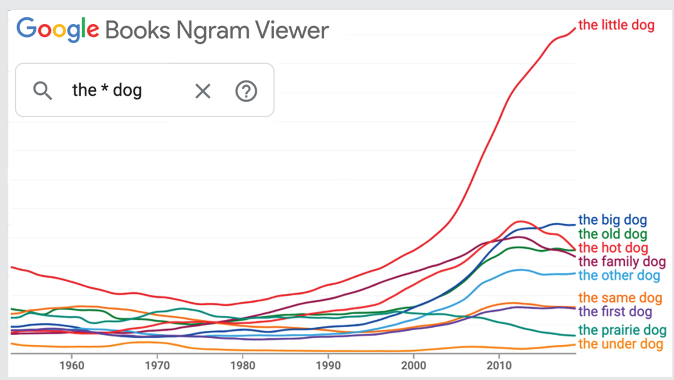
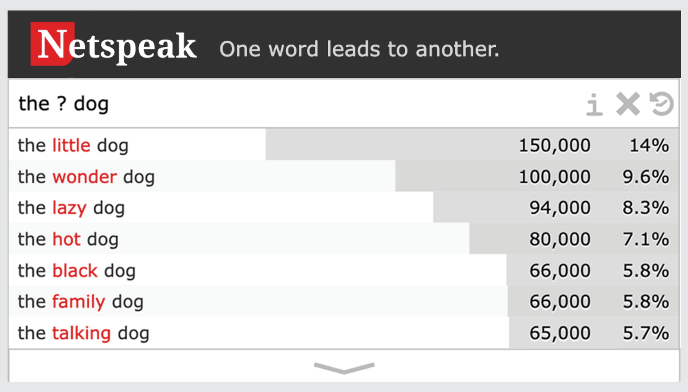
Matti Wiegmann, Michael Völske, Martin Potthast, Benno Stein

<firstname>.<lastname>@uni-weimar.de martin.potthast@uni-leipzig.de

Language Models as Context-sensitive Word Search Engines

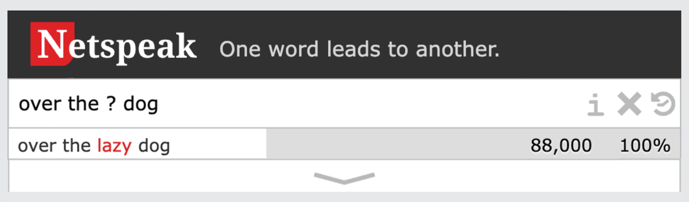
Context-sensitive word search engines retrieve words that match a given context



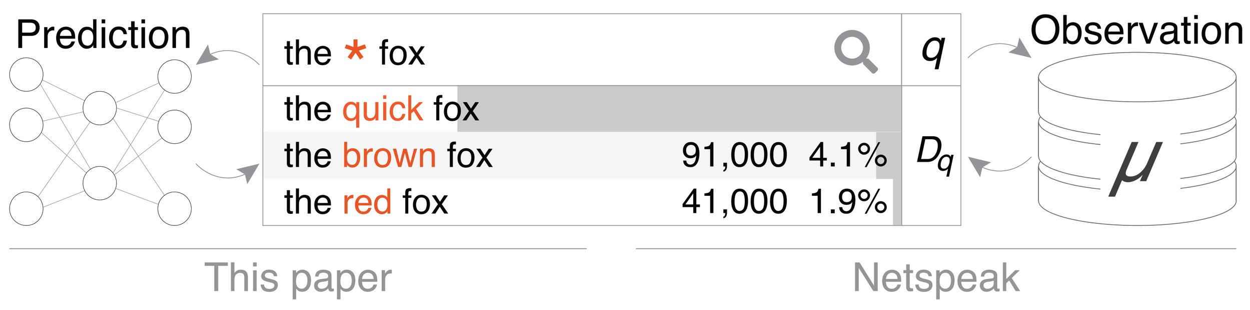


- They can answer wildcard queries $q = q_l \; ? \; q_r$
- They are usually build with n-gram collections

Problem: Increasing n requires exponential observations; We're limited to n <= 5



Contributions

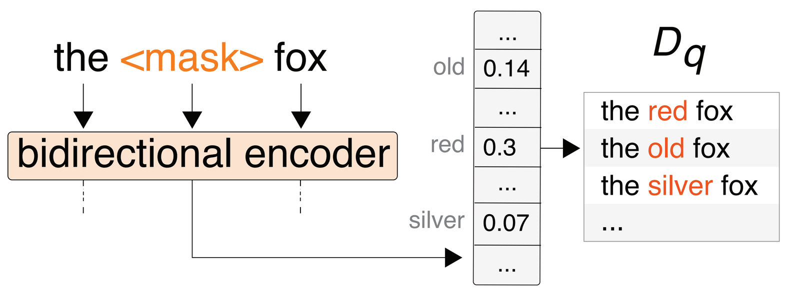


- Tune large language models to predict the answer of wildcard queries while preserving corpus characteristics
- Predict a list of plausible answers, ranked by their expected frequency and approximate this frequency

Language Modeling for Word Search

We propose two models strategies of using language models to predict word search results.

