Multi-domain Adaptation for Statistical Machine Translation Based on Feature Augmentation

Kenji Imamura kenji.imamura@nict.go.jp
Eiichiro Sumita eiichiro.sumita@nict.go.jp
National Institute of Information and Communications Technology,
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan

Abstract

Domain adaptation is a major challenge when applying machine translation to practical tasks. In this paper, we present domain adaptation methods for machine translation that assume multiple domains. The proposed methods combine two model types: a corpus-concatenated model covering multiple domains and single-domain models that are accurate but sparse in specific domains. We combine the advantages of both models using feature augmentation for domain adaptation in machine learning.

Our experimental results show that the BLEU scores of the proposed method clearly surpass those of single-domain models for low-resource domains. For high-resource domains, the scores of the proposed method were superior to those of both single-domain and corpus-concatenated models. Even in domains having a million bilingual sentences, the translation quality was at least preserved and even improved in some domains. These results demonstrate that state-of-the-art domain adaptation can be realized with appropriate settings, even when using standard log-linear models.

1 Introduction

Machine translation is used for translating a variety of text types, including speech. However, it remains challenging to appropriately translate texts across all domains and only a limited number of domains have been targeted.

The most promising approach to improve translation quality is to train the translator on massive bilingual corpora. However, collecting such corpora is challenging and expensive in several domains. Domain adaptation, which improves target domain quality by using data from another domain, has been proposed as a solution (Foster and Kuhn, 2007; Foster et al., 2010; Axelrod et al., 2011; Bisazza et al., 2011; Sennrich, 2012; Sennrich et al., 2013). This technique is important when applying machine translation to practical tasks.

This paper presents methods of domain adaptation for statistical machine translation (SMT) that assume multiple domains. The proposed methods combine multiple models using log-linear interpolation. These are simple yet effective approaches to take advantage of multiple domains based on feature augmentation (Daumé, 2007), a domain adaptation technique used in machine learning. We propose the following two methods.

1. Simultaneous optimization of multiple domains: this method uses an optimizer extended to multiple domains to optimize an augmented feature space.
2. Optimization of one domain at a time: this method restricts the feature space and regards

this space as that used in the standard log-linear model. This can be realized via a slight modification of existing translation systems.

Both methods use a corpus-concatenated model, which covers multiple domains and contains few unknown words, and single-domain models, which are accurate in their specific domains. In addition, we tune the hyper-parameter of the multiple-model combination. With appropriate settings, state-of-the-art domain adaptation can be realized even when using standard log-linear models.

In this study, we use phrase-based statistical machine translation (PBSMT) (Koehn et al., 2003, 2007) with preordering. The remainder of this paper is organized as follows. Section 2 briefly reviews domain adaptation in machine translation. Section 3 explains our proposed methods in detail. Section 4 discusses the characteristics of our methods through experiments, and Section 5 concludes the paper.

2 Domain Adaptation for Statistical Machine Translation

Domain adaptation is applied when the target domain (in-domain) data are insufficient but data from another domain (out-domain) are available in sufficient quantities. Domain adaptation in machine translation aims to improve the translation quality of in-domain texts using both in-domain and out-domain data.

There are two types of domains: those that are predefined, such as "News" and "Web," and those that are artificially created via automatic clustering. Even when using automatic clustering, the translation quality can be improved in some cases (Finch and Sumita, 2008; Sennrich et al., 2013). However, in this study, we have used predefined domains.

Corpus Concatenation The simplest approach to achieving domain adaptation for SMT is training the model using a concatenated corpus of in- and out-domain data. We refer to this method as corpus concatenation. The trained model is optimized using development (held-out) data of the in-domain.

In machine learning, a model trained on a concatenated corpus has features that are intermediate between the in- and the out-domains. Therefore, model accuracy is also generally intermediate between models trained individually on the in-domain or the out-domain data (i.e., single-domain models).

In contrast, for machine translation, translation quality achieved with corpus concatenation may be superior to that achieved with a single-domain model because the vocabulary coverage increases. The improvement represents a trade-off between reduction in the number of unknown words and greater inaccuracy of model parameters.

Linear/Log-linear Interpolation Statistical machine translation computes translation likelihood using linear or log-linear interpolation of feature values obtained from submodels such as phrase tables, language models, and lexicalized reordering models. The overall likelihood is computed by the following equation:

$$\log P(e|f) \quad \propto \quad \mathbf{w} \cdot \mathbf{h}(e, f) \tag{1}$$

where $\mathbf{h}(e, f)$ is a feature vector and $\mathbf{w}$ is a weight vector of the feature functions.

