Training Machine Translation with a Second-Order Taylor Approximation of Weighted Translation Instances

Aaron B. Phillips
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
aphillips@cmu.edu

Ralf D. Brown
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
ralf@cs.cmu.edu

Abstract

The Cunei Machine Translation Platform is an open-source MT system designed to model instances of translation. One of the challenges to this approach is effective training. We describe two techniques that improve the training procedure and allow us to leverage the strengths of instance-based modeling. First, during training we approximate our model with a second-order Taylor series. Second, we discount models based on the magnitude of their approximation. By reducing error in training, our model now consistently outperforms the standard SMT model with gains ranging from 0.51 to 3.77 BLEU on German-English and Czech-English test sets.

1 Introduction

Machine translation research over the years has explored the use of simple phrases (Och and Ney, 2004), Hiero grammars (Chiang, 2005), and complex S-CFG rules (Zollmann and Venugopal, 2006). These more specialized translation units can more accurately describe the translation process, but they are also less likely to occur in the corpus. The increased data sparsity makes it difficult to estimate the standard SMT features which are typically computed as relative frequencies. A significant challenge in building data-driven MT systems is identifying the right level of abstraction–to model translation units that both adequately reflect the data and can be estimated well.

Our approach pushes this trend of translation unit refinement to its logical end and models each *instance* of translation. An instance of translation is the realization of a source and corresponding target phrase at *one specific location in the corpus*. We score each translation instance with a series of features that examine the alignment, context, genre, and other surroundings. Our model then combines these translation instances in a weighted summation. This approach conveniently side-steps the challenges of estimation sparsity because our model is not based on relative frequency estimates. The weighting of translation instances relates to methods for domain-adaptation of SMT models, but our implementation is fundamentally different in that we do not alter or re-weight the training data. Instead, our model directly embodies the notion that not all translations are equal and individually evaluates the relevance of each translation instance.

We have presented this approach to translation before, but with performance that was on par with or indistinguishable from state-of-the-art SMT (Phillips and Brown, 2009; Phillips, 2010). In particular, the complexity of our model presents specific challenges in training. We have learned since that our training procedures did not fully leverage the capabilities of our model. In this paper we describe two new techniques for more effectively training our model. First, we utilize a second-order Taylor series to approximate the model during training. Second, we present a method for discounting models based on the magnitude of their approximation. We then proceed to show that by reducing error in training, our model outperforms the standard SMT model by 0.51 BLEU on German-English newswire, 0.75 BLEU on Czech-English newswire, and 3.77 BLEU on more diverse, multi-genre Czech-English data.

2 Cunei Machine Translation Platform

Our work has been implemented within the Cunei Machine Translation Platform. We developed this platform to facilitate research in modeling instances of translations. It is open-source and may be downloaded from `http://www.cunei.org`.

3 Approach

In order to compose a complete sentence, machine translation systems score small units of translation and select the fragments that when combined together yield the best score according to their model. We can abstractly describe this decision rule for the source sequence $s_1, s_2...s_n$ as:

$$\tilde{t} = \underset{t_1,t_2...t_n}{\arg\max} \sum_{i=0}^{n} m(s_i, t_i, \lambda) \tag{1}$$

Here model m scores each translation unit which consists of a target phrase t_i and a corresponding source span s_i.[1] The sequence of target phrases $t_i, t_2, ...t_n$ that maximizes the score composes the target sentence $\tilde{t}$. Within this framework, a typical log-linear SMT model with features θ and weights λ would be represented as:

$$m(s_i, t_i, \lambda) = \ln e^{\sum_k \lambda_k \cdot \theta_k(s_i,t_i)} \tag{2}$$

$$= \sum_k \lambda_k \cdot \theta_k(s_i, t_i) \tag{3}$$

The SMT feature function θ models each translation unit (s_i, t_i). Evidence for a translation unit will generally be present at multiple locations within the training data. The features for θ operate over this set of translation instances and are generally computed as relative frequencies. A common feature, for example, is the number of times s_i and t_i are aligned divided by the total occurrences of s_i.