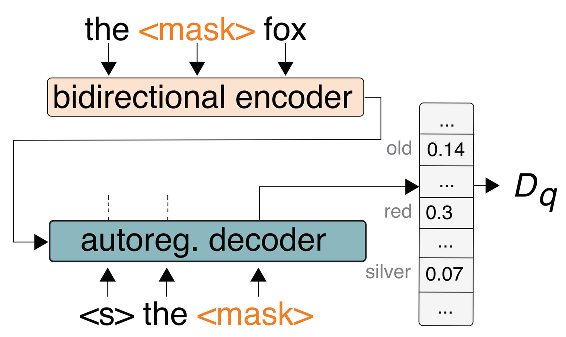
Method 1: Word search via masked language modeling (MLM)



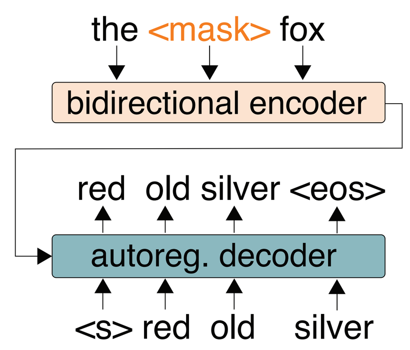
- Use a transformer encoder; We use DistillBert
- Pre-training is done via MLM on full sequences; fine-tuning is done on n-grams
- Result set is the sorted softmax output at the mask's position

Method 2: Word search via conditional language modeling (CDLM)

- Use a sequence2sequence transformer; We use BART
- Pre-training and Prediction is done via de-noising
- Result set is the sorted softmax output at the mask's position



- Fine-tuning the decoder is done by generating the result set of the query passed to the encoder



Evaluation

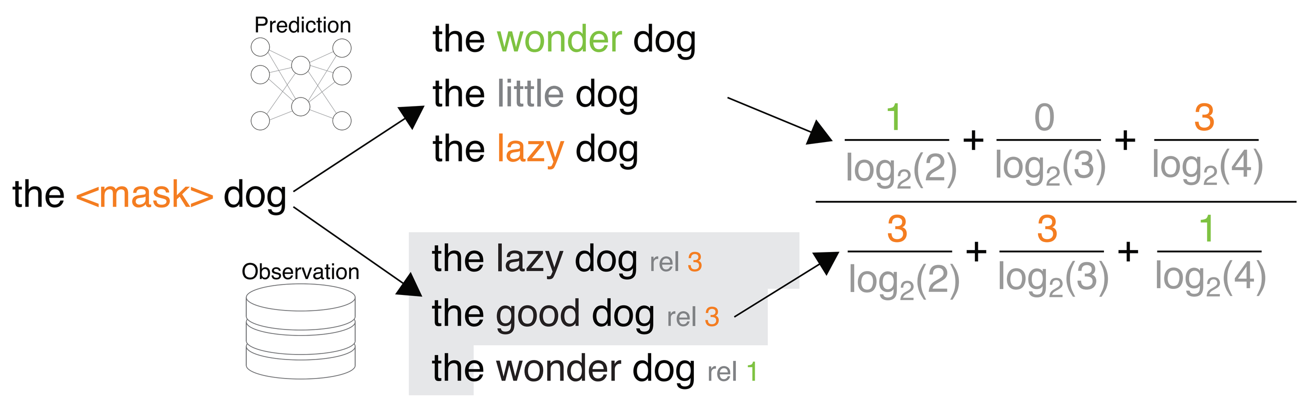
We compare both Methods, with and without fine-tuning, against Netspeak on two experiments

Data: 25 million wildcard queries from Wikitext and CLOTH

Experiment 1: The better model should assign, on average, a higher rank to a masked word

(I) For all n-grams, mask a random word to form a query
(II) Predict the results for the query
(III) Measure the *mean reciprocal rank (MRR)* of the masked token

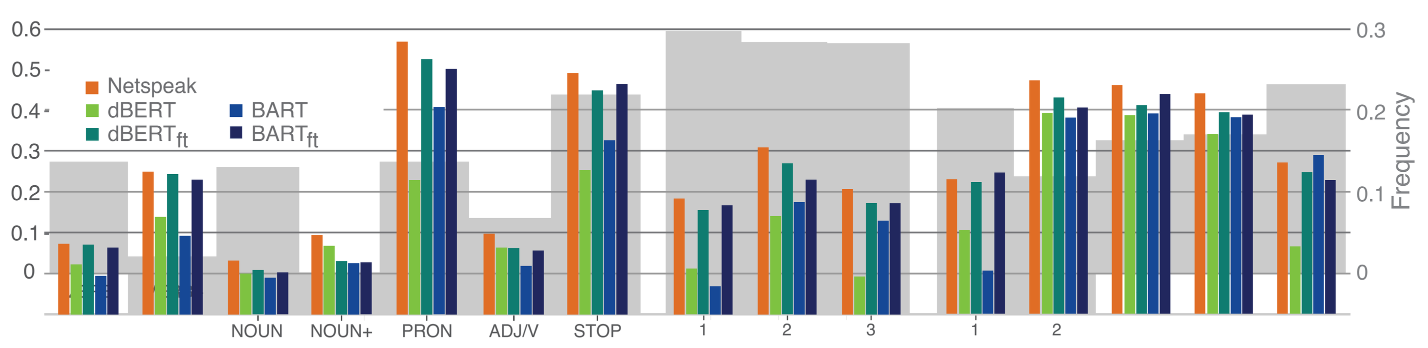
Experiment 2: The better model should predict the frequency-based ranking better.



(I) Get frequency based ranking and assign relevance scores
(II) Predict the results for the query
(III) Measure the *normalized discounted cumulative gain (nDCG)*

Results

- Finetuned models within 5 p.p. of Netspeak for queries with observable answers
- Finetuning doubles MRR and nDCG, depending on word class and wildcard position. No substantial difference between model types
- 80% of 5-gram queries have no obserable results:
  - Laguage models can answer, Netspeak can not;
  - Average MRR loss of 7 p.p. (20%)
- Runtime per Query: 5ms for BERT and Netspeak, 11 ms for BART



MRR and query frequency on Wikitextn by word class and mask position