Multi-domain Adaptation for SMT Using Multi-task Learning*

Lei Cui[1], **Xilun Chen**[2], **Dongdong Zhang**[3], **Shujie Liu**[3], **Mu Li**[3], and **Ming Zhou**[3]

[1]Harbin Institute of Technology, Harbin, P.R. China
leicui@hit.edu.cn
[2]Cornell University, Ithaca, NY, U.S.
xlchen@cs.cornell.edu
[3]Microsoft Research Asia, Beijing, P.R. China
{dozhang,shujliu,muli,mingzhou}@microsoft.com

Abstract

Domain adaptation for SMT usually adapts models to an individual specific domain. However, it often lacks some correlation among different domains where common knowledge could be shared to improve the overall translation quality. In this paper, we propose a novel multi-domain adaptation approach for SMT using Multi-Task Learning (MTL), with in-domain models tailored for each specific domain and a general-domain model shared by different domains. The parameters of these models are tuned jointly via MTL so that they can learn general knowledge more accurately and exploit domain knowledge better. Our experiments on a large-scale English-to-Chinese translation task validate that the MTL-based adaptation approach significantly and consistently improves the translation quality compared to a non-adapted baseline. Furthermore, it also outperforms the individual adaptation of each specific domain.

1 Introduction

Domain adaptation is an active topic in statistical machine learning and aims to alleviate the domain mismatch between training and testing data. Like many machine learning tasks, Statistical Machine Translation (SMT) assumes that the data distributions of training and testing domains are similar. However, this assumption does not hold for real world SMT systems since training data for SMT models may come from a variety of domains. The translation quality is often unsatisfactory when translating texts from a specific domain using a general model that is trained over a hotchpotch of bilingual corpora. Therefore, domain adaptation is crucial for SMT systems to achieve better performance.

Previous research on domain adaptation for SMT includes data selection and weighting (Eck et al., 2004; Lü et al., 2007; Foster et al., 2010; Moore and Lewis, 2010; Axelrod et al., 2011), mixture models (Foster and Kuhn, 2007; Koehn and Schroeder, 2007; Sennrich, 2012; Razmara et al., 2012), and semi-supervised transductive learning (Ueffing et al., 2007), etc. Most of these methods adapt SMT models to a specific domain according to testing data and have achieved good performance. It is natural that real world SMT systems should adapt the models to multiple domains because the input may be heterogeneous, so that the overall translation quality can be improved. Although we can easily apply these methods to multiple domains individually, it is difficult to use the common knowledge across different domains. To leverage the common knowledge, we need to devise a multi-domain adaptation approach that jointly adapts the SMT models.

Multi-domain adaptation has been proved quite effective in sentiment analysis (Dredze and Crammer, 2008) and web ranking (Chapelle et al., 2011), where the commonalities and differences across multiple domains are explicitly addressed by Multi-task Learning (MTL). MTL is an approach that learns one target problem with other related problems at the same time, using a shared feature representation. The key advantage of MTL is to enable implicit data sharing and regularization. Therefore, it often leads to a better model for each task. Analogously, we expect that the overall translation quality can be further improved by using an MTL-based

*This work was done while the first and second authors were visiting Microsoft Research Asia.

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1055–1065,
Seattle, Washington, USA, 18-21 October 2013.