Our model for translation is fundamentally different in that our translation units are not abstract phrase pairs or grammar rules. Similar to Equation 3, the core component of our model is a feature function which allows the user to easily add new sources of knowledge to the system. However, our feature function ϕ evaluates one specific instance of translation instead of scoring the entire set of translation instances. We model the translation unit as the weighed summation of the scores for all translation instances:

$$m(s_i, t_i, \lambda) = \ln \sum_{\eta} e^{\sum_k \lambda_k \cdot \phi_k(s_i,t_i,\eta)} \tag{4}$$

Here η represents an instance of translation which identifies a unique location within the training data where a source phrase s' translates as a target phrase t'. The feature function ϕ informs the model how relevant the translation instance η is for modeling the phrase pair (s_i, t_i).[2] The feature function ϕ may include information such as the alignment probability between s' and t', the similarity between s_i and s', the location of η in the corpus, or other contextual knowledge. For the experiments in this paper we used approximately 30 such features.

For illustration, consider that the translation instances for a given phrase pair occur in a variety of sentences within the training data. Some instances may include an inconsistent word alignment from within the selected phrase pair to a word in the remainder of the sentence. Our model allows us to learn from these translation instances, but discount them by including a feature in ϕ which measures the likelihood of the phrasal alignment given the words *outside* the phrase pair. This differs from the standard SMT approach where phrase alignment is a binary decision. The same principle also applies if we want to include additional non-local information such as genre or context within the model. A traditional SMT model requires new translation units conditioned on the extra information whereas our approach incorporates the extra information as features of ϕ and calculates a score over all instances.

One of the motivations for this model was to combine ideas from Statistical MT and Example-Based MT. Many EBMT systems rely on heuristics and lack a well-defined model, but our per-instance modeling is generally reflective of an 'EBMT approach.' On the other hand, we were motivated by SMT to

[1] For simplicity we only include the source span s_i, but both the SMT model and our approach can be extended to include the entire source sentence as a component of the model.

[2] For efficiency we usually only sum over instances where $s_i = s'$ and $t_i = t'$, but the model does not require this restriction and permits the use of translation instances that do not exactly match the input.

create a well-defined feature-based model in which the parameters could be estimated using development data. The result is that the standard log-linear model used in SMT is a special case of our model. When the features for all instances of a translation are constant such that $\phi_k(s,t,\eta) = \theta_k(s,t)\ \forall\eta\ \forall k$, then Equation 4 is exactly $|\eta|$ times Equation 3.[3]

Our approach differs from SMT only in how each translation unit is modeled. This is illustrated above in the different definitions for m (Equations 3 and 4). Both approaches use the same decision rule (Equation 1) to combine these translation units together and construct a complete translation.

3.1 Taylor Series Approximation

Given a set of weights, λ, we can easily compute the score of our model by iterating over the instances of translation and calculating the requisite features. However, learning the optimal λ for our model is not so straightforward. We estimate the weights for our model using the approach described in (Smith and Eisner, 2006) which minimizes the expected loss of BLEU over the n-best distribution of translations in a development set. However, this procedure requires us to compute the gradient and re-score the model frequently under a new λ. Storing the features ϕ for *every* translation instance consumes too much memory, and re-decoding under *every* new λ consumes too much time during training.

To address this problem, we approximate our model (Equation 4) during training with the second-order Taylor series:[4]

$$
\begin{aligned}
m(s,t,\lambda') \approx{} & \ln \sum_{\eta} e^{\sum_k \lambda_k \cdot \phi_k(s,t,\eta)} \\
& + \sum_q (\lambda'_q - \lambda_q) E_\eta[\phi_q(s,t,\eta)] \\
& + \frac{1}{2} \sum_q (\lambda'_q - \lambda_q) \sum_r (\lambda'_r - \lambda_r) \\
& (E_\eta[\phi_q(s,t,\eta) \cdot \phi_r(s,t,\eta)] \\
& - E_\eta[\phi_q(s,t,\eta)] \cdot E_\eta[\phi_r(s,t,\eta)])
\end{aligned} \quad (5)
$$

	Average	Variance
First Order Error	0.1751	0.2893
Second Order Error	0.1202	0.1391
Relative Improvement	*31.36%*	*51.94%*

