Discriminative Training of 150 Million Translation Parameters and Its Application to Pruning

Hendra Setiawan and Bowen Zhou
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598, USA
{*hendras, zhou*}*@us.ibm.com*

Abstract

Until recently, the application of discriminative training to log linear-based statistical machine translation has been limited to tuning the weights of a limited number of features or training features with a limited number of parameters. In this paper, we propose to scale up discriminative training of (He and Deng, 2012) to train features with 150 million parameters, which is one order of magnitude higher than previously published effort, and to apply discriminative training to redistribute probability mass that is lost due to model pruning. The experimental results confirm the effectiveness of our proposals on NIST MT06 set over a strong baseline.

1   Introduction

State-of-the-art statistical machine translation systems based on a log-linear framework are parameterized by $\{\lambda, \Phi\}$, where the feature weights λ are discriminatively trained (Och and Ney, 2002; Chiang et al., 2008b; Simianer et al., 2012) by directly optimizing them against a translation-oriented metric such as BLEU. The feature parameters Φ can be roughly divided into two categories: dense feature that measures the plausibility of each translation rule from a particular aspect, e.g., the rule translation probabilities $p(f|e)$ and $p(e|f)$; and sparse feature that fires when certain phenomena is observed, e.g., when a frequent word pair co-occured in a rule. In contrast to λ, feature parameters in Φ are usually modeled by generative models for dense features, or by indicator functions for sparse ones. It is therefore desirable to train the dense features for each rule in a discriminative fashion to maximize some translation criterion. The maximum expected BLEU training of (He and Deng, 2012) is a recent effort towards this

direction, and in this paper, we extend their work to a scaled-up task of discriminative training of the features of a strong hierarchical phrase-based model and confirm its effectiveness empirically.

In this work, we further consider the application of discriminative training to pruned model. Various pruning techniques (Johnson et al., 2007; Zens et al., 2012; Eck et al., 2007; Lee et al., 2012; Tomeh et al., 2011) have been proposed recently to filter translation rules. One common consequence of pruning is that the probability distribution of many surviving rules become deficient, i.e. $\sum_f p(f|e) < 1$. In practice, others have chosen either to leave the pruned rules as it-is, or simply to re-normalize the probability mass by distributing the pruned mass to surviving rules proportionally. We argue that both approaches are suboptimal, and propose a more principled method to re-distribute the probability mass, i.e. using discriminative training with some translation criterion. Our experimental results demonstrate that at various pruning levels, our approach improves performance consistently. Particularly at the level of 50% of rules being pruned, the discriminatively trained models performs better than the unpruned baseline grammar. This shows that discriminative training makes it possible to achieve smaller models that perform comparably or even better than the baseline model.

Our contributions in this paper are two-folded: First of all, we scale up the maximum expected BLEU training proposed in (He and Deng, 2012) in a number of ways including using 1) a hierarchical phrase-based model, 2) a richer feature set, and 3) a larger training set with a much larger parameter set, resulting in more than 150 million parameters in the model being updated, which is one order magnitude higher than the phrase-based model reported in (He and Deng, 2012). We are able to show a reasonable

335

improvement over this strong baseline. Secondly, we combine discriminative training with pruning techniques to reestimate parameters of pruned grammar. Our approach is shown to alleviate the loss due to pruning, and sometimes can even outperform the baseline unpruned grammar.

2 Discriminative Training of Φ

Given the entire training data $\{F_n, E_n\}_{n=1}^N$, and current parameterization $\{\lambda, \Phi\}$, we decode the source side of training data F_n to produce hypothesis $\{\hat{E}_n\}_{n=1}^N$. Our goal is to update Φ towards Φ' that maximizes the expected BLEU scores of the entire training data given the current λ:

$$U(\Phi) = \sum_{\forall \hat{E}_1 \ldots \hat{E}_N} \tilde{P}_\Phi(\hat{E}_1..\hat{E}_N | F_1..F_N) B(\hat{E}_1..\hat{E}_N) \quad (1)$$

where $B(\hat{E}_1...\hat{E}_N)$ is the BLEU score of the concatenated hypothesis of the entire training data, following (He and Deng, 2012).