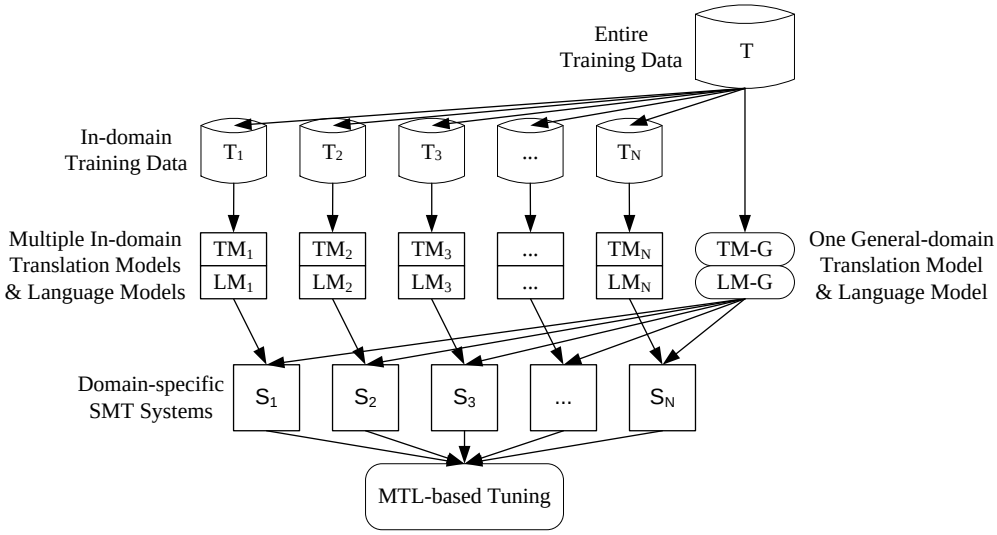

Figure 1: An example with N pre-defined domains, where T is the entire training corpus. T_i is the in-domain training data for the i-th domain selected from T using the bilingual cross-entropy based method (Axelrod et al., 2011). The in-domain TM$_i$ and LM$_i$ are trained using the in-domain training data T_i. The general-domain models TM-G and LM-G are trained using the entire training corpus T. S_i is the domain-specific SMT system for the i-th domain, leveraging the in-domain models and the general-domain models as features.

multi-domain adaptation approach.

In this paper, we use MTL to jointly adapt SMT models to multiple domains. Specifically, we develop multiple SMT systems based on mixture models, where each system is tailored for one specific domain with an in-domain Translation Model (TM) and an in-domain Language Model (LM). Meanwhile, all the systems share a same general-domain TM and LM. These SMT systems are considered as several related tasks with a shared feature representation, which fits well into a unified MTL framework. With the MTL-based joint tuning, general knowledge can be better learned by the general-domain models, while domain knowledge can be better exploited by the in-domain models as well. By using a distributed stochastic learning approach (Simianer et al., 2012), we can estimate the feature weights of multiple SMT systems at the same time. Furthermore, we modify the algorithm to treat in-domain and general-domain features separately, which brings regularization to multiple SMT systems in an efficient way. Experimental results have shown that our method can significantly improve the translation quality on multiple domains over a non-adapted baseline. Moreover, the MTL-based adaptation also outperforms the conventional individual adaptation approach towards each domain.

The rest of the paper is organized as follows: The proposed approach is explained in Section 2. Experimental results are presented in Section 3. Section 4 introduces some related work. Section 5 concludes the paper and suggests future research directions.

2 The Proposed Approach

Figure 1 gives an example with N pre-defined domains to illustrate the main idea. There are three steps in the training phase. First, in-domain training data is selected according to the pre-defined domains (Section 2.1). Second, in-domain models and general-domain models are trained to develop the domain-specific SMT systems (Section 2.2). Third, multiple domain-specific SMT systems are tuned jointly by using an MTL-based approach (Section 2.3).

2.1 In-domain Data Selection

In the first step, in-domain bilingual data is selected from all the bilingual data to train in-domain TMs. We use the bilingual cross-entropy based approach (Axelrod et al., 2011) to obtain the in-domain data:

$$[H_{I-src}(s)-H_{G-src}(s)]+[H_{I-tgt}(t)-H_{G-tgt}(t)] \quad (1)$$

where $\{s,t\}$ is a bilingual sentence pair in the entire bilingual corpus. $H_{I-\text{xxx}}(\cdot)$ and $H_{G-\text{xxx}}(\cdot)$ represent the cross-entropy of a string according to an in-domain LM and a general-domain LM, respectively. "xxx" denotes either the source language (src) or the target language (tgt). $H_{I-src}(s) - H_{G-src}(s)$ is the cross-entropy difference of string s between the in-domain and general-domain source-side LMs, and $H_{I-tgt}(t) - H_{G-tgt}(t)$ is the cross-entropy difference of string t between the in-domain and general-domain target-side LMs. This criterion biases towards sentence pairs that are like the in-domain corpus but unlike the general-domain corpus. Therefore, the sentence pairs with lower scores (larger differences) are presumed to be better.

Now, the question is how to find sufficient monolingual data to train in-domain LMs. A straightforward solution is to collect the data from the internet. There are a large number of monolingual webpages with domain information from web portal sites[1], which can be collected to train in-domain LMs. In large-scale real world SMT systems, practical domain adaptation techniques should target more domains rather than just one due to heterogeneous input. Therefore, we use a web crawler to collect monolingual webpages of N domains from web portal sites, for both the source language and the target language. The statistics of web-crawled data is given in Section 3.1. We use the web-crawled monolingual documents to train N in-domain source-side LMs and N in-domain target-side LMs. Additionally, we also train the source-side and target-side general-domain LMs with all the web-crawled documents from different domains. Finally, these in-domain and general-domain LMs are used to select in-domain bilingual data for different domains according to Formula (1).

