Fast and Scalable Decoding with Language Model Look-Ahead for Phrase-based Statistical Machine Translation

Joern Wuebker, Hermann Ney
Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University, Germany
surname@cs.rwth-aachen.de

Richard Zens*
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
zens@google.com

Abstract

In this work we present two extensions to the well-known dynamic programming beam search in phrase-based statistical machine translation (SMT), aiming at increased efficiency of decoding by minimizing the number of language model computations and hypothesis expansions. Our results show that language model based pre-sorting yields a small improvement in translation quality and a speedup by a factor of 2. Two look-ahead methods are shown to further increase translation speed by a factor of 2 without changing the search space and a factor of 4 with the side-effect of some additional search errors. We compare our approach with Moses and observe the same performance, but a substantially better trade-off between translation quality and speed. At a speed of roughly 70 words per second, Moses reaches 17.2% BLEU, whereas our approach yields 20.0% with identical models.

1 Introduction

Research efforts to increase search efficiency for phrase-based MT (Koehn et al., 2003) have explored several directions, ranging from generalizing the stack decoding algorithm (Ortiz et al., 2006) to additional early pruning techniques (Delaney et al., 2006), (Moore and Quirk, 2007) and more efficient language model (LM) querying (Heafield, 2011).

This work extends the approach by (Zens and Ney, 2008) with two techniques to increase translation speed and scalability. We show that taking a heuristic LM score estimate for pre-sorting the phrase translation candidates has a positive effect on both translation quality and speed. Further, we introduce two novel LM look-ahead methods. The idea of LM look-ahead is to incorporate the LM probabilities into the pruning process of the beam search as early as possible. In speech recognition it has been used for many years (Steinbiss et al., 1994; Ortmanns et al., 1998). *First-word LM look-ahead* exploits the search structure to use the LM costs of the first word of a new phrase as a lower bound for the full LM costs of the phrase. *Phrase-only LM look-ahead* makes use of a pre-computed estimate of the full LM costs for each phrase. We detail the implementation of these methods and analyze their effect with respect to the number of LM computations and hypothesis expansions as well as on translation speed and quality. We also run comparisons with the Moses decoder (Koehn et al., 2007), which yields the same performance in BLEU, but is outperformed significantly in terms of scalability for faster translation. Our implementation is available under a non-commercial open source licence[†].

2 Search Algorithm Extensions

We apply the decoding algorithm described in (Zens and Ney, 2008). Hypotheses are scored by a weighted log-linear combination of models. A beam search strategy is used to find the best hypothesis. During search we perform pruning controlled by the parameters *coverage histogram size*[‡] N_c and *lexical*

*Richard Zens's contribution was during his time at RWTH.
[†] www-i6.informatik.rwth-aachen.de/jane
[‡] number of hypothesized coverage vectors per cardinality

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 28–32,
Jeju, Republic of Korea, 8-14 July 2012.

histogram size[§] N_l.

2.1 Phrase candidate pre-sorting

In addition to the source sentence f_1^J, the beam search algorithm takes a matrix $E(\cdot,\cdot)$ as input, where for each contiguous phrase $\tilde{f} = f_j \dots f_{j'}$ within the source sentence, $E(j, j')$ contains a list of all candidate translations for $\tilde{f}$. The candidate lists are sorted according to their model score, which was observed to speed up translation by Delaney et al. (2006). In addition to sorting according to the purely phrase-internal scores, which is common practice, we compute an estimate $q_{\mathrm{LME}}(\tilde{e})$ for the LM score of each target phrase $\tilde{e}$. $q_{\mathrm{LME}}(\tilde{e})$ is the weighted LM score we receive by assuming $\tilde{e}$ to be a complete sentence without using sentence start and end markers. We limit the number of translation options per source phrase to the N_o top scoring candidates (observation histogram pruning).

The pre-sorting during phrase matching has two effects on the search algorithm. Firstly, it defines the order in which the hypothesis expansions take place. As higher scoring phrases are considered first, it is less likely that already created partial hypotheses will have to be replaced, thus effectively reducing the expected number of hypothesis expansions. Secondly, due to the observation pruning the sorting affects the considered phrase candidates and consequently the search space. A better pre-selection can be expected to improve translation quality.

