Large Language Models

INTRO AND DEMO

Agenda

- Disclaimers
- Main Take Aways
- Why Won't He Stop Talking About Data?
- Different Tools – Different Jobs
- Prompt Engineering
- Demos
- Q&A

Disclaimers

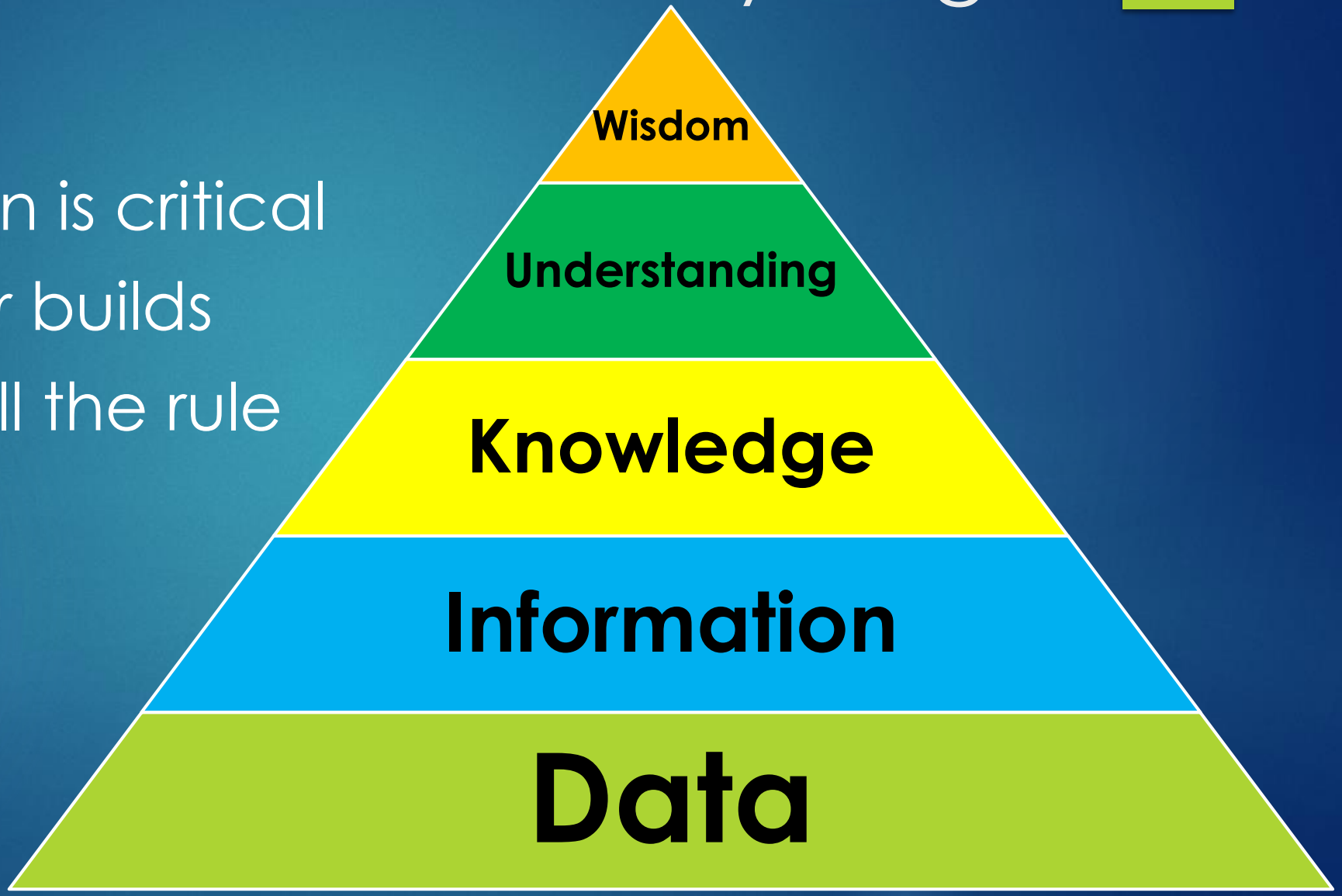
- My opinions unless specifically otherwise noted
- This is a highly dynamic environment
- YMMV – Your mileage may vary
- You will have homework
- I will include a pointless pie chart

Main Take Aways

- Data is everything
- Question LLM training data
- No “Holy Grail” – many Solo cups