Table 1: Approximation Error in Training Czech-English

Here q and r are indexes for the weights in λ and λ'. Both expectations can be computed efficiently with an online update that analyzes each translation instance once. Formally, the expectation is:

$$
E_\eta[X] = \sum_\eta X \cdot P(\eta \mid s,t,\lambda)
$$

$$
P(\eta \mid s,t,\lambda) = \frac{e^{\sum_k \lambda_k \phi_k(s,t,\eta)}}{\sum_{\eta'} e^{\sum_k \lambda_k \phi_k(s,t,\eta')}}
$$

In (Phillips, 2010) we used a first-order Taylor series as it was easier to implement and we assumed its approximation was 'close enough'. However, as shown in Table 1, the second-order Taylor approximation significantly decreases modeling error. The modeling error is measured as the absolute difference in log score between the approximation and the actual score of each model. The most compelling finding here was that we reduced the variance in error to slightly less than half of that present in the first-order Taylor approximation.

The statistics for Table 1 were collected from approximately 20,000 of the models used to train our Czech-English system (described in §5). After each training iteration, we recorded the log scores of the models according to their first-order and second-order Taylor approximations. These scores were the predictions of moving to the new optimum λ' when the original models were computed under λ. We then compared these approximations to their actual scores by iterating over the translation instances in the training data and re-computing the models at λ'.

3.2 Discounting Approximate Models

Cunei's training procedure, like SMT's training procedure, involves re-translating a small number of development sentences many times to locate the opti-

[3] The multiplicative factor can be eliminated by augmenting the model with one additional feature whose value is $-\log|\eta|$.

[4] To simplify the presentation s and t replace s_i and t_i.

mal λ. Each time we translate a sentence, we generate an n-best list of possible translations according to the model for the current λ. However, the n-best list contains at most a few hundred entries and is a very limited perspective of the search space. Thus, it is common practice to merge n-best lists over all iterations. This technique is necessary for stability, but it creates a new problem. Because our models are approximations, we risk learning λ that is optimal for models approximated from some λ' and not models computed with λ.

To address this second issue, we discount a model's score based on the magnitude of its approximation. Conveniently, the Taylor series is structured such that its degree of approximation is easy to identify. The first term in Equation 5 is the score of the model under λ while the latter terms multiply the change from λ to λ' by the first and second-order derivatives of our model. We measure the magnitude of approximation simply by summing the absolute values of these latter terms.

Figure 1 shows how the average approximation error increases as λ' moves away from λ. The individual data points are numerous and noisy, so we opted to bin the data. The x-axis displays the magnitude of approximation as calculated above. The y-axis represents the average error of the binned models. Each bin is labeled with the range of error it represents, and the bins further from the origin span larger increments due to fewer data points in those regions. Over 70% of the data has a distance magnitude less than 10; the last bin from 50-100 represents less than 1% of the data. We spread out the larger bins to provide a sense of distance, but the x-axis is not to scale due to limited space.

The trend is approximately linear so we compensate for it with a discount proportional to the degree of approximation. The multiplicative factor can be changed by the user, but the default configuration is to decrement the log score for each model by one tenth of its approximation magnitude. In general, the further λ' is from λ, the greater the model will be penalized and the less likely it is to be selected as the best candidate during training. This method intentionally biases the approximation toward underestimates, making the score of each approximation much closer to a lower bound of its actual score.

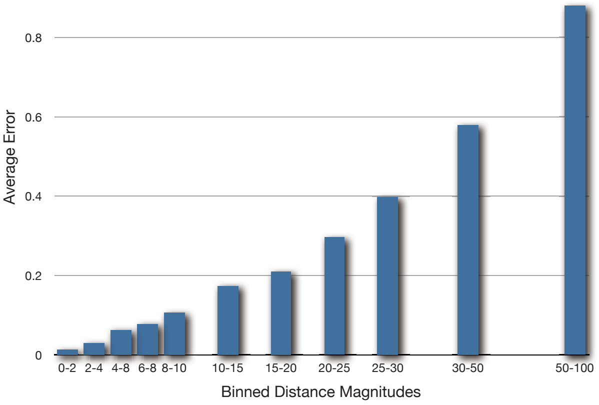

Figure 1: Average Modeling Error Increases Proportionally to Distance Magnitude (x-Axis Not to Scale)

4 Improvement in Training

The motivation behind implementing these techniques was that reducing the approximation error would stabilize and improve training. Training is a notoriously difficult task in machine translation and is an even more complex challenge with our model. In particular, the use of BLEU (Papineni et al., 2002) as the objective function results in a very bumpy error surface with many local minima. An additional, significant source of error due to approximations made during training is not something we can afford.

