Language Models as Context-sensitive Word Search Engines

Matti Wiegmann
Michael Völske
Martin Potthast
Benno Stein

1 Bauhaus-Universität Weimar  2 Leipzig University

webis.de
Language Models as Context-sensitive Word Search Engines

Motivation

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

- Trivially: Thesauri, idiom collections, ...
Language Models as Context-sensitive Word Search Engines

Motivation

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

- Trivially: Thesauri, idiom collections, ...
Context-sensitive word search engines retrieve words that match a given context.

- Trivially: Thesauri, idiom collections, ...
- Context allows wildcard queries $q = q_l \ ? \ q_r$ and ranking.
Language Models as Context-sensitive Word Search Engines

Motivation

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

- Trivially: Thesauri, idiom collections, ...
- Context allows wildcard queries \( q = q_l \ ? \ q_r \) and ranking.
- Counting frequencies beats predictions and smoothing for word search.
Language Models as Context-sensitive Word Search Engines

Motivation

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

- **Trivially**: Thesauri, idiom collections, ...
- Context allows wildcard queries $q = q_l \ ? q_r$ and ranking.
- Counting frequencies beats predictions and smoothing for word search.
  - Context-sensitive word search engines are build on $n$-gram collections.
Language Models as Context-sensitive Word Search Engines

Motivation

Problem: Increasing $n$ requires exponential observations; We’re limited to $n \leq 5$.

→ Infer the answers to wildcard queries and their probabilities from a (large) language model.
Language Models as Context-sensitive Word Search Engines

Motivation

Problem: Increasing $n$ requires exponential observations; We’re limited to $n \leq 5$.

→ Infer the answers to wildcard queries and their probabilities from a (large) language model.

Contributions:

- Tune large language models to $n$-grams while preserving corpus characteristics and idioms.
- Predict the ranking with frequency.
Language Models as Context-sensitive Word Search Engines

Language Modeling for Word Search

Solving wildcard queries $q = q_l \, ? \, q_r$ with:

1. Masked Language Modeling
   We used DistillBERT

2. Conditional Language Modeling
   We used BART
Language Models as Context-sensitive Word Search Engines

Language Modeling for Word Search

Solving wildcard queries $q = q_l ? q_r$ with:

1. Masked Language Modeling
   We used DistillBERT

   Pretrain and Predict

2. Conditional Language Modeling
   We used BART
Language Models as Context-sensitive Word Search Engines

Language Modeling for Word Search

Solving wildcard queries $q = q_l \ ? \ q_r$ with:

1. **Masked Language Modeling**
   - We used DistillBERT
   - Diagram: The masked fox (old, red, silver) is pre-trained and predicted.

2. **Conditional Language Modeling**
   - We used BART
   - Diagram: The masked fox (old, red, silver) is pre-trained and predicted with autoregressive decoder.
Language Models as Context-sensitive Word Search Engines

Language Modeling for Word Search

Solving wildcard queries $q = q_l ? q_r$ with:

1. Masked Language Modeling
   We used DistillBERT

2. Conditional Language Modeling
   We used BART
Language Models as Context-sensitive Word Search Engines

Experimental Evaluation

- **Data**: 3 and 5-grams from Wikitext and CLOTH.
- **Models**: DistillBERT, BART, DistillBERT\textsubscript{ft}, BART\textsubscript{ft}, Netspeak.
- Experiment 1: Predict masked word; Measure position in the result set via MRR.
- Experiment 2: Predict the observable ranking. Measure nDCG. High frequency results have a higher relevance.
Language Models as Context-sensitive Word Search Engines

Experimental Evaluation

- **Data:** 3 and 5-grams from Wikitext and CLOTH.
- **Models:** DistillBERT, BART, DistillBERT$_{ft}$, BART$_{ft}$, Netspeak.
- **Experiment 1:** Predict masked word; Measure position in the result set via MRR.
- **Experiment 2:** Predict the observable ranking. Measure nDCG. High frequency results have a higher relevance.

```
the lazy dog  ➔  the <mask> dog  ➔  the lazy dog  ➔  \frac{1}{2}
the little dog
the wonder dog
```
Language Models as Context-sensitive Word Search Engines

**Experimental Evaluation**

- **Data:** 3 and 5-grams from Wikitext and CLOTH.
- **Models:** DistillBERT, BART, DistillBERT\_ft, BART\_ft, Netspeak.
- **Experiment 1:** Predict masked word; Measure position in the result set via MRR.
- **Experiment 2:** Predict the observable ranking. Measure nDCG. High frequency results have a higher relevance.
Language Models as Context-sensitive Word Search Engines

Results

Core 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 observable results:
  - → Language models can answer, Netspeak can not;
  - → Average MRR loss of 7 p.p.
- Runtime per Query: 5ms for BERT and Netspeak, 11 ms for BART
Core 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 observable results:
  - Language models can answer, Netspeak can not;
  - Average MRR loss of 7 p.p.
- Runtime per Query: 5ms for BERT and Netspeak, 11 ms for BART
Language Models as Context-sensitive Word Search Engines

Results

Core 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 observable results:
  → Language models can answer, Netspeak can not;
  → Average MRR loss of 7 p.p.
- Runtime per Query: 5ms for BERT and Netspeak, 11 ms for BART
Language Models as Context-sensitive Word Search Engines

Results

Core 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 observable results:
  → Language models can answer, Netspeak can not;
  → Average MRR loss of 7 p.p.
- Runtime per Query: 5ms for BERT and Netspeak, 11 ms for BART

www.netspeak.org/demo