Sevi: Speech-to-Visualization through Neural Machine Translation

Jiawei Tang[†] American School of Doha, Qatar
Yuyu Luo[*] Tsinghua University Beijing, China
Mourad Ouzzani QCRI, HBKU Doha, Qatar
Guoliang Li Tsinghua University Beijing, China
Hongyang Chen Zhejiang Lab Hangzhou, China
23jtang@asd.edu.qa,{luoyy18@mails.,liguoliang@}tsinghua.edu.cn,mouzzani@hbku.edu.qa,hongyang@zhejianglab.com

ABSTRACT

Data visualization is a powerful tool for understating information through visual cues. However, allowing novices to create visualization artifacts for what they want to see is not easy, just as not everyone can write SQL queries. Arguably, the most natural way to specify *what to visualize* is through natural language or speech, similar to our daily search on Google or Apple Siri, leaving to the system the task of reasoning about *what to visualize and how*.

In this demo, we present **Sevi** an end-to-end data visualization system that acts as a virtual assistant to allow novices to create visualizations through either natural language or speech. **Sevi** is powered by two main components: `Speech2Text` which is based on Google Cloud Speech-to-Text Rest API, and `Text2VIS`, which uses an end-to-end neural machine translation model called **ncNet** trained using a cross-domain benchmark called **nvBench**. Both **ncNet** and **nvBench** have been developed by us. We will walk the audience through two general domain datasets, one related to COVID-19 and the other on NBA player statistics, to highlight how **Sevi** enables novices to easily create data visualizations. Because **nvBench** contains `Text2VIS` training samples from 105 domains (*e.g.,* sport, college, hospital, etc.), the audience can play with speech or text input with any of these domains.

CCS CONCEPTS

• **Information systems** → **Data analytics**; • **Human-centered computing** → **Visualization**; **Visualization systems and tools**.

KEYWORDS

Speech-to-Visualization; Natural Language-to-Visualization

ACM Reference Format:
Jiawei Tang[†], Yuyu Luo[*], Mourad Ouzzani, Guoliang Li, and Hongyang Chen. 2022.**Sevi**: Speech-to-Visualization through Neural Machine Translation. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22), June 12–17, 2022, Philadelphia, PA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3514221.3520150

[†] Work done while interning at QCRI, Qatar.
[*] Yuyu Luo is the corresponding author.

SIGMOD '22, June 12–17, 2022, Philadelphia, PA, USA.

ACM ISBN 978-1-4503-9249-5/22/06…$15.00
https://doi.org/10.1145/3514221.3520150

Figure 1: A user provides input in the form of voice (speech) or text (natural language). Sevi translates either input into a visualization.

