The CMU-ARK German-English Translation System

Chris Dyer Kevin Gimpel Jonathan H. Clark Noah A. Smith
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, 15213, USA
{cdyer,kgimpel,jhclark,nasmith}@cs.cmu.edu

Abstract

This paper describes the German-English translation system developed by the ARK research group at Carnegie Mellon University for the Sixth Workshop on Machine Translation (WMT11). We present the results of several modeling and training improvements to our core hierarchical phrase-based translation system, including: feature engineering to improve modeling of the derivation structure of translations; better handing of OOVs; and using development set translations into other languages to create additional pseudo-references for training.

1 Introduction

We describe the German-English translation system submitted to the shared translation task in the Sixth Workshop on Machine Translation (WMT11) by the ARK research group at Carnegie Mellon University.[1] The core translation system is a hierarchical phrase-based machine translation system (Chiang, 2007) that has been extended in several ways described in this paper.

Some of our innovations focus on modeling. Since German and English word orders can diverge considerably, particularly in non-matrix clauses, we focused on feature engineering to improve the modeling of long-distance relationships, which are poorly captured in standard hierarchical phrase-based translation models. To do so, we developed features that assess the goodness of the source language parse tree under the translation grammar (rather than of a "linguistic" grammar). To train the feature weights, we made use of a novel two-phase training algorithm that incorporates a probabilistic training objective and standard minimum error training (Och, 2003). These segmentation features were supplemented with a 7-gram class-based language model, which more directly models long-distance relationships. Together, these features provide a modest improvement over the baseline and suggest interesting directions for future work. While our work on parse modeling was involved and required substantial changes to the training pipeline, some other modeling enhancements were quite simple: for example, improving how out-of-vocabulary words are handled. We propose a very simple change, and show that it provides a small, consistent gain.

On the training side, we had two improvements over our baseline system. First, we were inspired by the work of Madnani (2010), who showed that when training to optimize BLEU (Papineni et al., 2002), overfitting is reduced by supplementing a single human-generated reference translation with additional computer-generated references. We generated supplementary pseudo-references for our development set (which is translated into many languages, but once) by using MT output from a secondary Spanish-English translation system. Second, following Foster and Kuhn (2009), we used a secondary development set to select from among many optimization runs, which further improved generalization.

We largely sought techniques that did not require language-specific resources (e.g., treebanks, POS

[1] http://www.ark.cs.cmu.edu

annotations, morphological analyzers). An exception is a compound segmentation model used for preprocessing that was trained on a corpus of manually segmented German. Aside from this, no further manually annotated data was used, and we suspect many of the improvements described here can be had in other language pairs. Despite avoiding language-specific resources and using only the training data provided by the workshop, an extensive manual evaluation determined that the outputs produced were of significantly higher quality than both statistical and rule-based systems that made use of language-specific resources (Callison-Burch et al., 2011).

2 Baseline system and data

Our translation system is based on a hierarchical phrase-based translation model (Chiang, 2007), as implemented in the `cdec` decoder (Dyer et al., 2010). Since German is a language that makes productive use of "closed" compounds (compound words written as a single orthographic token), we use a CRF segmentation model of to evaluate the probability of all possible segmentations, encoding the most probable ones compactly in a lattice (Dyer, 2009). For the purposes of grammar induction, the single most probable segmentation of each word in the source side of the parallel training data under the model was inferred.

The parallel data were aligned using the Giza++ implementation of IBM Model 4 run in both directions and then symmetrized using the `grow-diag-final-and` heuristic (Och and Ney, 2002; Brown et al., 1993; Koehn et al., 2003). The aligned corpus was encoded as a suffix array (Lopez, 2008) and lattice-specific grammars (containing just the rules that are capable of matching spans in the input lattice) were extracted for each sentence in the test and development sets, using the heuristics recommended by Chiang (2007).

A 4-gram modified Kneser-Ney language model (Chen and Goodman, 1996) was constructed using the SRI language modeling toolkit (Stolcke, 2002) from the English side of the parallel text, the monolingual English data, and the English version 4 Gigaword corpus (Parker et al., 2009). Since there were many duplicate segments in the training data (much of which was crawled from the web), duplicate segments and segments longer than 100 words were removed. Inference was carried out using the language modeling library described by Heafield (2011).

The `newstest-2009` set (with the 500 longest segments removed) was used for development,[2] and `newstest-2010` was used as a development test set. Results in this paper are reported on the devtest set using uncased BLEU_4 with a single reference translation. Minimum error rate training (Och, 2003) was used to optimize the parameters of the system to maximize BLEU on the development data, and inference was performed over a pruned hypergraph representation of the translation hypothesis space (Kumar et al., 2009).

