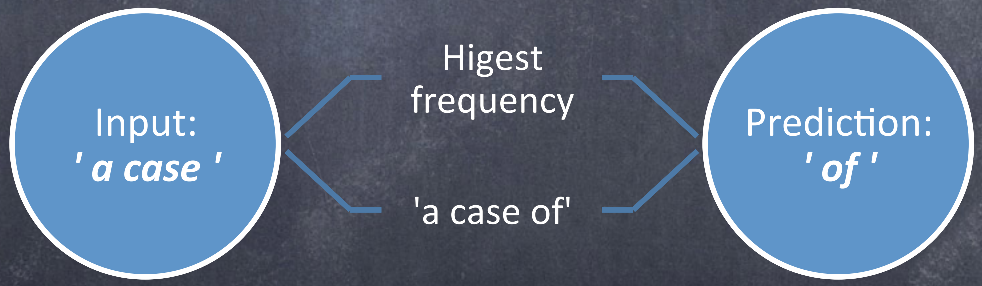
Next Word Prediction

Qi Shao

Overview

- The goal of this project is to allow a user to input a phrase into the application, and it would predict the next word that they “most likely” want to type.

- The primary use case for this application is text messaging on mobile phones.



Tasks

1. Data Acquisition
2. Data processing
3. Exploratory Analysis
4. Building Model
5. Prediction & Evaluation
6. Creative Exploratory
7. Data Product

Task 1 - Data Acquisition

- **Data Acquisition**
- Data processing
- Exploratory Analysis
- Building Model
- Prediction & Evaluation
- Creative Exploratory
- Data Product

Data Acquisition | Data processing | Exploratory Analysis | Building Model | Prediction & Evaluation | Creative Exploratory | Data Product

Data Acquisition

- HC Corpora (www.corpora.heliohost.org)
  - Blogs
  - News
  - Twitter

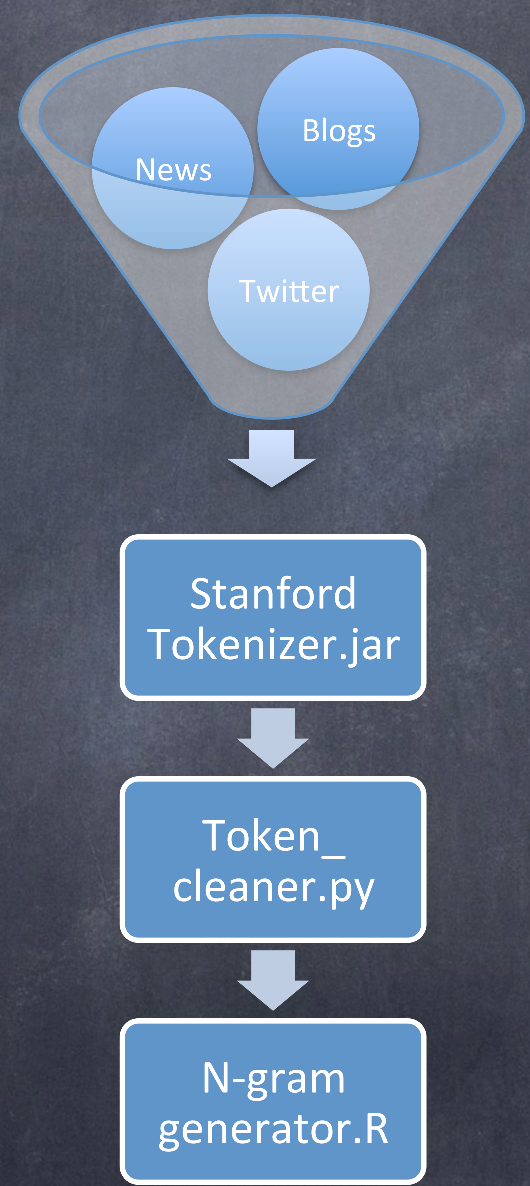
File	Size (MB)	Line Counts	Word Count	Average word length	Average word per line
Blogs	210.2	899,288	37,334,690	5.59	41.51
News	205.8	1,010,242	34,372,720	5.97	34.01
Twitter	167.1	2,360,148	30,374,206	5.49	12.86

Task 2 - Data Processing

- Data Acquisition
- Data processing
- Exploratory Analysis
- Building Model
- Prediction & Evaluation
- Creative Exploratory
- Data Product

Data Acquisition → Data processing → Exploratory Analysis → Building Model → Prediction & Evaluation → Creative Exploratory → Data Product

Processing Flow



Tokenization

- Stanford Tokenizer
  - Initially designed to largely mimic Penn Treebank 3 (PTB) tokenization
  - Mainly targets formal English writing rather than SMS-speak.

