Improved Language Modeling for Statistical Machine Translation

Katrin Kirchhoff and Mei Yang
Department of Electrical Engineering
University of Washington, Seattle, WA, 98195
{katrin,yangmei}@ee.washington.edu

Abstract

Statistical machine translation systems use a combination of one or more translation models and a language model. While there is a significant body of research addressing the improvement of translation models, the problem of optimizing language models for a specific translation task has not received much attention. Typically, standard word trigram models are used as an out-of-the-box component in a statistical machine translation system. In this paper we apply language modeling techniques that have proved beneficial in automatic speech recognition to the ACL05 machine translation shared data task and demonstrate improvements over a baseline system with a standard language model.

1 Introduction

Statistical machine translation (SMT) makes use of a noisy channel model where a sentence $\bar{e}$ in the desired language can be conceived of as originating as a sentence $\bar{f}$ in a source language. The goal is to find, for every input utterance $\bar{f}$, the best hypothesis $\bar{e}^*$ such that

$$\bar{e}^* = argmax_{\bar{e}} P(\bar{e}|\bar{f}) = argmax_{\bar{e}} P(\bar{f}|\bar{e}) P(\bar{e}) \tag{1}$$

$P(\bar{f}|\bar{e})$ is the translation model expressing probabilistic constraints on the association of source and target strings. $P(\bar{e})$ is a language model specifying the probability of target language strings. Usually, a standard word trigram model of the form

$$P(e_1, ..., e_l) \approx \prod_{i=3}^{l} P(e_i|e_{i-1}, e_{i-2}) \tag{2}$$

is used, where $\bar{e} = e_1, ..., e_l$. Each word is predicted based on a history of two preceding words.

Most work in SMT has concentrated on developing better translation models, decoding algorithms, or minimum error rate training for SMT. Comparatively little effort has been spent on language modeling for machine translation. In other fields, particularly in automatic speech recognition (ASR), there exists a large body of work on statistical language modeling, addressing e.g. the use of word classes, language model adaptation, or alternative probability estimation techniques. The goal of this study was to use some of the language modeling techniques that have proved beneficial for ASR in the past and to investigate whether they transfer to statistical machine translation. In particular, this includes language models that make use of morphological and part-of-speech information, so-called factored language models.

2 Factored Language Models

A factored language model (FLM) (Bilmes and Kirchhoff, 2003) is based on a representation of words as feature vectors and can utilize a variety of additional information sources in addition to words, such as part-of-speech (POS) information, morphological information, or semantic features, in a unified and principled framework. Assuming that each

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 125–128,
Ann Arbor, June 2005. ©Association for Computational Linguistics, 2005

word w can be decomposed into k features, i.e. $w \equiv f^{1:K}$, a trigram model can be defined as

$$p(f_1^{1:K}, f_2^{1:K}, ..., f_T^{1:K}) \approx \prod_{t=3}^{T} p(f_t^{1:K} | f_{t-1}^{1:K}, f_{t-2}^{1:K}) \tag{3}$$

Each word is dependent not only on a single stream of temporally preceding words, but also on additional parallel streams of features. This representation can be used to provide more robust probability estimates when a particular word n-gram has not been observed in the training data but its corresponding feature combinations (e.g. stem or tag trigrams) has been observed. FLMs are therefore designed to exploit sparse training data more effectively. However, even when a sufficient amount of training data is available, a language model utilizing morphological and POS information may bias the system towards selecting more fluent translations, by boosting the score of hypotheses with e.g. frequent POS combinations. In FLMs, word feature information is integrated via a new *generalized parallel backoff* technique. In standard Katz-style backoff, the maximum-likelihood estimate of an n-gram with too few observations in the training data is replaced with a probability derived from the lower-order $(n-1)$-gram and a backoff weight as follows:

$$\begin{aligned} & p_{BO}(w_t | w_{t-1}, w_{t-2}) \qquad (4) \\ &= \begin{cases} d_c p_{ML}(w_t | w_{t-1}, w_{t-2}) \text{ if } c > \tau \\ \alpha(w_{t-1}, w_{t-2}) p_{BO}(w_t | w_{t-1}) \text{ otherwise} \end{cases} \end{aligned}$$