2.2 SMT Systems with Mixture Models

In the second step, with the selected in-domain training data, we develop SMT systems based on mixture models. In particular, we use the mixture model based approach proposed by Koehn and Schroeder (2007). Specifically, we have developed N SMT systems for N domains respectively, where each system is a typical log-linear model. For each system, the best translation candidate $\hat{f}$ is given by:

$$\hat{f} = \arg\max_{f} \{P(f|e)\} \tag{2}$$

where the translation probability $P(f|e)$ is given by:

$$\begin{aligned} P(f|e) &\propto \sum_{i} w_i \cdot \log \phi_i(f,e) \\ &= \underbrace{\sum_{j \in \mathbf{I}} w_j \cdot \log \phi_j(f,e)}_{\textbf{I}\text{n-domain}} + \underbrace{\sum_{k \in \mathbf{G}} w_k \cdot \log \phi_k(f,e)}_{\textbf{G}\text{eneral domain}} \end{aligned} \tag{3}$$

where $\phi_j(f,e)$ is the in-domain feature function and w_j is the corresponding feature weight. $\phi_k(f,e)$ is the general-domain feature function and w_k is the feature weight. The detailed feature description is as follows:

In-domain features

- An in-domain TM, including phrase translation probabilities and lexical weights for both directions (4 features)
- An in-domain target-side LM (1 feature)
- word count (1 feature)
- phrase count (1 feature)
- NULL penalty (1 feature)
- Number of hierarchical rules used (1 feature)

General-domain features

- A general-domain TM, including phrase translation probabilities and lexical weights for both directions (4 features)
- A general-domain target-side LM (1 feature)

The feature description indicates that each SMT system contains two TMs and two LMs. The in-domain TMs are trained using the selected bilingual training data according to Formula (1), and the general-domain TM is trained using the entire bilingual training data. For the LMs, we re-use the target-side in-domain LMs and general-domain LM trained

[1]Many web portal sites contain domain information for webpages, such as "www.yahoo.com" in English and "www.sina.com.cn" in Chinese and etc. The webpages are often categorized by human editors into different domains, such as politics, sports, business, etc.

for data selection (Section 2.1). Compared with a normal single-model system, the system with mixture models can balance the contributions from the general-domain and in-domain knowledge. Hence it potentially benefits from both.

2.3 MTL-based Tuning

In the third step, the feature weights in multiple domain-specific SMT systems are estimated. Instead of tuning each domain-specific system separately, we treat different systems as related tasks and tune them jointly in an MTL framework. There are two main reasons for MTL-based tuning:

1. Domain-specific translation tasks share the same general-domain LM and TM. MTL often leads to better performance by leveraging commonalities among different tasks.

2. By enforcing that the general-domain LM and TM perform equally across different domains, MTL provides a kind of regularization to prevent over-fitting.

Formally, the objective function of the proposed MTL-based approach is described as follows:

$$\min_{\mathbf{W}} \left\{ \sum_{i=1}^{N} \textbf{Loss}(\mathbf{E}_i, \hat{\mathbf{e}}(\mathbf{F}_i, \mathbf{w}_i)) \right\} \quad (4)$$

where N is the number of pre-defined domains. $\{\mathbf{F}_i,\mathbf{E}_i\}$ is the in-domain development dataset for the i-th domain. $\mathbf{F}_i$ denotes the source sentences and $\mathbf{E}_i$ denotes the reference translations. $\mathbf{w}_i$ is a D-length feature weight column vector for the i-th domain, where D is the dimension of the feature space. $\mathbf{W}$ is a N-by-D matrix, representing $[\mathbf{w}_1|\mathbf{w}_2|\ldots|\mathbf{w}_N]^T$. $\hat{\mathbf{e}}(\mathbf{F}_i, \mathbf{w}_i)$ are the best translations obtained for $\mathbf{F}_i$ with parameters $\mathbf{w}_i$. $\textbf{Loss}(\cdot,\cdot)$ denotes the loss between the system's output and the reference translations. The basic idea of the objective function is to minimize the sum of loss functions for all the domains, rather than one domain at a time. Therefore, by adjusting the in-domain and general-domain feature weights, the translation quality is expected to be good across different domains.

To effectively tune SMT systems jointly, we modify the asynchronous Stochastic Gradient Descend (SGD) Algorithm (Simianer et al., 2012) to optimize objective function (4). We follow the pairwise ranking approach with the perceptron algorithm (Shen and Joshi, 2005) to update feature weights. Let a translation candidate be denoted by its feature vector $\mathbf{v} \in \mathbb{R}^D$, the pairwise preference for training is constructed by ranking two candidates according to the smoothed sentence-level BLEU (Liang et al., 2006). For a preference pair $\mathbf{v}_{[j]}$=($\mathbf{v}^{(1)}, \mathbf{v}^{(2)}$) where $\mathbf{v}^{(1)}$ is preferred, a hinge loss is used:

$$L(\mathbf{w}_i) = (-\langle \mathbf{w}_i, \mathbf{v}^{(1)} - \mathbf{v}^{(2)} \rangle)_+ \quad (5)$$

where $(x)_+ = \max(0, x)$ and $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors. With the perceptron algorithm (Shen and Joshi, 2005), the gradient of the hinge loss is:

$$\nabla L(\mathbf{w}_i) = \begin{cases} \mathbf{v}^{(2)} - \mathbf{v}^{(1)} & \text{if} \langle \mathbf{w}_i, \mathbf{v}^{(1)} - \mathbf{v}^{(2)} \rangle \leq 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