2.2 Language Model Look-Ahead

LM score computations are among the most expensive in decoding. Delaney et al. (2006) report significant improvements in runtime by removing unnecessary LM lookups via early pruning. Here we describe an LM look-ahead technique, which is aimed at further reducing the number of LM computations.

The innermost loop of the search algorithm iterates over all translation options for a single source phrase to consider them for expanding the current hypothesis. We introduce an LM look-ahead score $q_{\mathrm{LMLA}}(\tilde{e}|\tilde{e}')$, which is computed for each of the translation options. This score is added to the overall hypothesis score, and if the pruning threshold is exceeded, we discard the expansion without computing the full LM score.

First-word LM look-ahead pruning defines the LM look-ahead score $q_{\mathrm{LMLA}}(\tilde{e}|\tilde{e}') = q_{\mathrm{LM}}(\tilde{e}_1|\tilde{e}')$ to be the LM score of the first word of target phrase $\tilde{e}$ given history $\tilde{e}'$. As $q_{\mathrm{LM}}(\tilde{e}_1|\tilde{e}')$ is an upper bound for the full LM score, the technique does not introduce additional seach errors. The score can be reused, if the LM score of the full phrase $\tilde{e}$ needs to be computed afterwards.

We can exploit the structure of the search to speed up the LM lookups for the first word. The LM probabilities are stored in a *trie*, where each node corresponds to a specific LM history. Usually, each LM lookup consists of first traversing the trie to find the node corresponding to the current LM history and then retrieving the probability for the next word. If the n-gram is not present, we have to repeat this procedure with the next lower-order history, until a probability is found. However, the LM history for the first words of all phrases within the innermost loop of the search algorithm is identical. Just before the loop we can therefore traverse the trie once for the current history and each of its lower order n-grams and store the pointers to the resulting nodes. To retrieve the LM look-ahead scores, we can then directly access the nodes without the need to traverse the trie again. This implementational detail was confirmed to increase translation speed by roughly 20% in a short experiment.

Phrase-only LM look-ahead pruning defines the look-ahead score $q_{\mathrm{LMLA}}(\tilde{e}|\tilde{e}') = q_{\mathrm{LME}}(\tilde{e})$ to be the LM score of phrase $\tilde{e}$, assuming $\tilde{e}$ to be the full sentence. It was already used for sorting the phrases, is therefore pre-computed and does not require additional LM lookups. As it is not a lower bound for the real LM score, this pruning technique can introduce additional search errors. Our results show that it radically reduces the number of LM lookups.

3 Experimental Evaluation

3.1 Setup

The experiments are carried out on the German→English task provided for WMT 2011[*].

[§] number of lexical hypotheses per coverage vector

[*] http://www.statmt.org/wmt11

system	BLEU[%]	#HYP	#LM	w/s
$N_o = \infty$				
baseline	20.1	3.0K	322K	2.2
+pre-sort	20.1	2.5K	183K	3.6
$N_o = 100$				
baseline	19.9	2.3K	119K	7.1
+pre-sort	20.1	1.9K	52K	15.8
+first-word	20.1	1.9K	40K	31.4
+phrase-only	19.8	1.6K	6K	69.2

Table 1: Comparison of the number of hypothesis expansions per source word (#HYP) and LM computations per source word (#LM) with respect to LM pre-sorting, first-word LM look-ahead and phrase-only LM look-ahead on `newstest2009`. Speed is given in words per second. Results are given with ($N_o = 100$) and without ($N_o = \infty$) observation pruning.

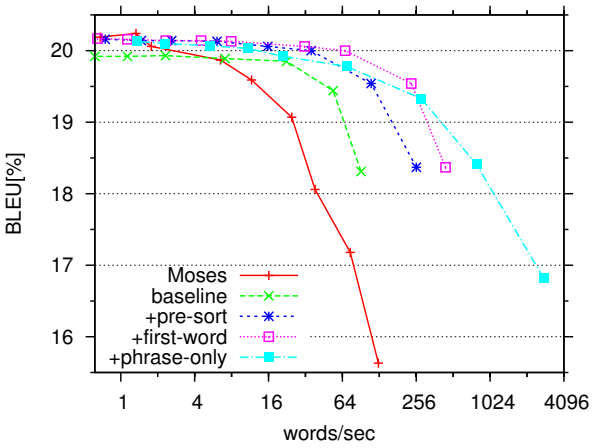

Figure 1: Translation performance in BLEU [%] on the `newstest2009` set vs. speed on a logarithmic scale. We compare Moses with our approach without LM look-ahead and LM score pre-sorting (baseline), with added LM pre-sorting and with either first-word or phrase-only LM look-ahead on top of +pre-sort. Observation histogram size is fixed to $N_o = 100$ for both decoders.