Eq. 1 summarizes over all possible combinations of $\hat{E}_1...\hat{E}_N$, which is intractable. Hence we make two simplifying approximations as follows. First, let the k-best hypotheses of the n-th sentence, $\hat{E}_n = \left\{ \hat{E}_n^1, ..., \hat{E}_n^K \right\}$, approximate all its possible translation. In other words, we assume that $\sum_{k=1}^K \tilde{P}(\hat{E}_n^k | F_n) = 1$, $\forall n$. Second, let the sum of sentence-level BLEU approximate the corpus BLEU. We note that corpus BLEU is not strictly decomposable (Chiang et al., 2008a), however, as the training data's size N gets big as in our case, we expect them to become more positively correlated.

Under these assumptions and the fact that each sentence is decoded independently, Eq. 1 can be algebraically simplified into:

$$U(\Phi) = \sum_{n=1}^N \sum_{k=1}^K P_\Phi(\hat{E}_n^k | F_n) B(\hat{E}_n^k) \quad (2)$$

where $P_\Phi(\hat{E}_n^k | F_n) = \tilde{P}_\Phi(\hat{E}_n^k | F_n) / \sum_{\forall k} \tilde{P}_\Phi(\hat{E}_n^k | F_n)$. We detail the process in the Appendix.

To further simplify the problem and relate it with model pruning, we consider to update a subset of $\theta \subset \Phi$ while keeping other parameterization of Φ unchanged, where $\theta = \{\theta_{ij} = p(e_j | f_i)\}$ denotes our parameter set that satisfies $\sum_j \theta_{ij} = 1$ and $\theta_{ij} \geq 0$. In experiments, we also consider $\{\theta_{ji} = p(f_i | e_j)\}$.

To alleviate overfitting, we introduce KL-distance based regularization as in (He and Deng, 2012). We thus arrive at the following objective function:

$$O(\theta) = \log(U(\theta)) - \tau \cdot KL(\theta || \theta^0) / N \quad (3)$$

where τ controls the regularization term's contribution, and θ^0 represents a prior parameter set, e.g., from the conventional maximum likelihood training.

The optimization algorithm is based on the Extended Baum Welch (EBW) (Gopalakrishnan et al., 1991) as derived by (He and Deng, 2012). The final update rule is as follow:

$$\theta'_{ij} = \frac{\sum_n \sum_k \gamma(n, k, i, j) + U(\theta) \tau \theta_{ij}^0 / \lambda + D_i \theta_{ij}}{\sum_n \sum_k \sum_j \gamma(n, k, i, j) + U(\theta) \tau / \lambda + D_i} \quad (4)$$

where θ'_{ij} is the updated parameter, $\gamma(n, k, i, j) = P_\theta(\hat{E}_n^k | F_n) \{B(\hat{E}_n^k) - U_n(\theta)\} \sum_l 1(f_{n,k,l} = f_i, e_{n,k,l} = e_j)$; $U_n(\theta) = \sum_{k=1}^K P_\theta(\hat{E}_n^k | F_n) B(\hat{E}_n^k)$; $D_i = \sum_{n,k,j} \max(0, -\gamma(n, k, i, j))$ and λ is the current feature's weight.

3 DT is Beneficial for Pruning

Pruning is often a key part in deploying large-scale SMT systems for many reasons, such as for reducing runtime memory footprint and for efficiency. Many pruning techniques have been proposed to assess translation rules and filter rules out if they are less plausible than others. While different pruning techniques may use different criterion, they all assume that pruning does not affect the feature function values of the surviving rules. This assumption may be suboptimal for some feature functions that have probabilistic sense since pruning will remove a portion of the probability mass that is previously assigned to the pruned rules. To be concrete, for the rule translation probabilities θ_{ij} under consideration, the constraint $\sum_j \theta_{ij} = 1$ will not hold for all source rules i after pruning. Previous works typically left the probability mass as it-is, or simply renormalize the pruned mass, i.e. $\bar{\theta}_{ij} = \theta_{ij} / \sum_j \theta_{ij}$.