The training instances for the discriminative learning in pairwise ranking are made by comparing the N-best list of the translation candidates scored by the smoothed sentence-level BLEU (Liang et al., 2006). Following Simianer et al. (2012), the N-best list is divided into three bins: the top 10% (High), the middle 80% (Middle), and the last 10% (Low). These bins are used for pairwise ranking where the translation preference pairs are built between the candidates in High-Middle, Middle-Low, and High-Low, but not the candidates within the same bin, which is shown in Figure 2. The idea is to guarantee that the ranker is more discriminative to prefer the good translations to the bad ones.

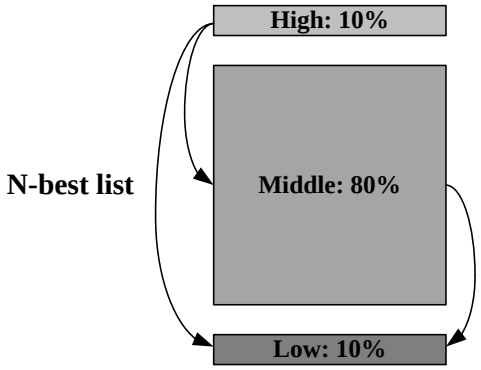

Figure 2: Training instances for pairwise ranking.

Algorithm 1 Modified Asynchronous SGD

1: Distribute N domain-specific decoders to N machines
2: Initialize $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N \leftarrow 0$
3: **for epochs** $t \leftarrow 0 \ldots T-1$ **do**
4: **for all domains** $d \in \{1 \ldots N\}$: **parallel do**
5: $\mathbf{u}_{d,t,0,0} = \mathbf{w}_d$
6: $S = |\mathbf{F}_d|$
7: **for all** $i \in \{0 \ldots S-1\}$ **do**
8: Decode i-th sentence with $\mathbf{u}_{d,t,i,0}$
9: P = No. of pairs built from the N-best list
10: **for all pairs** $\mathbf{v}_{[j]}, j \in \{0 \ldots P-1\}$ **do**
11: $\mathbf{u}_{d,t,i,j+1} \leftarrow \mathbf{u}_{d,t,i,j} - \eta \nabla L(\mathbf{u}_{d,t,i,j})$
12: **end for**
13: $\mathbf{u}_{d,t,i+1,0} \leftarrow \mathbf{u}_{d,t,i,P}$
14: **end for**
15: **end for**
16: **for all domains** $d \in \{1 \ldots N\}$ **do**
17: $\mathbf{w}_d = \mathbf{u}_{d,t,S,0}$
18: **end for**
19: $\mathbf{W}^G \leftarrow [\mathbf{w}_1^G | \ldots | \mathbf{w}_N^G]^T$
20: **for all domains** $d \in \{1 \ldots N\}$ **do**
21: **for** $k \leftarrow 1 \ldots |\mathbf{w}_d^G|$ **do**
22: $\mathbf{w}_d^G[k] = \frac{1}{N} \sum_{n=1}^{N} \mathbf{W}^G[n][k]$
23: **end for**
24: $\mathbf{w}_d \leftarrow \begin{bmatrix} \mathbf{w}_d^I \\ \mathbf{w}_d^G \end{bmatrix}$
25: **end for**
26: **end for**
27: **return** $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N$

Our modified algorithm is illustrated in Algorithm 1. Each column vector $\mathbf{w}_i$ is further split into two parts $\mathbf{w}_i^I$ and $\mathbf{w}_i^G$, representing the **I**n-domain and **G**eneral-domain feature weights respectively. In Algorithm 1, we first distribute the domain-specific SMT decoders to different machines and initialize the feature weights (line 1-2). Typically, the SGD algorithm runs in several iterations (In this study, we set the number of epochs T to 20) (line 3). Multiple SMT decoders run in parallel and each decoder updates its feature weights individually using its in-domain development data (line 4-15). For each domain, the domain-specific decoder translates each in-domain development sentence and determines the N-best translations (line 4-8). The preference pairs are built and used to update the parameters by gradient descent with $\eta = 0.0001$ (line 9-13). Each domain-specific decoder translates its in-domain development data multiple times. After each iteration, feature weights from all decoders are collected (line 16-19). In contrast to the original algorithm (Simianer et al., 2012), we only average the general-domain feature weights $\mathbf{w}_1^G, \ldots, \mathbf{w}_N^G$, but do not average the in-domain feature weights (line 20-25). The reason is we hope to leverage the commonalities among these systems. Meanwhile, general knowledge is enforced to be conveyed equally across different domains. Finally, the algorithm returns all the domain-specific feature weights $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N$ that are used for testing (line 27).

