In Proc. of the 50th Annual Meeting of the Association for Computational Linguistics (ACL-2012). July 9-11, 2012. Jeju Island. R. of Korea.

Mixing Multiple Translation Models in Statistical Machine Translation

Majid Razmara[1] **George Foster**[2] **Baskaran Sankaran**[1] **Anoop Sarkar**[1]

[1] Simon Fraser University, 8888 University Dr., Burnaby, BC, Canada
{razmara,baskaran,anoop}@sfu.ca
[2] National Research Council Canada, 283 Alexandre-Taché Blvd, Gatineau, QC, Canada
george.foster@nrc.gc.ca

Abstract

Statistical machine translation is often faced with the problem of combining training data from many diverse sources into a single translation model which then has to translate sentences in a new domain. We propose a novel approach, ensemble decoding, which combines a number of translation systems dynamically at the decoding step. In this paper, we evaluate performance on a domain adaptation setting where we translate sentences from the medical domain. Our experimental results show that ensemble decoding outperforms various strong baselines including mixture models, the current state-of-the-art for domain adaptation in machine translation.

1 Introduction

Statistical machine translation (SMT) systems require large parallel corpora in order to be able to obtain a reasonable translation quality. In statistical learning theory, it is assumed that the training and test datasets are drawn from the same distribution, or in other words, they are from the same domain. However, bilingual corpora are only available in very limited domains and building bilingual resources in a new domain is usually very expensive. It is an interesting question whether a model that is trained on an existing large bilingual corpus in a specific domain can be adapted to another domain for which little parallel data is present. Domain adaptation techniques aim at finding ways to adjust an *out-of-domain* (OUT) model to represent a target domain (*in-domain* or IN).

Common techniques for model adaptation adapt two main components of contemporary state-of-the-art SMT systems: the language model and the translation model. However, language model adaptation is a more straight-forward problem compared to translation model adaptation, because various measures such as perplexity of adapted language models can be easily computed on data in the target domain. As a result, language model adaptation has been well studied in various work (Clarkson and Robinson, 1997; Seymore and Rosenfeld, 1997; Bacchiani and Roark, 2003; Eck et al., 2004) both for speech recognition and for machine translation. It is also easier to obtain monolingual data in the target domain, compared to bilingual data which is required for translation model adaptation. In this paper, we focused on adapting only the translation model by fixing a language model for all the experiments. We expect domain adaptation for machine translation can be improved further by combining orthogonal techniques for translation model adaptation combined with language model adaptation.

In this paper, a new approach for adapting the translation model is proposed. We use a novel system combination approach called ensemble decoding in order to combine two or more translation models with the goal of constructing a system that outperforms all the component models. The strength of this system combination method is that the systems are combined in the decoder. This enables the decoder to pick the best hypotheses for each span of the input. The main applications of ensemble models are domain adaptation, domain mixing and system combination. We have modified *Kriya* (Sankaran et al., 2012), an in-house implementation of hierarchical phrase-based translation system (Chiang, 2005), to implement ensemble decoding using multiple translation models.

We compare the results of ensemble decoding with a number of baselines for domain adaptation. In addition to the basic approach of concatenation of in-domain and out-of-domain data, we also trained a log-linear mixture model (Foster and Kuhn, 2007)

as well as the linear mixture model of (Foster et al., 2010) for conditional phrase-pair probabilities over IN and OUT. Furthermore, within the framework of ensemble decoding, we study and evaluate various methods for combining translation tables.

2 Baselines

The natural baseline for model adaption is to concatenate the IN and OUT data into a single parallel corpus and train a model on it. In addition to this baseline, we have experimented with two more sophisticated baselines which are based on mixture techniques.

2.1 Log-Linear Mixture

Log-linear translation model (TM) mixtures are of the form:

$$p(\bar{e}|\bar{f}) \propto \exp\left(\sum_{m}^{M} \lambda_m \log p_m(\bar{e}|\bar{f})\right)$$

where m ranges over IN and OUT, $p_m(\bar{e}|\bar{f})$ is an estimate from a component phrase table, and each λ_m is a weight in the top-level log-linear model, set so as to maximize dev-set BLEU using minimum error rate training (Och, 2003). We learn separate weights for relative-frequency and lexical estimates for both $p_m(\bar{e}|\bar{f})$ and $p_m(\bar{f}|\bar{e})$. Thus, for 2 component models (from IN and OUT training corpora), there are $4 * 2 = 8$ TM weights to tune. Whenever a phrase pair does not appear in a component phrase table, we set the corresponding $p_m(\bar{e}|\bar{f})$ to a small epsilon value.

2.2 Linear Mixture

Linear TM mixtures are of the form:

$$p(\bar{e}|\bar{f}) = \sum_{m}^{M} \lambda_m p_m(\bar{e}|\bar{f})$$

Our technique for setting λ_m is similar to that outlined in Foster et al. (2010). We first extract a joint phrase-pair distribution $\tilde{p}(\bar{e}, \bar{f})$ from the development set using standard techniques (HMM word alignment with grow-diag-and symmeterization (Koehn et al., 2003)). We then find the set of weights $\hat{\lambda}$ that minimize the cross-entropy of the mixture $p(\bar{e}|\bar{f})$ with respect to $\tilde{p}(\bar{e}, \bar{f})$:

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmax}} \sum_{\bar{e},\bar{f}} \tilde{p}(\bar{e}, \bar{f}) \log \sum_{m}^{M} \lambda_m p_m(\bar{e}|\bar{f})$$

For efficiency and stability, we use the EM algorithm to find $\hat{\lambda}$, rather than L-BFGS as in (Foster et al., 2010). Whenever a phrase pair does not appear in a component phrase table, we set the corresponding $p_m(\bar{e}|\bar{f})$ to 0; pairs in $\tilde{p}(\bar{e}, \bar{f})$ that do not appear in at least one component table are discarded. We learn separate linear mixtures for relative-frequency and lexical estimates for both $p(\bar{e}|\bar{f})$ and $p(\bar{f}|\bar{e})$. These four features then appear in the top-level model as usual – there is no runtime cost for the linear mixture.

3 Ensemble Decoding

Ensemble decoding is a way to combine the expertise of different models in one single model. The current implementation is able to combine hierarchical phrase-based systems (Chiang, 2005) as well as phrase-based translation systems (Koehn et al., 2003). However, the method can be easily extended to support combining a number of heterogeneous translation systems e.g. phrase-based, hierarchical phrase-based, and/or syntax-based systems. This section explains how such models can be combined during the decoding.

Given a number of translation models which are already trained and tuned, the ensemble decoder uses hypotheses constructed from all of the models in order to translate a sentence. We use the bottom-up CKY parsing algorithm for decoding. For each sentence, a CKY chart is constructed. The cells of the CKY chart are populated with appropriate rules from all the phrase tables of different components. As in the Hiero SMT system (Chiang, 2005), the cells which span up to a certain length (i.e. the maximum span length) are populated from the phrase-tables and the rest of the chart uses *glue rules* as defined in (Chiang, 2005).

The rules suggested from the component models are combined in a single set. Some of the rules may be unique and others may be common with other component model rule sets, though with different scores. Therefore, we need to combine the scores of such common rules and assign a single score to

them. Depending on the mixture operation used for combining the scores, we would get different mixture scores. The choice of mixture operation will be discussed in Section 3.1.

Figure 1 illustrates how the CKY chart is filled with the rules. Each cell, covering a span, is populated with rules from all component models as well as from cells covering a sub-span of it.

In the typical log-linear model SMT, the posterior probability for each phrase pair $(\bar{e}, \bar{f})$ is given by:

$$p(\bar{e} \mid \bar{f}) \propto \exp\Bigg(\underbrace{\sum_i w_i \phi_i(\bar{e}, \bar{f})}_{\mathbf{w}\cdot\boldsymbol{\phi}}\Bigg)$$

Ensemble decoding uses the same framework for each individual system. Therefore, the score of a phrase-pair $(\bar{e}, \bar{f})$ in the ensemble model is:

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

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

3.1 Mixture Operations

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

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

  $$p(\bar{e} \mid \bar{f}) \propto \sum_m^M \lambda_m \exp\left(\mathbf{w}_m \cdot \boldsymbol{\phi}_m\right)$$

  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.

  $$p(\bar{e} \mid \bar{f}) \propto \max_m \left(\lambda_m \exp\left(\mathbf{w}_m \cdot \boldsymbol{\phi}_m\right)\right)$$

- **Model Switching (Switch):** 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 certain parts of sentence. In this method, we need to define a binary indicator function $\delta(\bar{f}, m)$ for each span and component model to specify rules of which model to retain for each span.

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

  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 best-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 scores of the best rules, the model with the highest weighted sum of the probabilities of the rules wins. This sum has to take into account the translation table limit (*ttl*), on the number of rules suggested by each model for each cell:

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

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

  $$p(\bar{e} \mid \bar{f}) = \sum_m^M \delta(\bar{f}, m)\ p_m(\bar{e} \mid \bar{f})$$

- **Product (prod):** in Product models or Product of Experts (Hinton, 1999), 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, i.e. log-linear mixture). Product models can also make use of weights to control the contribution of each component. These models are

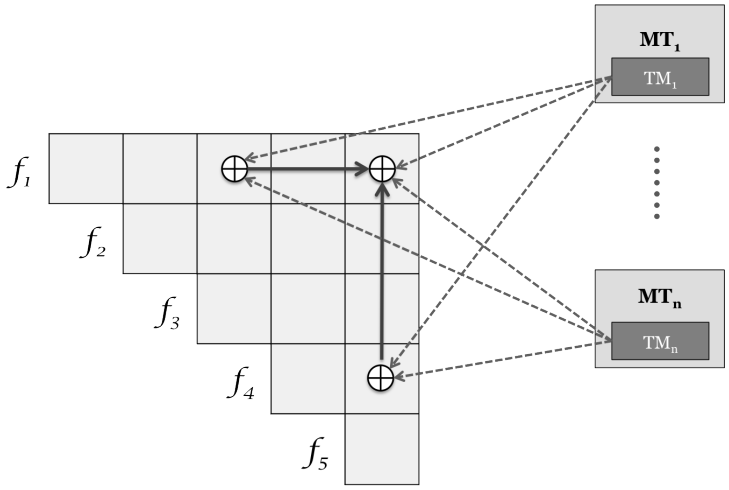

Figure 1: The cells in the CKY chart are populated using rules from all component models and sub-span cells.

generally known as *Logarithmic Opinion Pools (LOPs)* where:

$$p(\bar{e} \mid \bar{f}) \propto \exp\Big(\sum_{m}^{M} \lambda_m \left(\mathbf{w}_m \cdot \boldsymbol{\phi}_m\right)\Big)$$

Product models have been used in combining LMs and TMs in SMT as well as some other NLP tasks such as ensemble parsing (Petrov, 2010).

Each of these mixture operations has a specific property that makes it work in specific domain adaptation or system combination scenarios. For instance, LOPs may not be optimal for domain adaptation in the setting where there are two or more models trained on heterogeneous corpora. As discussed in (Smith et al., 2005), LOPs work best when all the models accuracies are high and close to each other with some degree of diversity. LOPs give veto power to any of the component models and this perfectly works for settings such as the one in (Petrov, 2010) where a number of parsers are trained by changing the randomization seeds but having the same base parser and using the same training set. They noticed that parsers trained using different randomization seeds have high accuracies but there are some diversities among them and they used product models for their advantage to get an even better parser. We assume that each of the models is expert in some parts and so they do not necessarily agree on correct hypotheses. In other words, product models (or LOPs) tend to have intersection-style effects while we are more interested in union-style effects.

In Section 4.2, we compare the BLEU scores of different mixture operations on a French-English experimental setup.

3.2 Normalization

Since in log-linear models, the model scores are not normalized to form probability distributions, 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 experimented with two different ways to deal with this normalization issue. A practical but inexact heuristic is to normalize the scores over a shorter list. So the list of rules coming from each model for a cell in CKY chart is normalized before getting mixed with other phrase-table rules. However, experiments showed changing the scores with the normalized scores hurts the BLEU score radically. So we use the normalized scores only for pruning and the actual scores are intact. We could also globally normalize the scores to obtain posterior probabilities using the inside-outside algorithm. However, we did not try it as the BLEU scores we got using the normalization heuristic was not promissing and it would impose a cost in decoding as well. More investigation on this issue has been left for future work.

A more principled way is to systematically find the most appropriate model weights that can avoid this problem by scaling the scores properly. We used a publicly available toolkit, CONDOR (Vanden Berghen and Bersini, 2005), a direct optimizer based on Powell's algorithm, that does not require

explicit gradient information for the objective function. Component weights for each mixture operation are optimized on the dev-set using CONDOR.

4 Experiments & Results

4.1 Experimental Setup

We carried out translation experiments using the European Medicines Agency (EMEA) corpus (Tiedemann, 2009) as IN, and the Europarl (EP) corpus[1] as OUT, for French to English translation. The dev and test sets were randomly chosen from the EMEA corpus.[2] The details of datasets used are summarized in Table 1.

Dataset	Sents	Words	
		French	English
EMEA	11770	168K	144K
Europarl	1.3M	40M	37M
Dev	1533	29K	25K
Test	1522	29K	25K

Table 1: Training, dev and test sets for EMEA.

For the mixture baselines, we used a standard one-pass phrase-based system (Koehn et al., 2003), Portage (Sadat et al., 2005), with the following 7 features: relative-frequency and lexical translation model (TM) probabilities in both directions; word-displacement distortion model; language model (LM) and word count. The corpus was word-aligned using both HMM and IBM2 models, and the phrase table was the union of phrases extracted from these separate alignments, with a length limit of 7. It was filtered to retain the top 20 translations for each source phrase using the TM part of the current log-linear model.

For ensemble decoding, we modified an in-house implementation of hierarchical phrase-based system, *Kriya* (Sankaran et al., 2012) which uses the same features mentioned in (Chiang, 2005): forward and backward relative-frequency and lexical TM probabilities; LM; word, phrase and glue-rules penalty. GIZA++(Och and Ney, 2000) has been used for word alignment with phrase length limit of 7.

In both systems, feature weights were optimized using MERT (Och, 2003) and with a 5-gram language model and Kneser-Ney smoothing was used in all the experiments. We used SRILM (Stolcke, 2002) as the langugage model toolkit. Fixing the language model allows us to compare various translation model combination techniques.

4.2 Results

Table 2 shows the results of the baselines. The first group are the baseline results on the phrase-based system discussed in Section 2 and the second group are those of our hierarchical MT system. Since the Hiero baselines results were substantially better than those of the phrase-based model, we also implemented the best-performing baseline, linear mixture, in our Hiero-style MT system and in fact it achieves the hights BLEU score among all the baselines as shown in Table 2. This baseline is run three times the score is averaged over the BLEU scores with standard deviation of 0.34.

Baseline	PBS	Hiero
IN	31.84	33.69
OUT	24.08	25.32
IN + OUT	31.75	33.76
LOGLIN	32.21	–
LINMIX	33.81	**35.57**

Table 2: The results of various baselines implemented in a phrase-based (PBS) and a Hiero SMT on EMEA.

Table 3 shows the results of ensemble decoding with different mixture operations and model weight settings. Each mixture operation has been evaluated on the test-set by setting the component weights uniformly (denoted by *uniform*) and by tuning the weights using CONDOR (denoted by *tuned*) on a held-out set. The tuned scores (3rd column in Table 3) are averages of three runs with different initial points as in Clark et al. (2011). We also reported the BLEU scores when we applied the span-wise normalization heuristic. All of these mixture operations were able to significantly improve over the concatenation baseline. In particular, *Switching:Max* could gain up to 2.2 BLEU points over the concatenation baseline and 0.39 BLEU points over the best performing baseline (i.e. linear mixture model implemented in Hiero) which is statistically significant based on Clark et al. (2011) ($p = 0.02$).

Prod when using with uniform weights gets the

[1] www.statmt.org/europarl

[2] Please contact the authors to access the data-sets.

Mixture Operation	Uniform	Tuned	Norm.
WMAX	35.39	35.47 (s=0.03)	35.47
WSUM	35.35	35.53 (s=0.04)	35.45
SWITCHING:MAX	35.93	**35.96** (s=0.01)	32.62
SWITCHING:SUM	34.90	34.72 (s=0.23)	34.90
PROD	33.93	35.24 (s=0.05)	35.02

Table 3: The results of ensemble decoding on EMEA for Fr2En when using uniform weights, tuned weights and normalization heuristic. The tuned BLEU scores are averaged over three runs with multiple initial points, as in (Clark et al., 2011), with the standard deviations in brackets .

lowest score among the mixture operations, however after tuning, it learns to bias the weights towards one of the models and hence improves by 1.31 BLEU points. Although *Switching:Sum* outperforms the concatenation baseline, it is substantially worse than other mixture operations. One explanation that *Switching:Max* is the best performing operation and *Switching:Sum* is the worst one, despite their similarities, is that *Switching:Max* prefers more peaked distributions while *Switching:Sum* favours a model that has fewer hypotheses for each span.

An interesting observation based on the results in Table 3 is that uniform weights are doing reasonably well given that the component weights are not optimized and therefore model scores may not be in the same scope (refer to discussion in §3.2). We suspect this is because a single LM is shared between both models. This shared component controls the variance of the weights in the two models when combined with the standard L-1 normalization of each model's weights and hence prohibits models to have too varied scores for the same input. Though, it may not be the case when multiple LMs are used which are not shared.

Two sample sentences from the EMEA test-set along with their translations by the IN, OUT and Ensemble models are shown in Figure 2. The boxes show how the Ensemble model is able to use n-grams from the IN and OUT models to construct a better translation than both of them. In the first example, there are two OOVs one for each of the IN and OUT models. Our approach is able to resolve the OOV issues by taking advantage of the other model's presence. Similarly, the second example shows how ensemble decoding improves lexical choices as well as word re-orderings.

5 Related Work

5.1 Domain Adaptation

Early approaches to domain adaptation involved information retrieval techniques where sentence pairs related to the target domain were retrieved from the training corpus using IR methods (Eck et al., 2004; Hildebrand et al., 2005). Foster et al. (2010), however, uses a different approach to select related sentences from OUT. They use language model perplexities from IN to select relavant sentences from OUT. These sentences are used to enrich the IN training set.

Other domain adaptation methods involve techniques that distinguish between general and domain-specific examples (Daumé and Marcu, 2006). Jiang and Zhai (2007) introduce a general instance weighting framework for model adaptation. This approach tries to penalize misleading training instances from OUT and assign more weight to IN-like instances than OUT instances. Foster et al. (2010) propose a similar method for machine translation that uses features to capture degrees of generality. Particularly, they include the output from an SVM classifier that uses the intersection between IN and OUT as positive examples. Unlike previous work on instance weighting in machine translation, they use phrase-level instances instead of sentences.

A large body of work uses interpolation techniques to create a single TM/LM from interpolating a number of LMs/TMs. Two famous examples of such methods are linear mixtures and log-linear mixtures (Koehn and Schroeder, 2007; Civera and Juan, 2007; Foster and Kuhn, 2007) which were used as baselines and discussed in Section 2. Other methods include using self-training techniques to exploit monolingual in-domain data (Ueffing et al., 2007;

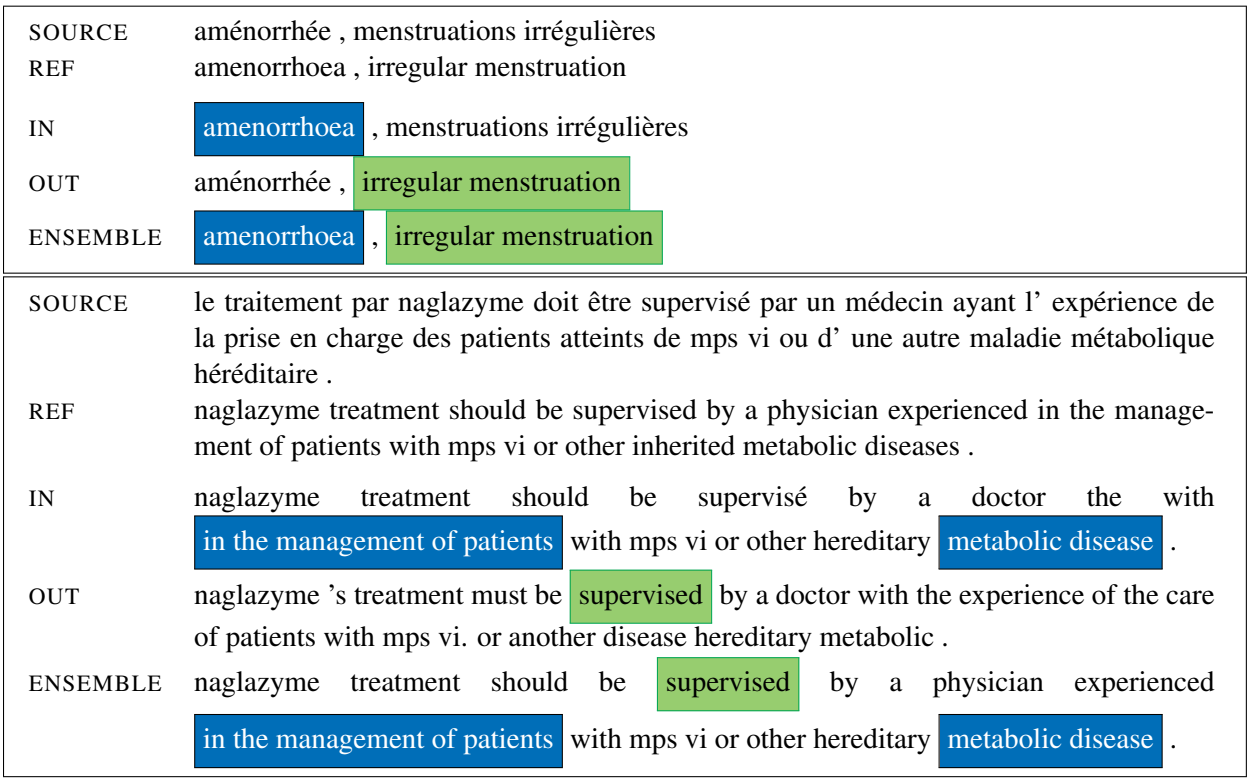

Figure 2: Examples illustrating how this method is able to use expertise of both out-of-domain and in-domain systems.

Bertoldi and Federico, 2009). In this approach, a system is trained on the parallel OUT and IN data and it is used to translate the monolingual IN data set. Iteratively, most confident sentence pairs are selected and added to the training corpus on which a new system is trained.

5.2 System Combination

Tackling the model adaptation problem using system combination approaches has been experimented in various work (Koehn and Schroeder, 2007; Hildebrand and Vogel, 2009). Among these approaches are sentence-based, phrase-based and word-based output combination methods. In a similar approach, Koehn and Schroeder (2007) use a feature of the factored translation model framework in Moses SMT system (Koehn and Schroeder, 2007) to use multiple alternative decoding paths. Two decoding paths, one for each translation table (IN and OUT), were used during decoding. The weights are set with minimum error rate training (Och, 2003).

Our work is closely related to Koehn and Schroeder (2007) but uses a different approach to deal with multiple translation tables. The Moses SMT system implements (Koehn and Schroeder, 2007) and can treat multiple translation tables in two different ways: *intersection* and *union*. In *intersection*, for each span only the hypotheses would be used that are present in all phrase tables. For each set of hypothesis with the same source and target phrases, a new hypothesis is created whose feature-set is the union of feature sets of all corresponding hypotheses. *Union*, on the other hand, uses hypotheses from all the phrase tables. The feature set of these hypotheses are expanded to include one feature set for each table. However, for the corresponding feature values of those phrase-tables that did not have a particular phrase-pair, a default log probability value of 0 is assumed (Bertoldi and Federico, 2009) which is counter-intuitive as it boosts the score of hypotheses with phrase-pairs that do not belong to all of the translation tables.

Our approach is different from Koehn and Schroeder (2007) in a number of ways. Firstly, unlike the multi-table support of Moses which only supports phrase-based translation table combination, our approach supports ensembles of both hierarchical and phrase-based systems. With little modification, it can also support ensemble of syntax-based systems with the other two state-of-the-art SMT sys-

tems. Secondly, our combining method uses the *union* option, but instead of preserving the features of all phrase-tables, it only combines their scores using various mixture operations. This enables us to experiment with a number of different operations as opposed to sticking to only one combination method. Finally, by avoiding increasing the number of features we can add as many translation models as we need without serious performance drop. In addition, MERT would not be an appropriate optimizer when the number of features increases a certain amount (Chiang et al., 2008).

Our approach differs from the model combination approach of DeNero et al. (2010), a generalization of consensus or minimum Bayes risk decoding where the search space consists of those of multiple systems, in that model combination uses forest of derivations of all component models to do the combination. In other words, it requires all component models to fully decode each sentence, compute n-gram expectations from each component model and calculate posterior probabilities over translation derivations. While, in our approach we only use partial hypotheses from component models and the derivation forest is constructed by the ensemble model. A major difference is that in the model combination approach the component search spaces are conjoined and they are not intermingled as opposed to our approach where these search spaces are intermixed on spans. This enables us to generate new sentences that cannot be generated by component models. Furthermore, various combination methods can be explored in our approach. Finally, main techniques used in this work are orthogonal to our approach such as Minimum Bayes Risk decoding, using n-gram features and tuning using MERT.

Finally, our work is most similar to that of Liu et al. (2009) where max-derivation and max-translation decoding have been used. Max-derivation finds a derivation with highest score and max-translation finds the highest scoring translation by summing the score of all derivations with the same yield. The combination can be done in two levels: translation-level and derivation-level. Their derivation-level max-translation decoding is similar to our ensemble decoding with *wsum* as the mixture operation. We did not restrict ourself to this particular mixture operation and experimented with a number of different mixing techniques and as Table 3 shows we could improve over *wsum* in our experimental setup. Liu et al. (2009) used a modified version of MERT to tune max-translation decoding weights, while we use a two-step approach using MERT for tuning each component model separately and then using CONDOR to tune component weights on top of them.

6 Conclusion & Future Work

In this paper, we presented a new approach for domain adaptation using ensemble decoding. In this approach a number of MT systems are combined at decoding time in order to form an ensemble model. The model combination can be done using various mixture operations. We showed that this approach can gain up to 2.2 BLEU points over its concatenation baseline and 0.39 BLEU points over a powerful mixture model.

Future work includes extending this approach to use multiple translation models with multiple language models in ensemble decoding. Different mixture operations can be investigated and the behaviour of each operation can be studied in more details. We will also add capability of supporting syntax-based ensemble decoding and experiment how a phrase-based system can benefit from syntax information present in a syntax-aware MT system. Furthermore, ensemble decoding can be applied on domain mixing settings in which development sets and test sets include sentences from different domains and genres, and this is a very suitable setting for an ensemble model which can adapt to new domains at test time. In addition, we can extend our approach by applying some of the techniques used in other system combination approaches such as consensus decoding, using n-gram features, tuning using forest-based MERT, among other possible extensions.

Acknowledgments

This research was partially supported by an NSERC, Canada (RGPIN: 264905) grant and a Google Faculty Award to the last author. We would like to thank Philipp Koehn and the anonymous reviewers for their valuable comments. We also thank the developers of GIZA++ and Condor which we used for our experiments.

References

M. Bacchiani and B. Roark. 2003. Unsupervised language model adaptation. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 1, pages I–224 – I–227 vol.1, april.

Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 182–189, Stroudsburg, PA, USA. ACL.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*. ACL.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270, Morristown, NJ, USA. ACL.

Jorge Civera and Alfons Juan. 2007. Domain adaptation in statistical machine translation with mixture modelling. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 177–180, Stroudsburg, PA, USA. ACL.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 176–181. ACL.

P. Clarkson and A. Robinson. 1997. Language model adaptation using mixtures and an exponentially decaying cache. In *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)-Volume 2 - Volume 2*, ICASSP '97, pages 799–, Washington, DC, USA. IEEE Computer Society.

Hal Daumé, III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *J. Artif. Int. Res.*, 26:101–126, May.

John DeNero, Shankar Kumar, Ciprian Chelba, and Franz Och. 2010. Model combination for machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 975–983, Stroudsburg, PA, USA. ACL.

Matthias Eck, Stephan Vogel, and Alex Waibel. 2004. Language model adaptation for statistical machine translation based on information retrieval. In *In Proceedings of LREC*.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for smt. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 128–135, Stroudsburg, PA, USA. ACL.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 451–459, Stroudsburg, PA, USA. ACL.

Almut Silja Hildebrand and Stephan Vogel. 2009. CMU system combination for WMT'09. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 47–50, Stroudsburg, PA, USA. ACL.

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 10th EAMT 2005*, Budapest, Hungary, May.

Geoffrey E. Hinton. 1999. Products of experts. In *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*, volume 1, pages 1–6.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, Prague, Czech Republic, June. ACL.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 224–227, Stroudsburg, PA, USA. ACL.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 127–133, Edmonton, May. NAACL.

Yang Liu, Haitao Mi, Yang Feng, and Qun Liu. 2009. Joint decoding with multiple translation models. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 576–584, Stroudsburg, PA, USA. ACL.

F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the ACL*, pages 440–447, Hongkong, China, October.

Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of the 41th Annual Meeting of the ACL*, Sapporo, July. ACL.

Slav Petrov. 2010. Products of random latent variable grammars. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 19–27, Stroudsburg, PA, USA. ACL.

Fatiha Sadat, Howard Johnson, Akakpo Agbago, George Foster, Joel Martin, and Aaron Tikuisis. 2005. Portage: A phrase-based machine translation system. In *In Proceedings of the ACL Worskhop on Building and Using Parallel Texts, Ann Arbor*. ACL.

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), April.

Kristie Seymore and Ronald Rosenfeld. 1997. Using story topics for language model adaptation. In George Kokkinakis, Nikos Fakotakis, and Evangelos Dermatas, editors, *EUROSPEECH*. ISCA.

Andrew Smith, Trevor Cohn, and Miles Osborne. 2005. Logarithmic opinion pools for conditional random fields. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 18–25, Stroudsburg, PA, USA. ACL.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings International Conference on Spoken Language Processing*, pages 257–286.

Jorg Tiedemann. 2009. News from opus - a collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia.

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2007. Transductive learning for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 25–32, Prague, Czech Republic, June. ACL.

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.