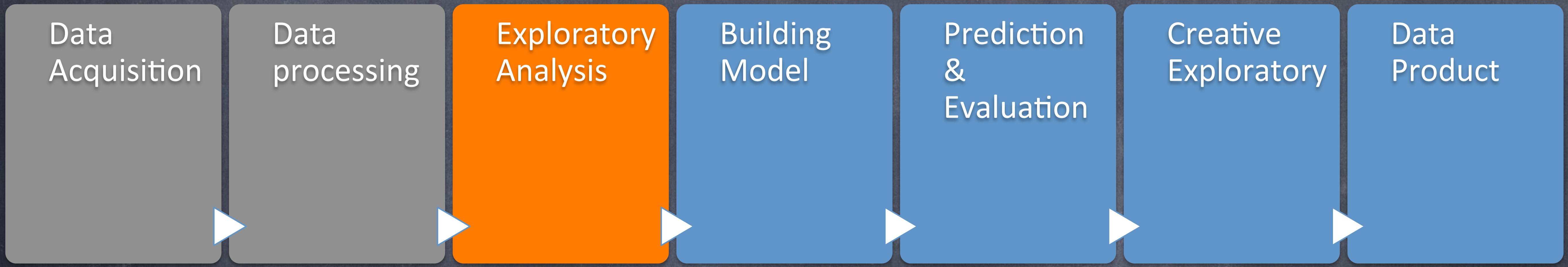
Token Cleaning

- Token cleaner
  - Converting to lower case
  - Removing numbers
  - Removing Punctuations
  - Removing Foreign words
  - Removing extra white spaces

Token Cleaning(con.)

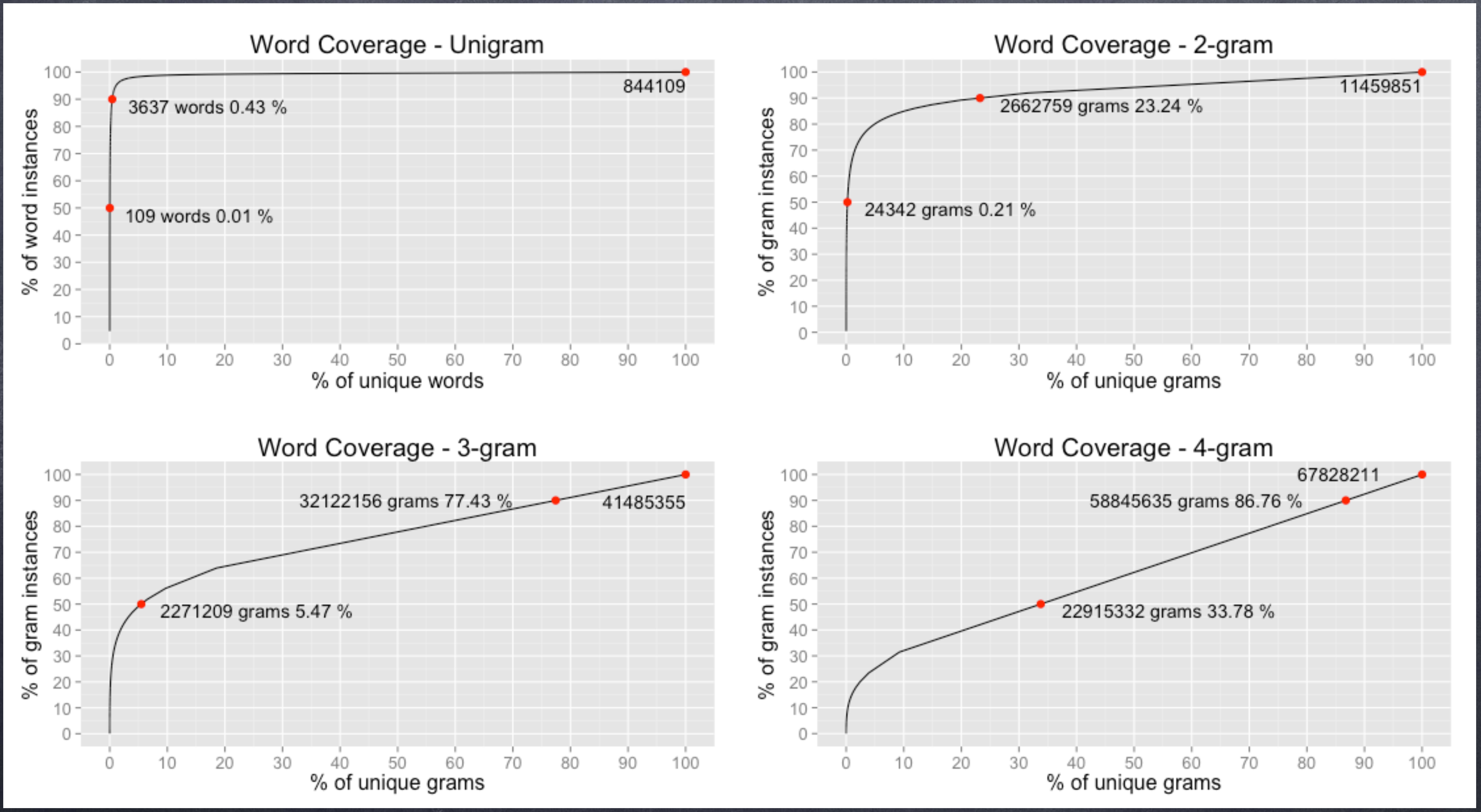
- N-gram
    - Stemming
    - Generate 1-4 grams termDocumentMatrix
    - Save to Rdata

