Combining Multi-Domain Statistical Machine Translation Models using Automatic Classifiers

Pratyush Banerjee,[1] **Jinhua Du,**[1] **Baoli Li,**[2] **Sudip Kr. Naskar,**[1] **Andy Way,**[1] **Josef van Genabith**[1]
[1]CNGL, School of Computing,
Dublin City University, Dublin, Ireland
{pbanerjee,jdu,snaskar,away,josef}@computing.dcu.ie
[2]CNGL, Department of Computer Science,
Trinity College, Dublin, Ireland
baoli.li@cs.tcd.ie

Abstract

This paper presents a set of experiments on Domain Adaptation of Statistical Machine Translation systems. The experiments focus on Chinese-English and two domain-specific corpora. The paper presents a novel approach for combining multiple domain-trained translation models to achieve improved translation quality for both domain-specific as well as combined sets of sentences. We train a statistical classifier to classify sentences according to the appropriate domain and utilize the corresponding domain-specific MT models to translate them. Experimental results show that the method achieves a statistically sig nificant absolute improvement of 1.58 BLEU (2.86% relative improvement) score over a translation model trained on combined data, and considerable improvements over a model using multiple decoding paths of the Moses decoder, for the combined domain test set. Furthermore, even for domain-specific test sets, our approach works almost as well as dedicated domain-specific models and perfect classification.

1 Introduction

Domain Adaptation in Machine Translation (MT) has gained considerable importance in recent years. Often a large corpus used to train Statistical Machine Translation (SMT) systems is comprised of many heterogeneous topics. An MT system trained over such a corpus might perform decently in the general domain, but fail to achieve good accuracy on domain-specific data. On the other hand, SMT systems trained and tuned on domain-specific data perform well for the respective domains, but performance deteriorates for out-of-domain sentences (Haque et al., 2009). In order to improve the overall quality of translation over multiple domains, we propose a method of combining multiple domain-specific SMT models using classifiers. The method not only gives better translation over multiple domains, but also provides an easier way of adding new domain-specific models to an existing system.

The paper presents a set of experiments conducted with the goal of building a framework for a large-scale MT system into which domain-specific models can be added upon availability. We focus on how systems trained separately on domain-specific corpora perform in comparison to systems trained with a combined-domain general corpus.

Our experiments were conducted for the Chinese–English language pair in both directions and we use domain-specific corpora comprising of two specific domains: 'Availability' and 'Security'. The data has been provided in the form of a translation memory by Symantec[1] with the specific domains consisting of sentences from two different streams of product documentation. In our work we create domain-specific translation models as well as a combined translation model from both domains. For comparison we also combined the domain-trained models using the multiple decoding paths of the Moses decoder (Koehn et al., 2007). The models and their combinations were tested using test sets comprising of 1000 sentences from each domain, as well as a combined test set of both domains.

[1]http://www.symantec.com/index.jsp

Our results show that a model trained on combined data outperforms the domain-trained models and even the combined models using multiple decoding paths. However, the picture changes when we utilize a sentence classifier to classify sentences according to domain and route them to domain-specific MT systems. This approach also outperforms the combined training model.

The remainder of the paper is organized as follows. In Section 2, we present previous research in the field of Domain Adaptation in SMT. Section 3 describes details of the classifier used in our experiment. Section 4 presents the experimental setup, the data and provides details of the specific experiments carried out. The results and analysis are presented in Section 5. In Section 6, we perform a manual analysis of the system performance using example sentences from the test corpora followed by conclusions and future research directions in Section 7.

2 Related Work

Langlais (2002) imported the concept of Domain Adaptation to SMT by integrating domain-specific lexicons in the translation model resulting in significant improvement in terms of Word Error Rate.

Eck et al. (2004) utilized information retrieval theories to propose a language model adaptation technique in SMT. Their approach was further refined by Zhao et al. (2004). Hildebrand (2005) also utilized the approach of Eck et al. shortciteeck:04 to select similar sentences from available training data to adapt translation models, which significantly improved translation performance over baseline systems.

Hasan and Ney (2005) proposed a method for building class-based language models by clustering sentences into specific classes and interpolating them with global language models achieving improvements in terms of perplexity reduction and error rates in MT. This work was further extended by Yamamoto and Sumita (2007) as well as Foster and Kuhn (2007) to include translation models. Using unsupervised clustering techniques on the bilingual training data, automatic clusters were created and each cluster was treated as a domain (Yamamoto and Sumita, 2007). Using domain-specific language models and translation models to translate sentences from the generated domains resulted in improved translation quality.

