systematic literature review, we pointed the readers to two most recent SLRs which also corroborate the gaps in the literature which we have identified.

5. **Add discussion around expressibility and prompt design (R1), ethical implications and risks (R2), and limitations, such as ability to handle dirty data (R3)**:
    a. The motivation behind our work was to solve the big problems identified in literature related to NL2VIS, instead of focusing on matters relating to the intricacies of large language models and prompt engineering. We have provided a prompting architecture that works and solves the key problem, and we believe it is for subsequent studies to investigate alternative prompting approaches that also work while being expressed differently. We have created a new 'Study limitations' section where we have discussed this.
    b. We have discussed ethical implications around risks of providing sensitive data to the LLMs and discussed the privacy-preserving capabilities of Chat2VIS. Other ethical issues around LLM and the challenges of handling dirty data are not within the scope of this study and have been addressed in study limitations.

We trust that these major revisions have significantly improved our manuscript and addressed the concerns raised by the reviewers. In revising the manuscript we have had to use three additional pages which we understand would incur a US$660 fee. We felt that this was the only way to incorporate all the revisions and we have acquired the funds to pay this fee if and when it comes to it.

We look forward to hearing from you soon and are prepared to address any further concerns that might be raised.

We thank you again for your kind consideration. Please let us know if you require anything else from us.

Yours sincerely

Teo Susnjak and Paula Maddigan

Dear Reviewers

We sincerely appreciate the time and effort you have invested in evaluating our work. Your insights have been invaluable in refining our research and manuscript. Your constructive feedback has certainly helped us improve the quality of the present paper. The following text details our responses to your feedback and suggestions. We have done our best to address each point in turn. Thank you for your contributions to the enhancement of our work.

Yours sincerely

Teo Susnjak and Paula Maddigan

REVIEWER #1 COMMENTS	**AUTHORS' RESPONSE**
The organization of the paper needs to be adjusted: For example, there is only one subsection under Section 1, and there is no 1.2 after 1.1, which is not common. Generally, subsections are only used when there are two or more subsections.	We accept your critique about the unconventional structure of the Introduction section. Our revised version addresses this by removing the numbering of Subsection 1.1 .We now just retain the heading 'Contribution' to make it easier for readers to identify the purpose and value of the paper a little more easily.
The title of the paper mentions fine-tuning, but the specific fine-tuning is only mentioned in Section 5 results. Perhaps it would be more appropriate to have a section on experimental settings before Section 5.	Using the term 'Fine-tuning' in the title maybe considered misleading given that it most frequently refers to adjusting the actual model parameters for LLMs. We are not doing this in the paper and we cannot since we cannot access the pretrained models from OpenAI. In order to mitigate the confusion, we have changed the title of the paper accordingly. We trust that this helps clarify the issue.
The title of the paper includes multilingual, but the multilingual capability is already inherent in the large models. What is the relevance of this to the paper? Please explain.	Thank you for raising the question regarding the relevance of multilingual capabilities of our system and we apologies for the weakness in explaining this in the paper. In response, we would like to highlight the findings of a recent comprehensive survey by Shen et al. (2022) on NL2VIS approaches (in IEEE Transactions on Visualization and Computer Graphics), where the lack of multilingual support in existing NL2VIS systems is identified as a gap in the literature. One of the key motivations of our paper is precisely to address this gap by focusing on enhancing the multilingual capacities of NL2VIS systems. While it is true that LLMs inherently possess multilingual capabilities, the existing

	literature highlights the need for further research on how to use them and the demonstration that they are effective. Indeed, Shen et al. (2022) call for exactly this – the exploration of GPT-like models for these purposes. By incorporating multilingual support in our research through LLMs, we respond to the most recent call to pursue this research path (Shen et al. (2022)) and show that it is viable at providing a robust solution to this otherwise outstanding problem. Our revised paper has highlighted this gap in literature and leaned more heavily of the findings by Shen et al. (2022) to justify our contribution and research motivation. Shen, L., Shen, E., Luo, Y., Yang, X., Hu, X., Zhang, X., Tai, Z. and Wang, J., 2022. Towards natural language interfaces for data visualization: A survey. IEEE transactions on visualization and computer graphics.
Section 5.4.1 mentions that chat2vis performs better in many scenarios, but these different categories are not distinguished in subsequent experiments. More detailed comparisons will make the experimental results more reliable.	We accept the critique. In the revised manuscript, we have rewritten this section in a way we hope is now clearer. Specifically, we have now included a new figure which demonstrates concretely some of the scenarios and categories under which Chat2VIS tends to produce more matches with the benchmarking dataset. We trust that this enhances the reliability of this subsection and provides readers with a nuanced understanding of the system's capabilities and the overall shortcomings as well as the challenges of performing automated benchmarking for the NL2VIS domain.
The related work of the paper has some shortcomings. Please carefully check and authors can refer to relevant reviews to check for missing citations: arxiv.org/pdf/2109.03506.pdf	We appreciate the feedback provided regarding the related work section of our paper. In response, we have made several improvements to address the comments. Firstly, we carefully reviewed the suggested paper arxiv.org/pdf/2109.03506.pdf and identified relevant content that aligns with our study. As a result, we have included an additional citation from this paper in our related work section to acknowledge its contribution to the NL2VIS domain.