Table 2 shows five randomized runs of training the same Czech-English system evaluated on the multi-genre test set described at the end of §5. The first-order and second-order approximations identified by the same run were initialized with the same random seed. All runs were initialized with the same λ val-

	First Order BLEU	Second Order BLEU
Run 1	30.03	30.39
Run 2	29.97	30.48
Run 3	30.16	30.43
Run 4	30.14	30.58
Run 5	30.37	29.99
Average	*0.3013*	*0.3037*

Table 2: BLEU Scores on Test Set (Tuned on One-Tenth of the Development Set)

ues and the seed was used to periodically generate new (random) λ values. Nonetheless, the two approximations predict different gradients, so even set to the same random seed, they will usually explore different λ values.

Approximating the models with a second-order Taylor series and then discounting them based on the magnitude of their approximation led to improved BLEU scores on our test set. Specifically, the average gain over five runs was 0.24 BLEU, which is statistically significant. Furthermore, comparing the first column to the second column of Table 2 we see these improvements helped stabilize training. While 'Run 5' reminds us that under-performing outliers still occur, most of the results using the second-order Taylor approximation are very close to one another.

Training is expensive and in most situations we do not have the time to run five separate runs; we are often lucky to run even two randomizations. In practical terms, the difference between the first-order and second-order approximations is quite important–it does not guarantee that a better λ will be found during training, but it does indicate we are more likely to find one.

5 Comparison to Moses

To demonstrate the effectiveness of our model when properly trained, we now compare it to a traditional SMT model. Moses is a widely-used and freely-available SMT toolkit (Koehn et al., 2007). We trained Cunei and Moses on the same data and compared their performance in German to English and Czech to English translation. The corpora for both language pairs included version 6 of the Europarl (Koehn, 2005) and the 2011 edition of parallel news commentary released by the 2011 Workshop on Statistical Machine Translation[5] (WMT). In addition, the Czech to English corpus included CzEng 0.9 (Bojar and Žabokrtský, 2009) which is a collection of many different texts including works of fiction, websites, subtitles, and technical documentation. The available data in Czech and English was quite large, so we sampled one quarter of the parallel text for training.[6] For monolingual data we combined the English text from all the above parallel corpora with years 2010 and 2011 of web-crawled news text released by WMT. Statistics describing our training resources are shown in Table 3.

We applied generic tokenization applicable to Western languages to all the training resources. We then aligned the parallel corpora using GIZA++ (Och and Ney, 2003) in both directions. With the SRILM toolkit (Stolcke, 2002) and the monolingual resources, we built a single 5-gram English language model using Kneser-Ney smoothing. The resulting corpus, word alignments, and language model were provided to Moses and Cunei for training. Each system used its respective phrase extraction and model estimation routines. Specifically, Moses used MERT and Cunei utilized the methods described in this paper for training.

In our first experiment, we selected text from a limited newswire domain. SMT is usually quite good at translating this type of data as the sentences are all similarly structured and the translations are often insensitive to variations in local context. Specifically, both systems were tuned with 632 sentences from the 2009 WMT test set and evaluated on 2489 sentences from the 2010 WMT test set. The results from these newswire experiments are reproduced in Tables 4 and 5. While the difference on the development set is marginal, Cunei clearly outperforms Moses on the unseen test sets with gains of 0.75 BLEU in German-English and 0.51 BLEU in Czech-English. This suggested that Cunei's model is more robust and that we might benefit from more variation within the development and test sets.