Integrating in-domain and out-of-domain language models one using log-linear features of an SMT model was carried out by Koehn and Schroeder (2007). This work also saw the first use of multiple decoding paths for combining multiple domain translation tables within the framework of the Moses decoder (Koehn et al., 2007). The same idea was explored using a different approach by Nakov (2008) using data-source indicator features to distinguish between phrases from different domains within the phrase tables.

Xu et al. (2007) investigated the usage of information retrieval approaches to classify the input test sentences based on the domains, along with domain-dependent language modeling or feature weights combination for domain-specific training of SMT models. This effort resulted in significant improvement in domain-dependent translation when compared to domain-independent translation. Bertoldi and Federico's (2009) experiments utilizing in-domain monolingual resources to improve domain adaptation also achieved considerable improvements. They used domain-specific baseline systems to translate in-domain monolingual data, creating a synthetic bilingual corpus and using the same for adapting the SMT system to perform better.

Our work is closely related to the work of Xu et al. (2007) as well as that of Yamamoto and Sumita (2007), but assumes a specific scenario where sufficient training data is available for separate domains and the test set data comprises multiple sentences from different domains. This setting is particularly relevant for the global industries where there is a high requirement for high-quality translation of text from multiple domains.

3 Domain Classification

One of the essential parts of our experiments is the classifier used to identify the domain of a test sentence. Since the premise of our experiments is to utilize domain-specific translation models to translate in-domain sentences, the accuracy of the classifier plays a vital role in the final translation score of the sentences from the combined test set data.

Considering various text categorization techniques, Support Vector Machines (SVM) perform well for the task of text categorization for a variety of requirements including topic-based text categorization (Joachims, 1999). Hence SVM was chosen to be the classifier for our experiments

3.1 The SVM Classifier

We use an SVM-based classifier with a linear kernel for domain-wise classification of the test set sentences. The classifier was initially trained on the training set data for each domain. The features used to train the classifier included the word stems occurring in the training data sets. For the feature values we utilized the Term Frequency-Inverse Document Frequency (tf-idf) values of the word stems appearing in the training corpus (Salton and Buckley, 1988). However, in our case the categorization is required for the sentence level instead of the document level. Hence we utilized Term Frequency Inverse Sentence Frequency (tf-isf) as the values of the features. For implementation we used the SVM Light application which is an open source implementation of the SVM.[2]

3.2 Feature Extraction

The very first step in document categorization is the transformation of documents into a representation suitable for the SVM learning algorithm and classification task. The requirement being topic-based categorization of sentences, the content words in the sentence constituted the most appropriate features. To capture some limited context in the features we use word bigrams as the feature sets for our classification task. The feature values are the frequency of the word bigrams in the corpus scaled by their inverse sentence frequency. Tf-isf is assumed to work well in the scenario owing to its closeness to the tf-idf (Salton and Buckley, 1988) features. However, we remove any 'stop-words' from the training data before actually creating the word bigrams.

The training sets combined with the development set sentences are pre-processed to extract the feature vectors. Pre-processing includes lowercasing, removing stop-words, and stemming all the words. We use the Porter Stemmer[3] (Rijsbergen et al., 1980) to convert the surface form of the words to their corresponding stems. Mapping different surface forms to a single stem also helps to reduce the dimensionality of the feature space. In order to compute the tf-isf feature vectors for every distinct word bigram in the corpus, the following formulae were used:

$$feature - value(b_i) = tf(b_i) \times isf(b_i)$$
$$tf(b_i) = \frac{n_i}{\sum_k n_k}$$
$$isf(b_i) = log\frac{N}{|\{s : b_i \epsilon s\}|}$$

Here, $tf(b_i)$ denotes the term-frequency of the bigram b_i. n_i is the count of the bigram in the sentence, k being all such bigrams in the same sentence, while N denotes the total number of sentences. $isf(b_i)$ is the Inverse Sentence Frequency for the bigram, and s denotes the set of all sentences containing the bigram within them.

The training set files for the Availability and Security domains contain 130,662 and 95,567 sentences respectively. The feature set comprising of unique stemmed word bigrams contains 250,154 entries. The test sets for the Availability and Security domains each contain 1000 sentences.

3.3 Classification Results

After training, the classifier was used to classify sentences from both domain-specific test corpora as well as a combined domain one. The combined domain test corpus was prepared by combining the two domain-specific test sets. The results of the classification tasks for the three different scenarios are reported in Table 1.