	Secondly, we acknowledge that the suggested paper is a comprehensive survey covering a wide range of related works. Due to space limitations in our manuscript, we were unable to provide an extensive literature review similar to the survey paper. However, we have cited the survey paper and emphasized its significance, guiding readers to explore it for a more comprehensive understanding of the related works in the field. Thirdly, our primary focus in the paper has been on citing the most recent advancements in NL2VIS research. We aimed to highlight state-of-the-art techniques and cutting-edge studies to provide readers with up-to-date information. However, if there are specific works or papers that we might have inadvertently omitted, we would greatly appreciate the reviewer's guidance in pointing them out. We will gladly consider including any additional relevant citations in the revised version.
In Table 1, the content of chat2vis is bolded, but RGVisNet actually performs better than chat2vis. According to usual practice, the accuracy of the best work should be bolded.	We appreciate your attention to detail and apologise for the unintended confusion. In the revised manuscript, we have highlighted the performance of RGVisNet, ensuring that the formatting accurately reflects the results.
In addition to directly generating visualization code, perhaps some structured expressions can have better performance, and the large models should have corresponding abilities. This may require setting up good visualization space.	We appreciate your insightful suggestion regarding the potential benefits of using structured expressions as an intermediate representation in the NL2VIS pipeline. The idea of converting the natural language query into a structured format, which might include a symbolic representation of the visualisation query before the final code generation indeed presents an interesting approach. This method could potentially yield a more consistent and accurate output, as it abstracts the desired visualisation into a structured format that could be more robust to variations in natural language phrasing. However, the current scope of our study is focused on an end-to-end solution, which translates natural language queries directly into visualisation code without an intermediate step. We believe your proposal constitutes an excellent avenue for future research and we may explore this in a subsequent study. Comparing the performance of our current end-to-end model with an approach that incorporates your proposed intermediate step would provide valuable insights into the trade-offs and benefits of each method. Thank you for your valuable input which

	we have now mentioned in the revised manuscript as an avenue for subsequent research to explore.
Some of the images have insufficient information. For example, the font of fig.1 and fig.8 is larger (compared to fig.6). Many images have a lot of white space, and the content should be adjusted to make the overall content more coordinated.	We thank you for your attention to detail. In the current manuscript, we have revised all images, addressing the issues to the best of our abilities and have particularly managed to reduce the amount of white space. We would like to note that in the images that represent the experimental outputs from Chat2VIS, we have retained fonts and dimensions of slightly different sizes because they were defined and generated by the LLMs and faithfully represent their abilities to produce visualisations so we did not think it was suitable for us to interfere with them. We trust that overall the images in the revised manuscript are now better.
The expressive ability of the large model is strongly dependent on the content of the prompt. Do the authors have a comparison and discussion of the content of the prompt?	Your observation regarding the influence of prompt content on the performance of large models is quite correct. In the context of this paper, we have constructed a generic prompt architecture, which is in itself a significant contribution, given its robustness across all the tested models without necessitating further customisation. This prompt architecture, which is illustrated in a new figure included in the appendix of the revised manuscript, has been an integral part of our methodology, facilitating consistent and effective utilisation of this family of LLMs. However, it is important to note that while we recognise the potential impact of varying prompting strategies, an extensive exploration of prompt engineering is not within the scope of this study. We have added a new 'Study Limitations' subsection to the manuscript, where we elaborate on this point and outline this as a promising directions for future research.

