N-gram Size in Auto-Suggestion

By Lauryn Fluellen

Main goal of work

1. Create N-gram model that suggests the next word given previous context

2. Investigate the performance of the model when you change the N-gram size

What are N-grams

1. A sequence of words or symbols in a text

2. A contiguous sequence of n items from a given text or speech

Ex.

Trigram:

“The quick brown frog jumped”

(the, quick, brown) (quick, brown, frog) (brown, frog, jumped)

The Task: N-gram Model for word suggestion

1. Words that appear in text are not independent of one another, but are related

2. Model predicts the next word in a sentence by considering the previous n-1 words

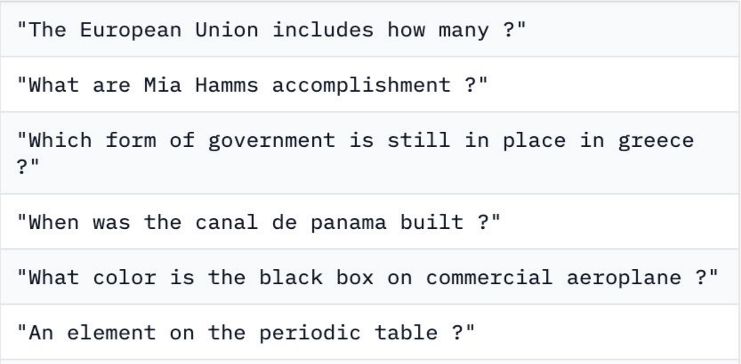
3. Predict the next word in a sentence by considering previous n-1 words

4. Suggest word based on the probability of words in training data appearing after previous words

Data

Training:

1. Dataset of 25K complete well formed Google queries
   a. Hugging Face - google_wellformed_query
   b. English



"The European Union includes how many ?"

"What are Mia Hamms accomplishment ?"

"Which form of government is still in place in greece ?"

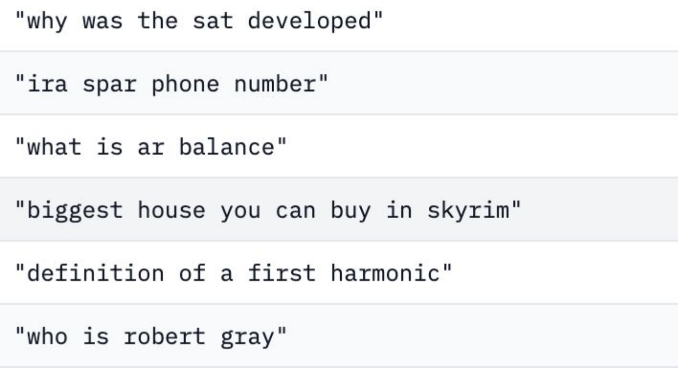
"When was the canal de panama built ?"

"What color is the black box on commercial aeroplane ?"

"An element on the periodic table ?"

Testing:

1. Dataset of 200 complete queries
   a. Hugging Face
   b. English



"why was the sat developed"

"ira spar phone number"

"what is ar balance"

"biggest house you can buy in skyrim"

"definition of a first harmonic"

"who is robert gray"

Models used

Compared 4 different n-gram models:

1. Unigram

2. Bigram

3. Trigram

4. n-gram with n size 4

How the model works

Full sentence **- “The quick brown frog jumped”**

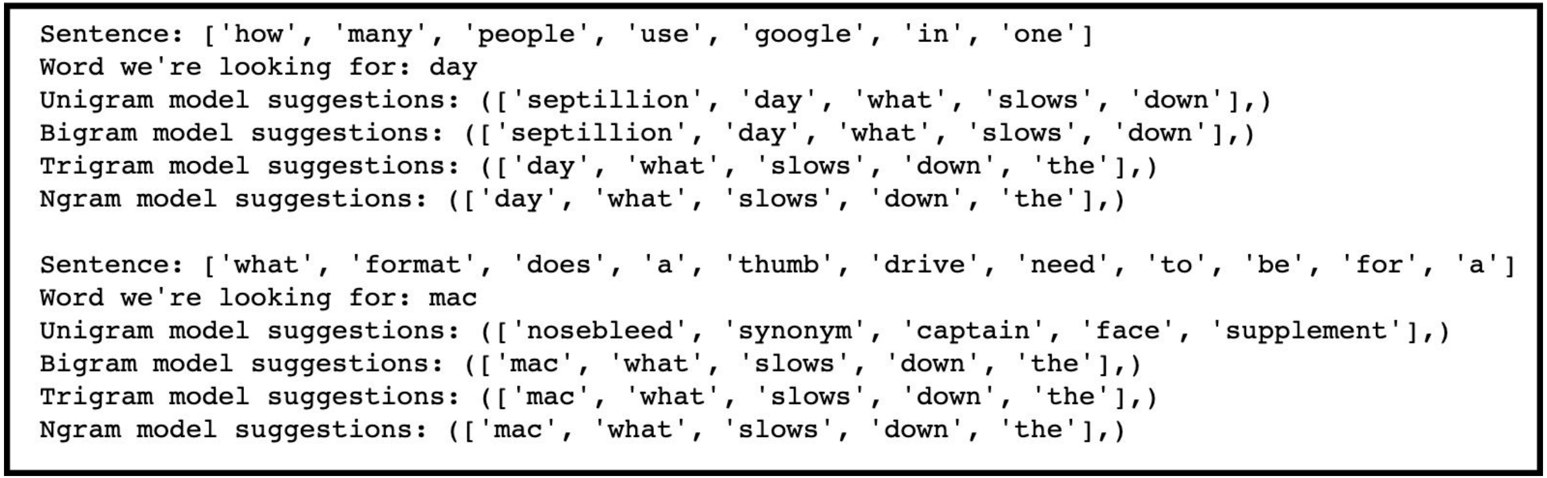
Sentence to predict **- “The quick brown frog”**