Test Set	Correct Predict	Wrong Predict	Accuracy (%)
Availability	963	37	96.30
Security	962	38	96.20
Combined	1925	75	96.25

Table 1: SVM classifier accuracy for domain-specific and combined test set data

4 Experimental Setup

4.1 Data Sets

The data sets used in our experiments are aligned Chinese–English bitexts, in the form of a translation

[2]http://svmlight.joachims.org/

[3]http://tartarus.org/ martin/PorterStemmer/index.html

memory for two specific domains: Availability and Security, provided by Symantec. Most of the sentences in both domains were obtained from the documentation of the two different product lines. The domain 'Availability' covers data protection, reliability and recovery and Symantec products in this area. The sentences in this domain mostly comprise instructions for usage and tuning of Symantec's data availability tools. The 'Security' domain data on the other hand covers system and data security as well as protection from vulnerability and attacks. Here sentences are primarily obtained from the product manuals instructing users about the usage of data or system security products. We split each of the datasets into training, development and test sets. For development and test set sentences, average sentence length was around 27–30 words since longer sentences are supposed to be more difficult to translate. Table 2 presents the number of sentences for each set as well as the average sentence length (A.S.L) for each data set.

Data Set	Stats	Availability	Security
Training Set	Count	129,662	94,576
	A.S.L.	13.36	12.99
Development Set	Count	1,000	1,000
	A.S.L.	28.95	27.07
Test Set	Count	1,000	1,000
	A.S.L.	28.74	27.47

Table 2: Training, Development and Test Corpus Details

4.2 Training, Tuning and Testing of SMT Models

For our experiments we used OpenMaTrEx (Stroppa and Way, 2006) – an open source SMT system[4] which provides a wrapper around the Moses decoder to train, tune, decode and evaluate our models. BLEU (Papineni et al., 2002) and NIST (Doddington, 2002) scores were used to evaluate the translation results.

BLEU scores measure the precision of unigrams, bigrams, trigrams and four grams with respect to reference translations with a penalty for too short sentences (Papineni et al., 2002). Usually this score reflects the fluency of the translated sentence. On the other hand NIST scores are quite similar to BLEU scores, but use an arithmetic average rather than a geometric one (Doddington, 2002). NIST weighs more informative n-grams higher than the others. Hence NIST scores reflect the adequacy of the translated sentences. In the context of our work, we report both BLEU and NIST scores to capture different aspects of the quality of a translation.

For training, word alignment was performed using Giza++ (Och and Ney, 2003), followed by the creation of the phrase and the re-ordering tables using Moses training (Koehn et al., 2007). 5-gram language models were built on the domain-specific training data using the SRILM toolset (Stolcke, 2002). After training, each of the model components was tuned using Minimum Error-Rate Training (MERT) (Och, 2003) on the BLEU metric. This process helps us tune models to domain-specific development sets.

Finally the models were tested on both domain-specific as well as combined domain test sets. Since the primary objective of our experiments was to achieve better translation of a mix of sentences coming from multiple domains, we tested all our translation models using a combined test set from both domains. Moreover, we also tested the same models using domain-specific test sets to get a clear understanding of the effect of domain-specific data.

4.3 Domain Adaptation Experiments

In this section we describe the different sets of experiments that we carried out with a brief description of the models used therein.

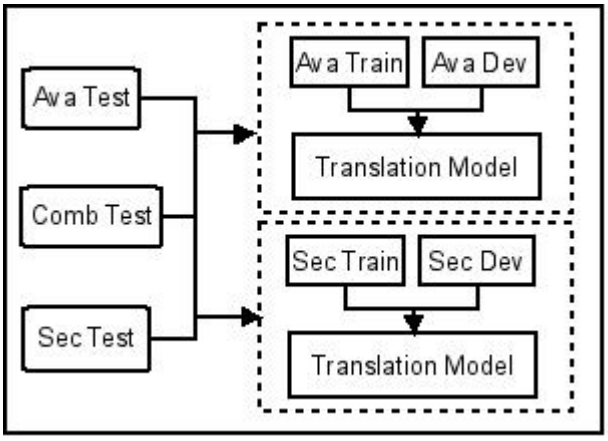

Figure 1: Phase 1 – Simple Domain-Specific Model

In the first phase of our experiments we start with

[4] http://www.openmatrex.org/

creating simple domain-specific models. As in Figure 1, we trained and tuned two different SMT models for both 'Ava(ailability)' and 'Sec(urity)' domains. These models were then subjected to decoding both domain-specific as well as combined domain test sets. Some cross-domain testing was also done by exposing the models trained in one domain to the test sets of the other domain. This provided insights about the importance of domain adaptation of SMT systems for the given sets of data. Henceforth in the paper, we refer to these models as 'Simple Domain-Specific' (SDS) models.

