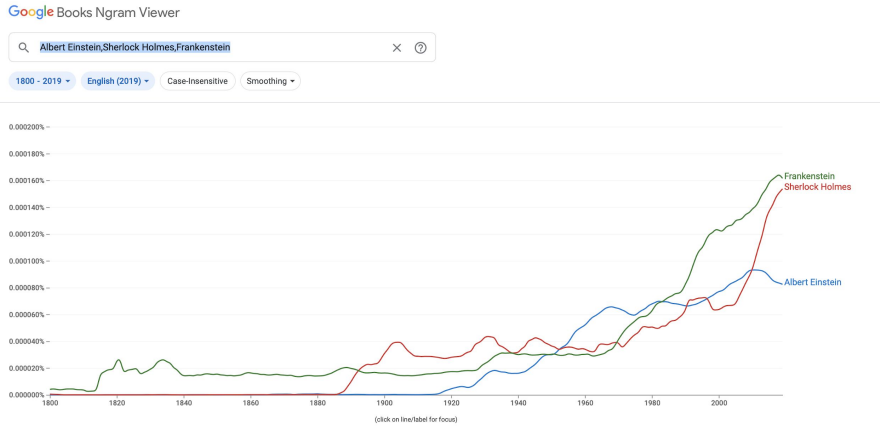
NGRAM Data Acquisition

1. Download small data set
   - Lacking performance
2. Download big data set
   a. ca. 9 TB to download = 270 GB after aggregation
   b. ca. 36 days processing time
3. Request Backend of Ngram Viewer
   a. Slow -> Caching
   b. Access to big data without download
   c. Simple switch between languages
   d. Vulnerable to changes by Google



Error Detection using N-GRAM Language Models

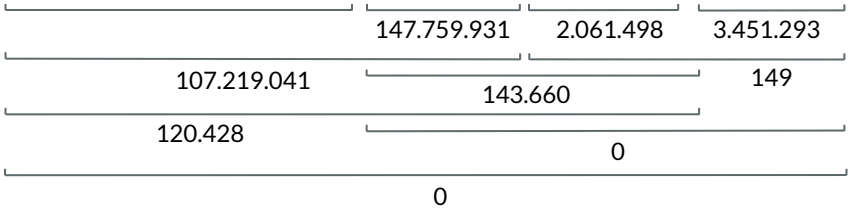
- n-gram models with up to 4-grams
- assumption: an error is detectable by the previous 3 words
- calculating an error value for every word if it is below a threshold mark it as error
- smoothing for unseen n-grams
- using “stupid backoff” a smoothing technique for huge language models
- according to the authors of the paper λ=0.4 works well

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

Brants, T., Popat, A. C., Xu, P., Och,F. J., and Dean, J. (2007). Large language models in machine translation

N-GRAM Language Model Example

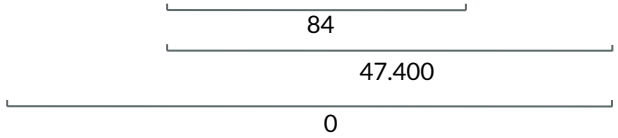
START The boy live on top of the hill.



$$S = \frac{\text{count(\_START\_ The boy live)}}{\text{count(\_START\_ The boy)}} => \frac{\lambda\ \text{count(The boy live)}}{\text{count(The boy)}} => \frac{\lambda\ \lambda\ \text{count(boy live)}}{\text{count(boy)}} => \frac{0.4 * 0.4 * 149}{2.061.498} \approx 0.0000115644$$

Smoothing Example

START The boy lives on top of the hill.



$$S = \frac{\text{count(boy lives on top)}}{\text{count(boy lives on)}} => \frac{\lambda\ \text{count(lives on top)}}{\text{count(lives on)}} => \frac{0.4 * 84}{47.400} \approx 0.0007148936$$

Error correction: The BERT Model

- Machine Learning model published 2018 by Google
- Transfer learning algorithm
- Mask Language Modelling:

Fill-Mask

Mask token: [MASK]

I like to play [MASK] when the weather is nice.

Compute

Computation time on cpu: 0.053 s

golf	0.098
football	0.077
hockey	0.040
here	0.039
tennis	0.037

Error correction: Implementation

- BERT Model from the NLP company Hugging Face
- Training data: 11.000 books and the complete English Wikipedia

Correction process:

1. Replace the wrong word by a mask token
2. Receive the twenty most likely suggestions from the BERT model
3. Evaluate the twenty suggestions using N-Grams and word distance
4. Show the four remaining suggestions to the user