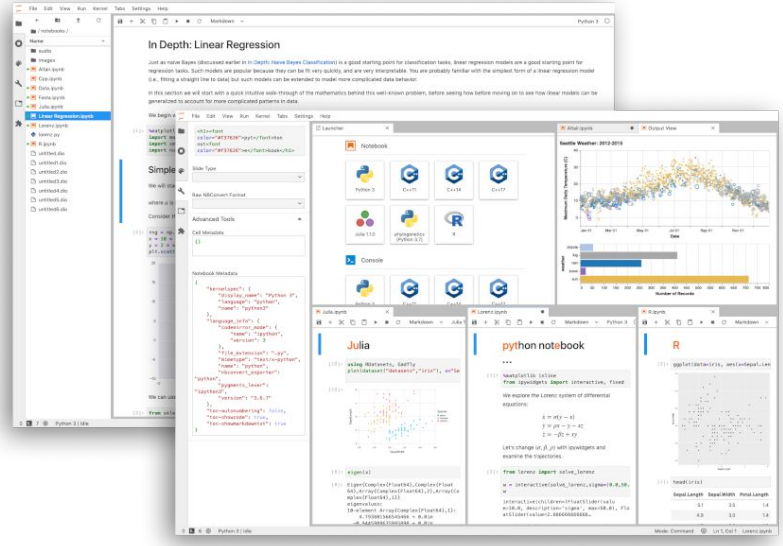
GENIE in a Notebook

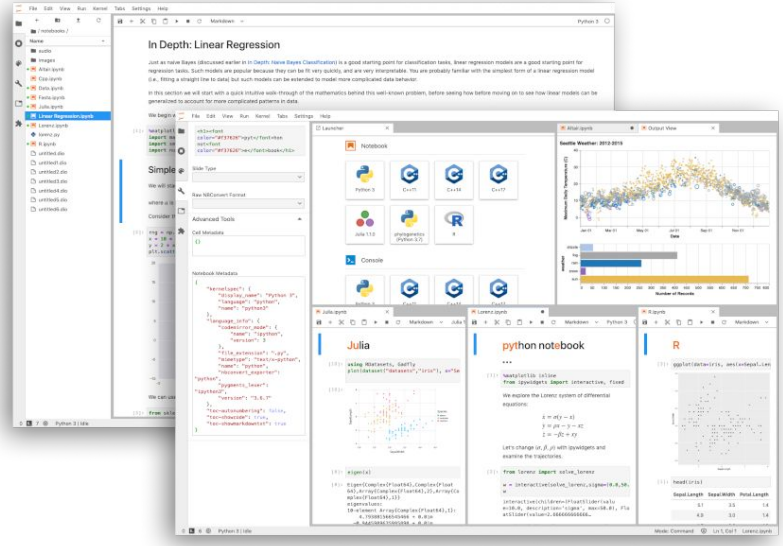
Speech-Based Code Generation in Computational Notebooks

Alice Yeh

Computational notebooks are a powerful tool



Recent work has added even more functionality



+

Auto-Suggest: Learning-to-Recommend Data Preparation Steps Using Data Science Notebooks

Cong Yan*
University of Washington
congy@cs.washington.edu

Yeye He
Microsoft Research
yeyehe@microsoft.com

ABSTRACT

Data preparation is widely recognized as the most time-consuming process in modern business intelligence (BI) and machine learning (ML) projects. Automating complex data preparation steps (e.g., Pivot, Unpivot, Normalize-JSON, etc.) holds the potential to greatly imp
has therefore become a central fo
We propose a novel approach t
alized data preparation steps, by

ACM Reference Format:
Cong Yan and Yeye He. 2020. Auto-Suggest: Learning-to-Recommend Data Preparation Steps Using Data Science Notebooks. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD'20), June 14–19, 2020, Portland, OR, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3318464.