Then, a domain-specific translation is generated by changing the weight vector $\mathbf{w}$ of each domain. For example, Foster and Kuhn (2007) trained single-domain PBSMT models and translated them while changing the weight vectors of the linear and log-linear interpolations. Although they used perplexities as objective functions to estimate the weights, optimization algorithms, such as minimum error rate training (MERT) (Och, 2003), have been used recently to estimate weight vectors (Foster et al., 2010).

Feature augmentation (Daumé, 2007) is a domain adaptation method used in machine learning that simultaneously optimizes the weight vector of each domain (cf., Section 3.1). Clark et al. (2012) applied it to machine translation as a type of log-linear interpolation; however, they only adapted the weight vectors of a model.

Model Adaptation There are basically two approaches to achieve domain adaptation by changing the feature vector $\mathbf{h}(e, f)$. The first is model adaptation, which modifies trained submodels, and the second is corpus filtering, which trains models using adapted corpora. The fill-up method (Bisazza et al., 2011), translation model combination (Sennrich, 2012), and instance weighting (Foster et al., 2010; Matsoukas et al., 2009) are well-known model adaptation methods.

The fill-up method changes feature values. If a phrase is contained in an in-domain phrase table, the feature values in that table are used. Otherwise, the feature values in the out-domain phrase table are used.

Translation model combination generates a new phrase table by combining two translation probabilities of in- and out- domains. The weights of the combination are determined using each feature function to minimize the perplexity on a development set.

Instance weighting modifies each model parameter in the phrase table to discriminate between the in- and the out-domains by additional learning.

These methods reduce the number of unknown words because the candidates for phrase translation are also altered when the phrase tables are modified. However, submodels other than the phrase table must be adapted using other methods.

Corpus Filtering The other approach to changing the feature vector $\mathbf{h}(e, f)$ is to train the models using the adapted corpora. Although corpus concatenation is one such approach, it uses all sentences in the out-domain corpora. Training data should be selected for better adaptation. Axelrod et al. (2011) selected training sentences similar to those in the in-domain from the out-domain corpora on the basis of cross-entropy difference (i.e., modified Moore-Lewis filtering). Then, they trained the models using the in-domain corpus with additional sentences.

Corpus filtering adapts not only phrase tables but also all submodels used in the translator. However, the ideal number of additional sentences cannot be estimated in advance.

Another Approach Another approach that does not require changing the likelihoods is connecting two translators in series. A translation result generated by the out-domain translator is re-translated by the in-domain translator (Jeblee et al., 2014). This method treats the generation of domain-specific translation as error correction.

3 Multi-domain Adaptation

3.1 Feature Augmentation

Feature augmentation is used to adapt feature weights to domains in machine learning. The feature space is segmented into the following subspaces: common, out-domain (source domain), and in-domain (target domain). In-domain features are copied to the in-domain and common spaces, and out-domain features are copied to the out-domain and common spaces. The adapted weight vector is obtained by optimizing the entire space. The in- and out-domain features deployed in the common space complement each other to improve likelihood accuracy.

Although feature augmentation is mainly used to adapt out-domain models to the in-domain, it can be easily extended to D domains because it treats the in- and the out-domains equivalently. In this case, the feature space is segmented into $D + 1$ subspaces: common, domain 1 , ..., and domain D (Figure 1), which is expressed as follows:

$$\mathbf{h}(f, e) \quad = \quad \langle \mathbf{h}_c, \mathbf{h}_1, \ldots, \mathbf{h}_i, \ldots, \mathbf{h}_D \rangle \tag{2}$$