We argue that applying the DT techniques to a pruned grammar, as described in Sec. 2, provides a more principled method to redistribute the mass, i.e. by quantizing how each rule contributes to the expected BLEU score in comparison to other competing rules. To empirically verify this, we consider

the significance test based pruning (Johnson et al., 2007), though our general idea can be appllied to any pruning techniques. For our experiments, we use the significance pruning tool that is available as part of Moses decoder package (Koehn et al., 2007).

4 Experiments

Our experiments are designed to serve two goals: 1) to show the performance of discriminative training of feature parameters θ in a large-scale task; and 2) to show the effectiveness of DT when applied to pruned grammar. Our baseline system is a state-of-the-art hierarchical phrase-based system as described in (Zhou et al., 2008), trained on six million parallel sentences corpora that are available to the DARPA BOLT Chinese-English task. The training corpora includes a mixed genre of news wire, broadcast news, web-blog and comes from various sources such as LDC, HK Hansard and UN data.

In total, there are 50 dense features in our translation system. In addition to the standard features which include the rule translation probabilities, we incorporate features that are found useful for developing a state-of-the-art baseline, e.g. provenance-based lexical features (Chiang et al., 2011). We use a large 6-gram language model, which we train on a 10 billion words monolingual corpus, including the English side of our parallel corpora plus other corpora such as Gigaword (LDC2011T07) and Google News. To prevent possible over-fitting, we only kept the rules that have at most three terminal words (plus up to two nonterminals) on the source side, resulting in a grammar with 167 million rules.

Our discriminative training procedure includes updating both λ and θ, and we follow (He and Deng, 2012) to optimize them in an alternate manner. That is, when we optimize θ via EBW, we keep λ fixed and when we optimize λ, we keep λ fixed. We use PRO (Hopkins and May, 2011) to tune λ.

For discriminative training of θ, we use a subset of 550 thousands of parallel sentences selected from the entire training data, mainly to allow for faster experimental cycle; they mainly come from news and web-blog domains. For each sentence of this subset, we generate 500-best of unique hypotheses using the baseline model. The 1-best and the oracle BLEU scores for this subset are 40.19 and 47.06 respec-

tively. Following (He and Deng, 2012), we focus on discriminative training of $p(f|e)$ and $p(e|f)$, which in practice affects around 150 million of parameters; hence the title.

For the tuning and development sets, we set aside 1275 and 1239 sentences respectively from LDC2010E30 corpus. The tune set is used by PRO for tuning λ while the dev set is used to decide the best DT model. As for the blind test set, we report the performance on the NIST MT06 evaluation set, which consists of 1644 sentences from news and web-blog domains. Our baseline system's performance on MT06 is 39.91 which is among the best number ever published so far in the community.

Table 1 compares the key components of our baseline system with that of (He and Deng, 2012). As shown, we are working with a stronger system than (He and Deng, 2012), especially in terms of the number of parameters under consideration $|\theta|$.

	He&Deng(2012)	This paper
Model	phrase-based	hierarchical
n-gram lm	3-gram	6-gram
# features	10	50
Max terminal	4	3
$|\theta|$	9.2 M	150M
# training data	750K	6M
N for DT	750K	550K
max K-best	100	500

Table 1: Our system compares to He&Deng's (2012).

4.1 DT of 150 Million Parameters

To ensure the correctness of our implementation, we show in Fig 2, the first five EBW updates with $\tau = 0.10$. As shown, the utility function $\log(U(\theta))$ increases monotonically but is countered by the KL term, resulting in a smaller but consistent increase of the objective function $O(\theta)$. This monotonically-increasing trend of the objective function confirms the correctness of our implementation since EBW algorithm is a bound-based technique that ensures growth transformations between updates.