REVIEWER #2 COMMENTS	**AUTHORS' RESPONSE**
W1: Novelty limited. This work combines different existing techniques to build the visualization recommendation system without any novel	We understand your concern about the novelty of Chat2VIS. While it is true that we integrated existing models (proprietary LLMs primarily), our contribution lies in (1)

ideas. The novelty of Chat2VIS is unclear and the overall technical contributions are limited. More justification and discussion are desired.	how we leverage these models, (2) the gaps in the literature we address, (3) comprehensive benchmarking experiments we conduct. As we noted in the previous comments, Shen et al. (2022) only recently produced a comprehensive survey of NL2VIS approaches published by this journal, and the authors highlighted clear gaps in the literature and proposed a forward-looking research agenda for the NL2VIS community. Their work specifically: 1. identified the lack of multilingual support in existing NL2VIS systems whose capabilities normally only encompass English. 2. They call for the use of advanced NLP models which includes LLMs, and the exploration of end-to-end approaches 3. Development of NL2VIS with conversational capabilities, since most systems lack the ability to integrate a user's interaction history into the reasoning. Each of the above key gaps and research opportunities are tackled by our work. These gaps justify our work and define the novelty and contribution of the study. We admit that we have not made this crystal clear and have taken steps in the introduction to highlight this more effectively in our revised manuscript.
W2: This study did not provide sufficient details about the Chat2VIS implementation and architecture. For example: In Section 3, the authors did not explain why they used a JSON specification as the intermediate representation for visualization generation, how they engineered the prompts for different LLMs, how they selected and configured different LLMs for various languages or tasks, and how they executed and rendered the generated scripts using Vega-Lite or Plotly libraries. Regarding Figure 2, ChatGPT API message structure, no concrete example was given to illustrate its meaning and function. The authors could have also provided a specific example of generating code to demonstrate how JSON specifications translated into executable code. These additions would have facilitated a better understanding of the workflow and output of Chat2VIS.	Thank you for your constructive feedback. We apologise for any confusion caused by the lack of clarity in our paper regarding the implementation and architecture of Chat2VIS. We would like to clarify that our system does not use JSON as the intermediate representation for visualisation generation which is why it is not discussed in Section 3. The JSON specification actually pertains to the nvBench dataset (discussed in Section 4.2.2), and we acknowledge that this distinction was not adequately emphasised in the paper, as we have tended to discuss what methods our research uses. We have tried to rectify this oversight to avoid further confusion in the revised manuscript. Regarding the prompts for different LLMs, we used a single prompt that was common across all LLMs – and this is one of the contributions since the prompt is generalisable. However, to provide a more concrete example and aid in better understanding, we have included an additional annotated figure in Appendix A that demonstrates the generation of a specific

	prompt for a given case and we have revised the explanation of the prompt structure as well. Additionally, we would like to clarify that our system does not utilize Vega-Lite or Plotly libraries. Instead, we rely on plain Python for executing and rendering the generated scripts and we hope that the new explanation and figure in the Appendix A clarifies this better.
W3: The paper did not discuss the ethical implications or potential risks of using LLMs for NL2VIS, such as their reliability, robustness, bias, or misuse.	We appreciate your comment highlighting the necessity to consider the ethical implications and potential risks associated with deploying LLMs in the context of NL2VIS, including aspects of reliability, robustness, bias, and misuse. We acknowledge the critical importance of these issues in the broader AI landscape. However, the focus of our current paper is primarily on the development and evaluation of an end-to-end NL2VIS system using LLMs. While we are acutely aware of the ethical dimensions, an in-depth exploration of these aspects falls outside the scope of this particular study. It is worth noting that these concerns are actively being investigated by dedicated research teams and projects globally. To ensure transparency about the study's limitations and our awareness of these important ethical considerations, we have added a new 'Study Limitations' subsection in the manuscript where we clarify this context. This addition serves to highlight the need for responsible and ethical AI use, despite not being the central focus of our research. We have however addressed a different aspect to the ethical use and risks of LLMs with respect to data privacy and have explained the strong privacy-preserving characteristics of Chat2VIS in section 3.2.
W4: In Section 5, the authors presented some case studies to demonstrate Chat2VIS's interaction and multilingual capabilities. However, the description and presentation of these case studies were not clear and consistent. For example, in Section 5.2, only a simple example was provided, without an analysis of the differences in the results generated by different LLMs. In Section 5.3, only two examples of different languages were given, without an explanation of language selection or how differences and transitions were handled.	Your comments on the case studies are appreciated. In response, we have expanded our case studies to include more languages and examples (see Appendix), providing a more thorough analysis and justification for our language selections (see Methodology). We have also included specific examples in Section 5.4, both for matches and mismatches, and enhanced the evaluation text with more details. We trust that these revisions address the concerns raised.