For the experiments reported in this paper, Viterbi (max-derivation) decoding was used. The system submitted for manual evaluation used segment-level MBR decoding with $1 - \text{BLEU}$ as the loss function, approximated over a 500-best list for each sentence. This reliably results in a small but consistent improvement in translation quality, but is much more time consuming to compute (Kumar and Byrne, 2004).

3 Source parse structure modeling

Improving phrase-based translation systems is challenging in part because our intuitions about what makes a "good" phrase or translation derivation are often poor. For example, restricting phrases and rules to be consistent with syntactic constituents consistently harms performance (Chiang, 2007; Galley et al., 2006; Koehn et al., 2003), although our intuitions might suggest this is a reasonable thing to do. On the other hand, it has been shown that incorporating syntactic information in the form of features *can* lead to improved performance (Chiang, 2010; Gimpel and Smith, 2009; Marton and Resnik, 2008). Syntactic features that are computed by assessing the overlap of the translation parse with a linguistic parse can be understood to improve translation because they lead to a better model of what a "correct" parse of the source sentence is *under the translation grammar*.

Like the "soft syntactic features" used in pre-

[2] Removing long segments substantially reduces training time and does not appear to negatively affect performance.

vious work (Marton and Resnik, 2008; Chiang et al., 2008), we propose features to assess the tree structure induced during translation. However, unlike that work, we do not rely on linguistic source parses, but instead only make use of features that are directly computable from the source sentence and the parse structure being considered in the decoder. In particular, we take inspiration from the model of Klein and Manning (2002), which models constituency in terms of the *contexts* that rule productions occur in. Additionally, we make use of salient aspects of the spans being dominated by a nonterminal, such as the words at the beginning and end of the span, and the length of the span. Importantly, the features do not rely on the target words being predicted, but only look at the structure of the translation derivation. As such, they can be understood as *monolingual parse features*.[3]

Table 1 lists the feature templates that were used.

Template	Description
CTX:f_{i-1}, f_j	context bigram
CTX:f_{i-1}, f_j, x	context bigram + NT
CTX:$f_{i-1}, f_j, x, (j-i)$	context bigram + NT + len
LU:f_{i-1}	left unigram
LB:f_{i-1}, f_i	left bigram (overlapping)
RU:f_j	right unigram
RB:f_{j-1}, f_j	right bigram (overlapping)

Table 1: Context feature templates for features extracted from every translation rule used; i and j indicate hypothesized constituent span, x is its nonterminal category label (in our grammar, X or S), and f_k is the k^{th} word of the source sentence, with $f_{<1} = \langle \text{s} \rangle$ and $f_{>|\mathbf{f}|} = \langle /\text{s} \rangle$. If a word f_k is not among the 1000 most frequent words in the training corpus, it is replaced by a special unknown token. The SMALLCAPS prefixes prevent accidental feature collisions.

3.1 Two-phase discriminative learning

The parse features just introduced are numerous and sparse, which means that MERT can not be used to infer their weights. Instead, we require a learning algorithm that can cope with millions of features and avoid overfitting, perhaps by eliminating most of the features and keeping only the most valuable (which would also keep the model compact). Furthermore, we would like to be able to still target the BLEU measure of translation quality during learning. While large-scale discriminative training for machine translation is a widely studied problem (Hopkins and May, 2011; Li and Eisner, 2009; Devlin, 2009; Blunsom et al., 2008; Watanabe et al., 2007; Arun and Koehn, 2007; Liang et al., 2006), no tractable algorithm exists for learning a large number of feature weights while directly optimizing a corpus-level metric like BLEU. Rather than resorting to a decomposable approximation, we have explored a new two-phase training algorithm in development of this system.

The two-phase algorithm works as follows. In phase 1, we use a non-BLEU objective to train a translation model that includes the large feature set. Then, we use this model to compute a small number of coarse "summary features," which summarize the "opinion" of the first model about a translation hypothesis in a low dimensional space. Then, in the second training pass, MERT is used to determine how much weight to give these summary features together with the other standard coarse translation features. At test time, translation becomes a multi-step process as well. The hypothesis space is first scored using the phase-1 model, then summary features are computed, then the hypothesis space is rescored with the phase-2 model. As long as the features used factor with the edges in the translation space (which ours do), this can be carried out in linear time in the size of the translation forest.