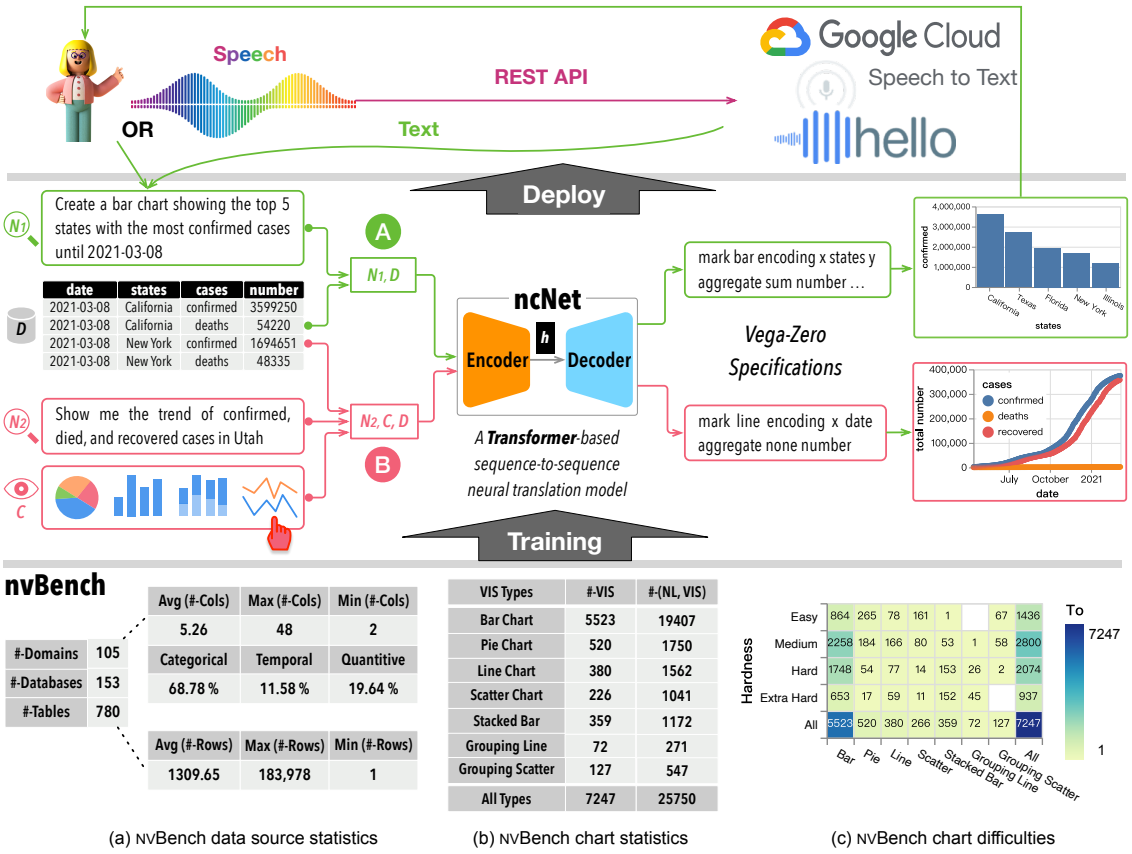
1 INTRODUCTION

Data is taking the world by storm, transforming virtually every industry, and is playing an important role in our daily lives. It is important to understand the insights that numbers alone cannot tell us. However, it is nontrivial to interpret the massive amounts of information being collected today.

Data visualization plays a key role in communicating information, through the use of visual elements such as bar charts, scatter plots, and histograms [16]. This makes the data more natural for the human mind to comprehend and therefore provides an accessible way for anyone, even those without statistical background, to identify trends, patterns, and outliers within large datasets [8, 9, 21]. In fact, we have been inundated with visual interpretations of the COVID-19 data, from early graphics urging us to flatten the pandemic curve to regularly updated dashboards [5, 7, 12].

Although there are many choices of interactive data visualization tools (*e.g.,* Tableau and Qlik) and easy-to-specify data visualization languages (*e.g.,* Vega-Lite [17] and ggplot2), only experts are able to to create good visualizations. In addition, this assumes that these experts know many details such as the meaning and the distribution of the data, the right combination of attributes, and the right type of charts.

The democratization of data visualization means that anyone can easily create data visualizations without the need to write code and with a very fast learning curve, similar to how Google democratized *search* using a natural language interface. In fact, both commercial vendors (*e.g.,* Tableau's Ask Data [18], Power BI [2], ThoughtSpot [3], and Amazon's QuickSight [1]) and academic researchers [4, 10, 15, 20] have investigated the translation from natural language queries to visualizations (`Text2VIS`). They mainly use statistical phrase-based translation that first employs natural language processing toolkits (*e.g.,* Stanford CoreNLP [14] and NER [6]) to parse a natural language query and produce a variety of linguistic annotations (*e.g.,* parts of speech, named entities, etc.), based on which they then devise algorithms to generate target visualizations.



nvBench

#-Domains	105
#-Databases	153
#-Tables	780

Avg (#-Cols)	Max (#-Cols)	Min (#-Cols)
5.26	48	2
Categorical	**Temporal**	**Quantitive**
68.78 %	11.58 %	19.64 %

Avg (#-Rows)	Max (#-Rows)	Min (#-Rows)
1309.65	183,978	1

(a) NVBench data source statistics

VIS Types	#-VIS	#-(NL, VIS)
Bar Chart	5523	19407
Pie Chart	520	1750
Line Chart	380	1562
Scatter Chart	226	1041
Stacked Bar	359	1172
Grouping Line	72	271
Grouping Scatter	127	547
All Types	7247	25750

(b) NVBench chart statistics

(c) NVBench chart difficulties

Figure 2: An overview of Sevi.