Additionally, in Section 5.4.1, cases where Chat2VIS performed well on the nvBench benchmark were listed without specific examples or explanations. The evaluation text was sparse and lacked sufficient details.	In Section 5.2, the chart refinement involves three parts which we regard as being realistic and typical of what an end-user might request. These are 1. Convert the bar chart to a pie chart 2. Then, a command to hide the axis label 3. Followed by a command to use pastel colours We have expanded the text in the above section to provide more analysis.
W5: The authors did not fine-tune the paper to ensure the best presentation quality. The figures had low resolution, and some of them featured very small font sizes that were difficult to discern. Additionally, some figures contained overlapping text labels.	Thank you for your feedback on the quality of the figures in our manuscript. In response to your comments, we have revised all figures to improve their resolution and readability as best as we can. We have endeavoured to reduce white space and increase the resolution where possible to ensure the best presentation quality. However, we would like to clarify the issue regarding the figures that represent the experimental outputs from Chat2VIS. These figures indeed feature small fonts, messy and overlapping labels, and dimensions of slightly different sizes, but it's important to note that these were directly generated by the LLMs in our experiments and represent experimental outputs with their imperfections. The variation in font size and other elements in these figures accurately reflect the strengths and weaknesses of the LLMs in producing visualisations. We therefore believe it is important to present these outputs as they were generated by the LLMs (despite their unappealing nature at times), without additional modifications, to provide an accurate representation of their capabilities. We appreciate your understanding on this matter and hope that this addresses your concerns.

REVIEWER #3 COMMENTS	AUTHORS' RESPONSE
Title is a bit misleading: When I initially read the title, I assumed that the paper is about fine-tuning language models for natural language interfaces, but after reading the paper more carefully, it became apparent that the paper first introduced Chat2VIS and then evaluated the system against two benchmark datasets. A more appropriate title should perhaps be Chat2VIS: A natural language interface for supporting conversational chart generation.	We appreciate your comment on the title. We agree that the original title may have been misleading in light of the meaning that 'fine-tuning' has with respect to adjusting parameter weights of LLMs. In order to dispel the confusion, we have dispensed with this term and have incorporated your advice to produce a more precise title, we hope.
The introduction of the paper can be tighter and talk specifically about the problems the system is attempting to solve. The initial portion of the introduction talks about the benefits of data insights and the need for NLIs. Given the precedence of NLIs for several years now, an elaborate justification is not needed. The contribution portion needs to be clearer about the takeaways from the benchmark studies, as well as the specific challenges for developing structured methodologies for developing baseline standards in NL2VIS.	Your feedback on the introduction is valuable. We have refined and tightened the introduction and we trust that we have addressed the critique sufficiently.
The related work section does not clearly describe how Chat2VIS simplifies the NL2VIS pipeline and makes claims on flexibility and robustness without specific details as to how when compared to previous systems.	We appreciate your observation about the related work section. In response, we have provided a deeper discussion of end-to-end solutions (like Chat2VIS) in NL2VIS and how these differ from many existing NL2VIS systems in the literature in terms of their holistic approach and comprehensive coverage of the entire pipeline. While previous NL2VIS systems often focus on specific components or aspects of the process, such as natural language understanding or visualisation generation, end-to-end solutions aim to provide a seamless and integrated workflow from start to finish. Unlike fragmented systems that require manual intervention or multiple separate steps, end-to-end solutions automate the entire process of translating natural language queries or instructions into visual outputs. They incorporate components for natural language understanding, information