3.1.1 Phase 1 training

For the first model, which includes the sparse parse features, we learn weights in order to optimize penalized conditional log likelihood (Blunsom et al., 2008). We are specifically interested in modeling an *unobserved* variable (i.e., the parse tree underlying a translation derivation), this objective is quite natural, since probabilistic models offer a principled account of unobserved data. Furthermore, because our features factor according to edges in the translation forest (they are "stateless" in standard MT terminology), there are efficient dynamic programming algorithms that can be used to exactly compute the expected values of the features (Lari and Young, 1990), which are necessary for computing the gradients used in optimization.

[3] Similar features have been proposed for use in discriminative monolingual parsing models (Taskar et al., 2004).

We are therefore optimizing the following objective, given a set $\mathcal{T}$ of parallel training sentences:

$$\mathcal{L} = \lambda R(\theta) - \sum_{\langle \mathbf{f}, \mathbf{e} \rangle \in \mathcal{T}} \log \sum_{\mathbf{d}} p_\theta(\mathbf{e}, \mathbf{d} \mid \mathbf{f})$$

$$\text{where } p_\theta(\mathbf{e}, \mathbf{d} \mid \mathbf{f}) = \frac{\exp \theta^\top \mathbf{h}(\mathbf{f}, \mathbf{e}, \mathbf{d})}{Z(\mathbf{f})} \quad ,$$

where $\mathbf{d}$ is a variable representing the *unobserved* synchronous parses giving rise to the pair of sentences $\langle \mathbf{f}, \mathbf{e} \rangle$, and where $R(\theta)$ is a penalty that favors less complex models. Since we not only want to prevent over fitting but also want a small model, we use $R(\theta) = \sum_k |\theta_k|$, the ℓ_1 norm, which forces many parameters to be exactly 0.

Although $\mathcal{L}$ is not convex in θ (on account of the latent derivation variable), we make use of an online stochastic gradient descent algorithm that imposes an ℓ_1 penalty on the objective (Tsuruoka et al., 2009). Online algorithms are often effective for non-convex objectives (Liang and Klein, 2009).

We selected 12,500 sentences randomly from the news-commentary portion of the training data to use to train the latent variable model. Using the standard rule extraction heuristics (Chiang, 2007), 9,967 of the sentence pairs could be derived.[4] In addition to the parse features describe above, the standard phrase features (relative frequency and lexical translation probabilities), and a rule count feature were included. Training was run for 48 hours on a single machine, which resulted in 8 passes through the training data, instantiating over 8M unique features. The regularization strength λ was chosen so that approximately $10,000$ (of the 8M) features would be non-zero.[5]

3.1.2 Summary features

As outlined above, the phase 1 model will be incorporated into the final translation model using a low dimensional "summary" of its opinion. Because we are using a probabilistic model, posterior probabilities (given the source sentence $\mathbf{f}$) under the parsing model are easily defined and straightforward to compute with dynamic programming. We made use of four summary features: the posterior log probability $\log p_\theta(\mathbf{e}, \mathbf{d}|\mathbf{f})$; for every rule $r \in \mathbf{d}$, the probability of its span being a constituent under the parse model; the probabilities that *some* span starts at the r's starting index, or that some rule ends at r's ending index.

Once these summary features have been computed, the sparse features are discarded, and the summary features are reweighted using coefficients learned by MERT, together with the standard MT features (language model, word penalty, etc.). This provides a small improvement over our already very strong baseline, as the first two rows in Table 2 show.

Condition	BLEU
baseline	25.0
+ parse features	25.2
+ parse features + 7-gram LM	25.4

Table 2: Additional features designed to improve model of long-range reordering.

3.2 7-gram class-based LM

The parsing features above were intended to improve long range reordering quality. To further support the modeling of larger spans, we incorporated a 7-gram class-based language model. Automatic word clusters are attractive because they can be learned for any language without supervised data, and, unlike part-of-speech annotations, each word is in only a single class, which simplifies inference. We performed Brown clustering (Brown et al., 1992) on 900k sentences from our language modeling data (including the news commentary corpus and a subset of Gigaword). We obtained 1,000 clusters using an implementation provided by Liang (2005),[6] as Turian et al. (2010) found that relatively large numbers clusters gave better performance for information extraction tasks. We then replaced words with their clusters in our language modeling data and built a 7-gram LM with Witten-Bell smoothing (Witten and Bell, 1991).[7] The last two rows of Ta-

[4] When optimizing conditional log likeligood, it is necessary to be able to exactly derive the training pair. See Blunsom et al. (2008) for more information.

[5] Ideally, λ would have been tuned to optimize held-out likelihood or BLEU; however, the evaluation deadline prevented us from doing this.

