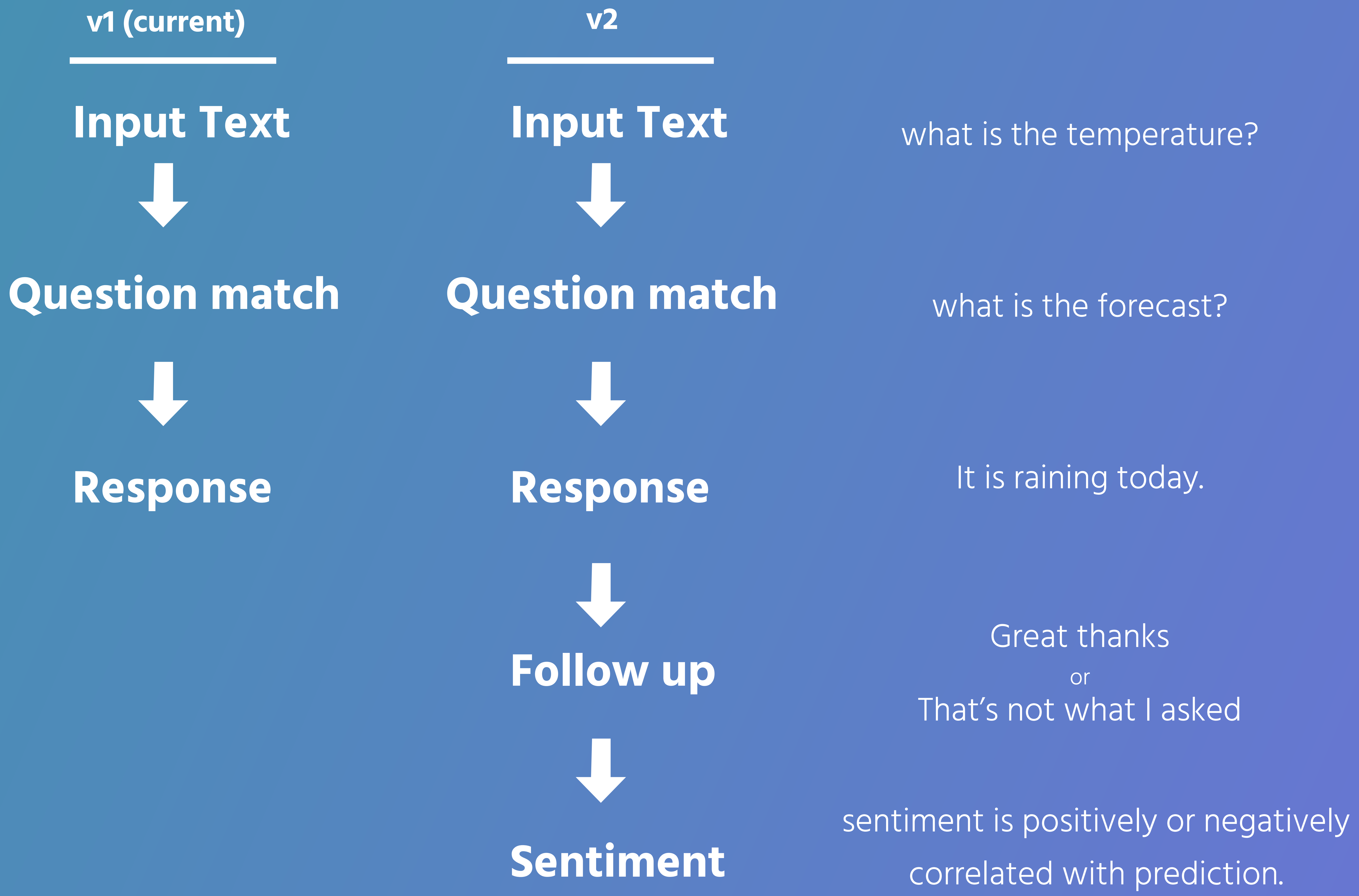
#DATA_SCIENCE

@GENERAL_ASSEMBLY

Final project by Samuel Delesque

[foreword] Baily Bot Structure



v1 (current)

Input Text

Question match

Response

v2

Input Text

Question match

Response

Follow up

Sentiment

what is the temperature?

what is the forecast?