The English language model is a 4-gram LM created with the SRILM toolkit (Stolcke, 2002) on all bilingual and parts of the provided monolingual data. `newstest2008` is used for parameter optimization, `newstest2009` as a blind test set. To confirm our results, we run the final set of experiments also on the English→French task of IWSLT 2011[†]. We evaluate with BLEU (Papineni et al., 2002) and TER (Snover et al., 2006).

We use identical phrase tables and scaling factors for Moses and our decoder. The phrase table is pruned to a maximum of 400 target candidates per source phrase before decoding. The phrase table and LM are loaded into memory before translating and loading time is eliminated for speed measurements.

3.2 Methodological analysis

To observe the effect of the proposed search algorithm extensions, we ran experiments with fixed pruning parameters, keeping track of the number of hypothesis expansions and LM computations. The LM score pre-sorting affects both the set of phrase candidates due to observation histogram pruning and the order in which they are considered. To separate these effects, experiments were run both with histogram pruning ($N_o = 100$) and without. From Table 1 we can see that in terms of efficiency both cases show similar improvements over the baseline, which performs pre-sorting with respect to the translation model scores only. The number of hypothesis expansions is reduced by ∼20% and the number of LM lookups by ∼50%. When observation pruning is applied, we additionally observe a small increase by 0.2% in BLEU.

Application of first-word LM look-ahead further reduces the number of LM lookups by 23%, resulting in doubled translation speed, part of which derives from fewer trie node searches. The heuristic phrase-only LM look-ahead method introduces additional search errors, resulting in a BLEU drop by 0.3%, but yields another 85% reduction in LM computations and increases throughput by a factor of 2.2.

3.3 Performance evaluation

In this section we evaluate the proposed extensions to the original beam search algorithm in terms of scalability and their usefulness for different application constraints. We compare Moses and four different setups of our decoder: LM score pre-sorting switched on or off without LM look-ahead and both LM look-ahead methods with LM score pre-sorting. We translated the test set with the beam sizes set to $N_c = N_l = \{1,2,4,8,16,24,32,48,64\}$. For Moses we used the beam sizes $2^i, i \in \{1,\ldots,9\}$. Transla-

[†] `http://iwslt2011.org`

setup	system	WMT 2011 German→English				IWSLT 2011 English→French			
		beam size (N_c,N_l)	speed w/s	BLEU [%]	TER [%]	beam size (N_c,N_l)	speed w/s	BLEU [%]	TER [%]
best	Moses	256	0.7	20.2	63.2	16	10	29.5	52.8
	this work: first-word	(48,48)	1.1	20.2	63.3	(8,8)	23	29.5	52.9
	phrase-only	(64,64)	1.4	20.1	63.2	(16,16)	18	29.5	52.8
BLEU:	Moses	16	12	19.6	63.7	4	40	29.1	53.2
$\geq$ -1%	this work: first-word	(4,4)	67	20.0	63.2	(2,2)	165	29.1	53.1
	phrase-only	(8,8)	69	19.8	63.0	(4,4)	258	29.3	52.9
BLEU:	Moses	8	25	19.1	64.2	2	66	28.1	54.3
$\geq$ -2%	this work: first-word	(2,2)	233	19.5	63.4	(1,1)	525	28.4	53.9
	phrase-only	(4,4)	280	19.3	63.0	(2,2)	771	28.5	53.2
fastest	Moses	1	126	15.6	68.3	1	116	26.7	55.9
	this work: first-word	(1,1)	444	18.4	64.6	(1,1)	525	28.4	53.9
	phrase-only	(1,1)	2.8K	16.8	64.4	(1,1)	2.2K	26.4	54.7