Task 3 - Exploratory Analysis



Data Acquisition → Data processing → Exploratory Analysis → Building Model → Prediction & Evaluation → Creative Exploratory → Data Product

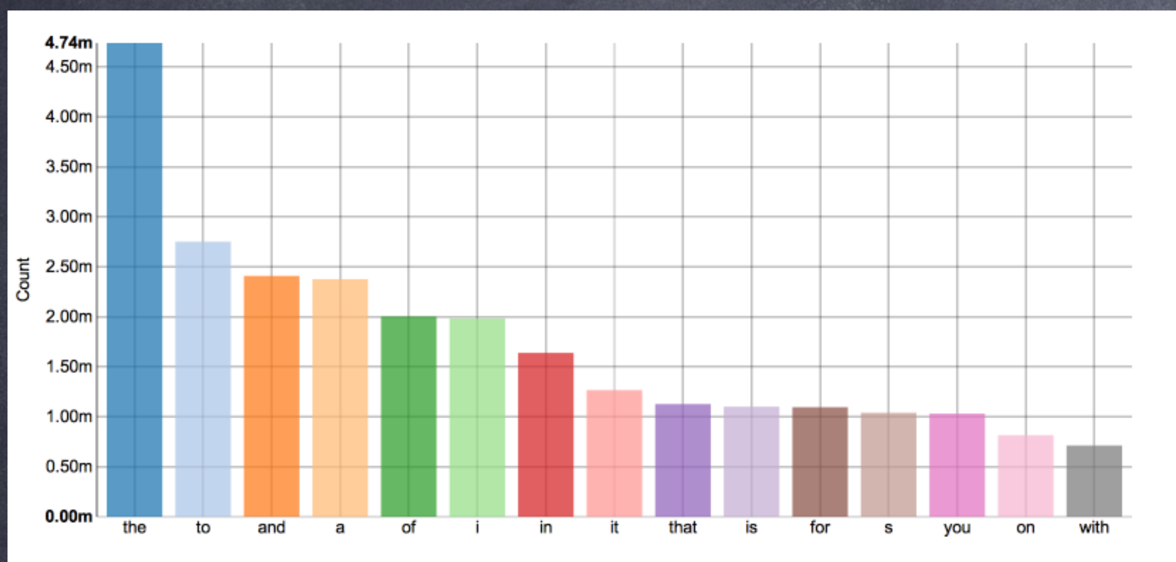
Exploratory Analysis - Word Coverage

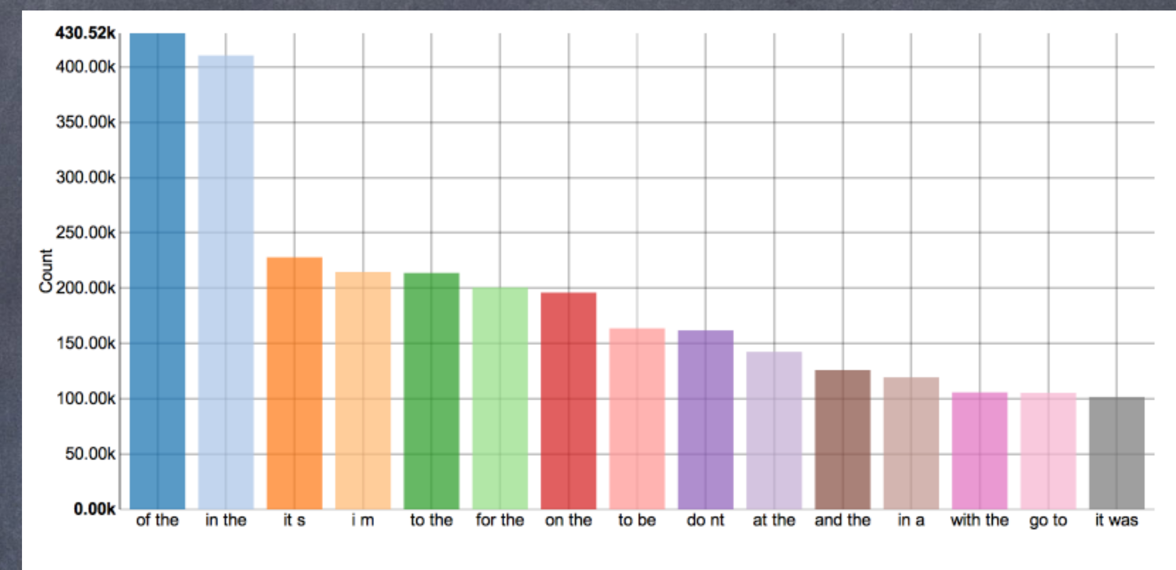