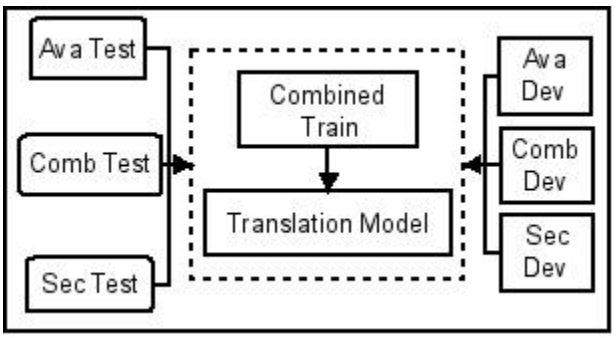

Figure 2: Phase 2 – Combined Data Model

In the second phase, training data from both domains were combined to create a single combined model, as shown in Figure 2. The combined model was then subjected to domain-specific as well as combined-domain data tuning. As in the previous phase, testing was carried out using both domain-specific as well as combined domain training data. The results for this phase not only provide insight into the system performance for the combined data, but also gave an idea about how increasing the training set with data from the other domain might affect the translation scores. We call this model the 'Combined Data' (CD) model in the rest of the paper.

In the third phase of the experiments, we combined the individually trained SDS models using multiple decoding paths of the PB-SMT Moses decoder (Koehn and Schroeder, 2007). The combined models were tuned using both domain-specific and combined-domain development sets. Evaluation of these models against domain-specific as well as combined test sets revealed the effect of model combination using Multiple Decoding Path. Comparing the results of this phase with those of the CD models,

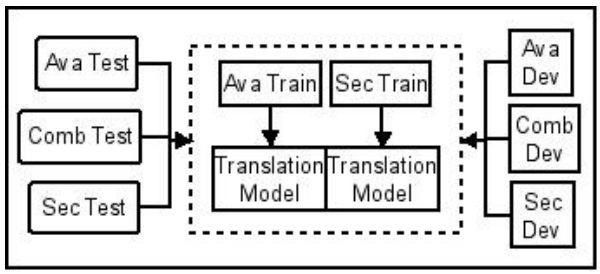

Figure 3: Phase 3 – Domain-Specific Models combined using Multiple Decoding Paths of Moses

gave us a relative idea of the effectiveness of combining pre-trained models with respect to domain adaptation. We call these models 'Multiple Decoding Path Combined' (MDPC) models as shown in Figure 3.

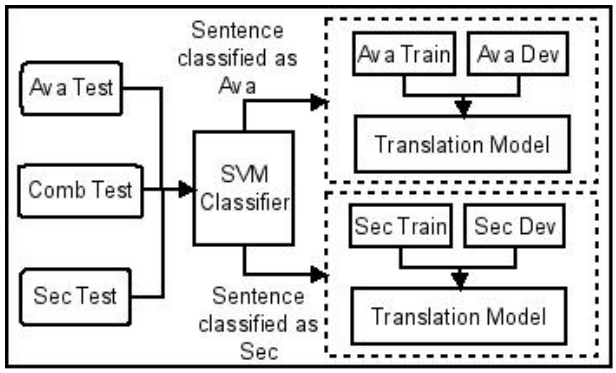

Figure 4: Phase 4 – Domain-Specific Models combined using an SVM-Based sentence classifier

Finally, in the fourth phase of our experiments, we combined the two SDS models (one for each domain) using the sentence classifier (Section 3), rerouting the classified sentences to the appropriate model to utilize the best of the domain-specific models as shown in Figure 4. We used this model to test both domain-specific as well as combined-domain test set data to observe the effect on translation quality. This model is referred to as the 'Classified Domain Specific' (CDS) model in the rest of the paper.

Automatically classifying a domain-specific test set results in labelling a part of the test set as out-of-domain. This basically means that the sentences which originally belonged to one domain were statistically closer to the training data for the other domain. Hence the translation model prepared for the other domain might translate them better. Sentences were routed as per their classification label to the

appropriate models and the translations were then combined for evaluation.