Table 2: Comparison of Moses with this work. Either first-word or phrase-only LM look-ahead is applied. We consider both the best and the fastest possible translation, as well as the fastest settings resulting in no more than 1% and 2% BLEU loss on the development set. Results are given on the test set (`newstest2009`).

tion performance in BLEU is plotted against speed in Figure 1. Without the proposed extensions, Moses slightly outperforms our decoder in terms of BLEU. However, the latter already scales better for higher speed. With LM score pre-sorting, the best BLEU value is similar to Moses while further accelerating translation, yielding identical performance at 16 words/sec as Moses at 1.8 words/sec. Application of first-word LM look-ahead shifts the graph to the right, now reaching the same performance at 31 words/sec. At a fixed translation speed of roughly 70 words/sec, our approach yields 20.0% BLEU, whereas Moses reaches 17.2%. For phrase-only LM look-ahead the graph is somewhat flatter. It yields nearly the same top performance with an even better trade-off between translation quality and speed.

The final set of experiments is performed on both the WMT and the IWSLT task. We directly compare our decoder with the two LM look-ahead methods with Moses in four scenarios: the best possible translation, the fastest possible translation without performance constraint and the fastest possible translation with no more than 1% and 2% loss in BLEU on the dev set compared to the best value. Table 2 shows that on the WMT data, the top performance is similar for both decoders. However, if we allow for a small degradation in translation performance, our approaches clearly outperform Moses in terms of translation speed. With phrase-only LM look-ahead, our decoder is faster by a factor of 6 for no more than 1% BLEU loss, a factor of 11 for 2% BLEU loss and a factor of 22 in the fastest setting. The results on the IWSLT data are very similar. Here, the speed difference reaches a factor of 19 in the fastest setting.

4 Conclusions

This work introduces two extensions to the well-known beam search algorithm for phrase-based machine translation. Both pre-sorting the phrase translation candidates with an LM score estimate and LM look-ahead during search are shown to have a positive effect on translation speed. We compare our decoder to Moses, reaching a similar highest BLEU score, but clearly outperforming it in terms of scalability with respect to the trade-off ratio between translation quality and speed. In our experiments, the fastest settings of our decoder and Moses differ in translation speed by a factor of 22 on the WMT data and a factor of 19 on the IWSLT data. Our software is part of the open source toolkit *Jane*.

Acknowledgments

This work was partially realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

References

[Delaney et al.2006] Brian Delaney, Wade Shen, and Timothy Anderson. 2006. An efficient graph search decoder for phrase-based statistical machine translation. In *International Workshop on Spoken Language Translation*, Kyoto, Japan, November.

[Heafield2011] Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the 6th Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, UK, July.

[Koehn et al.2003] P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Meeting of the North American chapter of the Association for Computational Linguistics (NAACL-03)*, pages 127–133, Edmonton, Alberta.

[Koehn et al.2007] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantine, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*, pages 177–180, Prague, Czech Republic, June.

[Moore and Quirk2007] Robert C. Moore and Chris Quirk. 2007. Faster beam-search decoding for phrasal statistical machine translation. In *Proceedings of MT Summit XI*.

[Ortiz et al.2006] Daniel Ortiz, Ismael Garcia-Varea, and Francisco Casacuberta. 2006. Generalized stack decoding algorithms for statistical machine translation. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 64–71, New York City, June.

[Ortmanns et al.1998] S. Ortmanns, H. Ney, and A. Eiden. 1998. Language-model look-ahead for large vocabulary speech recognition. In *International Conference on Spoken Language Processing*, pages 2095–2098, Sydney, Australia, October.

[Papineni et al.2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July.

[Snover et al.2006] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA, August.

[Steinbiss et al.1994] V. Steinbiss, B. Tran, and Hermann Ney. 1994. Improvements in Beam Search. In *Proc. of the Int. Conf. on Spoken Language Processing (ICSLP'94)*, pages 2143–2146, September.

[Stolcke2002] Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proceedings of the Seventh International Conference on Spoken Language Processing*, pages 901–904. ISCA, September.

[Zens and Ney2008] Richard Zens and Hermann Ney. 2008. Improvements in Dynamic Programming Beam Search for Phrase-based Statistical Machine Translation. In *International Workshop on Spoken Language Translation*, pages 195–205, Honolulu, Hawaii, October.