PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 103 APRIL 2015 5–20

Domain Adaptation for Machine Translation with Instance Selection

Ergun Biçici

ADAPT CNGL Centre for Global Intelligent Content
School of Computing
Dublin City University

Abstract

Domain adaptation for machine translation (MT) can be achieved by selecting training instances close to the test set from a larger set of instances. We consider 7 different domain adaptation strategies and answer 7 research questions, which give us a recipe for domain adaptation in MT. We perform English to German statistical MT (SMT) experiments in a setting where test and training sentences can come from different corpora and one of our goals is to learn the parameters of the sampling process. Domain adaptation with training instance selection can obtain 22% increase in target 2-gram recall and can gain up to 3.55 BLEU points compared with random selection. Domain adaptation with feature decay algorithm (FDA) not only achieves the highest target 2-gram recall and BLEU performance but also perfectly learns the test sample distribution parameter with correlation 0.99. Moses SMT systems built with FDA selected 10K training sentences is able to obtain F_1 results as good as the baselines that use up to 2M sentences. Moses SMT systems built with FDA selected 50K training sentences is able to obtain 1 F_1 point better results than the baselines.

1. Introduction

Machine translation (MT) performance is affected by tokens unseen in the training set, which may be due to specific use of vocabulary or grammatical structures observed in the test domain of interest. In this paper, we develop a recipe for domain adaptation for MT by comparing different strategies for the selection of training instances close to the test set from larger sets of in-domain (ID) and out-of-domain (OOD) training data. Each corpus has some characteristic distribution of vocabulary

 Corresponding author: ergun.bicici@computing.dcu.ie
Cite as: Ergun Biçici. Domain Adaptation for Machine Translation with Instance Selection. The Prague Bulletin of Mathematical Linguistics No. 103, 2015, pp. 5–20. doi: 10.1515/pralin-2015-0001.

and grammar use specific to its domain, reflected in the training instances selected for a given test corpus or for each test sentence per se. Our goal is to find the best mixture of the selected training instances in a setting where the training and test corpora can come from several different parallel corpora. We can view the test sentences as the result of a mixed selection from different domain corpora since n-grams of a sentence may come from different domains. Each test sentence defines a domain of interest that training sentences can be selected for. Therefore, the boundary between ID and OOD classes is blurred at the sentence level and in-domain or out-of-domainness is decided by a similarity function measuring the closeness of test sentences to training sentences from each domain. Each test sentence has a degree of closeness to the training domains and sampling accordingly can be a good idea. A sampling parameter for a test set specifies how much of it is selected from which domain.

Domain adaptation can be achieved by model weighting, which works with separate training and language models to obtain mixture translation models by linear combination of translation and language model probabilities with weights based on LM probabilities over training corpora split according to their genre (Foster and Kuhn, 2007). Adaptation can also be achieved by weighing the counts in the maximum likelihood estimation of phrase translation probabilities (Sennrich, 2012). Our approach is related to the instance weighting model (Foster et al., 2010). However, the instance selection models we use (Section 2) are based on scores over features consisting of n-grams in contrast to using phrases and relying on the extraction of phrase tables used during training of SMT models.

Biçici and Yuret (2011a) develop feature decay algorithm (FDA) and *dice* instance selection models, which can improve the SMT performance when compared with the performance of the SMT system using all of the training data. The results obtained demonstrate that SMT systems can improve their performance by transductive training set selection. Biçici and Yuret (2011a) focused on training instance selection for a single domain. By contrast, we demonstrate the effectiveness of instance selection for domain adaptation in a setting where test and training sets are selected from multiple separate domains generic enough to be extended to more than two domains. Previous results show that for translation at the sentence level, using only about 5000 training instances can be enough to achieve a performance close to the SMT system trained on millions of sentences (Biçici and Yuret, 2011a; Biçici, 2011).

Statistical MT (SMT) models can make use of various domain-specific training corpora to improve their performance. Adapting to a domain where parallel training resources are scarce can pose a problem for SMT performance. We provide a solution to domain adaptation with training instance selection where we retrieve relevant instances for the test set from a larger set of training instances. Our approach is transductive since we try to find training instances close to the test set and build an SMT model using the selected training set. We focus on how to pick training instances when the test set is a mixture of sentences from two different domains sampled according to a specific sampling parameter. Our goal is to closely mimic the sampling

process of the test set by creating a training set from a mixture of the two domains. We compare different MT training data selection strategies, the results of which reveal how to adapt to a new test set domain. We assume that we have two domains from which we can select training data from: domain A (D_A) and domain B (D_B). Test corpus sentences are sampled from either D_A or D_B. A sampling parameter α, $0 \leq \alpha \leq 1$, is randomly assigned to each test set where $(100 \times \alpha)\%$ of the data is selected from D_A and the rest is selected from D_B. We explore the following training data adaptation strategies:

$\mathbf{R}$	Randomly sampling from $D_A \cup D_B$.
$\mathbf{R}_\alpha$	Randomly sampling from D_A or D_B according to α.
$\mathbf{S}_{0.5}$	Selecting equally from each domain.
$\mathbf{S}_\alpha$	Selecting according to α.
$\mathbf{S}_O$	Selecting from the known, oracle, test sentence domain.
$\mathbf{S}_\cup$	Selecting from $D_A \cup D_B$.
$\mathbf{S}_{\cup\rightleftarrows}$	Selecting from $D_A \cup D_B$ using common cover link (CCL) (Section 2.4).