When repeating the experiments with combined domain test set data, since the classifier routes most of the in-domain sentences to the appropriate domain-specific SDS models, quality of translation is assumed to improve.

5 Results and Analysis

In this section we report the evaluation scores for the various models described above. Each subsection contains the results for a single model with respect to three different test sets (Availability, Security and Combined) followed by subsequent analysis of the results.

5.1 Simple Domain-Specific Model Translation Scores

The first phase of our experiments involved using domain-specific (SDS) models and testing them using both combined domain and single domain test sets. Table 3 shows the translation scores for these experiments.

Lang. Pair	Test Data	Training Data	BLEU	NIST
Zh-En	Ava	Ava	56.93	10.26
		Sec	27.47	7.05
	Sec	Ava	26.85	7.03
		Sec	57.07	10.28
	Comb	Ava	42.04	8.99
		Sec	41.86	9.09
En-Zh	Ava	Ava	52.27	9.95
		Sec	24.81	7.16
	Sec	Ava	25.53	7.09
		Sec	53.24	10.03
	Comb	Ava	38.9	8.89
		Sec	38.37	8.76

Table 3: Translation Scores for Simple Domain-Specific models tested with Availability, Security and Combined test data

Table 3 shows that a domain-specific model performs much better translating in-domain sentences compared to out-of-domain ones. The best translation scores are obtained from the models where the training data and test data are from the same domain. When comparing the results of the combined test set with those of the domain-specific ones, it can be observed that the scores are on average 15.05 absolute BLEU points (35.25% relative) lower than in-domain test scores. Again the same scores are on average 14.79 absolute BLEU points (26.4% relative) better than out-of-domain test sets. Since the test set contains an equal number of in-domain and out-of-domain sentences, the high-quality of translation of the in-domain sentences is offset by the poor quality translation of the out-of-domain sentences. Overall, these results strongly suggest the need for domain adaptation to ensure better quality translation for sentences from both domains.

5.2 Combined Data Model Translation Scores

The second phase of our experiments involves training a single system by combining the 'Availability' and 'Security' training data and then subjecting this model to domain-specific as well as combined-domain MERT tuning using the respective development sets. Like the previous model, these models are also tested against both domain-specific as well as combined domain data. Table 4 outlines the translation scores we achieve for these models.

Lang. Pair	Test Data	Dev Set Data	BLEU	NIST
Zh-En	Ava	Ava-MERT	55.57	10.16
		Sec-MERT	55.36	10.13
		Com-MERT	55.38	10.11
	Sec	Ava-MERT	55.57	10.15
		Sec-MERT	54.88	10.07
		Com-MERT	55.02	10.07
	Comb	Ava-MERT	55.42	10.73
		Sec-MERT	55.2	10.68
		Com-MERT	55.2	10.66
En-Zh	Ava	Ava-MERT	51.31	9.87
		Sec-MERT	50.99	9.74
		Com-MERT	51.02	9.79
	Sec	Ava-MERT	48.16	9.99
		Sec-MERT	51.87	9.85
		Com-MERT	52.05	9.89
	Comb	Ava-MERT	51.41	10.44
		Sec-MERT	51.44	10.33
		Com-MERT	51.53	10.38

Table 4: Translation Scores for Combined Data Trained models subjected to domain-specific and combined domain testing

In Table 4, 'Ava-MERT', 'Sec-MERT' and 'Com-MERT' indicate tuning the system using 'Availability', 'Security' and combined development sets re-

spectively. The results show an average improvement of 13.36 BLEU points absolute (31.84% relative improvement) for the combined test set Zh–En translation, and 12.79 BLEU points absolute (33.11% relative) for the En–Zh combined test set translation. Interestingly, combined-domain or domain-specific tuning does not significantly affect the scores for the model. Compared to the results in Table 3, the improvements are simply due to the model being trained on data from both domains. Since data from both domains have been used to train a single SMT model, combined-domain or domain-specific tuning does not significantly alter the component weights. However, when comparing in-domain scores with those of Table 3, a small drop in the respective scores could be attributed to the higher degree of generalization in the combined data model.

In spite of the significant improvement in cross-domain translation scores for the combined domain data, one major disadvantage of this approach is its lack of maintainability. Every time data for a new domain becomes available, we need to retrain the entire system. Moreover, as more domain data is added to the system, it increases the generality leading to poorer scores for domain-specific data.

5.3 Domain-Specific Models Combined Using Multiple Decoding Paths

In the third phase we investigate alternative ways of combining the individual models. We use multiple decoding paths of Moses to combine the two pre-built SDS models. The domain-specific translation models and the language models are combined together to create a new set of combined models, which are subjected to domain-specific and combined-domain tuning and testing just like in phase 2.