We then explore the optimal setting for τ which controls the contribution of the regularization term. Specifically, we perform grid search, exploring values of τ from 0.1 to 0.75. For each τ, we run several iterations of discriminative training where each iteration involves one simultaneous update of $p(f|e)$

and $p(e|f)$ according to Eq. 4, followed by one update of λ via PRO (as in (He and Deng, 2012)). In total, we run 10 such iterations for each τ.
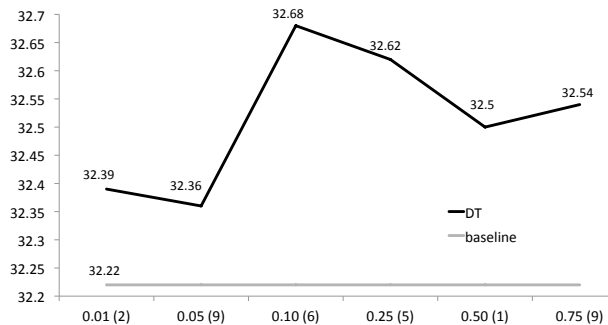

Figure 1: The dev set's BLEU score (y-axis) on different setting of τ (x-axis). The grey line indicates the baseline performance on dev set. The number in bracket on the x-axis indicates the iteration at which the score is obtained.

Across different τ, we find that the first iteration provides most of the gain while the subsequent iterations provide additional, smaller gain with occasional performance degradation; thus the translation performance is not always monotonically increasing over iteration. We report the best score of each τ in Fig. 1 and at which iteration that score is produced. As shown in Fig. 1, all settings of τ improve over the baseline and $\tau = 0.10$ gives the highest gain of 0.45 BLEU score. This improvement is in the same ballpark as in (He and Deng, 2012) though on a scaled-up task. We next decode the MT06 using the best model (i.e. $\tau = 0.10$ at 6-th iteration) observed on the dev set, and obtained 40.33 BLEU with an improvement of around 0.4 BLEU point. We see this result as confirming the effectiveness of discriminative training but on a larger-scale task, adding to what was reported by (He and Deng, 2012).

4.2 DT for Significance Pruning

Next, we show the contribution of discriminative training for model pruning. To do so, we prune the translation grammar so that its size becomes 50%, 25%, 10% of the original grammar. Respectively, we delete rules whose significance value below 15, 50 and 500. Table 2 compares the statistics of the pruned grammars and the unpruned one. In particular, columns 4 and 5 show the total averaged probability mass of the remaining rules. This statistics provides some indication of how deficient the fea-
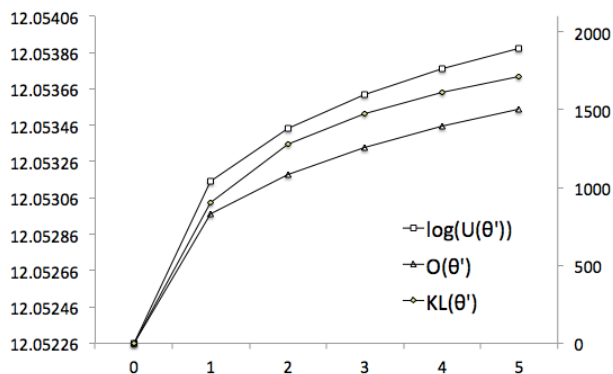

Figure 2: Objective function ($O(\theta')$), the regularization term ($KL(\theta')$) and the unregularized objective function ($\log(U(\theta'))$) for five EBW updates of updating $p(e_j|f_i)$

tures are after pruning. As shown, the total averaged probability mass after pruning is below 100% and even lower for the more aggressive pruning.

To show that the deficiency is suboptimal, we considers two baseline systems: models with/without mass renormalization. We tune a new λ for each model and use the new λ to decode the dev and test sets. The results are shown in columns 6 and 9 of Table 2 where we show the results for the unnormalized model in the brackets following the results for the re-normalized model. The results show that pruning degrades the performances and that naively re-normalizing the model provides no significant changes in performance. Subsequently, we will focus on the normalized models as the baseline as they represents the starting points of our EBW iteration.