R_α and S_α assume that α is known beforehand, making R_α a competitive baseline. S_O assumes perfect knowledge of the domain. We can also use a classifier to predict each test source sentence's domain and select from that domain. We use this perfect classification information in the oracle setting. We select either by FDA or *dice* (Section 2) for each test sentence, which also allows us to compare their performance under different domain adaptation strategies. Each training set is the union of the training sentences selected for each test sentence. One of our goals is to understand whether the sampling parameter α, reflected in the training data selections and learn α since we can use α to adapt to a target domain.

Mandal et al. (2008) use the language model (LM) perplexity and inter-SMT-system disagreement to select training data. Moore and Lewis (2010) select training data for LMs using the difference of the cross-entropy of ID and OOD training data: $H_{ID}(s) - H_{OOD}(s)$. OOD LM training data is randomly sampled to make its size close to the ID LM training data and the vocabulary used is set to the ID vocabulary items that are observed at least twice. Axelrod et al. (2011) use bilingual cross-entropy difference:

$$\phi_{aml}(s,t) = H^S_{ID}(s) - H^S_{OOD}(s) + H^T_{ID}(t) - H^T_{OOD}(t), \quad (1)$$

where S stands for the source language, T stands for the target language, and (s,t) is a training sentence pair being scored. Lower $\phi_{aml}(s,t)$ scores correspond to better training instances. Mansour et al. (2011) use IBM Model 1 (Brown et al., 1993) and LM perplexity to filter training data and the LM corpus. We also select according to Equation (1): S_{aml}.

We answer 7 main research questions addressing how much impact does sampling parameter α have on the domain adaptation performance, whether knowing

the domain from where each test sentence is selected from helps the performance, how much instance selection improves the performance, whether we can learn α by instance selection, and what can be the best recipe for domain adaptation in machine translation:

Q1 How much does knowing α improve the random sampling performance? (R vs. R_α)
Q2 Would the performance improve if we select from the exact domain where each test instance is sampled from? (S_O vs. $S_{0.5}$ or S_α)
Q3 How much do we gain by training instance selection? (R vs. $S_{0.5}$ and R_α vs. S_α)
Q4 How much does knowing α improve the selection performance? ($S_{0.5}$ vs. S_α)
Q5 What happens if only use instance selection methods? ($S_\cup$ and $S_{\cup_\rightleftarrows}$)
Q6 Does the selection α resemble the test set α? (Correlation of α vs. $\alpha_{S_\cup}$ and $\alpha_{S_{\cup_\rightleftarrows}}$)
Q7 How should we adaptively select SMT training data for a given test domain?

We use state-of-the-art instance selection models to learn a recipe for domain adaptation. We validate our the domain adaptation approach for not only a single SMT experiment but for 1400 different SMT systems and answer 7 important research questions while comparing 7 domain adaptation strategies. Our results demonstrate that using training instance selection over all of the instances available can increase target 2-gram recall, the percentage of test target 2-grams found in the training set, by 22% and BLEU (Papineni et al., 2002) by 3.55 points. Our results may generalize to other domain adaptation tasks in natural language processing as well such as parsing.

2. Instance Selection Algorithms

We use two training instance selection models for domain adaptation: feature decay algorithms and instance selection for alignment (*dice*), where both try to increase the recall of test target features in the training set. We use a scaling parameter for selecting shorter instances having similar source and target lengths. High coverage of target features in the training sets is important for achieving high BLEU performance (Biçici, 2011).

2.1. Feature Decay Algorithm (FDA)

Feature decay algorithms (Biçici and Yuret, 2011a, 2015) increase the diversity of the training set by decaying the weights of n-gram features that have already been included and try to maximize the coverage of source side features of the test set. FDA decays the initial feature weights as instances containing them are included in the se-

lected training data where the order by which sentences are selected is determined by a sentence score which is calculated by weighted sum of feature weights. Algorithm 1 presents the FDA algorithm.

Algorithm 1: The Feature Decay Algorithm

Input: Parallel training sentences $\mathcal{U}$, test set features $\mathcal{F}$, and desired number of training instances N.
Data: A priority queue $\mathcal{Q}$, sentence scores `score`, feature values `fval`.
Output: Subset of the parallel sentences to be used as the training data $\mathcal{L} \subseteq \mathcal{U}$.

foreach $f \in \mathcal{F}$ **do**
  $\texttt{fval}(f) \leftarrow \texttt{init}(f, \mathcal{U})$
foreach $S \in \mathcal{U}$ **do**
  $\texttt{score}(S) \leftarrow \frac{1}{z} \sum_{f \in \texttt{features}(S)} \texttt{fval}(f)$
  $\texttt{enqueue}(\mathcal{Q}, S, \texttt{score}(S))$
while $|\mathcal{L}| < N$ **do**
  $S \leftarrow \texttt{dequeue}(\mathcal{Q})$
  $\texttt{score}(S) \leftarrow \frac{1}{z} \sum_{f \in \texttt{features}(S)} \texttt{fval}(f)$
  if $\texttt{score}(S) \geq \texttt{topval}(\mathcal{Q})$ **then**
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{S\}$
    foreach $f \in \texttt{features}(S)$ **do**
      $\texttt{fval}(f) \leftarrow \texttt{decay}(f, \mathcal{U}, \mathcal{L})$
  else
    $\texttt{enqueue}(\mathcal{Q}, S, \texttt{score}(S))$

We summarize the initialization, decaying, and scoring used in the FDA version.