WUKID Pyramid: Data is everything

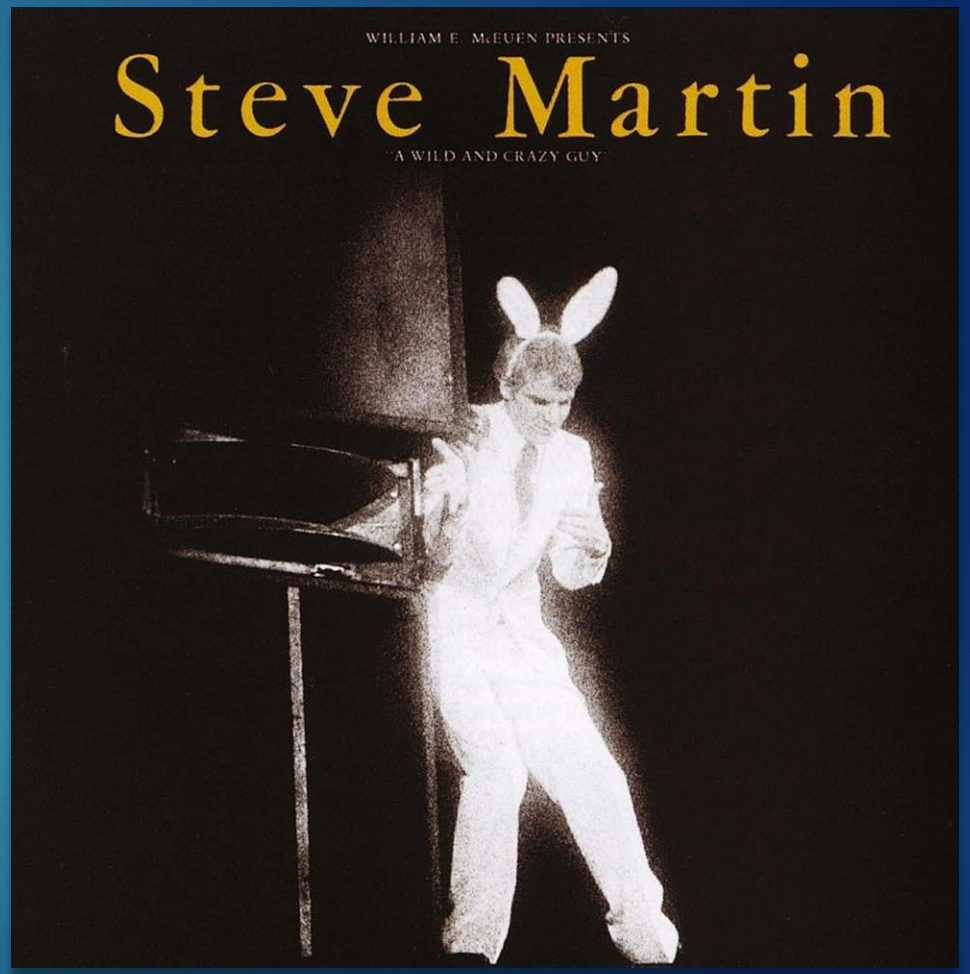
- Foundation is critical
- Each layer builds
- GIGO is still the rule



“Kids learn how to talk from listening to their parents.

So, if you have a three-year-old kid and you want to play a dirty trick on him, whenever you're around him, you talk wrong.