mage: Fluid Moves Between Code and Graphical Work in Computational Notebooks

Mary Beth Kery
Carnegie Mellon University
mkery@cs.cmu.edu

Donghao Ren
Apple Inc.
donghao@apple.com

Fred Hohman
Georgia Institute of Technology
fredhohman@gatech.edu

Dominik Moritz
Apple Inc.
domoritz@apple.com

Kanit Wongsuphasawat
Apple Inc.
kanitw@apple.com

Kayur Patel
Apple Inc
kayur@apple.com

Sep 2020

standard notebook | 1 mage : user edits table | 2 mage : edits reflect in code

***Notable*: On-the-fly Assistant for Data Storytelling in Computational Notebooks**

Haotian Li
The Hong Kong University of Science and Technology
Hong Kong SAR, China
Microsoft Research Asia
Beijing, China
haotian.li@connect.ust.hk

Lu Ying
Zhejiang University
Hangzhou, Zhejiang, China
Microsoft Research Asia
Beijing, China
yinglu@zju.edu.cn

Haidong Zhang
Microsoft Research Asia
Beijing, China
haizhang@microsoft.com

Yingcai Wu
Zhejiang University
Hangzhou, Zhejiang, China
ycwu@zju.edu.cn

Huamin Qu
The Hong Kong University of Science and Technology
Hong Kong SAR, China
huamin@cse.ust.hk

Yun Wang*
Microsoft Research Asia
Beijing, China
wangyun@microsoft.com

7 Mar 2023

But two major pain points still exist

Synchronous collaboration is difficult

But two major pain points still exist

Synchronous collaboration is difficult

Non-technical users are left out

Related work: speech-based code generation

- Enable users to write code using structured voice commands
- Works: VocalProgramming, Blockly, Serenade

\+ No need to remember syntax, can program hands-free

\- Users still need to understand general logic and syntax

Command	Code
add function int factorial	`def factorial() -> int:` `  pass`
add class page	`class Page:` `  pass`
add return say of string hello	`return say("hello")`

add else if x less than three	`if x > 3:`
	`  return`
	`elif x < 3:`
	`  pass`

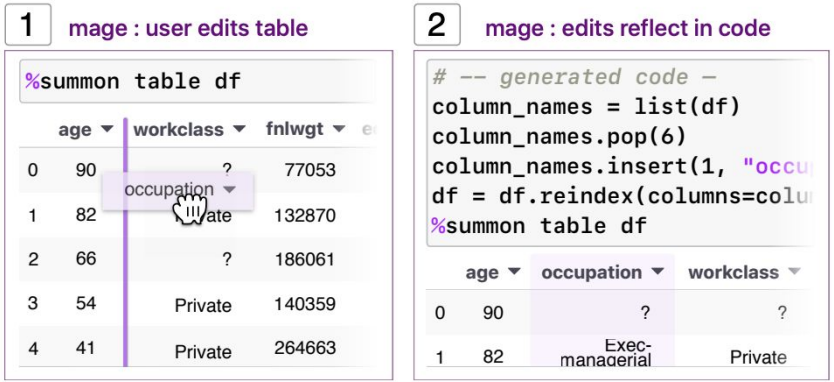
add import numpy as np	`import numpy as np`

Related work: computational notebook tools

- Some aimed at simplifying data workflows
- Works: Lux, mage

+ Simplifies computational notebook interactions

- Users need to have some understanding of code

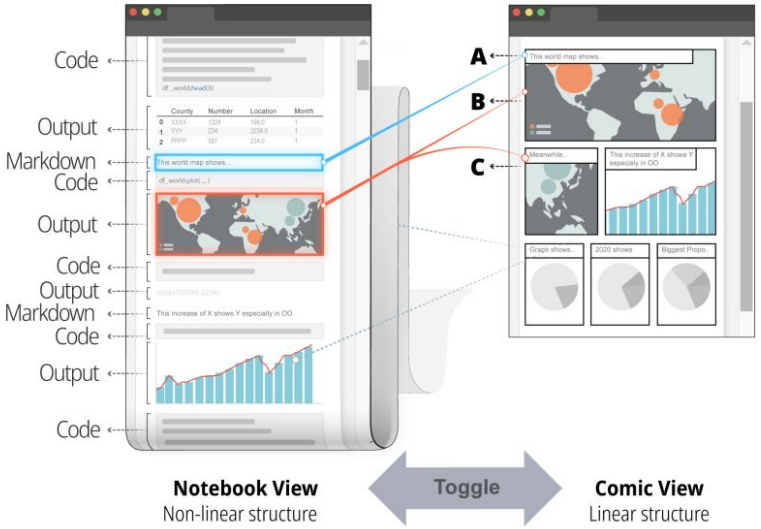


Related work: computational notebook tools

- Others work to broaden computational notebook audiences
- Works: ViDeTTe (data exploration), ToonNote (data understanding)