Next, we run discriminative training that would reassign the probability mass to the surviving rules. First, we normalize $p(f|e)$ and $p(e|f)$, so that they satisfy the sum to one constraint required by the algorithm. Then, we run discriminative training on these pruned grammars using $\tau = 0.10$ (i.e. the setting that gives the best performance for the unpruned grammar as discussed in Section 4.1). We report the results in columns 7 and 9 for the dev and test sets respectively, as well as the gain over the baseline system in columns 8 and 10.

As shown in Table 2, DT provides a nice improvement over the baseline model of no mass reassignment. For all pruning levels, DT can compensate the loss associated with pruning. In particular, at 50% level of pruning, there is a loss about 0.4

338

size	$|f|$	$|e|$	$p(*|e)$	$p(*|f)$	dev-set			test-set (MT06)		
(%)	(M)	(M)			baseline (un)	DT (iter)	gain	baseline (un)	DT	gain
100	59	50	1.00	1.00	*32.22(32.08)*	**32.68** (6)	+0.44	*39.91 (39.71)*	**40.33**	+0.42
50	38	35	0.92	0.94	31.84 (32.02)	**32.31** (6)	+0.57	39.61(39.72)	**40.08**	+0.47
25	14	14	0.87	0.91	31.39 (31.43)	31.68 (2)	+0.29	39.23 (39.17)	39.43	+0.20
10	4	3	0.77	0.84	27.27 (27.10)	27.82 (2)	+0.55	36.01 (36.04)	36.43	+0.42

Table 2: The statistics of grammars pruned at various level (column 1), including the number of unique source and target phrases (columns 2 & 3), total probability mass of the remaining rules for $p(f|e)$ and $p(e|f)$ (columns 4 & 5), the performance of the pruned model before and after discriminative training as well as the gain on the dev and the test sets (columns 6 to 11). The iteration at which DT gives the best dev set is indicated by the number enclosed by bracket in column 7. The baseline performance is in *italics*, followed by a number in the bracket which refers to the performance of using unnormalized model. The above-the-baseline performances are in **bold**.

BLEU point after pruning. With the DT on pruned model, all pruning losses are reclaimed and the new pruned model is even better than the unpruned original model. This empirical result shows that leaving probability mass unassigned after pruning is sub-optimal and that discriminative training provides a principled way to redistribute the mass.

5 Conclusion

In this paper, we first extend the maximum expected BLEU training of (He and Deng, 2012) to train two features of a state-of-the-art hierarchical phrase-based system, namely: $p(f|e)$ and $p(e|f)$. Compared to (He and Deng, 2012), we apply the algorithm to a strong baseline that is trained on a bigger parallel corpora and comes with a richer feature set. The number of parameters under consideration amounts to 150 million. Our experiments show that discriminative training these two features (out of 50) gives around 0.40 BLEU point improvement, which is consistent with the conclusion of (He and Deng, 2012) but in a much larger-scale system.

Furthermore, we apply the algorithm to redistribute the probability mass of $p(f|e)$ and $p(e|f)$ that is commonly lost due to conventional model pruning. Previous techniques either leave the probability mass as it is or distribute it proportionally among the surviving rules. We show that our proposal of using discriminative training to redistribute the mass empirically performs better, demonstrating the effectiveness of our proposal.

Appendix

We describe the process to simplify Eq. 1 to Eq. 2, which is omitted in (He and Deng, 2012). For conciseness, we drop the conditions and write $P(\hat{E}_i|F_i)$ as $P(\hat{E}_i)$. We write Eq. 1 again below as Eq. 5 .

$$\sum_{\forall \hat{E}_1 ... \hat{E}_N} \prod_{i=1}^{N} P(\hat{E}_i|F_i) \cdot \sum_{i=1}^{N} B(\hat{E}_i) \qquad (5)$$

We first focus on the first sentence E_1/F_1 and expand the related terms from the equation as follow:

$$\sum_{\forall \hat{E}_1} \sum_{\forall \hat{E}_2 ... \hat{E}_N} P(\hat{E}_1) \prod_{i=2}^{N} P(\hat{E}_i) . \left[B(\hat{E}_1) + \sum_{i=2}^{N} B(\hat{E}_i) \right]$$