Initialization:

$$\texttt{init}(f) = \log(|\mathcal{U}|/C_{\mathcal{U}}(f))$$

Decay:

$$\texttt{decay}(f) = \texttt{init}(f)(1 + C_{\mathcal{L}}(f))^{-1}$$

Sentence score:

$$\texttt{score}(S) = \frac{1}{z} \sum_{f \in F(S)} \textit{fvalue}(f)$$

The input to the algorithm is parallel training sentences, $\mathcal{U}$, the number of desired training instances, and the source-language features of the test set. The feature decay function, `decay`, is the most important part of the algorithm where feature weights are multiplied by $1/(1 + C_{\mathcal{L}}(f))$, where $C_{\mathcal{L}}(f)$ returns the count of f in $\mathcal{L}$, the subset of the corpus to be used as the training data. *fvalue*(.) is a function returning the weight of the argument feature. $F(S)$ returns the features of sentence S. The initialization function, `init`, calculates the log of inverse document frequency (idf), where $|\mathcal{U}|$ is the sum of the number of features appearing in the training corpus and $C_{\mathcal{U}}(f)$ is the number of times feature f appear in $\mathcal{U}$.

In the FDA version used in our experiments, we use a length scaling factor that prefers balanced shorter sentences defined as: $z = |S| \max(\frac{r|S|}{|T|}, \frac{|T|}{r|S|})$, where r is the ratio of the target-sentence length to the source-sentence length observed in the training set. FDA can be used in both transductive learning scenarios where test set is used to select the training data or in active learning scenarios where training set itself is used to obtain a sorting of the training data and select.

2.2. Using FDA5

FDA can be used to model new instance selection methods for natural language processing, information retrieval, machine translation, domain adaptation, or other related tasks where diverse and relevant selection of data is needed or phenomena with decaying feature weights are observed. FDA5 is a 5 parameter version of FDA providing a class of algorithms with feature decay and capability of modeling the behavior of other instance selection models as well (Biçici and Yuret, 2015). FDA5 is developed for efficient parameterization, optimization, and implementation of FDA. FDA5 allows a shift from developing general purpose SMT systems towards task adaptive SMT solutions.

FDA5 and instructions on how to use FDA5 are available at `github.com/bicici/FDA` and the FDA5 optimizer is available at `github.com/bicici/FDAOptimization`. The main parameters to the FDA5 algorithm are presented below:

```
-n (3): maximum n-gram order for features
-t (0): number of training words output, -t0 for no limit
-i (1.0): initial feature score idf exponent
-l (1.0): initial feature score ngram length exponent
-d (0.5): final feature score decay factor
-c (0.0): final feature score decay exponent
-s (1.0): sentence score length exponent

initial feature score: fscore0 = idf^i * ngram^l
final feature score  : fscore1 = fscore0 * d^cnt * cnt^(-c)
sentence score       : sscore  = sum_fscore1 * slen^(-s)
```

2.3. Instance Selection for Alignment

Dice's coefficient (Dice, 1945) is used as a heuristic word alignment technique giving an association score for each pair of word positions (Och and Ney, 2003). *Co-occurrence* of words in the parallel training sentences is used to retrieve sentences containing co-occurring items. Dice's coefficient score is defined as: $\mathrm{dice}(x,y) = \frac{2C(x,y)}{C(x)C(y)}$, where $C(x,y)$ is the number of times x and y co-occur and $C(x)$ is the number of times x appears in the selected training set. *dice* takes a test source sentence, S', and calculates the goodness of a training sentence pair, (S,T), by the sum of the alignment scores as in Equation (2):

$$\phi_{\mathrm{dice}}(S',S,T) = \frac{1}{z} \sum_{x \in X(S')} \sum_{j=1}^{|T|} \sum_{y \in Y(x)} \mathrm{dice}(y,T_j), \tag{2}$$

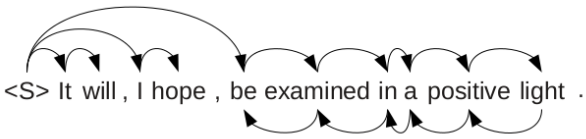

Figure 1. CCL output with arrows representing links, <S> representing the start of the sentence.

where $X(S')$ stores the features of S', $Y(x)$ lists the tokens in feature x, and

$$z = |S| \max(\frac{r|S|}{|T|}, \frac{|T|}{r|S|})(|T| \log |S| + |S| \log |T|)$$

is the scaling factor, which aims balanced shorter sentences that are not very difficult to align. $\phi_{\texttt{dice}}(S', S, T)$ favours the abundance of multiple cooccurring tokens. *dice* selects relevant training sentences for a given test sentence with a goal of improving word alignment performance (Biçici and Yuret, 2011a). SMT systems heavily rely on the word alignment of the parallel training sentences to derive a phrase table.

