Fast Domain Adaptation of SMT models without in-Domain Parallel Data

Prashant Mathur[*]
Fondazione Bruno Keseler
Povo - 38100 (IT)
`first@fbk.eu`

Sriram Venkatapathy
Xerox Research Center Europe
Meylan (FR)
`first.last@xrce.xerox.com`

Nicola Cancedda[*]
Microsoft, London (UK)
`first.last@gmail.com`

Abstract

We address a challenging problem frequently faced by MT service providers: creating a domain-specific system based on a purely source-monolingual sample of text from the domain. We solve this problem by introducing methods for domain adaptation requiring no in-domain parallel data. Our approach yields results comparable to state-of-the-art systems optimized on an in-domain parallel set with a drop of as little as 0.5 BLEU points across 4 domains.

1 Introduction

We consider the problem of creating the best possible statistical machine translation (SMT) system for a specific domain when no parallel sample or training data from such domain is available. We assume that we have access to a collection of phrase tables (PT) and other models independently created from now **unavailable** corpora, and we receive a monolingual source language sample from a text source we would like to optimize for.

For a MT provider to deliver a SMT system tailored to a customer's domain, a sample dataset is requested. In most cases, the customer is able to provide an in-domain mono-lingual sample from his operations. However, it is generally not feasible for the customer to provide the translations as well because the customer has to hire professional translators to do that. In such a scenario, the translations has to be generated by MT service provider itself by hiring human translators thus requiring an investment upfront. The methods proposed in this paper aim to avoid that by building a good quality pilot SMT system leveraging only sample mono-lingual source corpus, and previously trained library of models. This in turn postpones the task of generating in-domain parallel data to a later date when there is a commitment by the customer.

Unavailability of the raw parallel data could derive from a trading model where data owners share intermediate-level resources like PTs, Reordering Models (RM) and Language Models (LM), but can not, or do not want to, share the textual data such resources were derived from. This particular scenario has been explained in (Cancedda, 2012).

This scenario is similar to the multi-model framework studied in (Sennrich et al., 2013), with the additional challenge that no parallel development set is available. We build on the linear mixture model combination of the cited work, extending it to our more challenging environment:

1. We propose a new measure derived from the popular BLEU score (Papineni et al., 2002) to assess the fitness of a PT to cope with a given monolingual sample S. This measure is computed from n-gram statistics that can be easily extracted from a PT.

2. We propose a new method for tuning the parameters of a log linear model that does not require an in-domain parallel development set, and yet achieves results very close to traditional tuning on parallel in-domain data.

We present our proposed metric *BLEU-PT* and computation of multi-model in Section 2. The parameter estimation of log-linear parameters of the SMT system is described in Section 3. We present experiments and results in Sections 4 and 6 respectively.

[*] Major part of the work was performed when the authors were in Xerox Research Center Europe.

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 1114–1123, Dublin, Ireland, August 23-29 2014.

2 Building Multi-Model

Given a library of phrase tables, the goal of this step is to generate a domain adapted multi-model. The challenging aspect in our scenario is the lack of in-domain parallel data, as well as absence of original parallel corpora corresponding to the library of models. This rules out the possibility of using metrics such as cross-entropy (Sennrich, 2012b) or LM-perplexity for computing the mixing coefficients. We present our proposed metric in section 2.1, and interpolation of the phrase tables in section 2.2.

2.1 BLEU-PT

Given a source corpus s, and a set of phrase tables $\{pt_1, pt_2, \ldots, pt_n\}$, the goal is to measure the similarity of each of these tables with s. For measuring the similarity, we use BLEU-PT which is an adaptation of the popular BLEU score for measuring the similarity between a corpus and a phrase table. The metric BLEU-PT is measured as described in Equation 1.

$$\text{BLEU-PT}(PT, S) = \left(\prod_{n=1}^{4} \frac{match(n|pt, s)}{total(n|s)} \right)^{1/4} \tag{1}$$

where $match(n|pt, s)$ is the count of n-grams of order n in the source corpus s that exist in the source side of the phrase table pt. $total(n|s)$ is the number of n-grams of order n in the source corpus.

2.2 Interpolating Models

A state-of-the-art approach for building multi-models is through linear interpolation of component models, exemplified in Equation 2 for the case of the forward conditional phrase translation model.

$$h_{phr}(s, t) = log \sum_{j=1}^{N} \phi_j P_{phr,j}(t|s) \tag{2}$$

Various approaches have been suggested for computing the coefficients ϕ of the interpolated model, the most recent being perplexity minimization described in (Sennrich, 2012b), where each translation model feature is optimized separately on the parallel development set. Our work is set in a scenario where no parallel development set is available for optimizing the interpolation coefficients. We have also observed that perplexity minimization is computationally intensive, requires aligned parallel development set, and the optimization time increases rapidly with increasing number of component models (for details, see Section 4.2).

We propose a simple approach for computation of the mixing coefficients that relies on the similarity of each model with respect to the test set. The mixing coefficients are obtained by normalizing similarity values. The similarity between a model (phrase table) and a corpus is computed using the BLEU-PT metric proposed in the previous section. Another similarity metric that can be used is *LM Perplexity*. However, in the current scenario we do not have resources (training data) to build a source side LM for computing the perplexity.

We empirically compare our method for computing mixing coefficients with the the perplexity minimization method. We also experiment with applying the mixing coefficients obtained by using our method for mixing features of a reordering and language model.

3 Parameter Estimation

The overall quality of translation is strongly impacted by how optimized the weights of the log-linear combination of various translation features are for a domain of interest. MERT (Och, 2003) and MIRA (Watanabe et al., 2007) are popular solution to compute an optimal weight vector by minimizing the error on a held-out parallel development set. BLEU and its approximations are commonly used error metrics. In this paper we assume lack of a parallel development set, therefore the above methods cannot be used.

Pecina et. al. (2012) showed that the optimized log-linear weight vector [1] of a SMT system does not depend as much on the actual domain of the development set (on which the system was optimized), as

[1] Not to be confused with the mixing coefficients in a linear combination of model components.

on how "distant" the relevant domain is from the domain of the training corpus used to build the SMT models. This is an important finding. It means that the weight vector can be modeled as a function of the **distance/similarity** between the in-domain development set and the model built from the training set. In this work, we learn this function from examples of previous parameter optimizations, using our BLEU-PT as a similarity metric. Once we have retrieved the most relevant PTs (translation and reordering models) from our library, and we have linearly interpolated them using normalized BLEU-PT, we use the learned model to estimate the optimal value of the log-linear weights, instead of optimizing them.

In order to learn this mapping, we create a dataset of examples (pairs of the form <BLEU-PT, log-linear weight vector>, where weight vectors are normalized to ensure comparability across models) by performing repeated optimizations for out-domain models on a number of parallel development sets (see section 4 for more details of this data) using a traditional optimization method (MIRA in this work). Based on this dataset, the function of our interest can therefore be learnt using a supervised approach. We explore two parametric methods and a non-parametric method. We present these in Section 3.1, and 3.2 respectively. For a mono-lingual source in a new domain, the BLEU-PT can be computed, and then mapped to the appropriate weight vector using the methods presented below.

3.1 Parametric Methods

We considered two distinct parametric methods for estimating the mapping from model/corpus similarity into weight vectors. The first one makes the assumption that parameters can be estimated independently of one another, given the similarity, whereas the second tries to leverage known covariance between distinct parameters in the vector.

3.1.1 Linear Regression

Motivated by initial experiments highlighting strong correlation between BLEU-PT and optimal feature weights (see Section 5.1 below), we assumed here a simple linear relation of the form:

$$\lambda_i^* = W_i \mathrm{X} + b_i \tag{3}$$

where λ_i^* is the optimal log-linear weight for feature i, X is the feature vector (BLEU-PT vector), W_i and b_i are coefficients to be estimated. While a drastic assumption, this has the advantage of limiting the risk of overfitting in a situation like ours where there is only relatively few datapoints to learn from. We estimate a_i and b_i by simple least squares regression. Once these are available for all features, we can predict the log linear weights of any model given its BLEU-PT similarity to a monolingual source sample using Eq. 3.

3.1.2 Multi-Task learning

Optimal log-linear parameters might not be fully independent given BLEU-PT, especially since it is known that model features can be highly correlated. To account for correlation between parameter weights, we explore the use of multi-task lasso[2] (Caruana, 1997) where several functions corresponding to each parameter are learned jointly considering the correlation between their values observed in the training data. Multi-task lasso consists of a least square loss model trained along with a regularizer and the objective is to minimize the following:

$$\arg\min_{w} \frac{1}{2N} ||\mathrm{X} \cdot W - \lambda||_2^2 + \alpha ||W||_{21} \quad where; \quad ||W||_{21} = \sum_{j}^{M} \sqrt{\sum_{i} w_{ij}^2} \tag{4}$$

Here, N is the number of training samples, X is the feature vector(BLEU-PT score vector) λ is the label vector(log linear weights). $||W||_{21}$ is the l_{21} regularizer (Yang et al., 2011). The problem of prediction of log linear weights is reduced to prediction of i interlinked tasks where each task has M features[3]. Coefficients are calculated using coordinate descent algorithm in Multi-Task lasso. Once the coefficients are calculated we use Eq. 3 to predict the log linear weights.

[2] `http://scikit-learn.org/`

[3] In our case we only have 1 feature i.e. BLEU-PT score.

3.2 Non Parametric: Nearest Neighbor

Finally, instead of building a parametric predictor for log linear weights, we experimented with a simple nearest-neighbor approach:

$$\lambda_i^* = \lambda_i(M_{j^*}) \tag{5}$$

where M_j ranges over the linearly interpolated phrase tables, and $\lambda_i(M)$ returns the stored optimal value for the i^{th} log-linear weight, and:

$$j^* = \arg\min_j \min_{s'}(|\text{BLEU-PT}(M, s) - \text{BLEU-PT}'(M_j, s')|) \tag{6}$$

where s is the monolingual sample on which we want to calculate the BLEU-PT and s' ranges over the source sides of our available parallel development sets. In other words, a BLEU-PT of a model is calculated on the source sample to be translated and the log-linear weight is chosen which corresponds to BLEU-PT$'$, where BLEU-PT$'$ is a training data point closest to BLEU-PT. This approach is close to the cross-domain tuning of Pecina et. al. (2012).

4 Experimental Program

We conducted a number of experiments for English-French language pair, comparing the methods proposed in the previous sections among one another and against state-of-the-art baselines and oracles.

4.1 Datasets

In this section, we present the datasets (EN-FR) that we have used for our experiments and the training data that was created for the purpose of supervised learning. We collected a set of 12 publicly available corpora and 1 proprietary corpus, statistics of datasets are provided in Table 1.

Corpus	Train	Development	Test
Commoncrawl	78M	12.4K	12.6K
ECB	4.7M	13.9K	14K
EMEA	13.8M	14K	15.7K
EUconst	133K	8K	8.4K
Europarl	52.8M	13.5K	13.5K
P1	5M	35K	14.5K
KDE4	1.5M	12.8K	5.8K
News Comm.	4M	12.7K	65K
OpenOffice	400K	5.4K	5.6K
OpenSubs	156M	16K	15.7K
PHP	314K	3.5K	4K
TED	2.65M	21K	14.6K
UN	1.92M	21K	21K

Table 1: Statistics of parallel sets (# of source tokens)

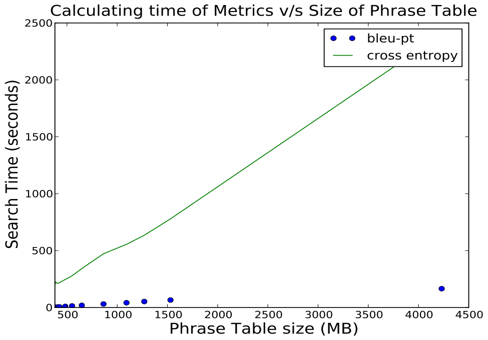

Figure 1: BLEU-PT v/s Cross-Entropy

Commoncrawl (CC) (Smith et al., 2013) and News Commentary (Bojar et al., 2013) corpora were provided in the 2013 shared translation task organized with workshop on machine translation. TED talks data was released as a part of IWSLT evaluation task (Cettolo et al., 2012). ECB, EMEA, EUconst, OpenOffice, OpenSubs 2011, PHP and UN corpora are provided as a part of OPUS parallel corpora (Tiedemann, 2012). The parallel corpora from OPUS were randomly split into training, development and testsets. Commoncrawl, News Commentary and TED datasets were used as they were provided in the evaluation task.

Out of 13 different domain datasets we selected 4 datasets randomly: Commoncrawl, KDE4, TED and UN (in bold in Table 1), to test our methods.

4.2 BLEU-PT v/s Cross-Entropy

We compared the overheads of calculating BLEU-PT and Cross-Entropy[4]. We are interested in estimating whether with increasing number of phrase tables the computation of both measures becomes slow or memory intensive.

[4] We used tmcombine.py script that comes along with the moses package to calculate the mixing coefficients.

Another advantage of using BLEU-PT apart from fast retrieval is that we can index the phrase tables using wFSA based indexing (explanation of indexing the phrase tables is not in the scope of this paper) and store the FSTs in binarised format on disk. When a source sample comes, we just load the indexed binaries and calculate the BLEU-PT while this cannot be achieved when we want to calculate cross entropy because we have to do one pass over all the phrase tables in question.

Experimental results depicted in Figure 1 shows that computation of BLEU-PT is fast (160 seconds) while computation of cross-entropy is slow (42 minutes) when we combine 12 phrase tables with total size of 4.2GB.

4.3 Training data for supervised learning and testing

As mentioned earlier, for estimating the parameters we require a training data containing the tuples of <BLEU-PT, log-linear-weight>. We perform parameter estimation on four of our datasets: Commoncrawl, KDE4, TED and UN. So, for obtaining evaluation results on say, UN, the rest of the resources are used for generating the training data. Our experimental setup can be explained well using the Venn diagram shown in Figure 2.

We set one of four domains as the test domain (in this case, UN) whose parallel set is not available to us and call it setup-UN. The training data tuples obtained from the rest of the 12 datasets are used to estimate parameters for the UN domain. From these 12 datasets we perform a round-robin experiment where one by one each dataset is considered as in-domain and the rest as out-domain. In-domain dataset provides the development set and the rest 11 out-domain models are linearly combined to build translation models. In figure 2, for example, the development set from the TED domain is taken as the development set of the multi-model build using the rest (i.e. excluding TED and UN). This multi-model is built by a weighted linear combination of the out-domain models (11 models). The parameters of this multi-model are tuned on the in-domain development set using MIRA. Simultaneously, we also calculate the BLEU-PT of the linear interpolated model on the source side of the in-domain development set (i.e. TED). This provides us the tuples of BLEU-PT and the log linear weights, which is our training data. So, four sets of experiments are conducted (one each for four datasets considered for testing), and for each set of experiments, there are 12 training data points. The final evaluation is done by measuring the BLEU score obtained on each test set using the predicted parameter estimates.

Reiterating, our optimizing method is fast, and hence, we are not not looking to learn the parameters apriori for all the domains based on a source side of the development set. The goal is to do a fast adaptation by predicting the parameters using statistical models for every new test in a particular domain even in the absence of a parallel development set.

4.4 Prediction

For prediction of parameters for a new domain, the BLEU-PT of the sample source corpus (UN in our example) is measured with the multi-model built on all the models (all the rest of 12 datasets including the TED model) and then the supervised predictor is applied. In our experiments, we test both parametric and non-parametric methods to estimate the parameters based on the training data obtained using the 12 domains.

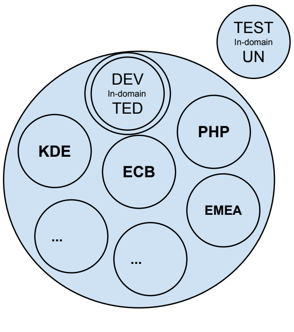

Figure 2: Cross domain tuning setup

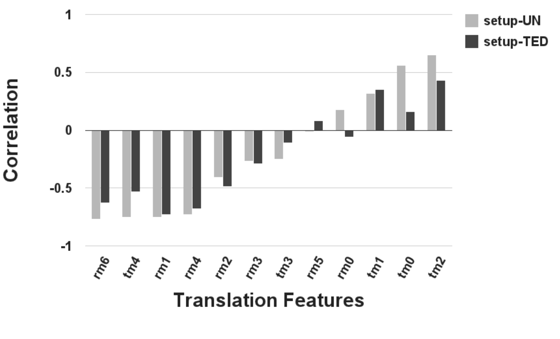

Figure 3: Correlation of log linear weights with BLEU-PT when indomain sets set to UN and TED

	Domain			**Linear Interpolation**		
System	**Train**	**Dev**	**Param. Est.**	**TM(coeff.)**	**RM(coeff.)**	**LM(coeff.)**
in-dom-train	In	In	mira	N.A	N.A	N.A
mira-bleupt-tm-rm	Out	In	mira	✓	✓	✗
mira-perp-tm-bleupt-rm	Out	In	mira	✓(Perp. Min)	✓	✗
mira-bleupt-tm-rm-perp-lm	Out	In	mira	✓	✓	✓(LM Perp. Min.)
mira-bleupt-all	Out	In	mira	✓	✓	✓
def-bleupt-all	Out	✗	def	✓	✓	✓
gen-reg-bleupt-all	Out	✗	regression	✓	✓	✓
gen-mtl-bleupt-all	Out	✗	multi-task	✓	✓	✓
gen-nn-bleupt-all	Out	✗	Near.Neigh.	✓	✓	✓
top5-reg-bleupt-all	Out	✗	regression	✓	✓	✓
top5-mtl-bleupt-all	Out	✗	multi-task	✓	✓	✓
top5-nn-bleupt-all	Out	✗	Near.Neigh.	✓	✓	✓

Table 2: System Description: Each system's training domain and development set domain along with the optimizer/predictor is mentioned. *def-bleupt-all* uses default weights from Moses decoder. Near.Neigh. shows that we used Nearest Neighbor predictor for optimizing weights. ✗ represent log linear interpolation of models while ✓ represents linear interpolation. The mixing coefficients for linear interpolation are calculated by normalizing bleu-pt scores unless mentioned otherwise.

5 Experiments and Results

5.1 Correlation analysis

Before embarking in the actual regression task, we examined the correlation between the similarity values (BLEU-PT) and the various weights in the training data. If there is good correlation between BLEU-PT and a particular parameter, then the linear regressor is expected to fit well and then predict an accurate parameter value for a new domain. For computing the correlation, we use Pearson correlation coefficient (PCC). Figure 3 shows the PCC between the feature weights and the BLEU-PT scores. The tm's are the translation model features, and rm's are the reordering model features.

We see that there is either a strong positive correlation or a strong negative correlation for most features in both the experimental setups shown in the figure 3. This validates our hypothesis that optimal parameters for a new test domain can indeed be estimated with good reliability. One can also observe that the correlation level also varies based on the mixture of training models. For example, the correlation is much higher in the training data that excluded UN (setup-UN) than the one that excluded TED (setup-TED).

In figure 3, one can also see that $tm0$ (forward phrase conditional probability) and $tm2$ (backward phrase conditional probability) which are shown in previous work to be the two most important features amongst all SMT features (Lopez and Resnik, 2006) in terms of their impact on translation quality, have a high correlation in setup-UN.

5.2 Systems

All SMT systems were built using the Moses toolkit (Koehn et al., 2007). To automatically align the parallel corpora we used MGIZA (Gao and Vogel, 2008). Aligned training data in each domain was then used to create the corresponding component translation models and lexical reordering models. We created 5-gram language models for every domain using SRILM (Stolcke, 2002) with improved Kneser-Ney smoothing (Chen and Goodman, 1999) on the target side of the training parallel corpora. Log linear weights for the systems were optimized using MIRA (Watanabe et al., 2007; Hasler et al., 2011) which is provided in the Moses toolkit. Performance of the systems are measured in terms of BLEU computed using the MultEval script (mteval-v13.pl).

We built one *in-dom-train* system where only in-domain training data is taken into account. This system shows the importance of in-domain training data in SMT (Haddow and Koehn, 2012). Three oracle systems are trained on out-domain training corpus and tuned on in-domain development data (in this case there are four domains we chose to test on: UN, TED, CommonCrawl and KDE4), thus 4 systems for each of the in-domain test sets.

We build another set of SMT systems in which language models are combined by linear interpolation[5].

[5] Linear interpolation of 12 LMs result in one single large LM, thus, one weight. So, a total of 14 weights have to be optimized or predicted

The systems using linear interpolated LM (mixing coefficients are normalized BLEU-PT scores) are *def-bleupt-all*, *mira-bleupt-all*, *gen-reg-bleupt-all*, *gen-mtl-bleupt-all* and *gen-nn-bleupt-all*. We compare *mira-bleupt-all* with *mira-bleupt-tm-rm-perp-lm* where mixing coefficients for LM interpolation are calculated by standard LM perplexity minimization method over target side of development set.

As mentioned earlier, ideally only a subset of all the models closer to the source sample should be taken into account for **quick** adaptation, so we select the top five domains related to the source sample and interpolate the respective models and address them as *top5-** systems. Adding more domains would unnecesary increase the size of the model and add more noise. Table 2 shows the configuration of different systems. In the next section we compare the performances of these systems and report the findings.

6 Results and Discussion

Table 3 presents results of the systems that use an in-domain parallel data. As expected, when an in-domain corpus is used both for training as well as for optimizing the log-linear parameters, the peformance is much higher than those systems that do not use in-domain parallel corpus for training (Koehn and Schroeder, 2007). We also observe that the use of normalized BLEU-PT for computing mixing coefficients gives comparable performance to using Cross-Entropy. The primary advantage in using BLEU-PT is that it can be compute much faster than Cross-Entropy (as shown in Figure 1). Evidently, normalized BLEU-PT scores as mixing coefficients performs at par with mixing coefficients retrieved by standard perplexity minimization method (Bertoldi and Federico, 2009). One can also use BLEU-PT for LM interpolation in cases where target side in-domain text is not available.

System	UN	TED	CC	KDE
in-dom-train	67.87	29.98	26.62	35.82
mira-bleupt-tm-rm	**44.14**	31.20	17.43	24.25
mira-perp-tm-bleupt-rm	43.56	31.36	17.54	**24.72**
mira-bleupt-tm-rm-perp-lm	43.96	31.85	**18.45**	23.39
mira-bleupt-all	43.66	**32.04**	18.44	23.09

Table 3: Comparison of In-Domain system versus the established Oracles in different setups.

System	UN	TED	CC	KDE
gen-reg-bleupt-all	43.27	32.18	17.95	21.05
gen-mtl-bleupt-all	43.35	32.61	18.26	20.67
gen-nn-bleupt-all	42.73	31.04	18.24	21.85

Table 4: Performance of generic systems (gen-*) in all setups.

Table 4 illustrates the impact of phrase table retrieval on the performance of multi-model. All the systems presented in this table use BLEU-PT for computing mixing coefficients, while the weights are computed using the three techniques that we explored in this paper. We see that in case of regression, the phrase table retrieval also results in a better MT performance. In the other two cases, the results are comparable. It shows that retrieval helps in building smaller sized multi-models while being more accurate on an average. Phrase table retrieval, thus, becomes particularly useful when a multi-model needs to be built from a library of dozens of pre-trained phrase tables of various domains.

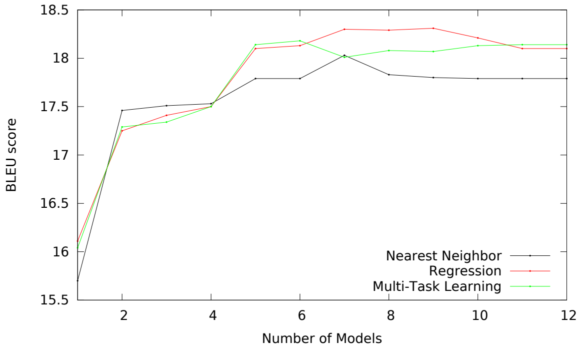

Figure 4: BLEU scores when top k models were used to evaluate commoncrawl test set where $k \in 1..12$.

System	UN	TED	CC	KDE
def-bleupt-all	42.03	30.82	17.97	19.66
mira-bleupt-all	43.66	32.04	18.44	23.09
top5-reg-bleupt-all	43.39$^{\blacktriangle}$	32.31$^{\blacktriangle}$	18.10	21.54$^{\blacktriangle}$
top5-mtl-bleupt-all	43.56$^{\blacktriangle}$	32.60$^{\blacktriangle}$	18.14	20.91$^{\blacktriangle}$
top5-nn-bleupt-all	42.96$^{\blacktriangle}$	30.89$^{\triangle}$	17.79	22.24$^{\blacktriangle}$

Table 5: Comparing the baseline system (def-bleupt-all) and Oracle (mira-bleupt-all) with domain specific multi-model systems trained on top5 domains. $^{\blacktriangle}$and $^{\triangle}$denotes significantly better results in comparison with def-bleupt-all system with p-value < 0.0001 and < 0.05 respectively.

Table 5 compares our approach of computing log-linear weights (in the absence of in-domain development set) to the state-of-art weight optimization technique MIRA (which requires an in-domain development set). As a baseline, we set default weights to all the parameters, which was shown to a strong baseline in (Pecina et al., 2012). We see that the methods proposed by us perform significantly better than the default weights baseline (improvement of more than 1.5 BLEU score on an average across 4 domains). Among the three approaches for computing weights, the method that uses multi-task lasso performs best (except in setup-KDE where the non-parametric method performs best), along the expected lines as multi-task lasso considers the correlation between various features. In comparison to MIRA, our methods result in an average drop of as little as 0.5 BLEU points across 4 domains (see Table 5).

Figure 4 shows BLEU score curve when we vary the k in top-k systems. BLEU score curve is almost tangential zero when k is between 5 and 6 which essentially means that selection of $k = 5$ is a good choice. For CommonCrawl test set, the top five domains used were Europarl, OpenSubs, NewsCommentary, TED and ECB. This is a significant result which indicates that one can build a good system for a domain even in the absence of the parallel data in the domain of interest.

7 Related Work

Domain adaptation in statistical machine translation has been widely studied and leveraged through adding more training data (Koehn and Knight, 2001), filtering of out of domain training data (Axelrod et al., 2011; Koehn and Haddow, 2012), fillup technique (Bisazza et al., 2011), language model adaptation by perplexity minimization over in-domain data (Bertoldi and Federico, 2009) and various other approaches. However, all the above adaptation approaches require either parallel in-domain corpus or monolingual in-domain target side corpus, thus, not applicable in our scenario.

In this paper we studied mixture modelling of heterogeneous translation models which was first proposed in Foster et. al. (2007). They showed various ways of computing mixing coefficients for linear interpolation using several distance based metrics borrowed from information theory. However, to calculate any such metrics it was required that one has an access to the source/target training corpus and source/target development corpus. Other noteable works in mixture modelling in SMT are (Civera and Juan, 2007; Razmara et al., 2012; Duan et al., 2010).

More recently, Sennrich (2012b) designed an approach to calculate mixing coefficients by minimizing the perplexity of translation models over an **aligned** development set for mixture modelling via linear interpolation or by weighting the corpora. Sennrich et. al. (2012a) clustered of a large heterogeneous development corpus and tuned a translation system on different clusters. In the decoding phase each sentence was assigned to a cluster and the translation system tuned on that cluster was used to translate that sentence.

(Banerjee et al., 2010) build several domain specific translation systems, and trained a classifier to assign each incoming sentence to a domain and use the domain specific system to translate the corresponding sentence. They assume that each sentence in test set belongs to one of the already existing domains which means it would fail in the case where the sentence doesn't belong to any of the existing domains. In our case we do not make any such assumptions.

Academically, above approaches are well suited for solving the problem of domain adaptation, but during the deployment of SMT systems in industrial scenario where the client is unable to deliver the parallel in-domain data these approaches fail to provide a quick solution.

8 Conclusion

We present an approach to multi-model domain adaptation in a particularly challenging setting where there is no parallel in-domain data. Parameter estimation without in-domain development set is a problem that, to the best of our knowledge, has not been addressed before. We designed a method for tuning model parameters without parallel development set and validated it through an experimental program for which we compared performances against an array of Oracles and Baselines. The effectiveness of the proposed method empirically supports the findings of (Pecina et al., 2012), who discovered that the log linear weights largely depend on the **distance** of training domain from the domain on which the models are being optimized on. As a side result, we designed in the process a novel similarity metric between a phrase table and a source sample and implemented it effectively using wFSAs. We empirically showed the excellent computation speed of BLEU-PT scores as compared to standard Cross-Entropy measure using standard toolkits.

Acknowledgement

The authors thank the three anonymous reviewers for their comments and suggestions.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 355–362, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pratyush Banerjee, Jinhua Du, Baoli Li, Sudip Kumar Naskar, Andy Way, and Josef Van Genabith. 2010. Combining multi-domain statistical machine translation models using automatic classifiers. In *Proceedings of 9th Conference of the Association for Machine Translation in the Americas*.

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. Association for Computational Linguistics.

Arianna Bisazza, Nick Ruiz, and Marcello Federico. 2011. Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation. In *International Workshop on Spoken Language Translation (IWSLT)*, pages 136–143, San Francisco, CA.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria, August. Association for Computational Linguistics.

Nicola Cancedda. 2012. Private access to phrase tables for statistical machine translation. In *ACL (2)*, pages 23–27.

Rich Caruana. 1997. Multitask learning. *Mach. Learn.*, 28(1):41–75, July.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 4(13):359–393.

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*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Nan Duan, Mu Li, Dongdong Zhang, and Ming Zhou. 2010. Mixture model-based minimum bayes risk decoding using multiple machine translation systems. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 313–321. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, SETQA-NLP '08, pages 49–57, Stroudsburg, PA, USA. Association for Computational Linguistics.

Barry Haddow and Philipp Koehn. 2012. Analysing the effect of out-of-domain data on smt systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 422–432, Montreal, Canada, June. Association for Computational Linguistics.

Eva Hasler, Barry Haddow, and Philipp Koehn. 2011. Margin Infused Relaxed Algorithm for Moses. *The Prague Bulletin of Mathematical Linguistics*, 96:69–78.

Philipp Koehn and Barry Haddow. 2012. Towards effective use of training data in statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, WMT '12, pages 317–321, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn and Kevin Knight. 2001. Knowledge sources for word-level translation models. In *In Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 27–35.

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. Association for Computational Linguistics.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Adam Lopez and Philipp Resnik. 2006. Word-based alignment, phrase-based translation: What's the link? In *In Proceedings of AMTA*, pages 90–99.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In Erhard Hinrichs and Dan Roth, editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-jing Zhu. 2002. BLEU : a Method for Automatic Evaluation of Machine Translation. In *Computational Linguistics*, volume pages, pages 311–318.

Pavel Pecina, Antonio Toral, and Josef van Genabith. 2012. Simple and effective parameter tuning for domain adaptation of statistical machine translation. In *COLING*, pages 2209–2224.

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: Long Papers-Volume 1*, pages 940–949. Association for Computational Linguistics.

Rico Sennrich, Holger Schwenk, and Walid Aransa. 2013. A multi-domain translation model framework for statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 832–840, Sofia, Bulgaria, August. Association for Computational Linguistics.

Rico Sennrich. 2012a. Mixture-modeling with unsupervised clusters for domain adaptation in statistical machine translation. In *Proceedings of the 16th Annual Conference of the European Association of Machine Translation (EAMT)*.

Rico Sennrich. 2012b. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 539–549, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jason R. Smith, Herve Saint-Amand, Magdalena Plamada, Philipp Koehn, Chris Callison-Burch, and Adam Lopez. 2013. Dirt cheap web-scale parallel text from the common crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1374–1383, Sofia, Bulgaria, August. Association for Computational Linguistics.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proceedings of ICSLP*, Denver, Colorado.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uur Doan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 764–773, Prague, Czech Republic, June. Association for Computational Linguistics.

Yi Yang, Heng Tao Shen, Zhigang Ma, Zi Huang, and Xiaofang Zhou. 2011. l2, 1-norm regularized discriminative feature selection for unsupervised learning. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two*, pages 1589–1594. AAAI Press.