Recall that the Czech and English parallel text is diverse and includes a variety of genres. This enabled us to set up a second experiment with our Czech-English system in which we could translate more than just newswire. We created a 763 sentence development set and 1506 sentence test set by uniformly sampling each *genre* from a held-out portion of the CzEng 0.9 corpus. The training data did not change, but we used the multi-genre development set to re-estimate the model weights for both Moses and Cunei. We expected that the use of multiple genres would leverage the strengths of Cunei as its model should adapt to variations within the text by

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

[6] Both Moses and Cunei are capable of handling the full dataset, but in the course of this research we needed to run many experiments so it was important that they run quickly.

	Parallel				*Monolingual*
	German	English	Czech	English	English
Vocab	386,567	120,967	434,361	236,654	1,068,172
Tokens	48,019,666	50,015,721	18,622,983	21,155,241	587,330,675
Sentences	1,822,910		1,658,723		28,900,163

Table 3: Statistics of Training Resources

	Development		*Test*	
	BLEU	NIST	BLEU	NIST
Moses	19.55	6.0262	20.41	6.4794
Cunei	19.91	6.1041	21.16	6.6221

Table 4: Newswire Evaluation of Czech-English

	Development		Test	
	BLEU	NIST	BLEU	NIST
Moses	19.12	5.9616	20.60	6.5102
Cunei	18.98	5.9694	21.11	6.5639

Table 5: Newswire Evaluation of German-English

preferring translation instances similar to the input. Indeed, as shown in Table 6, in this more challenging scenario Cunei strongly outperforms Moses on both the development and test sets.

We provide some examples of actual translations from these experiments in Tables 7 and 8. Overall, the translations from Moses and Cunei are very similar–which is to be expected as they are trained on the same data. Cunei's strengths appear to be slightly better lexical selection such as the use of "loaded" instead of "deploy" when discussing computer drivers. Similarly, Cunei correctly translates the term "state shackles" while Moses instead produces the words "government" and "bonds" (which is understandable given the banking context). In addition, Cunei's correct translation of "jeby" as "crane" suggests that by scoring each translation instance Cunei is able to pick out translations for words that Moses ignores. These modifications, while not dramatic, do consistently improve the quality of translations.

6 Related Work

The motivation for our work was to bring the concept of modeling each translation instance from EBMT into an SMT world. However, the most similar research to ours comes from the other end of the spectrum–training an SMT model that can adapt to new domains.

	Development		Test	
	BLEU	NIST	BLEU	NIST
Moses	30.46	6.7781	27.82	6.9530
Cunei	33.10	7.0221	31.59	7.3256

Table 6: Multi-Genre Evaluation of Czech-English

When dealing with corpora in multiple domains, perhaps the most natural extension of the SMT model is to build multiple models. (Foster and Kuhn, 2007) and (Lu et al., 2007) describe mixture-model approaches in which the corpus is partitioned and traditional SMT models are built on each component. (Lu et al., 2007) weight each component based on its TF-IDF similarity to the test set. (Foster and Kuhn, 2007) explore multiple distance metrics and finds that an EM approach maximizing the likelihood of the test set provides the best mixture weights.

An alternative technique has been to compute a single model, but uniquely weight sections or sentences of the corpora. An early approach by (Hildebrand et al., 2005) uses TF-IDF to compute the similarity between sentences in the training corpus and sentences in the test set. This work actually filters the training corpus so that it is maximally similar to the test set. Later, (Lu et al., 2007) extended this idea and used TF-IDF to re-weight the training corpus based on the test set.

More recent work has focused on learning weights for the corpus. (Shah et al., 2010) performs sampling to learn weights for the corpora and alignments. (Matsoukas et al., 2009) uses a perceptron model with several simple feature functions to assign a weight to each sentence pair in the corpus. These weights are learned as part of a discriminative

Cunei	the troubled us behemoths to touch their state shackles .
Moses	the troubled us behemoths swath of its government bonds .
Reference	the crisis-hit us major banks are breaking free from their state shackles .
Cunei	the countries must significantly more lecturers .
Moses	the countries must have a far more teachers to come .
Reference	the states must employ significantly more lecturers .

Table 7: German-English Newswire Examples

training process that minimizes an objective function on the development set.