Although deep learning-based natural language understanding and processing have achieved near-human intelligence and outperformed traditional statistical phrase-based machine translation techniques, using neural machine translation for `Text2VIS` was not well studied. There are two main challenges: **(C1)** *No benchmark.* Neural machine translation (for example, from English to French) needs a lot of training data that is usually easy to come by, but coming with a large sets of (`TEXT`, `VIS`) pairs is not straighforward. **(C2)** *No end-to-end* `Text2VIS` *neural machine translation systems.* This is partially because of the absence of a `Text2VIS` benchmark.

Recently, we tackled challenge **(C1)** by proposing the first `Text2VIS` benchmark, called **nvBench** [11], which consists of 25k+ (`TEXT`, `VIS`) pairs over 780 tables from 105 domains. For handling **(C2)**, and based on **nvBench**, we proposed the first neural machine translation system for `Text2VIS`, called **ncNet** [13], using Transformer [19] with special optimization techniques such as visualization-aware attention forcing and translation. In this demo, we take this a step further, streamlining the experience by directly allowing the user to speak into their computer rather than typing in text, resulting in a new system, called **Sevi**, that supports end-to-end `Speech2VIS`.

Demonstration Scenarios. We propose to demonstrate **Sevi** with two easy-to-specify scenarios, as shown in Figure 1: the users can either use speech or text input to create their desired visualizations. For the data sets, we use two general domain datasets, one for the COVID-19 pandemic data and the other for NBA player statistics. Better still, the audience can choose any dataset from the 105 domains (*e.g., Sport, Hospital*) that are provided in **nvBench**. **The code is available at https://github.com/Thanksyy/Sevi.**

2 SYSTEM OVERVIEW

In this section, we overview **Sevi**, as shown in Figure 2. We first introduce how **Sevi** works (Section 2.1). We then describe **ncNet** for `Text2VIS` neural machine translation (Section 2.2). We finally close this section by introducing **nvBench**, the `Text2VIS` benchmark used for training **ncNet** (Section 2.3).

2.1 Sevi: End-to-end Speech2VIS

Sevi works as follows, as shown at the top of Figure 2. The user specifies *what is desired* using a microphone to generate a speech query. This speech query will be sent to Google Cloud speech to text REST API (https://cloud.google.com/speech-to-text/). The transcribed text will be returned. Alternatively, the user can directly specify a text (*i.e.,* natural language) query. Afterwards, **Sevi** will send this text query to **ncNet**, which is responsible for interpreting this text query and creates a visualization for the user.

2.2 ncNet: Text2VIS Machine Translation

The architecture of **ncNet** is shown in the middle of Figure 2. **ncNet** adopts a Transformer-based [19] model that consists of an encoder and a decoder, which are both stacks of self-attention blocks.

How ncNet works. ncNet takes a text query N and a data set D as input, tokenizes and concatenates them as a sequence, and feeds them as the input of the encoder of **ncNet**. The encoder will convert this sequence of (N, D) into a hidden vector h as the input of the decoder. The decoder will then output a sequence in the form of Vega-Zero [13] (*e.g.,* "mark bar encoding x states ..."), as the visualization specification. Note that, Vega-Zero is the language we proposed in [13] as a simplification of Vega-Lite (https://vega.github.io/vega-lite/) and is more friendly to sequence-to-sequence models. The Vega-Zero specification will be converted to Vega-Lite by default (or other visualization languages such as ggplots) so as to be rendered and returned to the user.