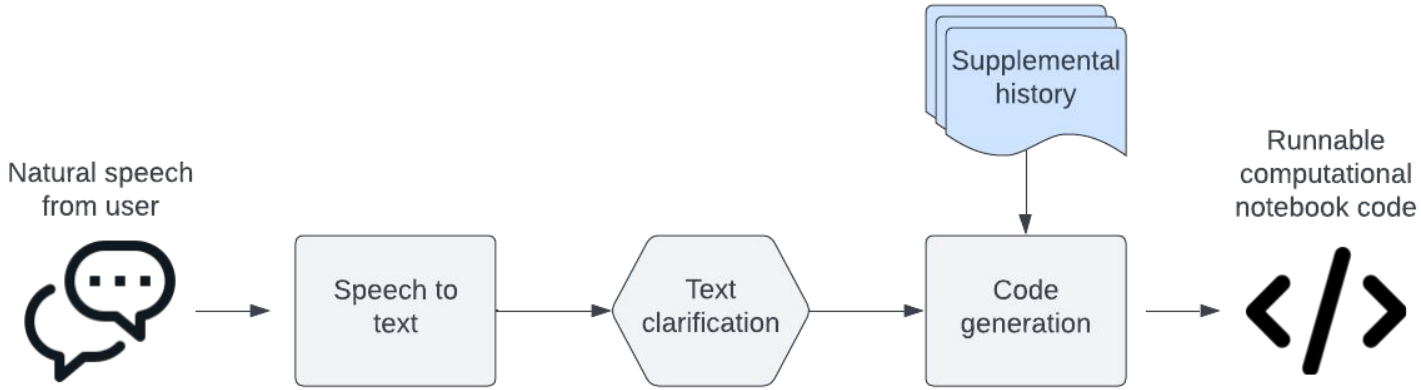
\+ Makes data work more accessible to non-technical audiences

\- Non-technical users still cannot partake in the data analysis process



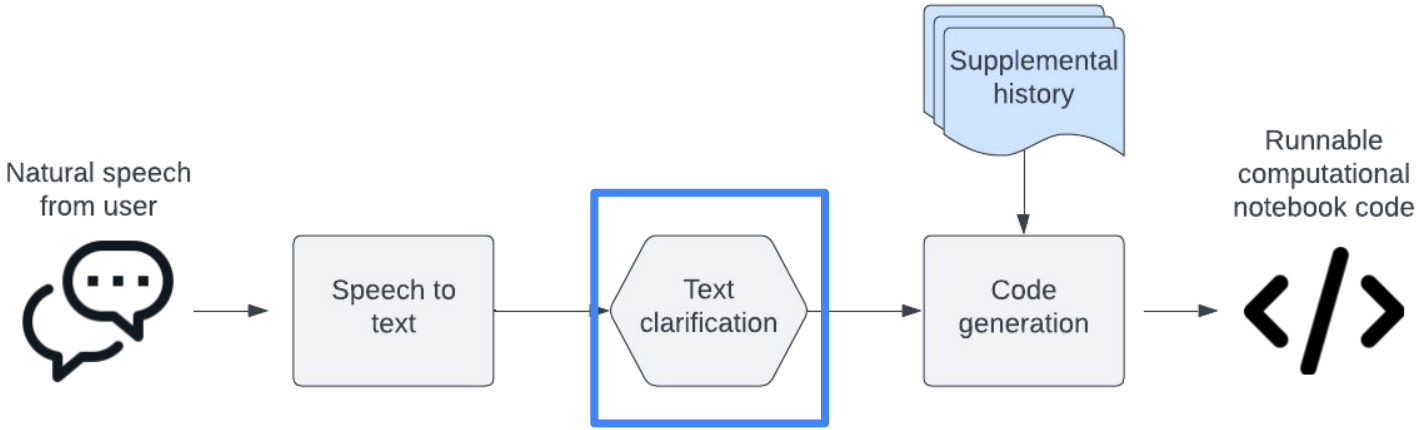
GENIE

Computational notebook tool that enables users to generate code through natural speech