2.4. Features for Instance Selection

We use n-gram features when selecting training instances with up to 3-grams. We also perform unsupervised parsing using the Common Cover Link (CCL) algorithm (Seginer, 2007) and extract links from the base words to the head words. CCL allows equivalent classes with reciprocal links between words. CCL structures allow us to obtain structures representing the grammar used in the training and test sentences. Figure 1 depicts the parsing output obtained by CCL for an example sentence. Reciprocal links increase the recall and help us find relevant sentences from the training set more easily.

3. Experiments

We run experiments comparing alternative training data adaptation strategies described, which help us answer the research questions we target in Section 1. We perform translation domain adaptation experiments using the phrase-based Moses SMT system (Koehn et al., 2007). We use two parallel corpus domains: domain A (D_A) and domain B (D_B), where the training and test instances can come from. D_A uses Europarl 7 and D_B uses the News Commentary corpus. Both of these corpora are available from the WMT'12 translation challenge website (Callison-Burch et al., 2012). [1] We

[1] Parallel corpora are available from `http://www.statmt.org/wmt12/translation-task.html`

use English-German parallel corpora for our experiments and translate from English to German. D_A contains $1,920,209$ sentences and D_B contains $158,840$ sentences with average target number of words being 23.2 and 22.0, respectively.

We obtain training data by transductively selecting $10,000$ training instances to translate test sets of 100 sentences sampled according to the domain adaptation strategies. The randomly selected α values used converge to 0.5 on average. Each test set defines a new domain that we try to adapt accordingly. For each domain adaptation strategy, we perform 100 training data selection experiments. In order to obtain $10,000$ training instances for a given test set of 100 sentences, we select 100 training instances for each test sentence. This corresponds to 50 times reduction in the number of training instances selected for each sentence but doubling the training data used for translating each compared to previous work (Biçici, 2011). We focus on how to pick the training instances from separate domains when the test set is a mixture of different domain corpora. We build SMT models using Moses for each training data experiment and perform tuning over randomly selected 500 instances separately for each experiment. The LM corpus is Common Crawl from WMT'13 (Bojar et al., 2013) and it is cleaned such that sentences from D_A and D_B are excluded and fixed for all experiments. We train a 4-gram language model using SRILM (Stolcke, 2002). Out of the 1400 SMT experiments, 500 each are run with *dice* or FDA, 300 are run for random selection, and 100 are run using $50,000$ training instances selected from $D_A \cup D_B$ using FDA, corresponding to $S_{\cup 50K}$.

We obtain results that span a wide range of distributional similarities between the training and the test set. In total, we perform 1400 training data selection and SMT experiments using 18 million training, 700 thousand development, and $10,000$ test sentences. We can think of a budgeted SMT training scenario where we have a budget of \$10,000 and pay \$1 per training sentence pair used but we do not pay for searching and picking the ones we want. We are solving the following problem: given a limited budget of \$10,000, a test set of 100 sentences, and two domains to choose training instances from, how should we construct the training set for SMT? The training corpora we use is the embodiment of larger domain corpora (e.g. web crawled corpora) from which training sentences can be selected.

3.1. Training Data Comparison

Table 1 compares the training data selected with each adaptation strategy according to the average source and target recall or coverage ($scov$ and $tcov$), the number of words per sentence they contain, and the number of target 2-grams found. $scov_n$ and $tcov_n$ refer to n-gram $scov$ and $tcov$, and $scov_{\rightleftarrows}$ refer to $scov$ over CCLs. Instance selection results in shorter sentences than the randomly sampled training data but more relevant due to higher recall, the percentage of test set features found in the training set. The columns represent the number of words per sentence (wps), the number of unique 2-grams found on the target side of the training sets, and source and target

		Target Stats		scov			tcov		tcov/n-gram $\times 10^5$	
	Exp	wps	2-grams	$scov_1$	$scov_2$	$scov_{\rightleftarrows}$	$tcov_1$	$tcov_2$	1-gram	2-gram
	R	25.8	120666	.9021	.5918	.5288	.8340	.5007	3.5077	.4149
	R_α	25.4	129838	.9246	.6127	.5448	.8577	.5148	3.2065	.3980
	S_{aml}	13.7	53314	.8046	.3653	.2918	.6747	.2769	4.4230	.5989
dice	$S_{.5}$	14.8	83857	.9929	.9125	.8053	.9069	.5849	4.7626	.6978
dice	S_α	15.2	84946	.9923	.9044	.7966	.9074	.5874	4.8480	.7057
dice	S_O	9.3	47563	.9789	.8388	.7209	.8498	.4965	7.1526	1.0525
dice	$S_\cup$	13.8	76385	.9943	.9248	.8162	.9064	.5884	5.2403	.7726
dice	$S_{\cup_{\rightleftarrows}}$	14.7	78044	.9684	.8534	.8155	.8863	.5652	5.1204	.7266
FDA	$S_{0.5}$	17.4	92070	.9935	.9190	.8252	.9101	.5980	4.5641	.6500
FDA	S_α	18.3	94564	.9927	.9081	.8082	.9110	.6026	4.6085	.6491
FDA	S_O	13.9	72231	.9898	.8858	.7694	.8913	.5665	5.4169	.7937
FDA	$S_\cup$	17.7	87480	**.9947**	**.9286**	.8307	**.9127**	**.6133**	4.9965	.7037
FDA	$S_{\cup_{\rightleftarrows}}$	16.0	81570	.9696	.8630	**.8715**	.8913	.5797	5.0665	.7133
	$S_{\cup_{50K}}$	21.8	366529	.9947	.9288	.8493	.9599	.7419	1.8684	.2032
	$D_{\alpha=1}$	25.6	5645724	.9329	.8087	.7979	.9164	.7547	.7910	.0134
	$D_{\alpha=0}$	23.9	1191613	.9058	.6915	.6790	.8412	.6233	.6557	.0523

Table 1. Training data comparison for each experiment. Numbers represent averages over 100 experiments except the last two rows. Target 2-grams count the number of unique 2-grams found.

1-gram and 2-gram recall. *dice* selects relatively shorter and less diverse training sentences than FDA and obtains slightly lower recall. Both selection models improve the recall significantly. Each coverage level shows the relationship between the test domain and the training domain. We obtain baseline training data, $D_{\alpha=1}$ and $D_{\alpha=0}$, by selecting all of the training instances from D_A ($\alpha = 1$) or D_B ($\alpha = 0$), excluding the test sentences.

dice achieves similar source and target recall levels to FDA using fewer target 1-grams and 2-grams. *dice* achieves higher scores than FDA for tcov / n-gram, which calculates the target recall per the n-grams found in the training set and shows the amount of recall we achieve per n-gram used in the training set. Source recall is the result of the sentence selection process as we select by looking at the source side but target recall is unknown. The strategy $S_\cup$ lets the instance selection model find the relevant instances, which achieves the best results. We observe that additional prior knowledge about the test distribution helps (S_α); even distributing the selections equally ($S_{0.5}$) improves the performance in comparison with S_O (**Q2**, see the next paragraph). We use $\phi_{aml}(s,t)$ with strategy S_O where for each test sentence, only the domain knowledge is used. We randomly select the OOD LM as having sim-

ilar size as the ID LM corpus. The vocabulary consists of the tokens appearing at least twice in D_B. We train an open-vocabulary LM and treat tokens not appearing in the vocabulary file as <UNK>. We use ϕ_{aml} in the oracle setting as a baseline where for each test sentence, we know the domain it is coming from and accordingly we calculate ϕ_{aml} for all sentences in $D_A \cup D_B$ and sort them using Equation (1). Top $tcov_2$ is achieved with strategy $S_\cup$ using FDA. Instance selection across different domains achieve remarkable results by obtaining larger $scov$ and $tcov$ levels than either the individual domains. The ordering obtained among the strategies is given in Equation (3), which forms a recipe for domain adaptation in MT:

$$S_\cup > S_\alpha > S_{0.5} > S_{\cup_\rightleftarrows} > S_O > R_\alpha > R. \quad (3)$$

The ordering in the recipe is obtained according to statistical significance tests with paired t-testing (Wasserman, 2004) using the $tcov_2$ obtained over 100 experiments with different strategies. A $>$ represents statistically significantly better performance and $\geq$ represents better but not statistically significant improvement.

$S_{0.5}$ gives close results to S_α since we selected α randomly and on average it converges to 0.5. We are surprised to see that S_O does not give the best results and obtains the least diverse training data, which reduces its recall. S_O is restricts the domain of the training sentences selected for each test sentence to the known oracle domain whereas S_α has more freedom when selecting by benefiting from relevant instances from the other domain as well. Table 1 shows that S_O is not the best strategy. If each sentence defines a domain of interest, its features may best be utilized by a mixture selection model for domain adaptation as we observe with the $S_\cup$ strategy. $S_{\cup_\rightleftarrows}$ obtains better results than S_O but obtains lower recall than $S_\cup$, which is likely to be due to a lot of CCLs being absent from the training set. Our recipe contains the essence of domain adaptation in a single line and abstracts the results obtained with different domain adaptation strategies. FDA with $S_{\cup_{50K}}$ improves $tcov_1$ by 5 percentage points and $tcov_2$ by 13.

As α converges to 0.5 over all 100 experiments, we have identified 4 cases restricting the α selection range and looked at the closeness of the training data to the test data in Table 2 in terms of the test target 2-grams recall. $\alpha \leq 0.1$ corresponds to selecting at least 90% of test set instances from D_B and $\alpha > 0.9$ selects at least 90% of them from D_A. The $tcov_2$ differences between $\alpha \leq 0.5$ and $\alpha > 0.5$ and between $\alpha \leq 0.1$ and $\alpha > 0.9$ are larger in setting $\cup$. Setting $S_\cup$ performs best when $\alpha > 0.9$, which is expected since it contains mostly sentences from D_A. Table 2 shows that $S_\cup$ achieves the best $tcov_2$ across all α ranges for FDA and most of them for *dice*.

3.2. Translation Results

Table 3 (left) shows the translation performance using a Moses SMT system trained with each training set to translate the test sets and the baseline system results with

	Exp	$\alpha \leq 0.5$	$\alpha > 0.5$	$\alpha \leq 0.1$	$\alpha > 0.9$
	R	.4684	.5330	.4386	.5550
	R_α	.4940	.5357	.4777	.5579
	ϕ_{aml}	.3422	.2116	.3908	.1890
dice	$S_{0.5}$	.5690	.6007	.5560	.6098
dice	S_α	.5692	.6056	.5618	.6237
dice	S_O	.4736	.5193	.4575	.5410
dice	$S_\cup$	.5666	.6101	.5507	.6263
dice	$S_{\cup\rightleftarrows}$	.5421	.5883	.5234	.6048
FDA	$S_{0.5}$	.5815	.6144	.5690	.6237
FDA	S_α	.5812	.6240	**.5757**	.6472
FDA	S_O	.5431	.5898	.5304	.6078
FDA	$S_\cup$	**.5914**	**.6352**	.5742	**.6518**
FDA	$S_{\cup\rightleftarrows}$	.5565	.6029	.5394	.6173
	$S_{\cup 50K}$	.7202	.7637	.7009	.7761

Table 2. Average tcov_2 *comparison of the training data for different* α *ranges.*

$D_{\alpha=1}$ and $D_{\alpha=0}$ [2]. tcov_2 results get reflected to the BLEU performance we obtain. FDA achieves better results than *dice* and both achieve significantly better BLEU performance than random sampling baselines. The BLEU gain becomes 3.55 points versus R and 3 points versus R_α. We present the BLEU and F_1 (Biçici, 2011) performance obtained for different α ranges in Table 3 (right). FDA using the $S_\cup$ strategy achieves the top performance. Instance selection across different domains in setting $S_{\cup 50K}$ achieve remarkable results by obtaining larger F_1 score than both of the domain specific systems. The ordering obtained among the strategies is given in Equation (4):

$$S_\cup > S_\alpha \geq S_{0.5} \geq S_{\cup\rightleftarrows} \geq S_O > R_\alpha > R. \quad (4)$$

The ordering is obtained according to statistical significance tests with paired t-testing using the corpus level BLEU and F_1 (Biçici and Yuret, 2011b; Biçici, 2011) scores. $S_{\cup\rightleftarrows}$, S_α, and $S_{0.5}$ strategies achieve close performance with each other using FDA. The $S_\cup > S_\alpha \geq S_O$ result in both recipes is very important, which shows that the boundaries defining a domain are not clear cut and we are better off using a strong instance selection model over all the available training data. We plot the BLEU performance for increasing α in Figure 2 for FDA. We observe that as $\alpha \to 1$, BLEU increases due to D_A being an easier translation domain. The gap between domain adaptation with FDA and random selection results is lowest around $\alpha = 0.4$. We are also able to obtain as good as the baseline results in terms of F_1 scores using strategy $S_\cup$ and FDA. F_1 score

[2] Baseline results are not an average but the translation performance over all of the 10K test sentences.

	BLEU		F_1	
$D_{\alpha=1}$	.1866		.2458	
$D_{\alpha=0}$	.1785		.2443	
	dice	FDA	*dice*	FDA
R	.1248		.1991	
R_α	.1305		.2066	
ϕ_{aml}	.0851		.1587	
$S_{0.5}$	.1530	.1589	.2385	.2428
S_α	.1542	.1572	.2391	.2410
S_O	.1438	.1549	.2289	.2401
$S_\cup$	**.1570**	**.1603**	**.2409**	**.2442**
$S_{\cup\rightleftarrows}$	.1520	.1537	.2329	.2342
$S_{\cup 50K}$	-	**.1770**	-	**.2559**

	Exp	$\alpha \leq 0.5$	$\alpha > 0.5$	$\alpha \leq 0.1$	$\alpha > 0.9$
	R	.1157	.1339	.1031	.1361
	R_α	.1248	.1363	.1193	.1409
	ϕ_{aml}	.1035	.0885	.1071	.0850
dice	$S_{0.5}$	.1463	.1597	**.1427**	.1654
dice	S_α	.1449	.1634	.1370	.1703
dice	S_O	.1348	.1528	.1257	.1561
dice	$S_\cup$	.1488	.1652	.1385	.1724
dice	$S_{\cup\rightleftarrows}$	.1423	.1617	.1336	.1680
FDA	$S_{0.5}$	.1511	.1667	.1386	.1702
FDA	S_α	.1483	.1660	.1384	.1727
FDA	S_O	.1466	.1632	.1334	.1673
FDA	$S_\cup$	**.1520**	**.1686**	.1397	**.1772**
FDA	$S_{\cup\rightleftarrows}$	.1430	.1644	.1329	.1697
FDA	$S_{\cup 50K}$	**.1690**	**.1850**	**.1610**	**.1915**

Table 3. Average BLEU and F_1 comparison for each experiment setting and baselines (left) and average BLEU comparison for each experiment setting for different α ranges (right).

can be easily interpreted and it correlates well with human judgments (Callison-Burch et al., 2011).

3.3. Instance Selection α

We compare the test sample distribution parameter α with the α present in the selected training sets in training data adaptation strategies $S_\cup$ and $S_{\cup\rightleftarrows}$, which select from $D_A \cup D_B$. We denote the corresponding learned αs as $\alpha_{S_\cup}$ and $\alpha_{S_{\cup\rightleftarrows}}$. Test set α affects the distribution of the features in the selected training sets such that the selection α may mimic the test set α. We use α to measure a given instance selection model's effectiveness in learning the inherent α of a new test domain. Table 4 presents the mean (μ) and variance (σ) of the α values obtained. μ for α is very close to 0.5 since it is randomly selected for each of the 100 experiments. μ for the learned αs are around 0.85 with σ around 0.055. Thus, $S_\cup$ and $S_{\cup\rightleftarrows}$ tend to select about 85% of the training data from D_A. This may be expected since the size of D_A is about 12 times the size of D_B and there may be more relevant instances in D_A. But as we show in the results, the instance selection models overcome this bias and manage to select with close to perfect correlation with the actual α.

Table 4 also presents the correlation results we obtain when we compare the actual αs for all of the 100 experiments with the selected αs. The results show that we can

	μ		σ	
α	0.5059		0.278	
	dice	FDA	*dice*	FDA
$\alpha_{S_\cup}$	0.8106	0.8326	0.062	0.066
$\alpha_{S_{\cup\rightleftarrows}}$	0.8595	0.8731	0.049	0.049

r	*dice*	FDA
$r(\alpha, \alpha_{S_\cup})$	0.9864	0.9857
$r(\alpha, \alpha_{S_{\cup\rightleftarrows}})$	0.9788	0.9783

Table 4. Mean (μ) and variance (σ) of the sampling parameter α values obtained (left) and their correlation (r) (right).

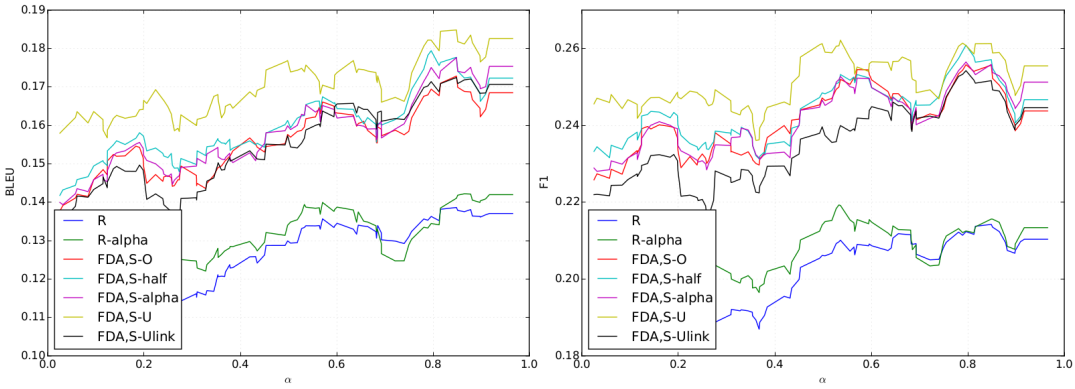

Figure 2. BLEU and F_1 for increasing sampling parameter α for FDA.

achieve close to perfect correlation with the actual αs. Thus, even though the training data adaptation strategies $S_\cup$ and $S_{\cup\rightleftarrows}$ select more from D_A and achieve larger μ for the selected αs, they perfectly correlate with the sampling parameter α. In other words, FDA and *dice* are able to mimic the sampling parameter successfully and still continue to retrieve relevant training instances at the same time.

4. Contributions

Our results answer the questions we have asked in Section 1, which we summarize below:

A1 Knowing α increases $tcov_2$ by 3% and BLEU by 0.26 points when sampling randomly.

A2 Knowing the domain of each test sentence does not improve the performance with FDA or *dice*.

A3 Instance selection can increase $tcov_2$ by 22% and BLEU by 3.55 points when compared with random sampling. Instance selection with known α increases $tcov_2$ by 20.4% and BLEU by 3.24 points.

A4 Knowing α improves BLEU by 0.3 points for *dice* and 0.1 points for FDA but does not significantly increase $tcov_2$.

A5 Instance selection without known α over all available training data using n-gram features achieves the best results with 22% increase in $tcov_2$ and BLEU gains of up to 3.55 points.

A6 Selection αs perfectly mimic the test set α with $r \sim 0.99$, which shows the effectiveness of the instance selection models.

A7 We arrive at a recipe to adapt SMT training data to a given new test domain based on the $tcov_2$ and BLEU performance: $S_{\cup} > S_{\alpha} \geq S_{0.5} \geq S_{\cup_{\rightleftarrows}} \geq S_O > R_{\alpha} > R$. Our results demonstrate that following the recipe can result in gains of up to 3.55 BLEU points and 22% increase in $tcov_2$.

