Kriya - The SFU System for Translation Task at WMT-12

Majid Razmara and Baskaran Sankaran and Ann Clifton and Anoop Sarkar
School of Computing Science
Simon Fraser University
8888 University Drive
Burnaby BC. V5A 1S6. Canada
{razmara, baskaran, aca69, anoop}@cs.sfu.ca

Abstract

This paper describes our submissions for the WMT-12 translation task using *Kriya* - our hierarchical phrase-based system. We submitted systems in French-English and English-Czech language pairs. In addition to the baseline system following the standard MT pipeline, we tried *ensemble* decoding for French-English. The ensemble decoding method improved the BLEU score by 0.4 points over the baseline in newstest-2011. For English-Czech, we segmented the Czech side of the corpora and trained two different segmented models in addition to our baseline system.

1 Baseline Systems

Our shared task submissions are trained in the hierarchical phrase-based model (Chiang, 2007) framework. Specifically, we use *Kriya* (Sankaran et al., 2012) - our in-house Hiero-style system for training and decoding. We now briefly explain the baseline systems in French-English and English-Czech language pairs.

We use GIZA++ for word alignments and the Moses (Koehn et al., 2007) phrase-extractor for extracting the initial phrases. The translation models are trained using the rule extraction module in Kriya. In both cases, we pre-processed the training data by running it through the usual pre-processing pipeline of tokenization and lowercasing.

For French-English baseline system, we trained a simplified hierarchical phrase-based model where the right-hand side can have at most one non-terminal (denoted as 1NT) instead of the usual two non-terminal (2NT) model. In our earlier experiments we found the 1NT model to perform comparably to the 2NT model for close language pairs such as French-English (Sankaran et al., 2012) at the same time resulting in a smaller model. We used the shared-task training data consisting of Europarl (v7), News commentary and UN documents for training the translation models having a total of 15 M sentence pairs (we did not use the Fr-En Giga parallel corpus for the training). We trained a 5-gram language model for English using the English Gigaword (v4).

For English-Czech, we trained a standard Hiero model that has up to two non-terminals on the right-hand side. We used the Europarl (v7), news commentary and CzEng (v0.9) corpora having 7.95M sentence pairs for training translation models. We trained a 5-gram language model using the Czech side of the parallel corpora and did not use the Czech monolingual corpus.

The baseline systems use the following 8 standard Hiero features: rule probabilities $p(e|f)$ and $p(f|e)$; lexical weights $p_l(e|f)$ and $p_l(f|e)$; word penalty, phrase penalty, language model and glue rule penalty.

1.1 LM Integration in Kriya

The kriya decoder is based on a modified CYK algorithm similar to that of Chiang (2007). We use a novel approach in computing the language model (LM) scores in Kriya, which deserves a mention here.

The CKY decoder in Hiero-style systems can freely combine target hypotheses generated in inter-

Proceedings of the 7th Workshop on Statistical Machine Translation, pages 356–361,
Montréal, Canada, June 7-8, 2012. ©2012 Association for Computational Linguistics

mediate cells with hierarchical rules in the higher cells. Thus the generation of the target hypotheses are fragmented and out of order in Hiero, compared to the left to right order preferred by n-gram language models.

This leads to challenges in estimating LM scores for partial target hypotheses and this is typically addressed by adding a sentence initial marker (<s>) to the beginning of each derivation path.[1] Thus the language model scores for the hypothesis in the intermediate cell are approximated, with the true language model score (taking into account sentence boundaries) being computed in the last cell that spans the entire source sentence.

Kriya uses a novel idea for computing LM scores: for each of the target hypothesis fragment, it finds the best position for the fragment in the final sentence and uses the corresponding score. Specifically, we compute three different scores corresponding to the three states where the fragment can end up in the final sentence, viz. sentence initial, middle and final and choose the best score. Thus given a fragment t_f consisting of a sequence of target tokens, we compute LM scores for (i) <s> t_f, (ii) t_f and (iii) t_f </s> and use the best score (*only*) for pruning.[2] While this increases the number of LM queries, we exploit the language model state information in KenLM (Heafield, 2011) to optimize the queries by saving the scores for the unchanged states. Our earlier experiments showed significant reduction in search errors due to this approach, in addition to a small but consistent increase in BLEU score (Sankaran et al., 2012).