[6] `http://www.cs.berkeley.edu/~pliang/software`

[7] The distributional assumptions made by the more commonly used Kneser-Ney estimator do not hold in the word-

ble 2 shows that in conjunction with the source parse features, a slight improvement comes from including the 7-gram LM.

4 Non-translating tokens

When two languages share a common alphabet (as German and English largely do), it is often appropriate to leave some tokens untranslated when translating. Named entities, numbers, and graphical elements such as emoticons are a few common examples of such "non-translating" elements. To ensure that such elements are well-modeled, we augment our translation grammar so that every token in the input can translate as itself and add a feature that counts the number of times such self-translation rules are used in a translation hypothesis. This is in contrast to the behavior of most other decoders, such as Moses, which only permit a token to translate as itself if it is learned from the training data, or if there is no translation in the phrase table at all.

Since many non-translating tokens are out-of-vocabulary (OOV) in the target LM, we also add a feature that fires each time the LM encounters a word that is OOV.[8] This behavior be understood as discriminatively learning the unknown word penalty that is part of the LM. Again, this is in contrast to the behavior of other decoders, which typically add a fixed (and very large) cost to the LM feature for every OOV. Our multi-feature parameterization permits the training algorithm to decide that, e.g., some OOVs are acceptable if they occur in a "good" context rather than forcing the decoder to avoid them at all costs. Table 3 shows that always providing a non-translating translation option together with a discriminative learned OOV feature improves the quality of German-English translation.[9]

Condition	BLEU
−OOV (baseline)	24.6
+OOV and non-translating rules	25.0

Table 3: Effect of discriminatively learned penalties for OOV words.

classified corpus.

[8]When multiple LMs are used, there is an extra OOV feature for each LM.

[9]Both systems were trained using the human+ES-EN reference set described below (§5).

5 Computer-generated references

Madnani (2010) shows that models learned by optimizing BLEU are liable to overfit if only a single reference is used, but that this overfitting can be mitigated by supplementing the single reference with supplemental computer-generated references produced by paraphrasing the human reference using a whole-sentence statistical paraphrase system. These computer-generated paraphrases are just used to compute "better" BLEU scores, but not directly as examples of target translations.

Although we did not have access to a paraphrase generator, we took advantage of the fact that our development set (`newstest-2009`) was translated into several languages other than English. By translating these back into English, we hypothesized we would get suitable pseudo-references that could be used in place of computer-generated paraphrases. Table 4 shows the results obtained on our held-out test set simply by altering the reference translations used to score the development data. These systems all contain the OOV features described above.

Condition	BLEU
1 human	24.7
1 human + ES-EN	**25.0**
1 human + FR-EN	24.0
1 human + ES-EN + FR-EN	24.2

Table 4: Effect of different sets of reference translations used during tuning.

While the effect is somewhat smaller than Madnani (2010) reports using a sentential paraphraser, the extremely simple technique of adding the output of a Spanish-English (ES-EN) system was found to consistently improve the quality of the translations of the held-out data. However, a comparable effect was not found when using references generated from a French-English (FR-EN) translation system, indicating that the utility of this technique must be assessed empirically and depends on several factors.

6 Case restoration

Our translation system generates lowercased output, so we must restore case as a post-processing step. We do so using a probabilistic transducer as implemented in SRILM's `disambig` tool. Each

lowercase token in the input can be mapped to a cased variant that was observed in the target language training data. Ambiguities are resolved using a language model that predicts true-cased sentences.[10] We used the same data sources to construct this model as were used above. During development, it was observed that many named entities that did not require translation required some case change, from simple uppercasing of the first letter, to more idiosyncratic casings (e.g., *iPod*). To ensure that these were properly restored, even when they did not occur in the target language training data, we supplement the true-cased LM training data and case transducer training data with the German *source* test set.

Condition	BLEU (Cased)
English-only	24.1
English+test-set	24.3

Table 5: Effect of supplementing recasing model training data with the test set *source*.

7 Model selection

Minimum error rate training (Och, 2003) is a stochastic optimization algorithm that typically finds a different weight vector each time it is run. Foster and Kuhn (2009) showed that while the variance on the development set objective may be narrow, the held-out test set variance is typically much greater, but that a secondary development set can be used to select a system that will have better generalization. We therefore replicated MERT 6 times and selected the output that performed best on NEWSTEST-2010. Since we had no additional blind test set, we cannot measure what the impact is. However, the BLEU scores we selected on varied from 25.4 to 26.1.

8 Summary

We have presented a summary of the enhancements made to a hierarchical phrase-based translation system for the WMT11 shared translation task. Some of our results are still preliminary (the source parse model), but a number of changes we made were quite simple (OOV handling, using MT output to provide additional references for training) and also led to improved results.