Target word for prediction **- “jumped”**

1. Unigram - **(the) (quick) (brown) (frog)**
2. Bigram - **(the, quick) (quick, brown) (brown, frog)**
3. Trigram - **(the, quick, brown) (quick, brown, frog)**
4. n-gram with n size 4 - **(the, quick, brown, frog)**

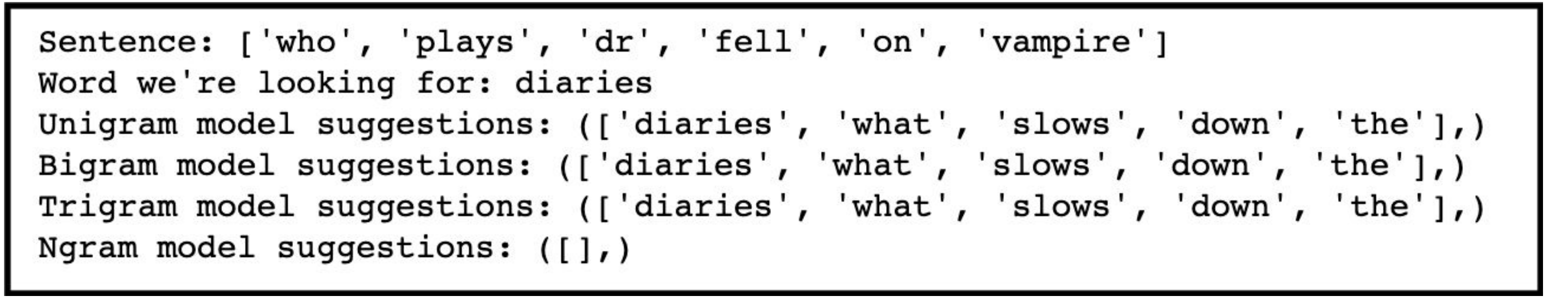
Use these n-grams as previous words to calculate the probability of words that would come after it to generate word predictions

Results



```
Sentence: ['how', 'many', 'people', 'use', 'google', 'in', 'one']
Word we're looking for: day
Unigram model suggestions: (['septillion', 'day', 'what', 'slows', 'down'],)
Bigram model suggestions: (['septillion', 'day', 'what', 'slows', 'down'],)
Trigram model suggestions: (['day', 'what', 'slows', 'down', 'the'],)
Ngram model suggestions: (['day', 'what', 'slows', 'down', 'the'],)

Sentence: ['what', 'format', 'does', 'a', 'thumb', 'drive', 'need', 'to', 'be', 'for', 'a']
Word we're looking for: mac
Unigram model suggestions: (['nosebleed', 'synonym', 'captain', 'face', 'supplement'],)
Bigram model suggestions: (['mac', 'what', 'slows', 'down', 'the'],)
Trigram model suggestions: (['mac', 'what', 'slows', 'down', 'the'],)
Ngram model suggestions: (['mac', 'what', 'slows', 'down', 'the'],)
```



```
Sentence: ['who', 'plays', 'dr', 'fell', 'on', 'vampire']
Word we're looking for: diaries
Unigram model suggestions: (['diaries', 'what', 'slows', 'down', 'the'],)
Bigram model suggestions: (['diaries', 'what', 'slows', 'down', 'the'],)
Trigram model suggestions: (['diaries', 'what', 'slows', 'down', 'the'],)
Ngram model suggestions: ([],)
```

Results Cont.

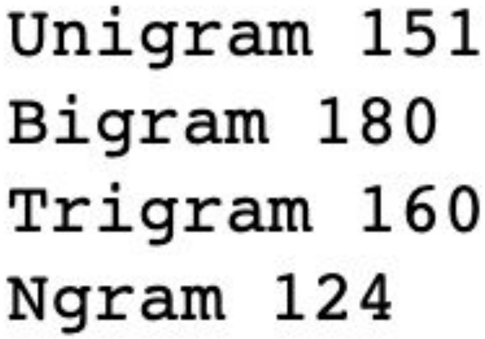
Which n-gram model performed the best?

1. From all results for each sentence predicted found which models the target word was suggested in

2. Kept a tally for each model when it found the target word

3. Whichever model had the most tallys was designated the best fit for the data because it was able to predict the words much more than the others

4. For the dataset used average the **Bigram** model performed the best



```
Unigram 151
Bigram 180
Trigram 160
Ngram 124
```

Limitations

1. Data sparsity

2. Limited Context

3. Overfitting

End