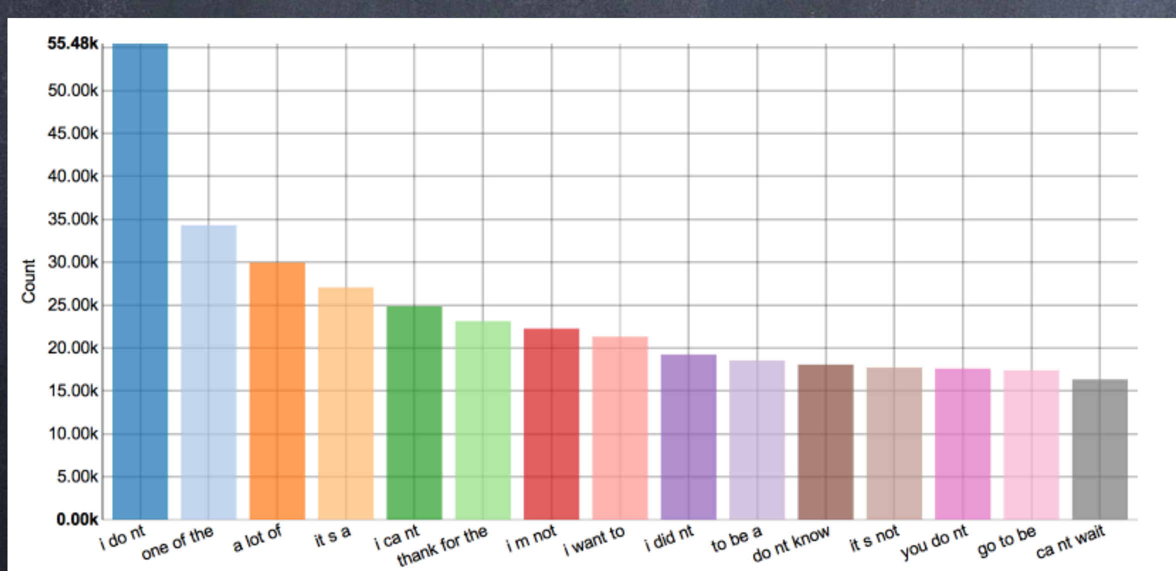
Exploratory Analysis - Word Cloud

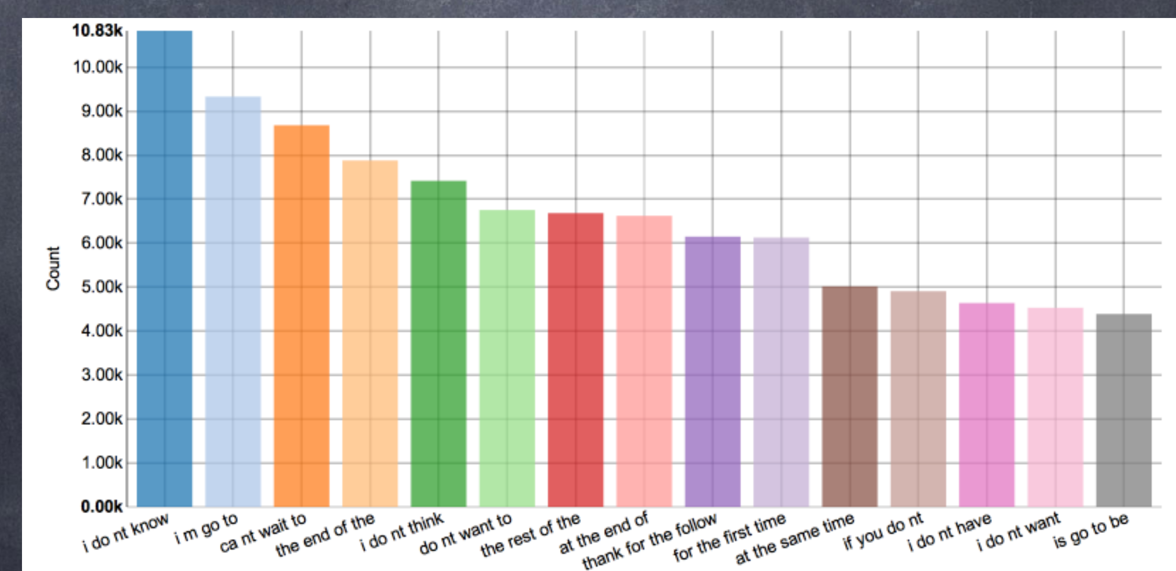
Data Acquisition | Data processing | Exploratory Analysis | Building Model | Prediction & Evaluation | Creative Exploratory | Data Product