The results in Table 5 indicate that for domain-specific tuning, results are on average 13.73 absolute BLEU points (24.81% relative loss) lower than those of the combined model for the combined test set Zh–En data, in Table 4. For En–Zh the figures are on average 8.48 absolute BLEU points lower (16.49% relative). Considering combined-domain tuning, the difference is much lower but still significant. Intuitively, the combined models tuned on domain-specific data biased the component weights towards the specific domain on which the tuning was performed. On the other hand, combined-domain tuning distributed the component weights evenly leading to better scores than domain-specific tuned models for combined-domain data, but still failed to match the respective scores in Table 4. This strongly indicates the limitation of multiple decoding paths in combining pre-trained models from different domains to produce high-quality translations.

Lang. Pair	Test Data	Dev Set Data	BLEU	NIST
Zh-En	Ava	Ava-MERT	56.86	10.31
		Sec-MERT	25.83	6.97
		Com-MERT	50.48	9.56
	Sec	Ava-MERT	21.53	6.96
		Sec-MERT	56.72	10.26
		Com-MERT	52.87	9.83
	Comb	Ava-MERT	40.71	9.25
		Sec-MERT	42.45	9.12
		Com-MERT	52.01	10.24
En-Zh	Ava	Ava-MERT	52.1	9.94
		Sec-MERT	31.55	7.99
		Com-MERT	43.59	8.81
	Sec	Ava-MERT	34.92	7.79
		Sec-MERT	52.05	9.82
		Com-MERT	51.12	9.81
	Comb	Ava-MERT	43.82	9.85
		Sec-MERT	42.07	9.51
		Com-MERT	47.3	9.75

Table 5: Domain-Specific and Combined Domain Translation Scores for Domain-Specific models combined using Multiple Decoding Paths

5.4 Combining Domain-Specific Models Using a Domain Classifier

In the final phase of our experiment we used a classifier (Section 3) to predict the domain of a particular input sentence. With the domain of the test set sentence predicted, it was routed to the SDS model trained for that particular domain. Considering the BLEU scores for all the models we experimented with, the SDS models provided the best translation for in-domain sentences (Table 3). The classifier attempts to predict the most appropriate domain-specific model to translate each sentence from the combined test set. We also tested the classifier-based model with the domain-specific test sets to obtain a better understanding of how well this technique performs for each domain data. Table 6 reports the re-

sults for this set of experiments.

The translation scores show a 1.58 BLEU point absolute improvement (2.86% relative improvement) for ZhEn and 1.095 BLEU point absolute (2.13% relative) improvement for EnZh, over the combined data trained model for the combined domain test set (Table 4). This improvement is also statistically significant with a reliability of 99.83% for the combined domain ZhEn data and 99.54% for the EnZh combined data. Since most of the sentences are handled by the appropriate domain-tuned translation model, the classifier-based model performs better than the combined model. This method utilizes the best of both domain-specific models.

Lang. Pair	Test Data	BLEU	NIST
Zh-En	Ava	56.79	10.21
	Sec	56.83	10.25
	Comb	56.89	10.84
En-Zh	Ava	52.02	9.93
	Sec	53.02	10.02
	Comb	52.52	10.52

Table 6: Translation Results of sentences classified and combined after translation using different models

It is interesting to observe the results of the domain-specific test sets when translated using this model with a classifier. Since the classifier and the translation models were trained on the same data, it appears that the classifier does a good job in deciding the appropriate translation model for a particular sentence. However, as the results suggest, the classifier-based domain-specific test set translation scores (Table 6) are not as good as the ones provided by the SDS models (Table 3), although the results are only marginally poorer. Furthermore, the results consistently outperform the combined data-trained models of phase 2, even for the domain-specific data sets (Table 4). Hence with this model we not only provide an alternative way of combining multiple domain-specific translation models, but also provide better translations compared to training a single model over combined data sets.

6 Manual Analysis

In Section 5 we compared the relative performance of the systems in terms of BLEU/NIST scores. In order to substantiate the claims made on the basis of the scores, we performed some manual analysis by primarily comparing the translations provided by the CDS and the CD systems.