Demo

GENIE Workflow

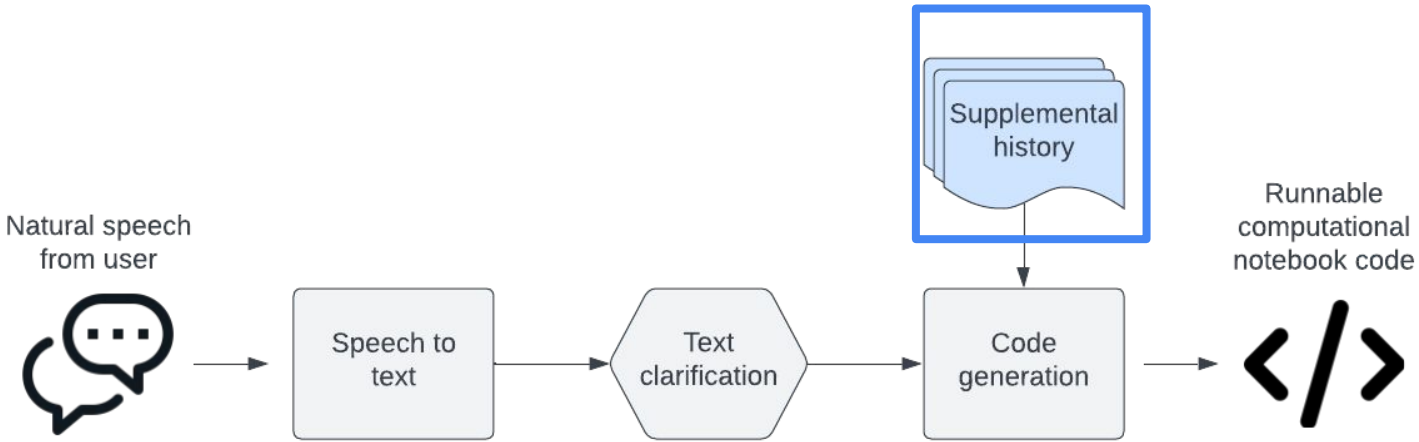


Text Clarification

- Target ambiguities in user speech that cannot be resolved by code generator
- Parse patterns in text and locate keywords frequently used in computational notebook contexts
  - Ex. preposition search

    ```
    <preposition> <”dataset”/”file”/”graph”>
    ```
- Challenges
  - Unbounded vocabulary as input
  - Code generation will occur regardless of validity of text

GENIE Workflow



Feeding Relevant History

- Provide user's prior queries and code as context to code generation stage
  - User will not need to track variables and can work on separate workflows simultaneously
- Create *user workflows* with k-means clustering
  - Group each interaction into workflows and feed in history based on workflow
  - Squash code from long/dense workflows (ex. care about variable assignment but not print statements)

Other Implementation Details

- Web Speech API for automatic speech recognition
- Chat Completions API for code generation

User Studies

- Have presented idea to a class of 18 students with mixed technical abilities
  - General interest from students, especially non-technical students
- User study design
  - User allotted 10 minutes to watch tutorial and play around with GENIE, then provided with a specific question to answer based on a dataset
  - User is given 30 minutes to solve the question
    - First 10 min: no GENIE access but access to the Internet and any tools (ChatGPT); last 20 min: full access

Next Steps

- Enhance functionality (clarification and history)
- Run user studies with technical and non-technical audiences
- Perform benchmarks on individual components to understand speed

Thanks! Questions?