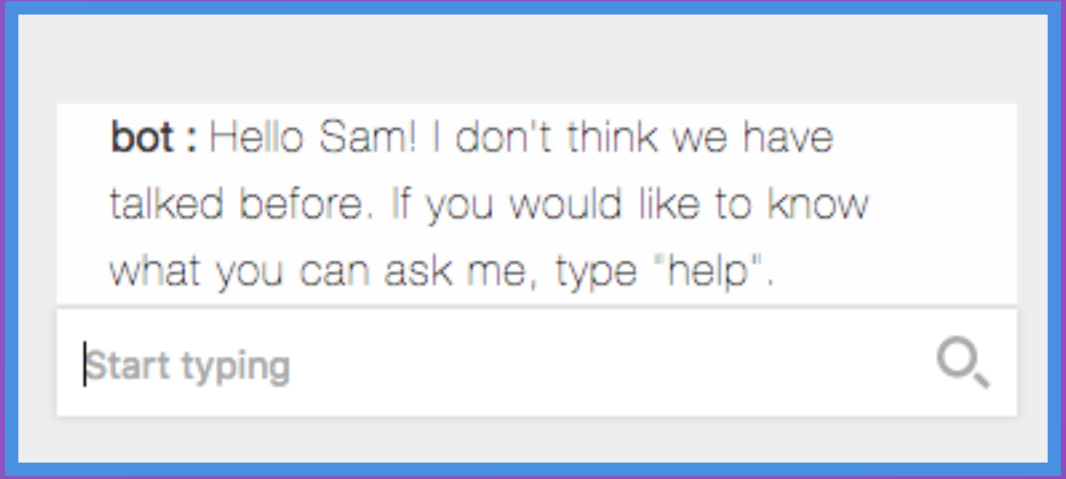
It is raining today.

Great thanks
or
That's not what I asked

sentiment is positively or negatively correlated with prediction.

BUILDING A FEEDBACK MODEL FOR BAILY

- better matching models (using fuzzywuzzy currently)
- context extractor for time, loc, names, subject… (covariables?)
- model predicting possible user follow-up after a response
- sentiment analysis to estimate weather a response was satisfactory