We randomly selected 50 sentences (25 from each domain) from the combined domain test set and manually inspected the translations provided by each of the two models. The analysis revealed that out of the 50 sentences, the translations provided by the CDS system were better for 29 sentences, the CD Model translations were better for 9 sentences and for the remaining 12 sentences the translations provided by both the models were similar. Out of these 29 'better' translations, 17 were due to better word ordering, 7 were due to better lexical selection and 5 were simply better due to inclusion of a keyword when compared with the reference translations. However, out of the 9 translations, where the CDS model actually performed worse than the CD model, 2 were worse in terms of lexical selection, 5 in terms of poor word ordering or syntax and 2 for missing one or more important key words. Table 7 shows a domain-wise breakdown of the manual analysis results, with the first column indicating if the CDS translation is 'better' or 'worse' or 'similar' to the translation provided by the CD system.

CDS Trans	Category	Ava	Sec	Total
Better	Better Lexical Selection	4	3	7
	Better Word Order	11	6	17
	Keyword Present	2	3	5
Worse	Poor Lexical Selection	1	1	2
	Poor Word Order	2	3	5
	Keyword Absent	1	1	2
Similar		4	8	12

Table 7: Categorical Distribution of Manual Analysis Observations

Table 8 shows better translations by the CDS model using an example from each of the 3 categories mentioned above. The first sentence is an example of better lexical choice, the second one of better word order and the last one being better due to presence of a keyword. Observing the breakdown of the three categories, the majority of the better translations (17 out of 29) occur due to improved word ordering which indicates that data in the two domains differ more in terms of style rather than content words. This observation not only supports

our assumption that the CDS model produces better translations, but also justifies our approach of combining multiple models instead of training a single model on combined data. Furthermore, the majority of the sentences translated by the CDS system we found to be better than those translated by the CD system, which corroborates our claim in Section 5 about the superiority of the CDS system over CD and the other systems.

Type	Sentence
Src	如果 文件 夹 大小 仍 超出 限制，则 会 删除 最 旧 的 文件，直 到 文件夹 大小 降到 限制 之内。
Ref	if the size of the folder still exceeds the limit , then the oldest files are deleted until the folder size falls below the limit.
CDS	if the folder size still exceeded limit, it deletes the oldest files until the folder size limit minimize.
CD	if you still folder size exceeded limit , it deletes the oldest files until the folder size limit down.
Src	如果 您 是 在 backup exec 的 一 个 早期 版本 上 进行 安装, 并且 想要 更改 当前 语言 , 请 键入y（是）。
Ref	if you install over a previous version of backup exec , type y if you want to change the current language.
CDS	if you are installing on a previous versions of backup exec, and you want to change the current language, type y.
CD	if you are in the backup exec a previous versions are installing and you want to change the current language, type y.
Src	不 能 手动 删除 由 策略 创建 的 作业。必须 首先 删除 策略 和 选择 项 列表 之间 的 关联 。
Ref	you cannot manually delete a job that was created by a policy. you must first remove the association between the policy and the selection list.
CDS	you cannot manually delete a job that was created by a policy. you must first remove the association between the policy and selection list.
CD	you cannot manually delete a job that was created by a policy. you must first remove policies and selection lists.

Table 8: Example translations where CDS Model outperforms CD Model Translations

Looking deeper into the translations produced by the CDS system for the first two example sentences in Table 8, we observe the differences in n-grams between the CDS translations and the reference translation. These differences cause the first sentence translation to score only 17.76 BLEU points while the second translation obtaining a score of just 43.77 points. Under manual evaluation, these sentences would appear as good translations, with the proper meaning conserved along with the valid syntactic structure. In fact for most of the sentences, the system output is better than what the BLEU scores indicate. The manual analyses of the translation quality of the different systems thereby confirm our previous conclusion that the Classified Domain-Specific (CDS) system outperforms all the other systems, not only in terms of BLEU/NIST scores, but in terms of actual translation quality.

7 Conclusions and Future Work

The set of experiments described in the paper prepares the groundwork for further research on domain adaptation using SMT. The results show that for combined domain test sentences, using a classifier to classify sentences and using domain-specific SMT models to translate them lead to improved translation quality, over both combined data trained models as well as models combined using multiple decoding paths. However, the biggest advantage of our approach is that it also provides a scalable framework to add more domain-specific translation models to an already existing system with ease of use and maintenance.

We intend to use more sophisticated feature sets for the classification task as well as various translation model combination techniques like data source indicator features (Nakov, 2008) in the future. Incorporating more than two domains in experiments to measure the scalability of the approach is also another objective. The goal is the development of a large-scale multi-domain SMT framework with the capability of adding different domain data and models on demand.

