Large Language Models

Madeline Remse and Sabrina Stehwien
Institute of Computational Linguistics
Heidelberg University

Winter term 2011/12
Software Project

A language model contains conditional word probabilities and can be used to assign probabilities to target language sentences as part of a statistical machine translation task. Usually the word probabilities are based on relative n-gram frequencies obtained from a corpus.
By using an extraordinarily large corpus we manage to create a language model which obtains particularly good results in open-domain statistical machine translation.

Based on: Brants et al. (2007)
Large Language Models in Machine Translation [1]
Basic idea:

- 2 trillion token training corpus
- up to 300 billion n-grams
- MapReduce implementation
- Stupid Backoff as a surprisingly successful smoothing method

Implementation

MapReduce:

- Realization in Java
- Hadoop framework (vers. 0.20.2, CL-Cluster)
- Large amounts of data are distributed and computed independently *(Map)*
- Map-results are combined to different reducers
- Combined results are joined and saved *(Reduce)*

Web1t corpus:

- 87gb of prepared English data
- 1-5-grams
- Source: World Wide Web – open domain

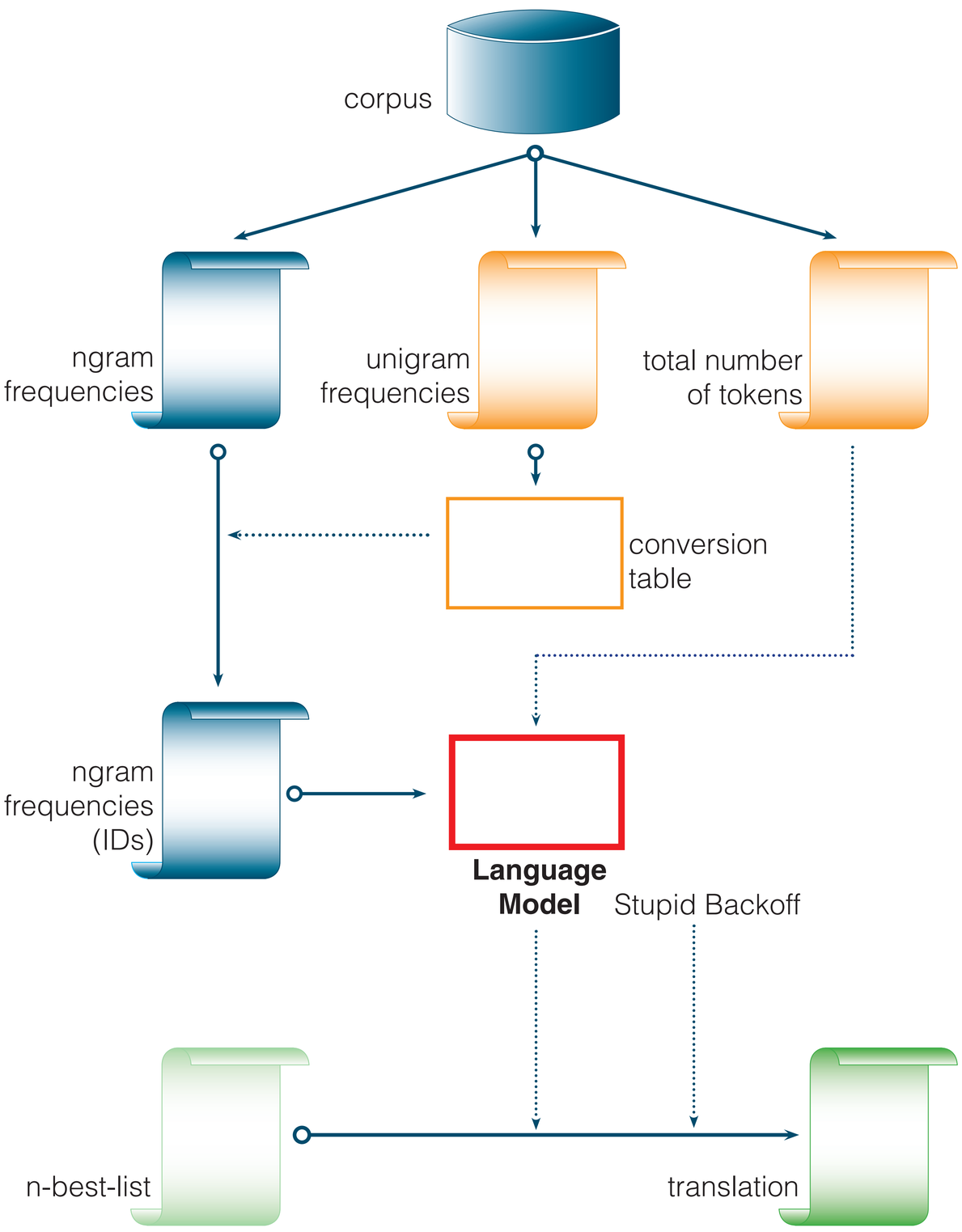
Stupid Backoff:

- Usually the probability of a string of tokens is computed as follows:

$$P(w_1^L) = \prod_{i=1}^{L} P(w_i|w_1^{i-1}) \approx \prod_{i=1}^{L} \hat{P}(w_i|w_{i-n+1}^{i-1}) \quad [1]$$

- If an n-gram is not part of the language model, its probability is 0, so the probability of the whole string of tokens becomes 0 as well
- There are several smoothing methods to solve the sparse data problem
- Stupid Backoff assigns scores to n-grams instead of probabilities

$$S(w_i|w_{i-k+1}^{i-1}) = \begin{cases} \frac{f(w_{i-k+1}^{i})}{f(w_{i-k+1}^{i-1})} & \text{if } f(w_{i-k+1}^{i}) > 0 \\ \alpha S(w_i|w_{i-k+2}^{i-1}) & \text{otherwise} \end{cases} \quad [1]$$



Evaluation

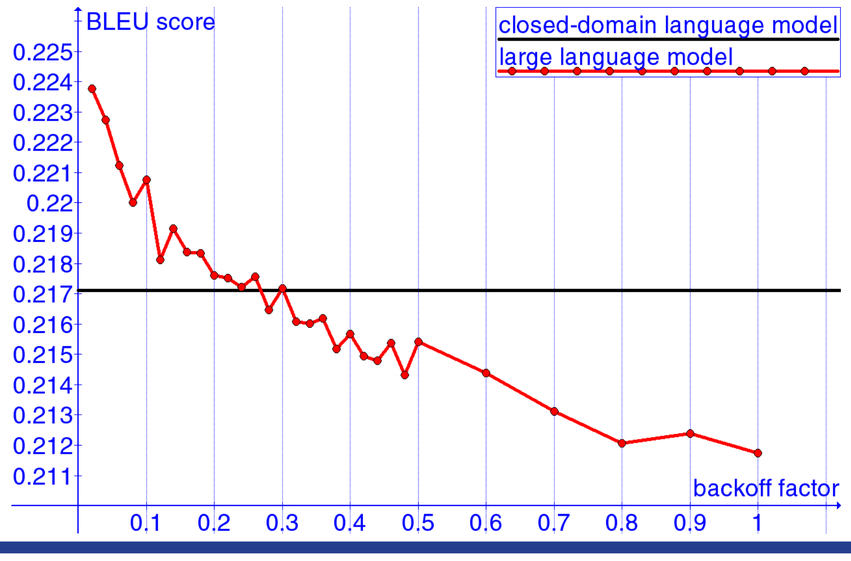
Method:

- Rerank N-best lists
- Generate N-best lists from a 2000 sentence test set with a machine translation system
- Use language model and Stupid Backoff to score sentences
- Extract final translation and compare BLEU scores

Machine Translation System [2]**:**

- Cdec Decoder
- Language model: SRILM
- Aligner: Berkeley Aligner
- Training and development corpus: Europarl (de-en)
- Decoder output: 100 best translation suggestions for each of the sentences in the development test set

Results:



Issues concerning Parallel Programming:

Converting the corpus to IDs:

To handle the large amount of ngrams that we want to convert to IDs, we have to process this step in the Hadoop framework. In the Hadoop framework data is distributed automatically between different mappers, so we cannot use the conversion table as a usual input file – that way it would not be available to every mapper. To make the conversion table available to all the data it has to be put into the *distributed cache*.

Calculate a complete language model:

The probability of a given n-gram is dependent on its history, but since the data is distributed between different mappers, we cannot guarantee that the n-gram histories we need are always available – the language model may be incomplete. So instead of immediately calculating conditional probabilities we wait until the reranking step and collect absolute frequencies. Then we can calculate conditional probabilities on the fly.

[1] Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, Jeffrey Dean. Large Language Models in Machine Translation. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. pp. 858-867, Prague, June 2007.
[2] http://cdec-decoder.org – http://www.speech.sri.com/projects/srilm – http://code.google.com/p/berkeleyaligner