Chart Templates. ncNet further proposes to use chart templates as additional hints, where a user can specify the output to be a pie chart or a scatter plot with a simple click, *e.g.,* the user can select a chart template C as shown in Figure 2. In this case, the encoder will take (N, D, C) as input. In practice, chart templates have been widely used in all commercial products, including Tableau, Excel, and Google Sheets. **ncNet** supports seven chart templates, {Bar, Stacked Bar, Pie, Line, Grouping Line, Scatter, Grouping Scatter}. Figure 2 showcases four sample chart templates. Essentially, the chart template is used as a **constraint** to reduce the search space of possible outputs.

ncNet is trained using **nvBench**, which is discussed below.

2.3 nvBench: The Text2VIS Benchmark

Dataset statistics. Figure 2(a) gives statistics of **nvBench** from the data sources and (TEXT, VIS) perspectives, where the TEXT is a natural language specification and the VIS is the corresponding visualization specification in the form of Vega-Zero. As shown in Figure 2 (a), **nvBench** has 153 databases containing a total of 780 tables that cover 105 domains (*e.g.,* sports, customers). The average number of columns/rows in the 780 tables is 5.26/1,309.65, and the maximum/minimum number of columns (rows) is 48/2 (183,978/1). Among the columns, 68.78% of columns are categorical columns, 11.58% of columns are temporal columns, and 19.64% of columns are quantitative columns. Figure 2 (a) also depicts the distributions of columns and rows, which tells us that most of the tables have 2 to 9 columns.

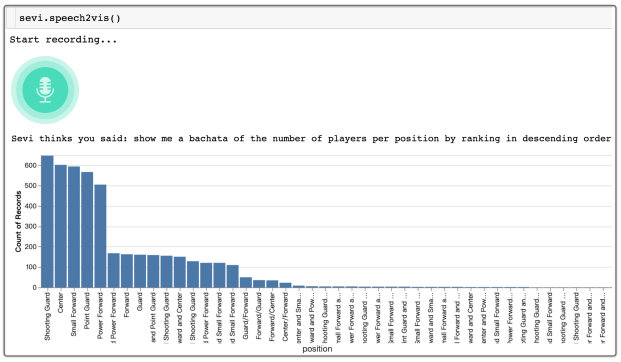
(TEXT, VIS) statistics. Given the 153 databases (Figure 2 (b)), **nvBench** contains 7,274 visualizations on 7 types of charts. For each VIS, **nvBench** provides one to several TEXT queries since different users might provide different TEXT queries for the same visualization. In total, **nvBench** consists of 25,750 (TEXT, VIS) pairs.

Visualization difficulties. nvBench further defines four-level of complexities, *i.e., easy, medium, hard, and extra hard*, for the visualizations based on the hardness of the query. For example, a visualization query with *filter*, *bin* and *aggregations* may be categorized as *Hard*. The heatmap in Figure 2(c) shows the distribution of visualizations in different chart types and hardness of visualizations.

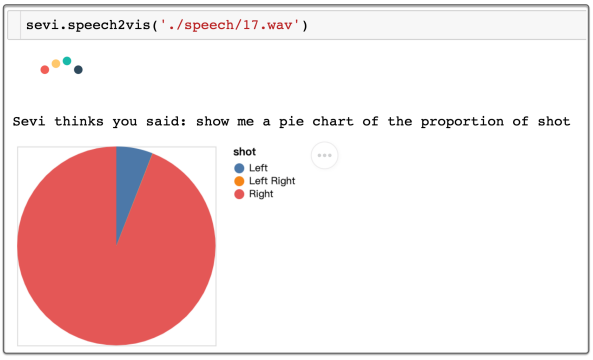
3 DEMONSTRATION SCENARIOS

We will walk through the audience through two datasets, one for NBA player statistics and the other for COVID-19 pandemic dataset , using the Jupyter Notebook and a Web-based interface.

