Language Model Rest Costs and Space-Efficient Storage

Kenneth Heafield    Philipp Koehn    Alon Lavie

Carnegie Mellon, University of Edinburgh

July 14, 2012
Complaint About Language Models

Make Search Expensive

\[
\frac{p_5(\text{is one of the})}{p_5(\text{is one})p_5(\text{of the})} \neq 1
\]

1 Better fragment scores
Complaints About Language Models

Make Search Expensive

\[ \frac{p_5(\text{is one of the})}{p_5(\text{is one})p_5(\text{of the})} \neq 1 \]

Better fragment scores

Use Too Much Memory

\[ \log p_5(\text{the} | \text{is one of}) = -0.5 \]
\[ \log b_5(\text{is one of the}) = -1.2 \]

Collapse probability and backoff
Language Model Probability of Sentence Fragments

\[ \log p_5(\text{is one of the few}) = -6.62 \]

Why does it matter?
Decoders prune hypotheses based on score.
Baseline: How to Score a Fragment

\[
\begin{align*}
\log p_5(\text{is}) &= -2.63 \\
\log p_5(\text{one} \mid \text{is}) &= -2.03 \\
\log p_5(\text{of} \mid \text{is one}) &= -0.24 \\
\log p_5(\text{the} \mid \text{is one of}) &= -0.47 \\
+ \log p_5(\text{few} \mid \text{is one of the}) &= -1.26 \\
\hline
= \log p_5(\text{is one of the few}) &= -6.62
\end{align*}
\]
The Problem: Lower Order Entries

5-Gram Model: \( \log p_5(\text{is}) = -2.63 \)

Unigram Model: \( \log p_1(\text{is}) = -2.30 \)

Same training data.
Backoff Smoothing

\( p_5(\text{is}) \) should be used when a bigram was not found.

**In the language model**

\[
\log p_5(\text{is} \mid \text{australia}) = -2.21
\]

**Not in the language model**

\[
\log p_5(\text{is} \mid \text{periwinkle}) = \log b_5(\text{periwinkle}) + \log p_5(\text{is}) = -2.95
\]
$p_5(\text{is})$ should be used when a bigram was not found.

In the language model

$$\log p_5(\text{is} \mid \text{australia}) = -2.21$$

Not in the language model

$$\log p_5(\text{is} \mid \text{periwinkle}) = \log b_5(\text{periwinkle}) + \log p_5(\text{is}) = -2.95$$

In Kneser-Ney smoothing, lower order probabilities assume backoff.
Use Lower Order Models for the First Few Words

| Term                                      | Baseline       | Lower          |
|-------------------------------------------|----------------|----------------|
| $\log p_5(\text{is})$                    | $-2.63$        | $-2.30 = \log p_1$ |
| $\log p_5(\text{one} \mid \text{is})$   | $-2.03$        | $-1.92 = \log p_2$ |
| $\log p_5(\text{of} \mid \text{is one})$| $-0.24$        | $-0.08 = \log p_3$ |
| $\log p_5(\text{the} \mid \text{is one of})$ | $-0.47$        | $-0.21 = \log p_4$ |
| $\log p_5(\text{few} \mid \text{is one of the})$ | $-1.26$        | $-1.26 = \log p_5$ |
| $\log p_5(\text{is one of the few})$     | $-6.62$        | $-5.77 = \log p_{\text{Low}}$ |
Which is Better?

Baseline: \( \log p_5(\text{is one of the few}) \)
Lower Order: \( \log p_{\text{Low}}(\text{is one of the few}) \)

\[
\begin{align*}
\text{Baseline:} & \quad \log p_5(\text{is one of the few}) &= -6.62 \\
\text{Lower Order:} & \quad \log p_{\text{Low}}(\text{is one of the few}) &= -5.77
\end{align*}
\]
Which is Better: Prediction Task

Baseline: $\log p_5(\text{is one of the few}) = -6.62$ 

Lower Order: $\log p_{\text{Low}}(\text{is one of the few}) = -5.77$

Actual: $\log p_5(\text{is one of the few } | \text{ <s> australia}) = -4.10$

Error $-2.52$ $-1.67$
The Lower Order Estimate is Better

Run the decoder and log error every time context is revealed.

| Length | 1   | 2   | 3   | 4   |
|--------|-----|-----|-----|-----|
| Baseline | .87 | .24 | .10 | .09 |
| Lower Order | .84 | .18 | .07 | .04 |

**Table**: Mean squared error in predicting log probability.
Storing Lower Order Models

One extra float per entry, except for longest order.

**Unigrams**

| Words  | $\log p_5$ | $\log b_5$ | $\log p_1$ |
|--------|-------------|-------------|-------------|
| australia | -3.9        | -0.6        | -3.6        |
| is      | -2.6        | -1.5        | -2.3        |
| one     | -3.4        | -1.0        | -2.9        |
| of      | -2.5        | -1.1        | -1.7        |

No need for backoff $b_1$

If backoff occurs, the Kneser-Ney assumption holds and $p_5$ is used.
Fragment scores are more accurate, but require more memory.
Score with and without sentence boundaries. [Sankaran et al, 2012]

Peek at future phrases. [Zens and Ney, 2008] [Wuebker et al, Wed.]

Coarse pass predicts scores for a finer pass. [Vilar and Ney, 2011]
Score with and without sentence boundaries. [Sankaran et al, 2012]
Peek at future phrases. [Zens and Ney, 2008] [Wuebker et al, Wed.]
Coarse pass predicts scores for a finer pass. [Vilar and Ney, 2011]

All of these use fragment scores as a subroutine.
Related Work II: Carter et al, Yesterday

This Work

\[ p(\text{is one of the}) \approx p(\text{is one})p(\text{of the}) \]

Their Work

\[ p(\text{is one of the}) \leq p(\text{is one})p(\text{of the}) \]

Implementing Upper Bounds Within This Work

- Store upper bound probabilities instead of averages
- Account for positive backoff with the context

Three values per \( n \)-gram instead of their four.
Lower Order Summary

Previously
Fragment scores are more accurate, but require more memory.

Next
Save memory but make fragment scores less accurate.
### Saving Memory

| Words    | \( \log p_5 \) | \( \log b_5 \) | \( \log q_5 \) |
|----------|----------------|----------------|-----------------|
| australia | -3.9           | -0.6           | -4.5            |
| is       | -2.6           | -1.5           | -4.1            |
| one      | -3.4           | -1.0           | -4.4            |
| of       | -2.5           | -1.1           | -3.6            |

One less float per entry, except for longest order.
Related Work

Store counts instead of probability and backoff [Brants et al, 2007] RandLM, ShefLM, BerkeleyLM

This Work
- Memory comparable to storing counts.
- Higher quality Kneser-Ney smoothing.
How Backoff Works

\[ p(\text{periwinkle} \mid \text{is one of}) = p(\text{periwinkle} \mid \text{of}) b(\text{is one of}) b(\text{one of}) \]

because “of periwinkle” appears but “one of periwinkle” does not.
Assume backoff all the way to unigrams.

\[ q(\text{is one of}) = p(\text{is one of})b(\text{is one of})b(\text{one of})b(\text{of}) \]
Assume backoff all the way to unigrams.

\[ q \text{(is one of)} = p \text{(is one of)} b \text{(is one of)} b \text{(one of)} b \text{(of)} \]

Sentence Scores Are Unchanged

\[ q(<s> \cdots </s>) = p(<s> \cdots </s>) \]

because \( b(\cdots </s>) = 1 \)
Incremental Pessimism

\[ q(is) = p(is)b(is) \]

\[ q(one \mid is) = p(one \mid is) \frac{b(is \ one)b(one)}{b(is)} \]

These are terms in a telescoping series:

\[ q(is \ one) = q(is)q(one \mid is) \]
Using $q$

\[
\begin{align*}
\log q(\text{is}) &= -4.10 \\
\log q(\text{one } | \text{ is}) &= -2.51 \\
\log q(\text{of } | \text{ is one}) &= -0.94 \\
\log q(\text{the } | \text{ is one of}) &= -1.61 \\
+ \log q(\text{few } | \text{ is one of the}) &= 1.03 \\
= \log q(\text{is one of the few}) &= -8.13
\end{align*}
\]

Store $q$, forget probability and backoff.
Using $q$

\[
\begin{align*}
\log q(\text{is}) & = -4.10 \\
\log q(\text{one } | \text{ is}) & = -2.51 \\
\log q(\text{of } | \text{ is one}) & = -0.94 \\
\log q(\text{the } | \text{ is one of}) & = -1.61 \\
+ \log q(\text{few } | \text{ is one of the}) & = 1.03 \\
\hline
= \log q(\text{is one of the few}) & = -8.13
\end{align*}
\]

Store $q$, forget probability and backoff.

$q$ is not a proper probability distribution.
Collapse probability and backoff from two values to one value.
Stacking

Lower Order and Pessimistic Combined

- Same memory (one extra float, one less float).
- Better on the left, worse on the right.
Cube Pruning: Approximate Search

For each constituent, going bottom-up:

1. Make a priority queue over possible rule applications.
2. Pop a fixed number of hypotheses: the pop limit.

Larger pop limit $\implies$ more accurate search.
Cube Pruning: Approximate Search

For each constituent, going bottom-up:

1. Make a priority queue over possible rule applications.
2. Pop a fixed number of hypotheses: the pop limit.

Larger pop limit
Accurate fragment scores $\implies$ more accurate search.
Experiments

Task  WMT 2011 German-English
Decoder  Moses
LM  5-gram from Europarl, news commentary, and news
Grammar  Hierarchical and target-syntax systems
Parser  Collins
Hierarchical Model Score and BLEU

![Graph showing Hierarchical Model Score and BLEU](image)

- Average model score
- CPU seconds/sentence
- Uncased BLEU
- Lower
- Combined
- Baseline
- Pessimistic
Target-Syntax Model Score and BLEU

![Graphs showing the relationship between CPU seconds per sentence and model scores or BLEU scores. The graphs display different baseline and pessimistic models, with lines representing Lower and Combined scores.](image-url)
Memory

Cost to add or savings from removing a float per entry.

| Structure                  | Baseline (MB) | Change (MB) | %   |
|----------------------------|---------------|-------------|-----|
| Probing                    | 4,072         | 517         | 13% |
| Trie                       | 2,647         | 506         | 19% |
| 8-bit quantized trie       | 1,236         | 140         | 11% |
| 8-bit minimal perfect hash | 540           | 140         | 26% |
Summary

- **Lower Order Models**
  - 21-63% less CPU
  - 13-26% more memory

- **Pessimistic Backoff**
  - 27% more CPU
  - 13-26% less memory

- **Lower Order + Pessimistic**
  - 3% less CPU
  - Same memory as baseline
Code

kheafield.com/code/kenlm
Also distributed with Moses and cdec.

Lower Order
build_binary -r “1.arpa 2.arpa 3.arpa 4.arpa” 5.arpa 5.binary

Pessimistic Backoff
Release planned