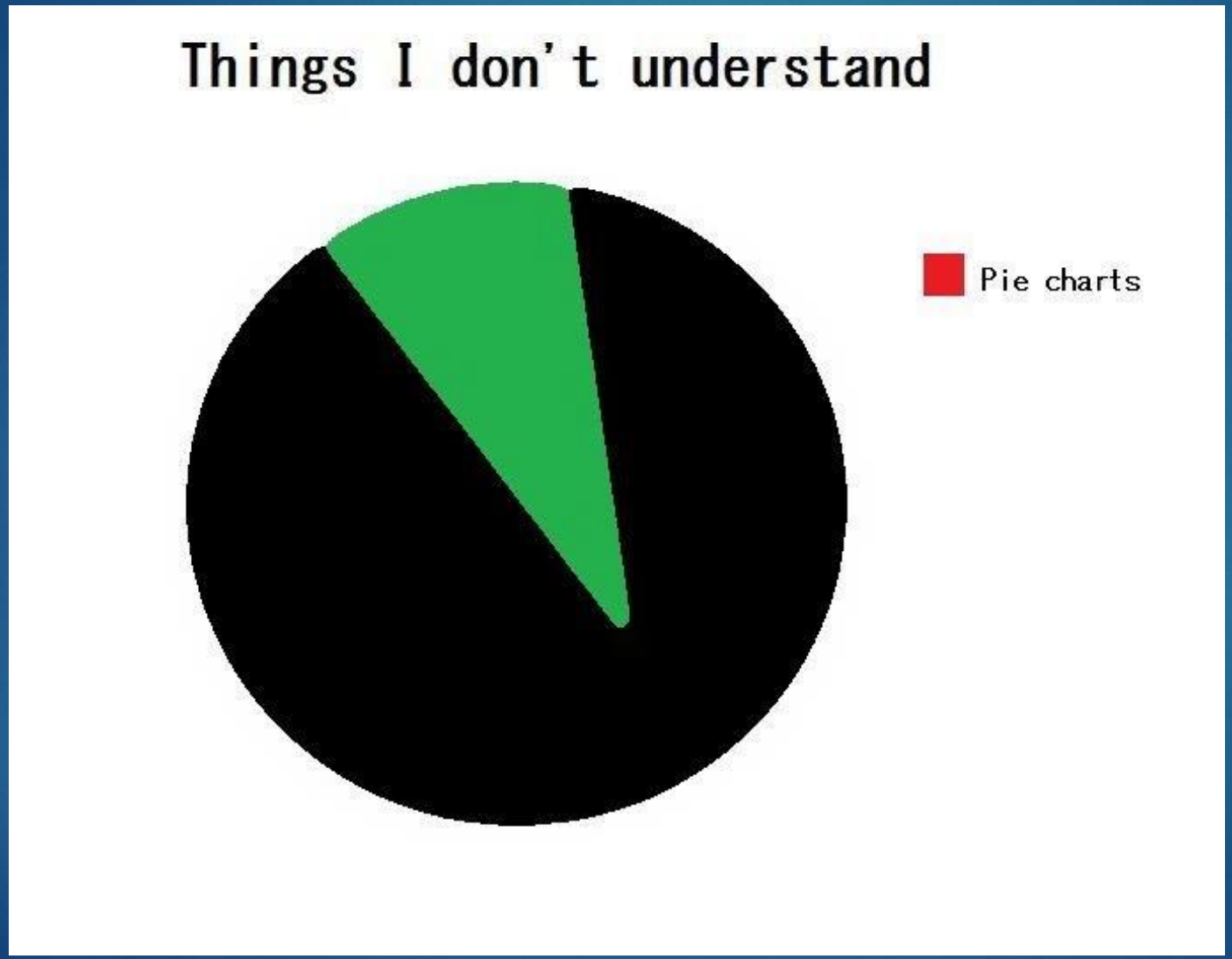
So now it's like his first day in school and he raises his hand: *‘May I mambo dogface to the banana patch?’*"



LLM Development Key Steps

- Data Collection and Preprocessing
- Model Architecture Selection***
- Pre-training
- Fine-tuning
- Model Evaluation
- Deployment and Inference

Pointless Pie Chart



Different Tools – Different Jobs

Model	Core differentiator	Pre-training objective	Para-meters	Access	Information Extraction	Text Classification	Conversa-tional AI	Summari-zation	Content generation
BERT	First transformer-based LLM	AE	370M	Source code					
RoBERTa	More robust training procedure	AE	354M	Source code					
GPT-3	Parameter size	AR	175B	API					
BART	Novel combination of pre-training objectives	AR and AE	147M	Source code					
GPT-2	Parameter size	AR	1.5B	Source code					
T5	Multi-task transfer learning	AR	11B	Source code					
LaMDA	Dialogue; safety and factual grounding	AR	137B	No access					
XLNet	Joint AE and AR	AE and AR	110M	Source code					
DistilBERT	Reduced model size via knowledge distillation	AE	82M	Source code					
ELECTRA	Computational efficiency	AE	335M	Source code					
PaLM	Training infrastructure	AR	540B	No access					
MT-NLG	Training infrastructure	AR and AE	530B	API					

Prompt Engineering

- Be as specific as possible
- Supply the AI with examples
- Get better answers by providing data
- Specify your desired output
- Provide instructions on what to do instead of what not to do
- Give the model a persona or frame of reference
- Try chain of thought prompting
- Split complex tasks into simpler ones
- Understand the model's shortcomings
- Take an experimental approach to prompting


Demo

Main Take Aways - Reprise

- Data is everything
- Question LLM training data
- No “Holy Grail” – many Solo cups

Q&A

Thank You!

ANDREW.GIBBS@LIBERTY-SOURCE.COM (WORK)

ANDREWMCLEANGIBBS@GMAIL.COM (NON-WORK)

HTTPS://WWW.LINKEDIN.COM/IN/ANDREWMGIBBS/

GRAB TIME ON MY CALENDAR: