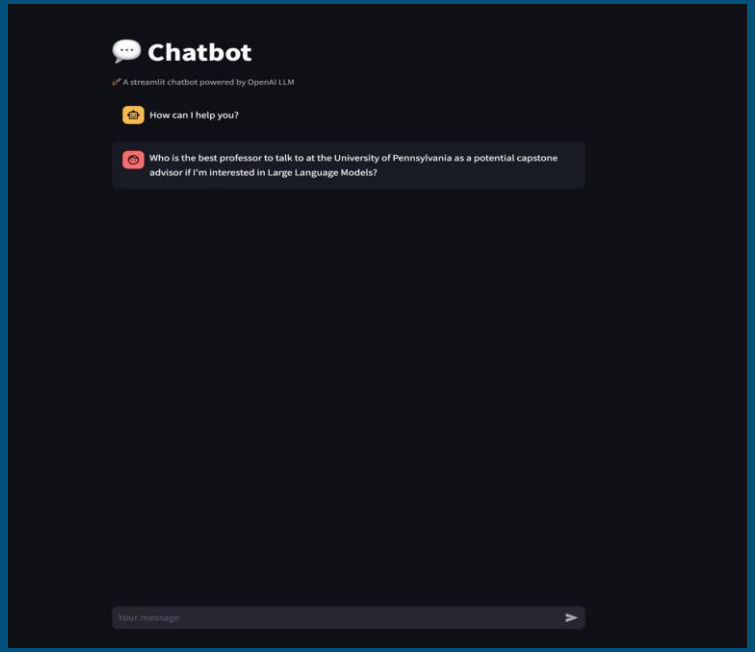
PennPalGPT:
ChatBot for students seeking research advisor

Grace Benner, Mikaela Spaventa, Natalie Gilbert,
Yash Nakadi

PennPalGPT

- Function: Q&A ChatBot
- Users: Penn students interested in research
- Model: GPT-4 (gpt-4-0613)
- UI: Streamlit (once completed)



Motivation

Utility:

- Many students at Penn are interested in research
- Finding the right advisor for your research interests is crucial

NLP:

- Popularity of chatbots has exploded in the past 12 months
- Utilize knowledge of LLMs, apply GPT models in useful setting

What we learned:

- Generating features
- How to group similar professors together
- Fine-tuning prompt to get desired output
- Intricacies of the NLP Model Pipeline

Training Data (Pretraining)

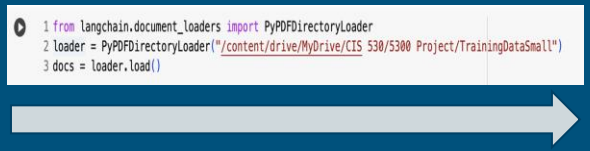
- Academic papers, website bios, contact info, etc. for 102 Penn Engineering faculty in the CIS department (2410 documents in total, not including metadata)
- PyPDF2 was used for text extraction

Research paper

Piazza: Data Management Infrastructure for Semantic Web Applications

Alon Y. Halevy Zachary G. Ives* Peter Mork Igor Tatarinov
University of Washington
Box 352350
Seattle, WA 98195-2350
{alon,zives,pmork,igor}@cs.washington.edu

Extraction via PyPDF2



```
from langchain.document_loaders import PyPDFDirectoryLoader
loader = PyPDFDirectoryLoader("/content/drive/MyDrive/CIS 530/5300 Project/TrainingDataSmall")
docs = loader.load()
```

Text file for pretraining

Document(page_content='The Piazza PeerData Mana gement Project\nIgor Tatarinov1,Zachar yIves2,JayantMadha van1,\nAlon Hale vy1,Dan Suciu1,Nilesh Dalvi1,Xin(Luna) Dong1,\nYana Kadiyska1,Gerome Miklau1,Peter Mork1\n1Depar tment ofComputer Science and Engineer ing\nUniv ersity ofWashington, Seattle ,WA98195\nfigor,jayant,alon,suciu,nilesh,lunadong,ykadiysk,gerome ,pmor kg@cs .washington.edu

Dev/Test Data

- Description: Each dataset contains 50 examples (input, answer)
- Source: Research interests from PhD students at Penn (input), and their actual advisor (answer)
- Dev set function: fine tuning
- Test set function: evaluation