where c is the count of (w_t, w_{t-1}, w_{t-2}), p_{ML} denotes the maximum-likelihood estimate, τ is a count threshold, d_c is a discounting factor and $\alpha(w_{t-1}, w_{t-2})$ is a normalization factor. During standard backoff, the most distant conditioning variable (in this case w_{t-2}) is dropped first, followed by the second most distant variable etc., until the unigram is reached. This can be visualized as a backoff *path* (Figure 1(a)). If additional conditioning variables are used which do not form a temporal sequence, it is not immediately obvious in which order they should be eliminated. In this case, several backoff paths are possible, which can be summarized in a backoff *graph* (Figure 1(b)). Paths in this graph can be chosen in advance based on linguistic knowledge, or at run-time based on statistical criteria such as counts in the training set. It is also possible to choose multiple paths and combine their probability estimates. This is achieved by replacing the backed-off probability p_{BO} in Equation 2 by a general function g, which can be any non-negative function applied to the counts of the lower-order n-gram. Several different g functions can be chosen, e.g. the mean, weighted mean, product, minimum or maximum of the smoothed probability distributions over all subsets of conditioning factors. In addition to different choices for g, different discounting parameters can be selected at different levels in the backoff graph. One difficulty in training FLMs is the choice of the best combination of conditioning factors, backoff path(s) and smoothing options. Since the space of different combinations is too large to be searched exhaustively, we use a guided search procedure based on Genetic Algorithms (Duh and Kirchhoff, 2004), which optimizes the FLM structure with respect to the desired criterion. In ASR, this is usually the perplexity of the language model on a held-out dataset; here, we use the BLEU scores of the oracle 1-best hypotheses on the development set, as described below. FLMs have previously shown significant improvements in perplexity and word error rate on several ASR tasks (e.g. (Vergyri et al., 2004)).

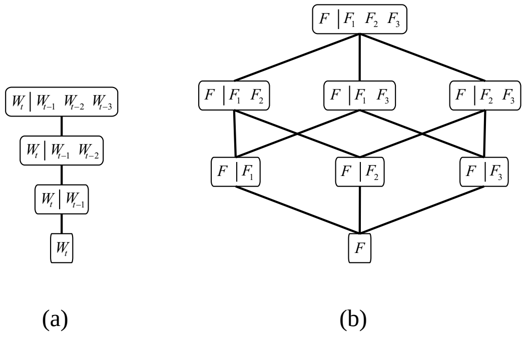

Figure 1: Standard backoff path for a 4-gram language model over words (left) and backoff graph over word features (right).

3 Baseline System

We used a fairly simple baseline system trained using standard tools, i.e. GIZA++ (Och and Ney, 2000) for training word alignments and Pharaoh (Koehn, 2004) for phrase-based decoding. The training data was that provided on the ACL05 Shared MT task website for 4 different language pairs (translation from Finnish, Spanish, French into English); no additional data was used. Preprocessing consisted of lowercasing the data and filtering out sentences with a length ratio greater than 9. The total number of training sentences and words per language pair ranged between 11.3M words (Finnish-English) and 15.7M words (Spanish-English). The development data consisted of the development sets provided on the website (2000 sentences each). We trained our own word alignments, phrase table, language model, and model combination weights. The language model was a trigram model trained using the SRILM toolkit, with modified Kneser-Ney smoothing and interpolation of higher- and lower-order ngrams. Combination weights were trained using the minimum error weight optimization procedure provided by Pharaoh. We use a two-pass decoding approach: in the first pass, Pharaoh is run in N-best mode to produce N-best lists with 2000 hypotheses per sentence. Seven different component model scores are collected from the outputs, including the distortion model score, the first-pass language model score, word and phrase penalties, and bidirectional phrase and word translation scores, as used in Pharaoh (Koehn, 2004). In the second pass, the N-best lists are rescored with additional language models. The resulting scores are then combined with the above scores in a log-linear fashion. The combination weights are optimized on the development set to maximize the BLEU score. The weighted combined scores are then used to select the final 1-best hypothesis. The individual rescoring steps are described in more detail below.

4 Language Models

We trained two additional language models to be used in the second pass, one word-based 4-gram model, and a factored trigram model. Both were trained on the same training set as the baseline system. The 4-gram model uses modified Kneser-Ney smoothing and interpolation of higher-order and lower-order n-gram probabilities. The potential advantage of this model is that it models n-grams up to length 4; since the BLEU score is a combination of n-gram precision scores up to length 4, the integration of a 4-gram language model might yield better results. Note that this can only be done in a rescoring framework since the first-pass decoder can only use a trigram language model.