Expanding the inner summation, we arrive at:

$$\sum_{\forall \hat{E}_1} P(\hat{E}_1) B(\hat{E}_1) \sum_{\forall \hat{E}_2 ... \hat{E}_N} \prod_{i=2}^{N} P(\hat{E}_i) +$$

$$\sum_{\forall \hat{E}_1} P(\hat{E}_1) \sum_{\forall \hat{E}_2 ... \hat{E}_N} \prod_{i=2}^{N} P(\hat{E}_i) \sum_{i=2}^{N} B(\hat{E}_i)$$

Due to the that $\sum_{k=1}^{K} \tilde{P}(\hat{E}_n^K|F_n) = 1$, we can equate $\sum_{\forall \hat{E}_2 ... \hat{E}_N} \prod_{i=2}^{N} P(\hat{E}_i)$ and $\sum_{\forall \hat{E}_1} P(\hat{E}_1)$ to 1. Thus, we arrive at:

$$\sum_{\forall \hat{E}_1} P(\hat{E}_1) B(\hat{E}_1) + \sum_{\forall \hat{E}_2 ... \hat{E}_N} \prod_{i=2}^{N} P(\hat{E}_i) \sum_{i=2}^{N} B(\hat{E}_i)$$

Notice that the second term has the same form as Eq. 5 except that the starting index starts from the second sentence. The same process can be performed and at the end, thus we can arrive at Eq. 2.

Acknowledgments

References

David Chiang, Steve DeNeefe, Yee Seng Chan, and Hwee Tou Ng. 2008a. Decomposability of translation metrics for improved evaluation and efficient algorithms. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 610–619, Honolulu, Hawaii, October. Association for Computational Linguistics.

David Chiang, Yuval Marton, and Philip Resnik. 2008b. Online large-margin training of syntactic and structural translation features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 224–233, Honolulu, Hawaii, October. Association for Computational Linguistics.

David Chiang, Steve DeNeefe, and Michael Pust. 2011. Two easy improvements to lexical weighting. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 455–460, Portland, Oregon, USA, June. Association for Computational Linguistics.

Matthias Eck, Stephan Vogel, and Alex Waibel. 2007. Translation model pruning via usage statistics for statistical machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 21–24, Rochester, New York, April. Association for Computational Linguistics.

P. S. Gopalakrishnan, Dimitri Kanevsky, Arthur Nádas, and David Nahamoo. 1991. An inequality for rational functions with applications to some statistical estimation problems. *IEEE Transactions on Information Theory*, 37(1):107–113.

Xiaodong He and Li Deng. 2012. Maximum expected bleu training of phrase and lexicon translation models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 292–301, Jeju Island, Korea, July. Association for Computational Linguistics.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empir-ical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. 2007. Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 967–975, Prague, Czech Republic, June. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Seung-Wook Lee, Dongdong Zhang, Mu Li, Ming Zhou, and Hae-Chang Rim. 2012. Translation model size reduction for hierarchical phrase-based statistical machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 291–295, Jeju Island, Korea, July. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Patrick Simianer, Stefan Riezler, and Chris Dyer. 2012. Joint feature selection in distributed stochastic learning for large-scale discriminative training in smt. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–21, Jeju Island, Korea, July. Association for Computational Linguistics.

N. Tomeh, M. Turchi, G. Wisniewski, A. Allauzen, and F. Yvon. 2011. How good are your phrases? assessing phrase quality with single class classification. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 261–268.

Richard Zens, Daisy Stanton, and Peng Xu. 2012. A systematic comparison of phrase table pruning techniques. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*,

pages 972–983, Jeju Island, Korea, July. Association for Computational Linguistics.

Bowen Zhou, Bing Xiang, Xiaodan Zhu, and Yuqing Gao. 2008. Prior derivation models for formally syntax-based translation using linguistically syntactic parsing and tree kernels. In *Proceedings of the ACL-08: HLT Second Workshop on Syntax and Structure in Statistical Translation (SSST-2)*, pages 19–27, Columbus, Ohio, June. Association for Computational Linguistics.