2 French-English System

In addition to the baseline system, we also trained separate systems for *News* and *Non-News* genres for applying *ensemble* decoding (Razmara et al., 2012). The news genre system was trained only using the news-commentary corpus (about 137K sentence pairs) and the non-news genre system was trained on the Europarl and UN documents data (14.8M sentence pairs). The ensemble decoding framework combines the models of these two systems dynamically when decoding the testset. The idea is to effectively use the small amount of news genre data in order to maximize the performance on the news-based testsets. In the following sections, we explain in broader detail how this system combination technique works as well as the details of this experiment and the evaluation results.

2.1 Ensemble Decoding

In the ensemble decoding framework we view translation task as a domain mixing problem involving news and non-news genres. The official training data is from two major sources: news-commentary data and Europarl/UN data and we hope to exploit the distinctive nature of the two genres. Given that the news data is smaller comparing to parliamentary proceedings data, we could tune the ensemble decoding to appropriately boost the weight for the news genre mode during decoding. The ensemble decoding approach (Razmara et al., 2012) takes advantage of multiple translation models with the goal of constructing a system that outperforms all the component models. The key strength of this system combination method is that the systems are combined dynamically at decode time. This enables the decoder to pick the best hypotheses for each span of the input.

In ensemble decoding, given a number of translation systems which are already trained and tuned, all of the hypotheses from component models are used in order to translate a sentence. The scores of such rules are combined in the decoder (i.e. CKY) using various mixture operations to assign a single score to them. Depending on the mixture operation used for combining the scores, we would get different mixture scores.

Ensemble decoding extends the log-linear framework which is found in state-of-the-art machine translation systems. Specifically, the probability of a phrase-pair $(\bar{e}, \bar{f})$ in the ensemble model is:

$$p(\bar{e} \mid \bar{f}) \propto \exp\left(\underbrace{\mathbf{w_1} \cdot \boldsymbol{\phi_1}}_{1^{st} \text{ model}} \oplus \underbrace{\mathbf{w_2} \cdot \boldsymbol{\phi_2}}_{2^{nd} \text{ model}} \oplus \cdots \right)$$

[1] Alternately systems add sentence boundary markers (<s> and </s>) to the training data so that they are explicitly present in the translation and language models. While this can speed up the decoding as the cube pruning is more aggressive, it also limits the applicability of rules having the boundary contexts.

[2] This ensures the the LM score estimates are never underestimated for pruning. We retain the LM score for fragment (case ii) for estimating the score for the full candidate sentence later.

where $\oplus$ denotes the mixture operation between two or more model scores.

Mixture operations receive two or more scores (probabilities) and return the mixture score (probability). In this section, we explore different options for this mixture operation.

Weighted Sum (wsum): in *wsum* the ensemble probability is proportional to the weighted sum of all individual model probabilities.

$$p(\bar{e} \,|\, \bar{f}) \;\; \propto \;\; \sum_{m}^{M} \lambda_m \, \exp\big(\mathbf{w}_m \cdot \boldsymbol{\phi}_m\big)$$

where m denotes the index of component models, M is the total number of them and λ_i is the weight for component i.

Weighted Max (wmax): where the ensemble score is the weighted max of all model scores.

$$p(\bar{e} \,|\, \bar{f}) \;\; \propto \;\; \max_{m} \big(\lambda_m \, \exp\big(\mathbf{w}_m \cdot \boldsymbol{\phi}_m\big)\big)$$

Product (prod): in *prod*, the probability of the ensemble model or a rule is computed as the product of the probabilities of all components (or equally the sum of log-probabilities). When using this mixture operation, ensemble decoding would be a generalization of the log-linear framework over multiple models. Product models can also make use of weights to control the contribution of each component. These models are generally known as *Logarithmic Opinion Pools (LOPs)* where:

$$p(\bar{e} \,|\, \bar{f}) \;\; \propto \;\; \exp\big(\sum_{m}^{M} \lambda_m \, \mathbf{w}_m \cdot \boldsymbol{\phi}_m\big)$$

Model Switching: in model switching, each cell in the CKY chart gets populated only by rules from one of the models and the other models' rules are discarded. This is based on the hypothesis that each component model is an expert on different parts of sentence. In this method, we need to define a binary indicator function $\delta(\bar{f}, m)$ for each span and component model.

$$\delta(\bar{f}, m) = \begin{cases} 1, & m = \underset{n \in M}{\text{argmax}} \; \psi(\bar{f}, n) \\ 0, & \text{otherwise} \end{cases}$$

The criteria for choosing a model for each cell, $\psi(\bar{f}, n)$, could be based on:

Max: for each cell, the model that has the highest weighted top-rule score wins:

$$\psi(\bar{f}, n) = \lambda_n \max_{\bar{e}} (\mathbf{w}_n \cdot \boldsymbol{\phi}_n(\bar{\mathbf{e}}, \bar{\mathbf{f}}))$$

Sum: Instead of comparing only the score of the top rules, the model with the highest weighted sum of the probability of the rules wins (taking into account the *ttl*(translation table limit) limit on the number of rules suggested by each model for each cell):

$$\psi(\bar{f}, n) = \lambda_n \sum_{\bar{e}} \exp\big(\mathbf{w}_n \cdot \boldsymbol{\phi}_n(\bar{\mathbf{e}}, \bar{\mathbf{f}})\big)$$

The probability of each phrase-pair $(\bar{e}, \bar{f})$ is computed as:

$$p(\bar{e} \,|\, \bar{f}) = \sum_{m} \delta(\bar{f}, m) \; p_m(\bar{e} \,|\, \bar{f})$$

Since log-linear models usually look for the best derivation, they do not need to normalize the scores to form probabilities. Therefore, the scores that different models assign to each phrase-pair may not be in the same scale. Therefore, mixing their scores might wash out the information in one (or some) of the models. We applied a heuristic to deal with this problem where the scores are normalized over a shorter list. So the list of rules coming from each model for a certain cell in the CKY chart is normalized before getting mixed with other phrase-table rules. However, experiments showed using normalized scores hurts the BLEU score radically. So we use the normalized scores only for pruning and for mixing the actual scores are used.

As a more principled way, we used a toolkit, CONDOR (Vanden Berghen and Bersini, 2005), to optimize the weights of our component models on a dev-set. CONDOR, which is publicly available, is a direct optimizer based on Powell's algorithm that does not require explicit gradient information for the objective function.

2.2 Experiments and Results

As mentioned earlier all the experiments reported for French-English use a simpler Hiero translation

Method	Devset	Test-11	Test-12
Baseline Hiero	26.03	27.63	**28.15**
News data	24.02	26.47	26.27
Non-news data	26.09	27.87	28.15
Ensemble PROD	25.66	**28.25**	28.09

Table 1: French-English BLEU scores. Best performing setting is shown in **Boldface**.

model having at most one non-terminal (1NT) on the right-hand side. We use 7567 sentence pairs from news-tests 2008 through 2010 for tuning and use news-test 2011 for testing in addition to the 2012 test data. The feature weights were tuned using MERT (Och, 2003) and we report the devset (IBM) BLEU scores and the testset BLEU scores computed using the official evaluation script (mteval-v11b.pl).

The results for the French-English experiments are reported in Table 1. We note that both baseline Hiero model and the model trained from the non-news genre get comparable BLEU scores. The news genre model however gets a lesser BLEU score and this is to be expected due to the very small training data available for this genre.

Table 2 shows the results of applying various mixture operations on the devset and testset, both in normalized (denoted by Norm.) and un-normalized settings (denoted by Base). We present results for these mixture operations using uniform weights (i.e. untuned weights) and for PROD we also present the results using the weights optimized by CONDOR. Most of the mixture operations outperform the Test-11 BLEU of the baseline models (shown in Table 1) even with uniform (untuned) weights. We took the best performing operation (i.e. PROD) and tuned its component weights using our optimizer which lead to 0.26 points improvement over its uniform-weight version.

The last row in Table 1 reports the BLEU score for this mixture operation with the tuned weights on the Test-12 dataset and it is marginally less than the baseline model. While this is disappointing, this also runs counter to our empirical results from other datasets. We are currently investigating this aspect as we hope to improve the robustness and applicability of our ensemble approach for different datasets and language pairs.

Mix. Operation	Weights	Base	Norm.
WMAX	uniform	27.67	27.94
WSUM	uniform	27.72	27.95
SWITCHMAX	uniform	27.96	26.21
SWITCHSUM	uniform	27.98	27.98
PROD	uniform	**27.99**	**28.09**
PROD	optimized	**28.25**	28.11

Table 2: Applying ensemble decoding with different mixture operations on the Test-11 dataset. Best performing setting is shown in **Boldface**.

3 English-Czech System

3.1 Morpheme Segmented Model

For English-Czech, we additionally experimented using morphologically segmented versions of the Czech side of the parallel data, since previous work (Clifton and Sarkar, 2011) has shown that segmentation of morphologically rich languages can aid translation. To derive the segmentation, we built an unsupervised morphological segmentation model using the Morfessor toolkit (Creutz and Lagus, 2007).