After the joint MTL-based tuning, the feature weights tailored for domain-specific SMT systems are used to translate the testing data. We collect in-domain testing data for each domain to evaluate the domain-specific systems. Although this is not always the case in real applications where the testing domain is known, this study mainly focuses on the effectiveness of the MTL-based tuning approach.

3 Experiments

3.1 Data

We evaluated our MTL-based domain adaptation approach on a large-scale English-to-Chinese machine translation task. The training data consisted of two parts: monolingual data and bilingual data. The monolingual data was used to train the source-side and target-side LMs, both of which were used for data selection in Section 2.1. In addition, the target-side LMs were re-used in the SMT systems as features. As mentioned in Section 2.1, we built a web crawler to collect a large number of webpages from web portal sites in English and Chinese respectively. In the experiments, we mainly focused on six popular domains, namely Business, Entertainment, Health, Science & Technology, Sports, and Politics. For both English and Chinese webpages, the HTML tags were removed and the main content was extracted. The data statistics are shown in Table 1.

The bilingual data we used was mainly mined from the web using the method proposed by Jiang et al. (2009), with a post-processing step using our bilingual data cleaning method (Cui et al., 2013). Therefore, the data quality is pretty good. In addition, we also used the English-Chinese parallel corpus released by LDC[2]. In total, the bilingual data

[2]LDC2003E07, LDC2003E14, LDC2004E12, LDC2005T06, LDC2005T10, LDC2005E83, LDC2006E26,

Domain	English		Chinese	
	Docs	Words	Docs	Words
Business	21M	10.4B	7.91M	2.73B
Ent.	18.3M	8.29B	4.16M	1.31B
Health	8.7M	4.73B	0.9M	0.42B
Sci&Tech	10.9M	5.33B	5.28M	1.6B
Sports	18.9M	9.58B	2.49M	0.59B
Politics	10.3M	5.56B	1.67M	0.39B

Table 1: Statistics of web-crawled monolingual data, in numbers of documents and words (main content). "M" refers to million and "B" refers to billion.

contained around 30 million sentence pairs, with 404M words in English and 329M words in Chinese. For each domain, we used the cross-entropy based method in Section 2.1 to rank the entire bilingual data, and the top 10% sentence pairs from the ranked bilingual data were selected as the in-domain data to train the in-domain TM. Moreover, we prepared 2,000 in-domain sentences for development and 1,000 in-domain sentences for testing in each domain. The details are shown in Table 2.

Domain	Train		Dev		Test	
	En	Ch	En	Ch	En	Ch
Business	30M	28M	36K	35K	19K	19K
Ent.	25M	22M	21K	18K	13K	12K
Health	23M	20M	33K	33K	21K	22K
Sci&Tech	28M	26M	46K	45K	27K	27K
Sports	19M	16M	18K	14K	10K	9K
Politics	28M	24M	19K	17K	13K	12K

Table 2: Statistics of in-domain training, development and testing data, in number of words.

3.2 Setup

An in-house hierarchical phrase-based SMT decoder was implemented for our experiments. The CKY decoding algorithm was used and cube pruning was performed with the same default parameter settings as in Chiang (2007). We used a 100-best list from the decoder for the pairwise ranking algorithm. Translation models were trained over the bilingual data that was automatically word-aligned using GIZA++ (Och and Ney, 2003) in both directions, and the diag-grow-final heuristic was used to refine the symmetric word alignment. The phrase tables were filtered to retain top-20 translation candidates for each source phrase for efficiency. An in-house language modeling toolkit was used to train the 4-gram language models with modified Kneser-Ney smoothing (Kneser and Ney, 1995) over the web-crawled data. The evaluation metric for the overall translation quality was case-insensitive BLEU4 (Papineni et al., 2002). A statistical significance test was performed using the bootstrap resampling method (Koehn, 2004).

LDC2006E34, LDC2006E85, LDC2006E92.

3.3 Baseline

We have two baselines. The first baseline is a non-adapted Hiero using our implementation. It contained the general-domain TM and LM, as well as other standard features. In addition, the fix-discount method (Foster et al., 2006) for phrase table smoothing was also used. The system was general-domain oriented and it was tuned by using MERT (Och, 2003) with a combination of six in-domain development datasets. The second baseline is Google Online Translation Service[3]. We obtained the English-to-Chinese translations of the testing data from Google Translation to have a more solid comparison.

Moreover, we also compared our method with the adapted systems towards each domain individually (Koehn and Schroeder, 2007). This is to demonstrate the superiority of our MTL-based tuning approach across different domains.

