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

G <mark>oogle</mark> Books	Ngram Vie	wer		/	the little dog
Q the * dog	× ?				
				/	
					the big dog
				\frown	the big dog the old dog the hot dog the family dog the other dog
				<	the same dog the first dog the prairie dog the under dog
1960 1970	1980	1990	2000	2010	the under dog

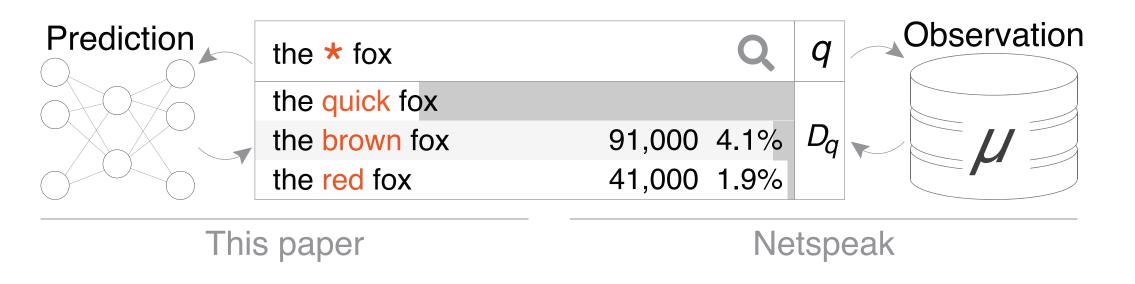
Netspeak	One word leads t	to another.		
the ? dog			i	ХЭ
the little dog			150,000	14%
the wonder dog			100,000	9.6%
the lazy dog			94,000	8.3%
the <mark>hot</mark> dog			80,000	7.1%
the <mark>black</mark> dog			66,000	5.8%
the family dog			66,000	5.8%
the talking dog			65,000	5.7%

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

Netspeak	One word leads to another.		
over the ? dog		i	ХЭ
over the lazy dog		88,000	100%

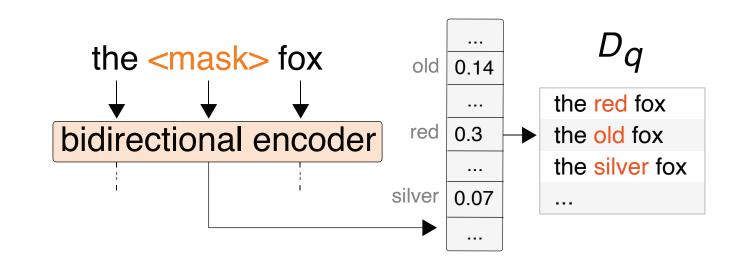
Contributions



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)



- 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

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

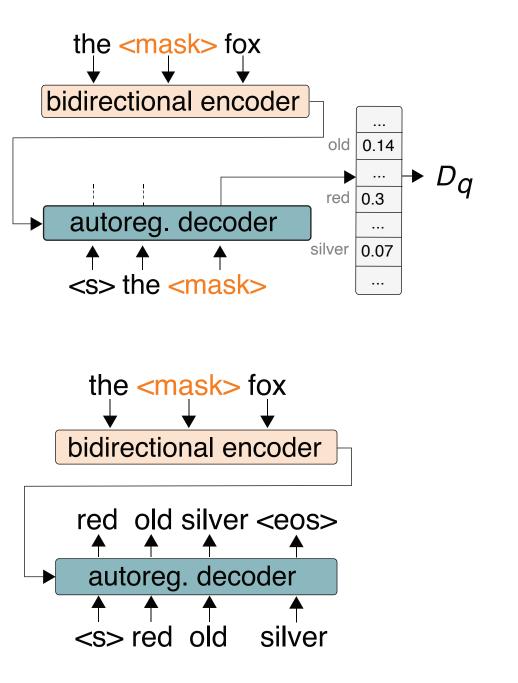
the little dog the lazy dog \longrightarrow the <mask> dog \longrightarrow the lazy dog $\longrightarrow \frac{1}{2}$ the wonder dog

- 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

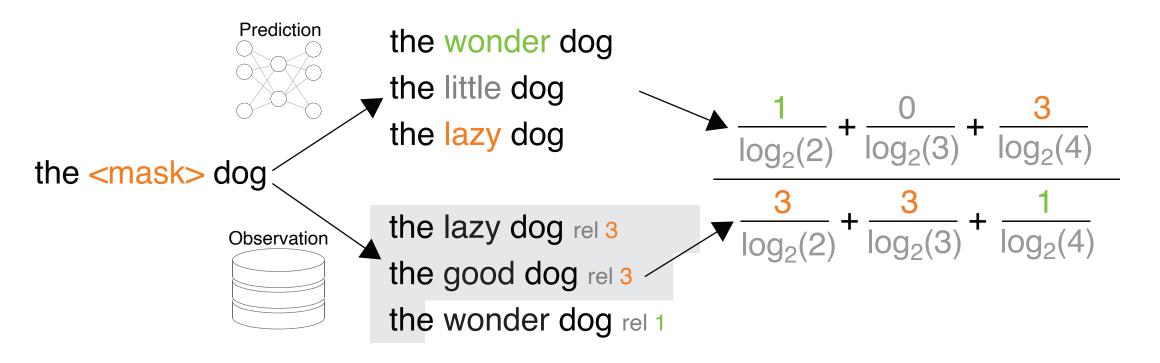
Fine-tuning the decoder is done
by generating the result set of the query passed to the encoder



(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.

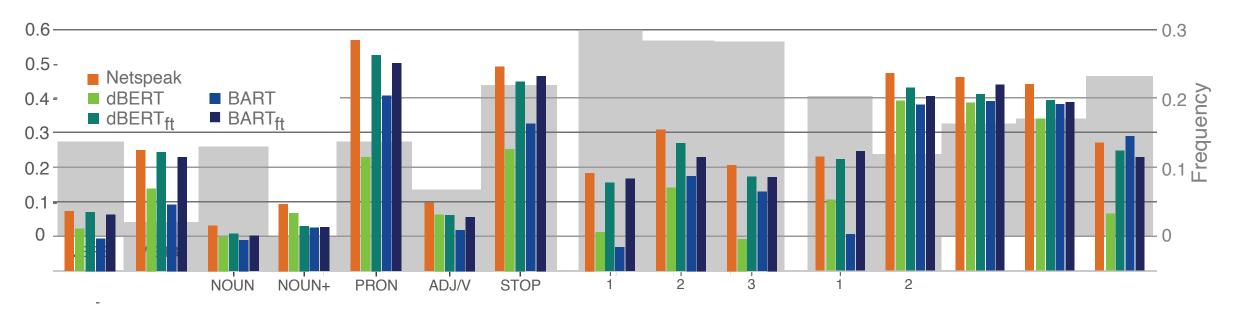


(I) Get frequency based ranking and assign relevance scores(II) Predict the results for the query

(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

Web Technology and Information Systems www.webis.de

Bauhaus-Universität Weimar