Morfessor uses minimum description length criteria to train a HMM-based segmentation model. Varying the perplexity threshold in Morfessor does not segment more word types, but rather oversegments the same word types. We hand tuned the model parameters over training data size and perplexity; these control the granularity and coverage of the segmentations. Specifically, we trained different segmenter models on varying sets of most frequent words and different perplexities and identified two sets that performed best based on a separate held-out set. These two sets correspond to 500k most frequent words and a perplexity of 50 (denoted SM1) and 10k most frequent words and a perplexity of 20 (denoted SM2). We then used these two models to segment the entire data set and generate two different segmented training sets. These models had the best combination of segmentation coverage of the training data and largest segments, since we found empirically that smaller segments were less meaningful in the translation model. The SM2 segmentation segmented more words than SM1, but more frequently segmented words into single-character units.

For example, the Czech word ‘dlaební’ is broken into the useful components ‘dlaeb + ní’ by SM1, but is oversegmented into ‘dl + a + e + b + ní’ by SM2. However, SM1 fails to find a segmentation at all for the related word ‘dlaebními’, while SM2 breaks it up similiarly with an additional suffix: ‘dl + a + e + b + ní + mi’.

With these segmentation models, we segmented the target side of the training and dev data before training the translation model. Similarly, we also train segmented language models corresponding to the two sets SM1 and SM2. The MERT tuning step uses the segmented dev-set reference to evaluate the segmented hypotheses generated by the decoder for optimizing the weights for the BLEU score. However for evaluating the test-set, we stitched the segments in the decoder output back into unsegmented forms in a post-processing step, before performing evaluation against the original unsegmented references. The hypotheses generated by the decoder can have incomplete dangling segments where one or more prefixes and/or suffixes are missing. While these dangling segments could be handled in a different way, we use a simple heuristic of ignoring the segment marker ’+’ by just removing the segment marker. In next section, we report the results of using the unsegmented model as well as its segmented counterparts.

3.2 Experiments and Results

In the English-Czech experiments, we used the same datasets for the dev and test sets as in French-English experiments (dev: news-tests 2008, 2009, 2010 with 7567 sentence pairs and test: newstest2011 with 3003 sentence pairs). Similarly, MERT (Och, 2003) has been used to tune the feature weights and we report the BLEU scores of two testsets computed using the official evaluation script (mteval-v11b.pl).

Table 3.2 shows the results of different segmentation schemes on the WMT-11 and WMT-12 test-sets. SM1 slightly outperformed the other two models in Test-11, however the unsegmented model performed best in Test-12, though marginally. We are currently investigating this and are also considering the possibility employing the idea of morpheme prediction in the post-decoding step in combination with this morpheme-based translation as suggested by Clifton and Sarkar (2011).

Segmentation	**Test-11**	**Test-12**
Baseline Hiero	14.65	**12.40**
SM1 : 500k-ppl50	**14.75**	12.34
SM2 : 10k-ppl20	14.57	12.34

Table 3: The English-Czech results for different segmentation settings. Best performing setting is shown in **Boldface**.

4 Conclusion

We submitted systems in two language pairs French-English and English-Czech for WMT-12 shared task. In French-English, we experimented the ensemble decoding framework that effectively utilizes the small amount of *news* genre data to improve the performance in the testset belonging to the same genre. We obtained a moderate gain of 0.4 BLEU points with the ensemble decoding over the baseline system in newstest-2011. For newstest-2012, it performs comparably to that of the baseline and we are presently investigating the lack of improvement in newstest-2012. For Cz-En, We found that the BLEU scores do not substantially differ from each other and also the minor differences are not consistent for Test-11 and Test-12.

References

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33.

Ann Clifton and Anoop Sarkar. 2011. Combining morpheme-based machine translation with post-processing morpheme prediction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 32–42.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1):3:1–3:34, February.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197.

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 Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the Annual Meeting of Association of Computational Linguistics*, pages 160–167.

Majid Razmara, George Foster, Baskaran Sankaran, and Anoop Sarkar. 2012. Mixing multiple translation models in statistical machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju, Republic of Korea, July. Association for Computational Linguistics. *To appear*.

Baskaran Sankaran, Majid Razmara, and Anoop Sarkar. 2012. Kriya an end-to-end hierarchical phrase-based mt system. *The Prague Bulletin of Mathematical Linguistics*, 97(97):83–98, April.

Frank Vanden Berghen and Hugues Bersini. 2005. CONDOR, a new parallel, constrained extension of powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm. *Journal of Computational and Applied Mathematics*, 181:157–175, September.