[10]The model used is $p(\mathbf{y} \mid \mathbf{x})p(\mathbf{y})$. While this model is somewhat unusual (the conditional probability is backwards from a noisy channel model), it is a standard and effective technique for case restoration.

Acknowledgments

This research was supported in part by the NSF through grant IIS-0844507, the U. S. Army Research Laboratory and the U. S. Army Research Office under contract/grant number W911NF-10-1-0533, and Sandia National Laboratories (fellowship to K. Gimpel). We thank the anonymous reviewers for their thorough feedback.

References

A. Arun and P. Koehn. 2007. Online learning methods for discriminative training of phrase based statistical machine translation. In *Proc. of MT Summit XI*.

P. Blunsom, T. Cohn, and M. Osborne. 2008. A discriminative latent variable model for statistical machine translation. In *Proc. of ACL-HLT*.

P. F. Brown, P. V. de Souza, R. L. Mercer, V. J. Della Pietra, and J. C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479.

P. F. Brown, V. J. Della Pietra, S. A. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.

C. Callison-Burch, P. Koehn, C. Monz, and O. F. Zaidan. 2011. Findings of the 2011 workshop on statistical machine translation. In *Proc. of the Sixth Workshop on Statistical Machine Translation*.

S. F. Chen and J. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proc. of ACL*, pages 310–318.

D. Chiang, Y. Marton, and P. Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proc. EMNLP*, pages 224–233.

D. Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

D. Chiang. 2010. Learning to translate with source and target syntax. In *Proc. of ACL*, pages 1443–1452.

J. Devlin. 2009. Lexical features for statistical machine translation. Master's thesis, University of Maryland.

C. Dyer, A. Lopez, J. Ganitkevitch, J. Weese, F. Ture, P. Blunsom, H. Setiawan, V. Eidelman, and P. Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proc. of ACL (demonstration session)*.

C. Dyer. 2009. Using a maximum entropy model to build segmentation lattices for MT. In *Proc. of NAACL*.

G. Foster and R. Kuhn. 2009. Stabilizing minimum error rate training. *Proc. of WMT*.

M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc. of ACL*, pages 961–968.

K. Gimpel and N. A. Smith. 2009. Feature-rich translation by quasi-synchronous lattice parsing. In *Proc. of EMNLP*, pages 219–228.

K. Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proc. of the Sixth Workshop on Statistical Machine Translation*.

M. Hopkins and J. May. 2011. Tuning as ranking. In *Proc. of EMNLP*.

D. Klein and C. D. Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proc. of ACL*, pages 128–135.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. of NAACL*.

S. Kumar and W. Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Processings of HLT-NAACL*.

S. Kumar, W. Macherey, C. Dyer, and F. Och. 2009. Efficient minimum error rate training and minimum bayes-risk decoding for translation hypergraphs and lattices. In *Proc. of ACL-IJCNLP*.

K. Lari and S. Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*.

Z. Li and J. Eisner. 2009. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proc. of EMNLP*, pages 40–51.

P. Liang and D. Klein. 2009. Online EM for unsupervised models. In *Proc. of NAACL*.

P. Liang, A. Bouchard-Côté, D. Klein, and B. Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proc. of ACL*.

P. Liang. 2005. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

A. Lopez. 2008. Tera-scale translation models via pattern matching. In *Proc. of COLING*.

N. Madnani. 2010. *The Circle of Meaning: From Translation to Paraphrasing and Back*. Ph.D. thesis, Department of Computer Science, University of Maryland College Park.

Y. Marton and P. Resnik. 2008. Soft syntactic constraints for hierarchical phrased-based translation. In *Proc. of ACL*, pages 1003–1011, Columbus, Ohio.

F. J. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL*, pages 295–302.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL*, pages 160–167.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*.

R. Parker, D. Graff, J. Kong, K. Chen, and K. Maeda. 2009. English gigaword fourth edition.

A. Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Intl. Conf. on Spoken Language Processing*.

B. Taskar, D. Klein, M. Collins, D. Koller, and C. Manning. 2004. Max-margin parsing. In *Proc. of EMNLP*.

Y. Tsuruoka, J. Tsujii, and S. Ananiadou. 2009. Stochastic gradient descent training for l_1-regularized log-linear models with cumulative penalty. In *Proc. of ACL-IJCNLP*.

J. Turian, L. Ratinov, and Y. Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proc. of ACL*, pages 384–394.

T. Watanabe, J. Suzuki, H. Tsukuda, and H. Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proc. of EMNLP*.

I. H. Witten and T. C. Bell. 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Trans. Information Theory*, 37(4).