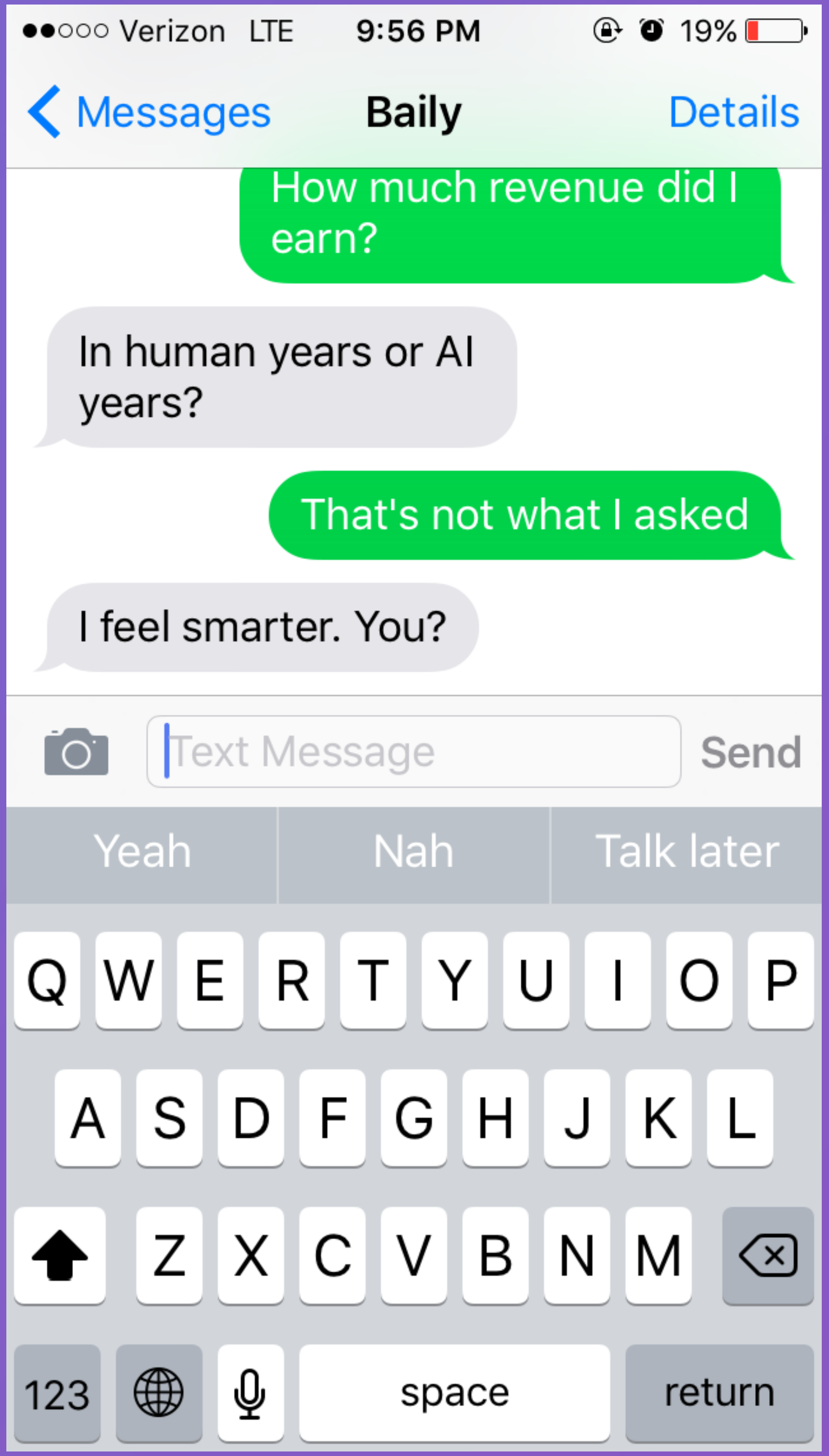
bot : Hello Sam! I don't think we have talked before. If you would like to know what you can ask me, type "help".

Data:

- conversation history from Baily

Objective: Combining a prediction of possible user follow up text with sentiment analysis, to determine if response was well received.

Hypothesis: Positively correlated sentiments between follow-up prediction and actual follow up text should result in higher perceived value in bot responses and can help build the response model over time.



How much revenue did I earn?

In human years or AI years?

That's not what I asked

I feel smarter. You?

Our goal will be to write a model that can predict wether a response was well received.

A few data points that we will collect in order to predict wether the response was successfull or not will be:

- Classification of text: Different question, short feedback (thanks..), related question,
- Sentiment of text (testable on http://text-processing.com/demo/sentiment/)
- Score of initial response (is there a certain threshold at which the feedback becomes better?)

Let’s have a more detailed look at our variables.

Question (Q) = user input text

Q.score = match model score to existing question

Response (R) = bot response

Subject (S) = subject of the conversation (i.e. weather forecast)

S.score = similarity of subject between R and F

Feedback (F) = user follow up text (F ~ Q[n-1])

F.class = classification of text (question, short affirmation, long affirmation…)

F.sentiment = sentiment score of follow up text

F.score = result (0 to 1)

Our initial assumption is that the model would follow something like:

Q.score++, F.sentiment ++ == F.score++

Positive F.classes:

- short affirmation (ex: “thanks”)
- different Q (we assume first question was answered)

Negative F.classes:

- same Q, similar Q or S (“how cold is it?” -> “no - I mean what is the temperature?”)