Exploratory Analysis - Top Frequency Words









Task 4 - Building Model

1. Data Acquisition
2. Data processing
3. Exploratory Analysis
4. Building Model
5. Prediction & Evaluation
6. Creative Exploratory
7. Data Product

Data Acquisition | Data processing | Exploratory Analysis | Building Model | Prediction & Evaluation | Creative Exploratory | Data Product

Modeling - Computing Probability

- Estimate probability from relative frequency counts?

$$P(\text{too} \mid \text{nice to meet you}) = \frac{C(\text{nice to meet you too})}{C(\text{nice to meet you})}$$

- Chain rule of probability?

$$P(w_1 w_2 \ldots w_n) = \prod_i P(w_i \mid w_1 w_2 \ldots w_{i-1})$$

- Markov Assumption!

$$P(w_i \mid w_1 w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-k} \ldots w_{i-1})$$

Modeling - N-Gram

- An n-gram is a contiguous sequence of n items from a given sequence of text or speech. (Wikipedia)
- Example : 3-grams of “The quick brown fox jumps over” are “The quick brown”, “quick brown fox”, “brown fox jumps”, “fox jumps over”.
- The Frequencies of the N-Grams are stored in a table.

word	count
the	4739361
to	2752048
and	2409487
of	2003983

Modeling - Maximum Likelihood Estimation

Quadgram ML estimate

$$q_{ML}(w_i \mid w_{i-3}, w_{i-2}, w_{i-1}) = \frac{\text{Count}(w_{i-3}, w_{i-2}, w_{i-1}, w_i)}{\text{Count}(w_{i-3}, w_{i-2}, w_{i-1})}$$

Trigram ML estimate

$$q_{ML}(w_i \mid w_{i-2}, w_{i-1}) = \frac{\text{Count}(w_{i-2}, w_{i-1}, w_i)}{\text{Count}(w_{i-2}, w_{i-1})}$$

Bigram ML estimate

$$q_{ML}(w_i \mid w_{i-1}) = \frac{\text{Count}(w_{i-1}, w_i)}{\text{Count}(w_{i-1})}$$

Unigram ML estimate

$$q_{ML}(w_i) = \frac{\text{Count}(w_i)}{\text{Count}()}$$

Modeling - Discrimination vs Reliability

- larger n: more information about the context of the specific instance (great discrimination)
- smaller n: more instances in training data,better statistical estimate(more reliability)

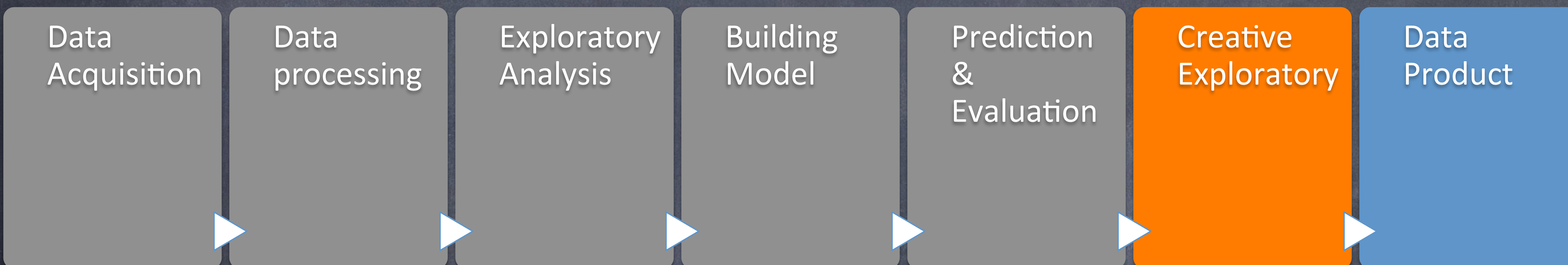
Task 5 - Prediction & Evaluation

Data Acquisition → Data processing → Exploratory Analysis → Building Model → Prediction & Evaluation → Creative Exploratory → Data Product

Prediction & Evaluation

- Extrinsic Evaluation
  - End to end compare performance of two models
  - Time consuming
- Intrinsic Evaluation
  - Perplexity $PP(W) = P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}}$ → $\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_1 \ldots w_{i-1})}}$
    - Minimizing perplexity is the same as maximizing probability
  - Bad approximation
    - unless test data looks just like training data
  - But is helpful to think about

Task 6 - Creative Exploratory



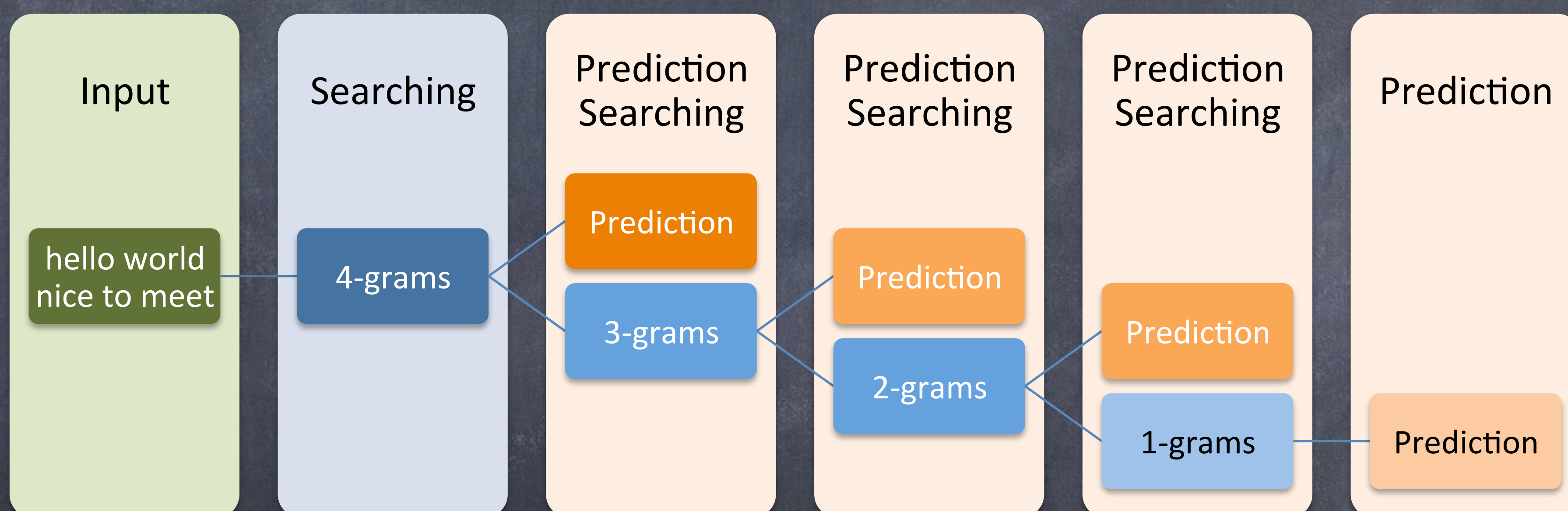
Creative Exploratory - Additive Smoothing

MLE estimate: $P_{MLE}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$

Add-1 estimate: $P_{Add-1}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i) + 1}{c(w_{i-1}) + V}$

- Add-One(Laplace) Smoothing
  - Pretend we saw each word one more time than we did
  - Too much probability mass is moved to all the zeros
- Add-k Smoothing
  - Similar to Add One, choosing k can be done on optimizing on a development dataset.
  - Generating counts with poor variances and often inappropriate discounts(Gale and Church,1994)

Creative Exploratory - Backoff



Creative Exploratory - Backoff

- Katz back-off model
  - Good Turing
    - reallocate the probability mass of n-grams that occur r + 1 times in the training data to the n-grams that occur r times

$$r^* = (r+1)\frac{n_{r+1}}{n_r} \longrightarrow r^* = (r+1)\frac{E[n_{r+1}]}{E[n_r]}$$