(a) Import the package and specify a dataset



(b) Perform Speech2VIS by recording in the notebook



(c) Perform Speech2VIS by loading from an audio file

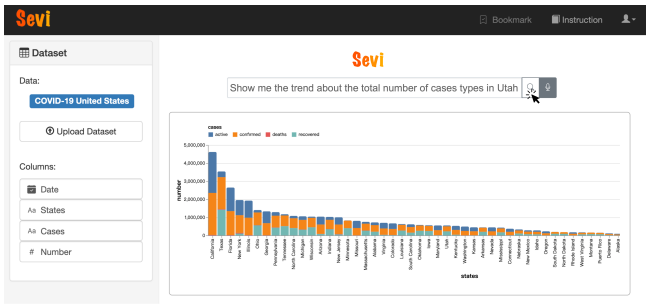
Figure 3: Sevi in Jupyter Notebook.

Sevi in Jupyter Notebook. Data science practitioners often perform interactive data visualization using a Jupyter Notebook (or Jupyter Lab). To make **Sevi** easy-to-use for this type of users, we have developed a Python package for **Sevi** to be used in the Jupyter Notebook ecosystem.

Figure 3 is a screenshot that shows how to create desired visualizations with speech queries for the NBA Players Stats dataset. First, the user needs to import the **Sevi** package and load the trained model, and then she can specify the dataset for visualization via `specify_dataset()`. Using **Sevi**, one can load datasets from SQLite databases, CSV files, or JSON files (Figure 3(a)).

Next, the user can issue a speech query via `speech2vis()`. Finally, **Sevi** renders the best-inferred visualization relevant to the speech query using the Vega-Lite (Figure 3(b)). In our current implementation, **Sevi** first transcribes the speech query into the corresponding TEXT query, and then performs Text2VIS in the backend.

(a) An example of Speech2VIS in the web-based **Sevi**



(b) An example of Text2VIS in the web-based **Sevi**

Figure 4: Sevi in Web-based Interface.

However, this process may introduce "noise" in the transcribed TEXT query. For example, as shown in Figure 3(b), the speech "bar chart" is transcribed as "bachata" due to the cacoepy. Given this observation, further work is needed to support robust Text2VIS from "noisy" and "ambiguous" TEXT queries. Furthermore, acquiring a (Speech, VIS) corpus and directly training deep learning models to support Speech2VIS in an end-to-end way is another promising direction.

The user can also directly specify the speech query by loading a pre-recorded audio file (Figure 3(c)). Note that the user can also perform Text2VIS directly by specifying TEXT queries via `text2vis()`. For example, if the user types in "`show me a bar chart of the number of players per position by ranking in descending order`", **Sevi** will return the same bar chart as shown in Figure 3(b).

Sevi in Web-based Interface. For non-coders or users who want an easy-to-use interface, **Sevi** offers a simple web-based interface that will be also demonstrated. Figure 4 are screenshots of **Sevi**'s web interface. First, the user can upload their relational dataset (table) by clicking the ⓘ Upload Dataset button. Next, the user can overview the schema information of the dataset in the *Columns* panel. Alternatively, the user can browse the full dataset by clicking the COVID-19 United States button.

As shown in Figure 4(a), the user can click the 🎙 button to record a speech query. Similar to **Sevi** in Jupyter notebook, **Sevi** will transcribe the speech query and generate the best visualization result. Alternatively, as depicted in Figure 4(b), the user can also directly input a text query and click the 🔍 button to get the visualization result.

4 CONCLUDING REMARKS

We demonstrated **Sevi**, a system to enable novices to create data visualizations through either a speech or a natural language input. So far, **Sevi** only supports speech in English, and we are planning to extend it to further support Chinese and Arabic. Moreover, through quantitative evaluation, we found that the performance of **Sevi** (and **ncNet**) clearly downgrades for the domains that are not included by **nvBench**, which shows a common hard-to-generalize problem. Therefore, we are also planning to significantly expand **nvBench** to cover more domains with more training examples. Furthermore, from the demonstration, we observe that supporting robust Text2VIS from the "noisy" and "ambiguous" queries is an important problem. Acquiring (Speech, VIS) pairs to train deep learning models to support Speech2VIS in an end-to-end fashion is another promising research direction.