	extraction, visualization generation, and user interaction, creating a cohesive framework that handles the entire pipeline within a single system – in our case LLMs. We trust that the expanded discussion on this draws a clearer distinction now between Chat2VIS and previous systems.
The architecture described in Section 3 has details that are not clearly explained. For instance, how does setting the temperature parameter to zero specifically encourage LLMs to be more consistent with code generation? Figure 1 is too big and too illustrative to understand the specific inputs and outputs that go into the Chat2VIS pipeline. What is Streamlit? Also, the prompt shown in Figure 2 is too generic to help understand the relevant conversational interactions that are passed as input to ChatGPT to generate appropriate visualizations. Having specific examples would be helpful.	Your thoughtful critique and the identification of opportunities for clarification is appreciated. In response to your query on the temperature parameter, in LLMs, this parameter controls the randomness of the model's predictions. Specifically, a temperature setting of zero encourages deterministic behaviour, promoting more consistent code generation. This detail has now been elaborated on in the revised manuscript to ensure reader understanding. In addressing your comments on Figure 1, we revised the narrative around the figure in order to supplement it with an extended description in the manuscript. This adjustment aimed to enhance its comprehensibility and better illustrate the Chat2VIS pipeline. We hope this helps. For your question regarding Streamlit, it is an open-source app framework that enables developers to build interactive and user-friendly web applications for machine learning and data science. We have included this explanation in the revised manuscript for further clarity and provided a reference. We apologise for the oversight. Regarding the prompt shown in Figure 2, we appreciate your observation about its generic nature and have removed the figure. To provide a clearer perspective, we included a new, more detailed figure in the appendix that offers a concrete representation of the prompt structure. This addition, we believe, enriches the understanding of the types of conversational interactions that are input to LLMs for visualisation generation.
Looking at the example illustrated in Figure 4, I am not convinced why an LLM is needed to interpret the utterance. "Plot the outcome" is a very concrete, imperative command that can be solved by previous	Thank you for your critique of Figure 4. We acknowledge that the command "Plot the outcome" may appear to be straightforward at first glance. However, the choice of this specific command in our study was intentional. While it may seem simple, it actually presents certain challenges due to its inherent ambiguity and lack of precision in terms of specific instructions on how to plot the data.

symbolic state-of-the-art NLI systems. Also, there are no examples of language pragmatics in follow-up utterances that are typical of a conversational interface.	The utterance does not explicitly state that a ‘group by’ operation over the ‘outcome’ column needs to be first executed and that the aggregation should be counts rather than something else. Neither is it explicitly stated that a bar graph is the most suitable chart type for the result. While this is all obvious to us, it in fact requires a significant amount of reasoning and inference. The purpose was to showcase that LLMs combined with our prompting architecture is able to exhibit the system's end-to-end capabilities in handling such ambiguity and correctly reasoning about the most suitable chart type to generate. As for a lack of examples of language pragmatics in Figure 4, we would like to refer to Figure 7 which carries on this example and demonstrates how context and more precisely expressed intentions influence the final visualisations from the starting point of "Plot the outcome" on this dataset. We maintain that the example in Figure 7 which continues the conversational refinement of the visualisation from Figure 4 is a reasonable example of conversational interactions with these systems. In our revised manuscript, we have included additional examples to further illustrate this advantage of utilizing LLMs in the Chat2VIS approach and have improved the text to express the above points more clearly. We appreciate your feedback, as it has allowed us to emphasise the significance of LLMs in our revised manuscript and highlight the system's ability to handle ambiguity and reason about the most appropriate chart type for a given prompt.
The rationale for decisions made in benchmarking is not clearly explained or motivated. It is unclear why accurate comparisons cannot be made across other chart types and the choice to pick only bar charts. That decision limits the type of analytical intents that can be used for benchmarking as it would not account for correlation or temporal analysis. Also, why is only the first option picked among the various nvBench variants? The examples used in Case Study 1 (Figure 7) are very specific and concrete, even prescriptive about the type of chart that needs to be generated. What about higher-order intents that are ambiguous such as "which model is better?" without specifying	Thank you for your constructive critique. Firstly, on the matter of choosing to focus on bar charts, it is important to note that the decision was guided by practical and technical considerations. The nvBench benchmark had to be reduced initially since many NL queries did not use a flat table like Chat2VIS but instead relied on database joins across multiple tables and sub-queries. The choice to concentrate on bar charts was also not arbitrary, but a calculated decision driven by the challenge of automating the testing of various chart types, each with its unique complexities. Bar charts represented an overwhelming majority of chart types in the dataset and was therefore a good starting point and represents a coverage of 3,003 instances out of some 7 thousand. Writing software for automating the comparisons for other chart types like line, pie and scatter would involve would necessitate devising a different and custom comparison mechanism, an enormous amount of testing and software and resources which warrant a new and separate study in order to fully solve.

how it should be interpreted and the chart that needs to be generated? The example of ambiguity described in Section 5.4.4 - "high to low" is a very limited example of ambiguity differences between nvBench and ChatVIS. There should be additional examples on a spectrum of concreteness to fuzziness that should be considered as part of the benchmarking process. Also, how dirty can the underlying data be? What if there are no synonyms for the attributes (e.g., domain-specific pharmaceutical acronyms)? More information needs to be included to describe these use cases. The specific language choices described under Case Study 3 feel rather arbitrary. Why Croatian vs. more prevalent languages such as Spanish? How about non-Latin languages such as Mandarin or Japanese, for instance?