  - Katz Smoothing
    - discount ratio dr, which is approximately r*/r

$$c_{katz}(w_{i-1}^{i}) = \begin{cases} d_r r & \text{if } r > 0 \\ \alpha(w_{i-1}) p_{ML}(w_i) & \text{if } r = 0 \end{cases}$$

$$p_{katz}(w_i|w_{i-1}) = \frac{c_{katz}(w_{i-1}^{i})}{\sum_{w_i} c_{katz}(w_{i-1}^{i})}$$

Creative Exploratory - Backoff

- Stupid back-off model
    - State of Art smoothing uses variations of context-dependent back with the following scheme where ρ(·) are pre-computed and stored probabilities and λ(·) are back-off weights

$$P(w_i|w_{i-k+1}^{i-1}) = \begin{cases} \rho(w_{i-k+1}^{i}) & \text{if } (w_{i-k+1}^{i}) \text{ is found} \\ \lambda(w_{i-k+1}^{i-1})P(w_{i-k+2}^{i}) & \text{otherwise} \end{cases}$$

$$S(w_i|w_{i-k+1}^{i-1}) = \begin{cases} \dfrac{f(w_{i-k+1}^{i})}{f(w_{i-k+1}^{i-1})} & \text{if } f(w_{i-k+1}^{i}) > 0 \\ \alpha S(w_i|w_{i-k+2}^{i-1}) & \text{otherwise} \end{cases}$$

Creative Exploratory - Interpolation

- Simple Linear Interpolation
  - Contrast to backoff, we always mix the probability estimates from all the N-gram estimators, weighting and combining the trigram, bigram and unigram counts.

Simple Linear Interpolation:

$$q(w_i|w_{i-2},w_{i-1}) = \lambda_1 \times q_{ML}(w_i|w_{i-2},w_{i-1}) + \lambda_2 \times q_{ML}(w_i|w_{i-1}) + \lambda_3 \times q_{ML}(w_i)$$

$$\text{where } \lambda_1+\lambda_2+\lambda_3=1 \text{ and } \lambda_i \geq 0 \ \forall\, i$$

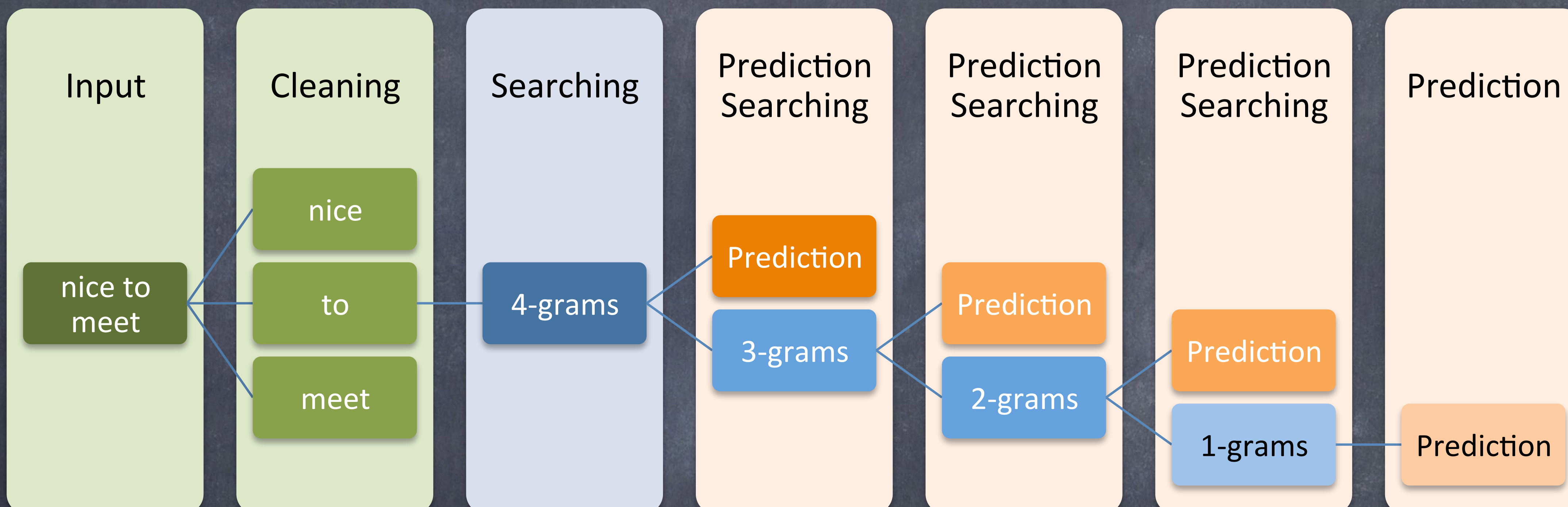
  - Choose λ values that maximize the likelihood of the held-out corpus