Acknowledgments

This work is supported by Science Foundation Ireland (Grant No. 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University. We thank the reviewers for

their insightful comments. We also thank Symantec for kindly providing us with the data sets.

References

Bertoldi, N. and Federico, M. 2009. Domain Adaptation for Statistical Machine Translation with Monolingual Resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation, StatMT '09* Morristown, NJ, pp 182–189

Doddington, George 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second international conference on Human Language Technology Research* San Diego, CA, pages 138–145.

Eck, M., S. Vogel and A. Waibel 2004. Language model adaptation for statistical machine translation based on information retrieval In *Proceedings of the 4th International Conference on language resources and evaluation(LREC-2004)* Lisbon, Portugal, pp. 327–330

Foster, G. and R. Kuhn. 2007. Mixture-model adaptation for SMT. in *ACL 2007: Proceedings of the Second Workshop on Statistical Machine Translation* Prague, Czech Republic, pp.128–135.

Haque, Rejwanul, Sudip Kumar Naskar, Josef van Genabith and Andy Way. 2009. Experiments on Domain Adaptation for English-Hindi SMT. In *Proceedings of PACLIC 23: the 23rd Pacific Asia Conference on Language, Information and Computation* Hong Kong, pp.670–677.

Hasan, S. and H. Ney. 2005. Clustered language models based on regular expressions for SMT. In *Proceedings of EAMT 2005 (10th Annual Conference of the European Association for Machine Translation)* Budapest, Hungary, pp.133–142.

Hildebrand, A.S.,M. Eck, S. Vogel and A. Waibel 2005. Adaptation of the translation model for statistical machine translation based on information retrieval In *Proceedings of EAMT 2005 (10th Annual Conference of the European Association for Machine Translation)* Budapest, Hungary, pp.119–125.

Joachims, T. 1999. Making large-Scale SVM Learning Practical. In *Advances in Kernel Methods – Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.)* Cambridge, MA, pp.169–184

Koehn, P., 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 *ACL-2007: Proceedings of demo and poster sessions* Prague, Czech Republic, pp.177–180.

Koehn, P. and J. Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation* Prague, Czech Republic, pp.224–227.

Langlais, P. 2002. Improving a general-purpose statistical translation engine by terminological lexicons. In *Proceedings of Coling-2002: Second international workshop on computational terminology (COMPUTERM 2002)* Taipei, Taiwan, pp.1–7.

Nakov, P. 2008. Improving English-Spanish statistical machine translation: experiments in domain adaptation, sentence paraphrasing, tokenization, and recasing. *In Proceedings of ACL-08: HLT. Third Workshop on Statistical Machine Translation* The Ohio State University, Columbus, OH. pp.147–150.

Och, F. J. and H. Ney 2003. A Systematic Comparison of Various Statistical Alignment Models In *Computational Linguistics* volume 29, (1), pp. 19–51.

Och, F. J. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association For Computational Linguistics - Volume 1* Sapporo, Japan, pp.160–167.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. 2002. BLEU: a method for automatic evaluation of machine translation In *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics* Philadelphia, PA, pp. 311–318

Rijsbergen C.J. van, Robertson S.E. and Porter M.F. 1980. New models in probabilistic information retrieval. *British Library Research and Development Report, no. 5587*. London: British Library.

Salton, G. and C. Buckley 1988. Term weighting approaches in automatic text retrieval. In *Information Processing and Management* 24(5):513–523

Stolcke, A. 2002. SRILM-an extensible language modelling toolkit. In *Proceedings of the International Conference On Spoken Language Processing* Denver, CO, pp. 901–904

Stroppa, N. and A. Way. 2006. MaTrEx: DCU machine translation system for IWSLT 2006. In *Proceedings of the International Workshop on Spoken Language Translation* Kyoto, Japan, pages 31–36.

Xu J., Y. Deng, Y. Gao, and H. Ney. 2007. Domain dependent statistical machine translation. In *Proceedings of the MT Summit XI* Copenhagen, Denmark. pp.515–520

Yamamoto, H. and E. Sumita. 2007. Bilingual cluster based models for statistical machine translation. In *Proceedings of EMNLP-CoNLL'07* Prague, Czech Republic, pp.514–523.

Zhao B.,M. Eck and S. Vogel. 2004. Language model adaptation for statistical machine translation with structured query models. In *Proceedings of 20th International Conference on Computation Linguistics (Coling 2004)* University of Geneva, Switzerland, pp.1–7.