3.4 Results

The end-to-end translation performance is shown in Table 3. We found that the baseline has a similar performance to Google Translation, with certain domains performed even better (Business, Sci&Tech, Sports, Politics). This demonstrates that the translation quality of our baseline is state-of-the-art. Moreover, we can answer three questions according to the experimental results as follow:

First, is domain mismatch a significant problem for a real world SMT system? We used the same system only with general-domain TM and LM, but tuned towards each domain individually using in-domain dev data. Table 3 shows that the setting "[A] *G*-TM + *G*-LM" performs much better than

[3]http://translate.google.com

	Business	Ent.	Health	Sci&Tech	Sports	Politics
[N] Baseline (*G*-TM + *G*-LM)	27.19	17.87	25.79	25.34	25.53	23.01
Google Translation	26.01	18.44	27.71	25.07	24.08	22.97
[A] *G*-TM + *G*-LM	29.58	19.08	28.80	26.84	30.28	25.64
[A] *I*-TM + *I*-LM	28.20	17.25	27.20	25.41	30.12	22.97
[A] (*G*+*I*)-TM + *G*-LM	29.45	19.22	28.93	27.01	31.01	25.40
[A] (*G*+*I*)-TM + *I*-LM	29.60	19.43	28.94	27.05	**34.36**	25.98
[A] (*G*+*I*)-LM + *G*-TM	29.66	19.50	29.00	27.10	33.60	26.03
[A] (*G*+*I*)-LM + *I*-TM	28.50	17.66	27.58	25.99	30.44	23.30
[A] (*G*+*I*)-TM + (*G*+*I*)-LM	29.82	19.53	29.03	26.94	33.77	26.09
[A,MTL] (*G*+*I*)-TM + (*G*+*I*)-LM	**30.26**	**19.94**	**29.08**	**27.17**	34.11	**26.50**

Table 3: End-to-end experimental results (BLEU4%) with large-scale training data ($p < 0.05$). "[N]" means the system is non-adapted and tuned using MERT on general-domain dev data. "[A]" denotes that the system is adapted towards each domain individually using MERT on in-domain dev data. "[A,MTL]" indicates that the system was tuned using our MTL-based approach on in-domain dev data. "*I*-TM" and "*G*-TM" denote the in-domain and general-domain translation model. "*I*-LM" and "*G*-LM" denote the in-domain and general-domain language model. We also obtained translations of the testing data using Google Translation for comparison.

the non-adapted baseline across all domains with at least 1.2 BLEU points. In addition, the setting "[A] *G*-TM + *G*-LM" also outperforms Google Translation on all domains. Analogous to previous research, this confirms that the domain mismatch indeed exists and the parameter estimation using in-domain dev data is quite useful.

Second, does the mixture models based adaptation work for a variety of domains? We experimented with different settings with multiple TMs or LMs, or both. It is interesting to note that for large-scale SMT systems, using in-domain models alone is inferior to using the general models alone. The setting "[A] *G*-TM + *G*-LM" is better than the setting "[A] *I*-TM + *I*-LM" across different domains. The reason is the data for general models has already included the in-domain data and the data coverage is much larger, thus the probability estimation is more reliable and the translation quality is much better.

For the LM, the in-domain LM performs better than the general-domain LM because our monolingual data (Table 1) for each domain is already sufficient for training an in-domain LM with good performance. From Table 3, we observed that the setting "[A] (*G*+*I*)-TM + *I*-LM" outperforms "[A] (*G*+*I*)-TM + *G*-LM", with the "Sports" domain being the most significant. For the TM, the performance of the in-domain TM is inferior to the general-domain TM. The results show that the setting "[A] (*G*+*I*)-LM + *G*-TM" is significantly better than "[A] (*G*+*I*)-LM + *I*-TM". The main reason is the data coverage for in-domain TM is much smaller than the general model. When each system uses two TMs and two LMs, it consistently results in better performance, indicating that mixture models are crucial for domain adaptation in SMT.

Third, can MTL further improve the translation quality? We used the MTL-based approach to jointly tune multiple domain-specific systems, leveraging the commonalities among different but related tasks. From Table 3, the MTL-based approach significantly improve the translation quality over the non-adapted baseline, and also outperforms conventional mixture models based methods. In particular, the "Sports" domain benefits the most from the in-domain knowledge, which confirms that domain discrepancy should be addressed and may bring large improvements on certain domains.