For the factored language models, a feature-based word representation was obtained by tagging the text with Rathnaparki's maximum-entropy tagger (Ratnaparkhi, 1996) and by stemming words using the Porter stemmer (Porter, 1980). Thus, the factored language models use two additional features per word. A word history of up to 2 was considered (3-gram FLMs). Rather than optimizing the FLMs on the development set references, they were optimized to achieve a low perplexity on the oracle 1-best hypotheses (the hypotheses with the best individual BLEU scores) from the first decoding pass. This is done to avoid optimizing the model on word combinations that might never be hypothesized by the first-pass decoder, and to bias the model towards achieving a high BLEU score. Since N-best lists differ for different language pairs, a separate FLM was trained for each language pair. While both the 4-gram language model and the FLMs achieved a 8-10% reduction in perplexity on the dev set references compared to the baseline language model, their perplexities on the oracle 1-best hypotheses were not significantly different from that of the baseline model.

5 N-best List Rescoring

For N-best list rescoring, the original seven model scores are combined with the scores of the second-pass language models using the framework of discriminative model combination (Beyerlein, 1998). This approach aims at an optimal (with respect to a given error criterion) integration of different information sources in a log-linear model, whose combination weights are trained discriminatively. This combination technique has been used successfully in ASR, where weights are typically optimized to minimize the empirical word error count on a held-out set. In this case, we use the BLEU score of the N-best hypothesis as an optimization criterion. Optimization is performed using a simplex downhill method known as amoeba search (Nelder and Mead, 1965), which is available as part of the SRILM toolkit.

Language pair	1st pass	oracle
Fi-En	21.8	29.8
Fr-En	28.9	34.4
De-En	23.9	31.0
Es-En	30.8	37.4

Table 1: First-pass (left column) and oracle results (right column) on the dev set (% BLEU).

Language pair	4-gram	FLM	both
Fi-En	22.2	22.2	22.3
Fr-En	30.2	30.2	30.4
De-En	24.6	24.2	24.6
Es-En	31.4	31.0	31.3

Table 2: Second-pass rescoring results (% BLEU) on the dev set for 4-gram LM, 3-gram FLM, and their combination.

6 Results

The results from the first decoding pass on the development set are shown in Table 1. The second column in Table 1 lists the oracle BLEU scores for the N-best lists, i.e. the scores obtained by always selecting the hypothesis known to have the highest individual BLEU score. We see that considerable improvements can in principle be obtained by a better second-pass selection of hypotheses. The language model rescoring results are shown in Table 2, for both types of second-pass language models individually, and for their combination. In both cases we obtain small improvements in BLEU score, with the 4-gram providing larger gains than the 3-gram FLM. Since their combination only yielded negligible additional improvements, only 4-grams were used for processing the final evaluation sets. The evaluation results are shown in Table 3.

Language pair	baseline	4-gram
Fi-En	21.6	22.0
Fr-En	29.3	30.3
De-En	24.2	24.8
Es-En	30.5	31.0

Table 3: Second-pass rescoring results (% BLEU) on the evaluation set.

7 Conclusions

We have demonstrated improvements in BLEU score by utilizing more complex language models in the rescoring pass of a two-pass SMT system. We noticed that FLMs performed worse than word-based 4-gram models. However, only trigram FLM were used in the present experiments; larger improvements might be obtained by 4-gram FLMs. The weights assigned to the second-pass language models during weight optimization were larger than those assigned to the first-pass language model, suggesting that both the word-based model and the FLM provide more useful scores than the baseline language model. Finally, we observed that the overall improvement represents only a small portion of the possible increase in BLEU score as indicated by the oracle results, suggesting that better language models do not have a significant effect on the overall system performance unless the translation model is improved as well.

Acknowledgements

This work was funded by the National Science Foundation, Grant no. IIS-0308297. We are grateful to Philip Koehn for assistance with Pharaoh.

References

P. Beyerlein. 1998. Discriminative model combination. In *Proc. ICASSP*, pages 481–484.

J.A. Bilmes and K. Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *Proceedings of HLT/NAACL*, pages 4–6.

K. Duh and K. Kirchhoff. 2004. Automatic learning of language model structure. In *Proceedings of COLING*.

P. Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proceedings of AMTA*.

J.A. Nelder and R. Mead. 1965. A simplex method for function minimization. *Computing Journal*, 7(4):308–313.

F.J. Och and H. Ney. 2000. Giza++: Training of statistical translation models. http://www-i6.informatik.rwth-aachen.de/ och/software/GIZA++.html.

M.F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

A. Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings EMNLP*, pages 133–141.

D. Vergyri et al. 2004. Morphology-based language modeling for Arabic speech recognition. In *Proceedings of ICSLP*.