Test/Dev Input	Expected Response	Name (if applicable)
My name is Weiqiu You (尤玮秋). I am a 4th year PhD student in Computer and Information Science at University of Pennsylvania. Previously, I was very fortunate to have worked with Prof. Mohit Iyyer while I was a Master's student at UMass, Prof Jon May while I was an intern at USC/ISI, and Dr. Youngja Park while I was an intern at IBM Research. I am broadly interested in * Machine learning * Explainable AI Who would you recommend be my advisor based on my interests?	Eric Wong	Weiqui Yu

Models Leading Up to Final Model

Simple Baseline

- Model: GPT 3.5 Turbo, msmarco-bert-base-dot-v5 embeddings
- Quality: relevant, grammatically correct, very unhelpful

Strong Baseline

- Model: GPT 3.5 Turbo 16k, same embeddings
- Quality: much more helpful and accurate

Concept Bottleneck Model (Yang et al., 2023)

- Advantage: Increase interpretability of results, identifies key features
- Disadvantage: Less accurate
- Variations:
  - Random Forest
  - SVC
  - KNN

Evaluation Metrics

Quantitative:

- Each answer is scored 1-5 depending on how similar the chatbot answer is to the golden standard answer
- We take into account if the outputted professors are in the same groups
- Given our set of scores S = {s_1, s_2, … s_n} and the probability of each score is $p(s_i)$, we calculate the final score as follows:

$$\sum_{i=1}^{n} \frac{p(s_i)*s_i}{n}$$

Qualitative:

- Evaluating quality of the response
  - Understandable
  - Grammatically error-free
  - Valid answer
  - Interpretability (justification of choice)

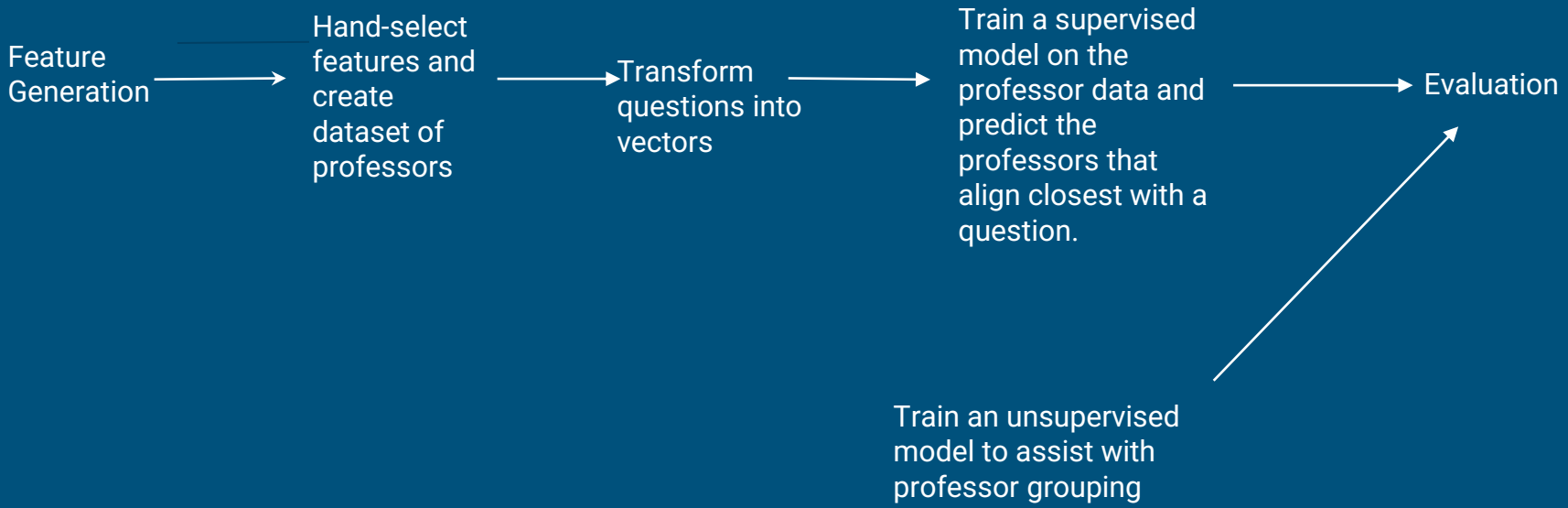
Inspiration: Liu et al. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment

Groups of Professors to Assist with Evaluation

Group	Professors
1	Hamed Hassani, Sanjeev Khanna, Michael Kearns, Robin Pemantle, Aaron Roth, Jacob Gardner, Shivani Agarwal, Anindya De, Eric Weingarten, Rakesh Vohra, Weijie Su
2	Lingjie Liu, Eric Eaton, Nadia Figueroa, Dinesh Jayaraman, Mingmin Zhao, Jianbo Shi, Dan Roth, Andrew Head, Rajeev Alur, Justin Gottschlich, Charles Yang, Michael Posa, Kostas Daniilidis, Alejandro Ribeiro, Cynthia Sung, Camillo Taylor, M. Ani Hsieh, Vijay Kumar, Daniel E. Koditschek
3	Linh Thi Xuan Phan, Jonathan Smith, Ryan Marcus, Tal Rabin, Vincent Liu, Mayur Naik, Benjamin C. Pierce, Susan Davidson, Andreas Haeberlen, Zachary Ives, Sebastian Angel, Lin Thi Xuan Phan
4	Rajiv Gandhi, Brett Hemenway, Harry Smith, Adam David Mally, Andre Scedrov, Bong Ho Kim, Jérémie O. Lumbroso, Nikolai Matni, Travis Q. McGaha, Pratyush Mishra, Shirin Saeedi Bidokhti, Boon Thau Loo, Jing Li, Benjamin Lee, Stephen Lane, Joe Devietti, Pratik Chaudhari, Jean Gallier, Thomas Farmer, Val B. Tannen, Christopher S. Yoo, Scott Weinstein, Stephanie Weirich, Steven Zdancemi
5	Konrad Kording, Qi Long, Victor M. Preciado, Insup Lee, Yoseph Barash
6	Daniel Hashimoto, Junhyong Kim, Harvey Rubin, Joshua B. Plotkin, Mark L. Liberman, Kevin B. Johnson, Sampath K. Kannan, James Gee, Norman I. Badler, Li-San Wang, Oleg Sokolsky, Rene Vidal
7	Eric Fouh, Gushu Li, Yasmin Kafai, Ryan Baker, Swapneel Seth
8	Osbert Bastani, Surbi Goel, Chris Callison-Burch, Mark Yatskar, Eric Wong, Danaë Metaxa, Damon Centola, Sharath Chandra Guntuku, Daniel J Hopkins, Duncan Watts, Lyle Ungar

Used K-means++ to help us find these groups

Concept Bottleneck Model Approach



Final Model Pipeline

Feature Generation

Model: gpt-4-0613

Hand-select features and create data_professors.csv, dataset of all professors and their feature assignments

Transform questions into vectors

Model: gpt-4-0613

Train a KNN model on the professor data, where each professor corresponds to a group and predict the 4 closest neighbors for a question.

Model: KNN

Assigns a score between 1 and 5 for each chatbot answer compared to its golden answer and a final answer between 0 and 1

Model: gpt-4-0613

Assisted us with grouping of similar professors (which helped us more score the quality of our chatbot answers)

Model: K-Means++

Model	Improvements	Quantitative Performance	Qualitative Performance
Simple Baseline	n/a	.458 (test)	- A lot of 'I do not know' answers - Many hallucinations Ex:'Professor Alice'. or professors at other universities. - Not answering in the correct format -None of the answers are correct
Strong Baseline	-Used gpt-3.5-turbo-16k instead of gpt-3.5-turbo -Instructed to only return a professor's name from UPenn and gave a list of all possible professors names	.644 (test)	- Less 'I don't know' answers - Less hallucinations but they are still frequent - Still not always answering in the correct format - Not many of the answers are correct - We don't trust the quantitative evaluation since this model doesn't have context on how similar the professors are
Extension One	-Used bottleneck concept approach -Changed evaluation prompt to account for professors with similar interests	Random Forest - .456 (dev) SVC - .598 (dev) KNN - .608 (dev)	- No hallucinations - Always answering in the right format - Many answers are incorrect or outputting very unrelated professors
Extension Two	-Used gpt-4-0613 instead of gpt-3.5 for feature generation and changed the extraction approach -Created 112 features instead of 93 -Used KNN model only - Changed grouping of professors in evaluation prompt	.7 (dev) .718 (test)	- Most guess the expected professor or some that are very similar to the expected professor -Still often receive a list of professors for answers we're not expecting one for

Future Work

- Create a UI that runs our model in the backend
- Allow the chatbot to output a wider array of answers
- Create more features

Thank you for watching our presentation!