Our results demonstrate that the boundaries defining a domain are not clear cut and domain selection at the corpus level or the sentence level is not as effective as sentence-level training instance selection using all of the available corpora. Each sentence defines a domain of interest and we show that its features are best utilized by a mixture selection model with strategy $S_{\cup}$ using FDA. FDA selected 10K training sentences using strategy $S_{\cup}$ is able to obtain F_1 results as good as the baseline systems using 2M sentences. FDA selected 50K training sentences is able to obtain BLEU results as good as the baseline and obtains 1 F_1 point better results. We also show that our instance selection techniques are able to perfectly learn the sampling parameter of the test set. Matching orderings in the recipes obtained according to coverage and translation performance supports that high coverage is important for achieving high BLEU performance.

We obtain remarkable results showing that instance selection across different domains achieve better $scov$ and $tcov$ than either the individual domains and better F_1 score than both of the domain specific systems in setting $S_{\cup_{50K}}$ using FDA. Our results show that sharing data across different domains is providing an advantage over competing domain specific corpora. Instance selection for domain adaptation is diminishing the competitive advantage of domain specific corpora and encouraging data sharing. We provide our SMT experiments' datasets via a link at `github.com/bicici/MTPPDAT`.

Acknowledgments

This work is supported in part by SFI as part of the ADAPT CNGL Centre for Global Intelligent Content (`www.adaptcentre.ie`, 07/CE/I1142) at Dublin City University and in part by the European Commission through the QTLaunchPad FP7 project (`www.qt21.eu`, No: 296347). We also thank the SFI/HEA Irish Centre for High-

End Computing (ICHEC, www.ichec.ie) for the provision of computational facilities and support.

Bibliography

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

Biçici, Ergun. *The Regression Model of Machine Translation*. PhD thesis, Koç University, 2011. Supervisor: Deniz Yuret.

Biçici, Ergun and Deniz Yuret. Instance selection for machine translation using feature decay algorithms. In *Proc. of the Sixth Workshop on Statistical Machine Translation*, pages 272–283, Edinburgh, Scotland, July 2011a. Association for Computational Linguistics.

Biçici, Ergun and Deniz Yuret. RegMT system for machine translation, system combination, and evaluation. In *Proc. of the Sixth Workshop on Statistical Machine Translation*, pages 323–329, Edinburgh, Scotland, July 2011b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W11-2137.

Biçici, Ergun and Deniz Yuret. Optimizing instance selection for statistical machine translation with feature decay algorithms. *IEEE/ACM Transactions On Audio, Speech, and Language Processing (TASLP)*, 23:339–350, 2015. doi: 10.1109/TASLP.2014.2381882.

Bojar, Ondřej, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proc. of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-2201.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, June 1993.

Callison-Burch, Chris, Philipp Koehn, Christof Monz, and Omer F. Zaidan. Findings of the 2011 Workshop on Statistical Machine Translation. In *Proc. of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, England, July 2011. Association for Computational Linguistics.

Callison-Burch, Chris, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. Findings of the 2012 workshop on statistical machine translation. In *Proc. of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June 2012. Association for Computational Linguistics.

Dice, Lee R. Measures of the amount of ecologic association between species. *Ecology*, 26(3): 297–302, 1945. ISSN 00129658. URL http://www.jstor.org/stable/1932409.

Foster, George and Roland Kuhn. Mixture-model adaptation for SMT. In *Proc. of the Second Workshop on Statistical Machine Translation*, pages 128–135, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W07/W07-0717.

Foster, George F., Cyril Goutte, and Roland Kuhn. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proc. of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459, Cambridge, MA, October 2010. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D10-1044`.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *ACL*, pages 177–180, Prague, Czech Republic, June 2007.

Mandal, A., D. Vergyri, W. Wang, J. Zheng, A. Stolcke, G. Tur, D. Hakkani-Tur, and N.F. Ayan. Efficient data selection for machine translation. In *Spoken Language Technology Workshop, 2008. SLT 2008. IEEE*, pages 261 –264, Dec 2008. doi: 10.1109/SLT.2008.4777890.

Mansour, Saab, Joern Wuebker, and Hermann Ney. Combining translation and language model scoring for domain-specific data filtering. In *International Workshop on Spoken Language Translation*, pages 222–229, San Francisco, California, USA, Dec. 2011.

Moore, Robert C. and William Lewis. Intelligent selection of language model training data. In *Proc. of the Association for Computational Linguistics 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden, July 2010. URL `http://www.aclweb.org/anthology/P10-2041`.

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

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

Seginer, Yoav. *Learning Syntactic Structure*. PhD thesis, Universiteit van Amsterdam, 2007.

Sennrich, Rico. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proc. of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 539–549, Avignon, France, April 2012. URL `http://www.aclweb.org/anthology/E12-1055`.

Stolcke, Andreas. Srilm - an extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 901–904, 2002.

Wasserman, Larry. *All of Statistics: A Concise Course in Statistical Inference*. Springer, 2004.

Address for correspondence:
Ergun Biçici
`ergun.bicici@computing.dcu.ie`
ADAPT CNGL Centre for Global Intelligent Content
School of Computing
Dublin City University
Dublin 9, Dublin, Ireland.