While we acknowledge that our approach considering a subset of samples might have limited the variety of analytical intents that could be benchmarked, but it is important to stress other studies have also followed similar constrained benchmarking while recognising that bar charts do make up a significant portion of the dataset and provide a robust initial benchmarking study. Thus, focusing on them still allowed us to conduct a meaningful, large-scale evaluation of Chat2VIS. Furthermore, this selection does not imply that Chat2VIS is incapable of generating other chart types—it merely reflects the practical constraints of our evaluation approach.

Regarding the critique about picking only the first NL query option out of several possible and equivalent NL queries which have an equivalent visualisation intent on the nvBench dataset, this decision was again motivated by several factors. Chiefly, the Codex API we used for nvBench was throttled by OpenAI and only a handful of requests could be executed per minute, with requests frequently being denied. The possibility existed of API requests being blocked completely and undermining our research due to too many requests being initiated and thus, we were forced to be frugal with this service. However, we operated under the assumption that the first NL query was just as expressive of the visualisation intent as any of the other queries, and therefore we believe that it was nonetheless reasonable to work with the first NL query anyway.

Thus, we want to emphasise that we are aware of the limitations of our current evaluation method and the choices we had to make. Our study should be seen as a step towards a more comprehensive evaluation framework, highlighting the challenges and paving the way for future research in this direction. To that end, we have revised the manuscript to clarify some of the points expressed here with some more clarity in the methodology and the new limitations subsection.

With respect to ambiguous queries, we briefly demonstrated this with the "Plot the outcome." Query. Despite its lack of detail regarding chart type or the need for specific operations (like a group-by), Chat2VIS leveraged its language understanding capabilities to generate meaningful visualisations. However, ambiguity tends to lead to mismatches when evaluated against benchmark datasets. These mismatches do not signify incorrect visualisations, but rather reflect divergent interpretations of the ambiguous NL queries.

	We also appreciate the reviewer's feedback regarding the specific language choices in Case Study 3. Due to space limitations, it was not feasible to cover all alternative languages extensively in the main paper. However, we want to clarify that our language selection was based on several factors. Firstly, we chose languages that we were familiar with and could ensure the correctness and meaningfulness of the generated requests. Secondly, we aimed to strike a balance between high-resource languages like German and French, and low-resource languages like Croatian and especially te reo Māori, in order to showcase the capabilities of the LLMs across languages with which it has differing levels of training data exposure. In response to the reviewer's comment, we have included additional examples in the Appendix that demonstrate multilingual plots using Spanish, as well as non-Latin languages such as Mandarin and Japanese. We have also clarified in the Methodology some of our justifications more clearly. In our study, we focused on evaluating the performance of Chat2VIS against multiple benchmark datasets (the first study to conduct this level of benchmarking), aiming to establish a comprehensive assessment of the system's capabilities within the context of NL2VIS. While we acknowledge the importance of considering additional examples on a spectrum of concreteness to fuzziness, we believe that conducting a more nuanced examination of the reliability of the Chat2VIS artifact warrants a separate study. Exploring a wider range of examples and scenarios would require dedicated research efforts beyond the scope of this current work. Regarding the use of dirty data and the absence of synonyms for attributes, we recognize the significance of these factors in real-world NL2VIS applications. However, investigating the visualization of dirty data and addressing the challenges posed by domain-specific acronyms or similar issues would also be a distinct research topic meriting its own dedicated study.
Finally, there are several typos in the paper: Section 1: "These LLMs were primarily trained *on* English corpora" Section 3.2: Extra period after "Plot the outcome" Section 4.2.2: page 5 "error" should not be capitalized Section 5.4.1: "below" alludes to content that is actually on the next page. Replace with "as	We appreciate your meticulous attention to detail. We believe we have now corrected the typographical errors you pointed out, with one exception. Where the writing of “Error” in Section 4.2.2 depicts a categorical value used in the benchmarking program and directly reflects how the example is marked in the generated output, we have retained its capitalisation and changed the font to a typewritter font to indicate this representation. In places in Section 4.2.2 where we have

follows" The references need to be reviewed closely. Several words in the references are not capitalized correctly. E.g., "stanford," "corenlp," conference names, "gpt," "monte carlo"	discussed error when talking about incorrect code, for example “Python code error”, we have not capitialised the word “error”. We hope this addresses the critique. We also worked through the references and while we have addressed the capitalisation issues, we have found that the prescribed \bibliographystyle{IEEEtran} overrides this and forces lower cases in those instances and thus we have not been able to rectify this.