Task 7 - Data Product

Data Acquisition → Data processing → Exploratory Analysis → Building Model → Prediction & Evaluation → Creative Exploratory → **Data Product**

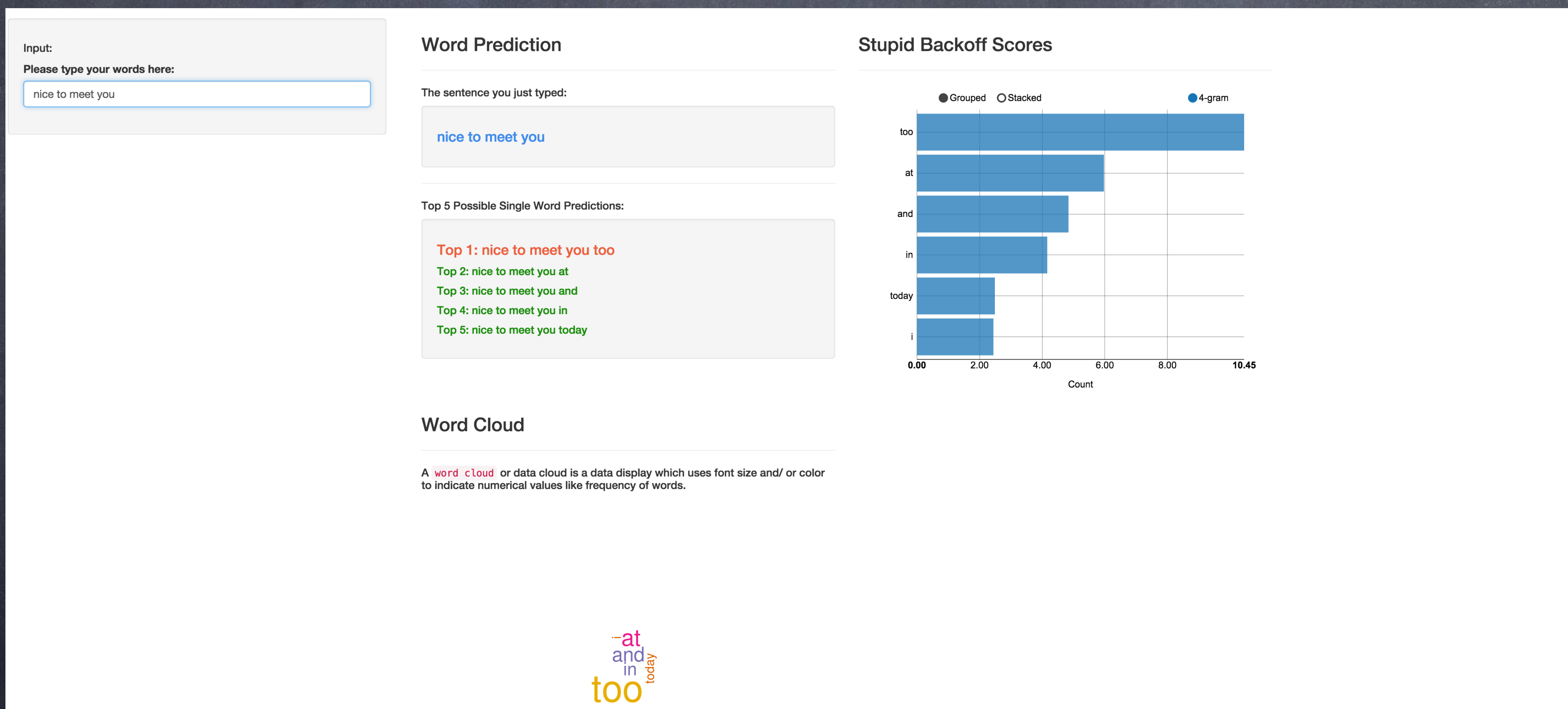
Data Acquisition | Data Processing | Exploratory Analysis | Building Model | Prediction & Evaluation | Creative Exploratory | Data Product

Data Product - shiny app



Data Acquisition | Data Processing | Exploratory Analysis | Building Model | Prediction & Evaluation | Creative Exploratory | Data Product

Data Product - shiny app



More…

- Customized tokenizer based on Stanford tokenizer.jar
- Using MapReduce for N-gram generation(Java,Pig)
- Using database to store N gram table
- More complicate models since N-gram can’t capture long range syntactic dependencies and semantic dependencies
- Profanity filtering

Thanks!

Qi Shao