All of these weighting schemes simply modify the probability distributions of phrase pairs and do not alter which phrases are extracted from the corpus. In our approach, changing the weights *can* change which phrases are extracted. However, this is only because Cunei locates translation instances at runtime and samples which phrases to extract.

The idea of weighting components in the corpus captures the essence of what we are trying to achieve, but the implementation is quite different. Our approach is most similar to (Matsoukas et al., 2009) in that we use multiple features and learn weights for them based on a development set. However, our features are more specific in that they operate over translation instances and not just sentences.

The most important distinction of our work is that we do *not* calculate the standard SMT feature functions *on top of* weighted sentences or corpora. In all of the related work, the distributions for each feature function are skewed by the weighting of the corpus. In addition, the weights applied to the corpus are separate from the weights applied to the feature functions of the SMT model. Our approach constructs a single unified model.

Cunei	right or wrong , i did n't want this !
Moses	well or badly , i did n't want this !
Reference	right or wrong it was not what i wanted !
Cunei	what with all those paper cranes ?
Moses	what with all those paper jeby ?
Reference	what 's with all these paper cranes ?
Cunei	it was clear that i 'll have to steal much more .
Moses	it was obvious that it for me to have to steal it much more .
Reference	it was clear that i was going to need to steal more stuff .
Cunei	driver could not be loaded .
Moses	driver can not be deploy .
Reference	the driver could not load .

Table 8: Czech-English Multi-Genre Examples

7 Conclusion

While research in the field of machine translation has been dominated lately by Statistical MT, we still believe it is beneficial for the research community to explore and understand a diversity of modeling approaches. Training MT models in general is difficult due to many local minima, but our model particularly exacerbates the problem by requiring an approximation in which the features are dependent on λ. The techniques described in this paper–using a second-order Taylor approximation and discounting models proportional to their approximation–have not only made training our model feasible, but they have enabled us to effectively leverage the strengths of modeling translation instances. In two different evaluations we found that our approach yielded higher quality translations than the traditional SMT approach. First, Cunei outperformed Moses on Czech-English and German-English translation of newswire text–a scenario in which SMT usually excels. Second, when we created a more complex evaluation set by varying the genres of translation in Czech-English, Cunei outperformed Moses by 3.77 BLEU. These results encourage us to further explore this modeling approach and enrich Cunei with more instance-specific features.

References

Ondřej Bojar and Zdeněk Žabokrtský. 2009. CzEng 0.9: Large parallel treebank with rich annotation. *Prague Bulletin of Mathematical Linguistics*, 92.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 263–270, Ann Arbor, USA, June.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Prague, Czech Republic, June. Association for Computational Linguistics.

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proceedings of the Tenth Annual Conference of the European Assocation for Machine Translation*, pages 133–142, Budapest, Hungary, May.

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*, pages 177–180, Prague, Czech Republic, June.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit X Proceedings*, pages 79–86, Phuket, Thailand, September.

Yajuan Lu, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 343–350, Prague, Czech Republic, June.

Spyros Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *2009 Conference on Empirical Methods in Natural Language Processing*, pages 708–717, Suntec, Singapore, August.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449, December.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, USA, July.

Aaron B. Phillips and Ralf D. Brown. 2009. Cunei Machine Translation Platform: System description. In Mikel L. Forcada and Andy Way, editors, *Proceedings of the 3rd Workshop on Example-Based Machine Translation*, pages 29–36, Dublin, Ireland, November.

Aaron B. Phillips. 2010. The Cunei Machine Translation Platform for WMT '10. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 155–160, Uppsala, Sweden, July. Association for Computational Linguistics.

Kashif Shah, Loïc Barrault, and Holger Schwenk. 2010. Translation model adaptation by resampling. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 392–399, Uppsala, Sweden, July. Association for Computational Linguistics.

David A. Smith and Jason Eisner. 2006. Minimum risk annealing for training log-linear models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 787–794, Sydney, Australia, July.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *7th International Conference on Spoken Language Processing*, pages 901–904, Denver, USA, September.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 138–141, New York City, USA, June. Association for Computational Linguistics.