Acknowledgement. This work is supported by NSF of China (61925205, 61632016), Huawei, BNRist, TAL Education, and Zhejiang Lab's International Talent Fund for Young Professionals. Hongyang Chen is supported by research initiation project of Zhejiang Lab (No. 2020LC0PI01).

REFERENCES

[1] Amazon's QuickSight, https://aws.amazon.com/cn/blogs/aws/amazon-quicksight-q-to-answer-ad-hoc-business-questions/.
[2] Microsoft Power BI Q&A. https://docs.microsoft.com/en-us/power-bi/create-reports/power-bi-tutorial-q-and-a.
[3] SpotIQ AI-Driven Insights (2nd Edition). https://www.thoughtspot.com/resources#white_paper.
[4] W. Cui, X. Zhang, and et al. Text-to-viz: Automatic generation of infographics from proportion-related natural language statements. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):906–916, 2020.
[5] E. Dong, H. Du, and L. Gardner. An interactive web-based dashboard to track covid-19 in real time. In *The Lancet infectious diseases*, volume 20, 2020.
[6] J. R. Finkel, T. Grenager, and C. D. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, 2005.
[7] Y. Luo, W. Li, T. Zhao, X. Yu, L. Zhang, G. Li, and N. Tang. Deeptrack: Monitoring and exploring spatio-temporal data - A case of tracking COVID-19 -. *Proc. VLDB Endow.*, 13(12):2841–2844, 2020.
[8] Y. Luo, X. Qin, C. Chai, N. Tang, G. Li, and W. Li. Steerable self-driving data visualization. *IEEE Transactions on Knowledge and Data Engineering*, 2020.
[9] Y. Luo, X. Qin, N. Tang, and G. Li. Deepeye: Towards automatic data visualization. In *ICDE 2018, Paris, France, April 16-19, 2018*, pages 101–112, 2018.
[10] Y. Luo, X. Qin, N. Tang, G. Li, and X. Wang. Deepeye: Creating good data visualizations by keyword search. In *SIGMOD*, 2018.
[11] Y. Luo, N. Tang, G. Li, C. Chai, W. Li, and X. Qin. Synthesizing natural language to visualization (NL2VIS) benchmarks from NL2SQL benchmarks. In *SIGMOD' 21, China, June 20-25, 2021*, pages 1235–1247. ACM, 2021.
[12] Y. Luo, N. Tang, G. Li, and et al. Deepeye: A data science system for monitoring and exploring COVID-19 data. *IEEE Data Eng. Bull.*, 2020.
[13] 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*, 28(1):217–226, 2022.
[14] C. D. Manning and et al. The stanford corenlp natural language processing toolkit. In *ACL*, pages 55–60, 2014.
[15] A. Narechania and et al. NL4DV: A toolkit for generating analytic specifications for data visualization from natural language queries. In *VIS*, 2020.
[16] X. Qin, Y. Luo, N. Tang, and G. Li. Making data visualization more efficient and effective: a survey. *VLDB J.*, 29(1):93–117, 2020.
[17] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE TVCG*, 23(1):341–350, 2017.
[18] V. Setlur, M. Tory, and A. Djalali. Inferencing underspecified natural language utterances in visual analysis. In *IUI*, pages 40–51, 2019.
[19] A. Vaswani and et al. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 2017.
[20] B. Yu and C. T. Silva. Flowsense: A natural language interface for visual data exploration within a dataflow system. *IEEE TVCG*, pages 1–11, 2020.
[21] H. Yuan and G. Li. A survey of traffic prediction: from spatio-temporal data to intelligent transportation. *Data Sci. Eng.*, 6(1):63–85, 2021.