3.5 Discussion

According to our experiments, only averaging over the out-of-domain feature weights returned robust and converged results. We do not have theoretically grounded guarantee. However, we observed that the BLEU score of our method on DEV data was slightly lower than that in the baseline system, which indicates the out-of-domain features are less over-fitting on the domain-specific DEV data since

SOURSE	A point begins with a **player** serving the ball. This means one **player** hits the ball towards the other **player**. (The **serve** must be played from behind the baseline and must **land** in the **service box**. Players get two attempts to make a good **serve**.)
REF	得分由一个**球员**发球开始，这是指一个**球员**向另一个**球员**击球。(**发球**时选手必须站在底线之外，球必须要**落在**对方的**发球区**内，每次**发球**允许有一次失误。)
[N] Baseline (*G*-TM + *G*-LM)	舞会始于**玩家**服务的一个点。这意味着**玩家**对其他**玩家**的击球。(该**服务**必须从背后打的基线和必须**降落在**服务框。球员两次试图成为一个好的**服务**。)
[A] (*G+I*)-TM + (*G+I*)-LM	一开始球的**球员**，这意味着一名**球员** 球打向其他**球员**。(必须从底线**发球**，必须在**发球区**的**区域**。球员只有两次尝试去做一个好的**发球**。)
[A,MTL](*G+I*)-TM + (*G+I*)-LM	第一球的**球员**，这意味着一名 **球员**对另一个**球员**击球。(必须在底线后面**发球**，并且必须**降落在**发球区。球员两次试图成为一个好的**发球**。)

Table 4: Examples illustrating some different translations, where the Chinese phrases are translated from the English phrases with the same symbols (e.g., underline, wavy-line, and box). The details are explained in Section 3.5.

we enforced them to play the same role across different domains. It seems that averaging the out-of-domain feature weights can be considered as a kind of regularization.

An example sentence from the Sports domain with translations from different methods is shown in Table 4. In this sentence, the baseline always translates "player" to "玩家" (game player), which should be "球员" (ball player). And, the baseline translates "serve" to "服务" (work for), which should be "发球" (put the ball into play). The phrase "service box" here means "发球区", which denotes the zone where the ball is to be served. However, the baseline incorrectly splits them into two words, then translates "service" to "服务" and "box" to "框". In contrast, the approaches with adapted models are able to translate these words very well.

Both our MTL-based approach and the conventional adaptation methods leverage the mixture models. A natural question is why our MTL-based approach performs better than the individual adaptation. To answer this question, we looked into the details of the tuning and decoding procedures in the MTL-based approach. We observed that the BLEU score on the development data for each system was lower than the score when conducting individual adaptation. Considering that the algorithm enforcing the general features play the same role across different domains, we suspect that MTL-based approach introduces a kind of regularization for each domain-specific system. The regularization prevents the general features from biasing towards certain domains to the extreme. This property is quite important for real world SMT systems. Usually, a sentence is composed of some domain-specific words and some general words, so it is often improper to translate every word in the sentence using the in-domain knowledge. For the example in Table 4, the individual adaptation method "[A] (*G+I*)-TM + (*G+I*)-LM" translates "land" to "区域" (zone) improperly, because "区域" appears more often in the Sports text than the general-domain text. This shows that the individual adaptation methods tend to overfit the in-domain development data. In contrast, the MTL-based approach "[A,MTL](*G+I*)-TM + (*G+I*)-LM" just translates "land" to "降落在" (fall on), which is more appropriate.

4 Related Work

4.1 Domain Adaptation

One direction of domain adaptation explored the data selection and weighting approach to improve the performance of SMT on specific domains. Eck

et al. (2004) first decoded the testing data with a general TM, and then used the translation results to train an adapted LM, which was in turn used to re-decode the testing data. Lü et al. (2007) tried to weight the training data according to the similarity with test data using information retrieval models, while Foster et al. (2010) trained a discriminative model to estimate a weight for each sentence in the training corpus. Other methods conducted data selection based on cross-entropy (Moore and Lewis, 2010), and Axelrod et al. (2011) further extended their cross-entropy based method to the selection of bilingual corpus in the hope that more relevant corpus to the target domain could yield smaller models with better performance. Other methods included using semi-supervised transductive learning techniques to exploit the monolingual in-domain data (Ueffing et al., 2007).

Adaptation methods also involved the utilization of mixture models. Foster and Kuhn (2007) explored a number of variants of utilizing multiple TMs and LMs by interpolation. Koehn and Schroeder (2007) used MERT to simultaneously tune two TMs or LMs. Sennrich (2012) investigated the TM perplexity minimization as a method to set model weights in mixture modeling. In addition, inspired by system combination approaches, Razmara et al. (2012) used the ensemble decoding method to mix multiple translation models, which outperformed a variety of strong baselines.

Generally, most previous methods merely conducted domain adaption for a single domain, rather than multiple domains at the same time. One could also simply build multiple SMT systems that were adapted to multiple domains, but they were often separated and not tuned together. So far, there has been little research into the multi-domain adaptation problem over mixture models for SMT systems, as proposed in this paper.

4.2 Multi-task Learning

In machine learning, MTL is an approach to learn one target problem with other related problems at the same time. This often leads to a better model for the main task because it allows the learner to use the commonality among the tasks. MTL is performed by learning tasks in parallel while using a shared representation. Therefore, what is learned for each task can help other tasks be learned better.

MTL was successfully applied in some Natural Language Processing (NLP) tasks. For example, Blitzer et al. (2006) extended the MTL approach (Ando and Zhang, 2005) to domain adaptation tasks in part-of-speech tagging. Collobert and Weston (2008) proposed using deep neural networks to train a set of tasks, including part-of-speech tagging, chunking, named entity recognition, and semantic roles labeling. They reported that jointly learning these tasks led to superior performance. MTL was also applied in sentiment analysis (Dredze and Crammer, 2008) and web ranking (Chapelle et al., 2011) to address the multi-domain learning and adaptation. In SMT, Duh et al. (2010) proposed using MTL for N-best re-ranking on sparse feature sets, where each N-best list corresponded to a distinct task. Simianer et al. (2012) proposed distributed stochastic learning with feature selection inspired by MTL. The distributed learning approach outperformed several other training methods including MIRA and SGD.

Inspired by these methods, we used MTL to tune multiple SMT systems at the same time, where each system was composed of in-domain and general-domain models. Through a shared feature representation, the commonalities among the SMT systems were better learned by the general models. In addition, domain-specific translation knowledge was also better characterized by the in-domain models.

5 Conclusion and Future Work

In this paper, we propose an MTL-based approach to address multi-domain adaptation for SMT. We first use the cross-entropy based data selection method to obtain in-domain bilingual data. After that, in-domain TMs and LMs are trained for each domain-specific SMT system. In addition, the general-domain TM and LM are also trained and shared across different systems. Finally, MTL is leveraged to tune multiple systems jointly. Experimental results have shown that our approach is quite promising for the multi-domain adaptation problem, and it brings significant improvement over both the non-adapted baselines and the conventional domain adaptation methods with mixture models.

We assume the domain information for testing

data is known beforehand in this study. However, this is not always the case for real world SMT systems. Therefore, to apply our approach in real applications, the domain information needs to be identified automatically. In the future, we will pre-define more popular domains and develop automatic domain classifiers. For those domains that are identified with high confidence, we use the domain-specific system to translate the texts. For other texts, we use the general system to translate them. Furthermore, since our approach is a general training method, we may also combine this approach with other domain adaptation methods to get more performance improvement.

Acknowledgments

We are especially grateful to Nan Yang, Yajuan Duan, Hong Sun and Danran Chen for the helpful discussions. We also thank the anonymous reviewers for their insightful comments.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research*, 6:1817–1853.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, Sydney, Australia, July. Association for Computational Linguistics.

Olivier Chapelle, Pannagadatta Shivaswamy, Srinivas Vadrevu, Kilian Weinberger, Ya Zhang, and Belle Tseng. 2011. Boosted multi-task learning. *Machine learning*, 85(1-2):149–173.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Lei Cui, Dongdong Zhang, Shujie Liu, Mu Li, and Ming Zhou. 2013. Bilingual data cleaning for smt using graph-based random walk. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 340–345, Sofia, Bulgaria, August. Association for Computational Linguistics.

Mark Dredze and Koby Crammer. 2008. Online methods for multi-domain learning and adaptation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 689–697, Honolulu, Hawaii, October. Association for Computational Linguistics.

Kevin Duh, Katsuhito Sudoh, Hajime Tsukada, Hideki Isozaki, and Masaaki Nagata. 2010. N-best reranking by multitask learning. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 375–383, Uppsala, Sweden, July. Association for Computational Linguistics.

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

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Prague, Czech Republic, June. Association for Computational Linguistics.

George Foster, Roland Kuhn, and Howard Johnson. 2006. Phrasetable smoothing for statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 53–61, Sydney, Australia, July. Association for Computational Linguistics.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459, Cambridge, MA, October. Association for Computational Linguistics.

Long Jiang, Shiquan Yang, Ming Zhou, Xiaohua Liu, and Qingsheng Zhu. 2009. Mining bilingual data from the web with adaptively learnt patterns. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 870–878, Suntec, Singapore, August. Association for Computational Linguistics.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 224–227, Prague, Czech Republic, June. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 761–768, Sydney, Australia, July. Association for Computational Linguistics.

Yajuan Lü, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 343–350, Prague, Czech Republic, June. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden, July. Association for Computational Linguistics.

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

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July. Association for Computational Linguistics.

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

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 (Volume 1: Long Papers)*, pages 940–949, Jeju Island, Korea, July. Association for Computational Linguistics.

Rico Sennrich. 2012. 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*, pages 539–549, Avignon, France, April. Association for Computational Linguistics.

Libin Shen and Aravind K Joshi. 2005. Ranking and reranking with perceptron. *Machine Learning*, 60(1-3):73–96.

Patrick Simianer, Stefan Riezler, and Chris Dyer. 2012. Joint feature selection in distributed stochastic learning for large-scale discriminative training in smt. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–21, Jeju Island, Korea, July. Association for Computational Linguistics.

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