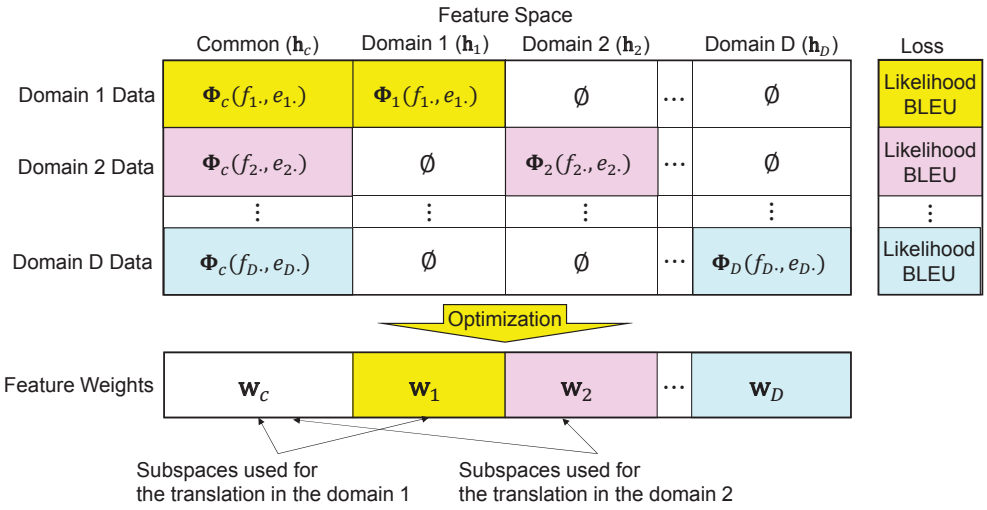

Figure 1: Feature Augmentation Incorporating Corpus-Concatenated Model and Single-Domain Models

where $\mathbf{h}_c$ and $\mathbf{h}_i$ denote the feature vectors of the common and the domain-specific spaces, respectively. All features are deployed to the common space, but only features that match the domain are copied to the domain space.

$$\mathbf{h}_c = \mathbf{\Phi}(f, e) \tag{3}$$

$$\mathbf{h}_i = \begin{cases} \mathbf{\Phi}(f, e) & \text{if domain}(f) = i \\ \emptyset & \text{otherwise} \end{cases} \tag{4}$$

where $\mathbf{\Phi}(f, e)$ denotes the subvector that stores the model scores and so on. It is equal to $\mathbf{h}(f, e)$ if no feature augmentation is applied. We obtain the weight vector by optimizing this feature matrix.

We use the default features of the Moses toolkit (Koehn et al., 2007) (15 dimensions) in the experiments reported in Section 4. The number of dimensions in the augmented feature space is 15 in the common space and 14 in each of the domain spaces[1].

Clark et al. (2012) applied feature augmentation to machine translation from Arabic to English (with News and Web domains) and Czech to English (six domains, e.g., Fiction). Only a corpus-concatenated model was used to obtain features so that feature functions were not changed to reflect the different domains.

3.2 Core of Proposed Methods

3.2.1 Corpus-Concatenated Model and Single-domain Models

In machine translation, compared to feature weights, feature functions have a greater effect on translation quality. Therefore, it is natural to change the submodels depending on the space. Similar to the feature deployment, we assign the corpus-concatenated model, which is constructed from all domain data, to the common space and the single-domain models, which are constructed from one domain data, to the domain-specific spaces. Our approach is as follows.

[1] `UnkPenalty`, which indicates the number of unknown words, is only deployed to the common space because it is not tunable.

- For all submodels, the corpus-concatenated model and single-domain models are constructed in advance.
- In feature augmentation, the scores obtained from the corpus-concatenated model are deployed to the common space as the feature function values, while those from the single-domain models are deployed to the domain spaces (Figure 1). Equations 3 and 4 are then rewritten as follows:

$$\mathbf{h}_c = \mathbf{\Phi}_c(f, e) \tag{5}$$

$$\mathbf{h}_i = \begin{cases} \mathbf{\Phi}_i(f, e) & \text{if domain}(f) = i \\ \emptyset & \text{otherwise} \end{cases} \tag{6}$$

  where $\mathbf{\Phi}_c(f, e)$ and $\mathbf{\Phi}_i(f, e)$ denote feature vectors obtained from the corpus-concatenated model and single-domain model i, respectively.
- While decoding, phrase pairs are first retrieved from both the corpus-concatenated and single-domain phrase tables. The likelihood of each translation hypothesis is computed using only the common space and domain space of the input sentence.

Use of the corpus-concatenated phrase table reduces the number of unknown words because phrase pairs appearing in other domains can be used to generate hypotheses. In addition, precise values of the feature functions can be obtained if the hypotheses exist in the single-domain models. All submodels used in the translator can be adapted without considering their types (i.e., phrase tables, reordering models, and language models) because adaptation is achieved by optimizing the augmented feature space. Therefore, this method can be easily applied to other translation methods, such as tree-to-tree translation. Moreover, unlike corpus filtering, this method does not need to consider the optimal number of additional data entries.

Note that, in machine translation, language models are sometimes constructed from very large monolingual corpora. Such models are regarded as corpus-concatenated models that cover broad domains. In this case, i.e., when we add models (feature functions) acquired from external knowledge, they are located in the common space, which increases the number of dimensions.

3.2.2 Empty Value

In our method, several phrases appear only in one of the phrase tables of the corpus-concatenated and single-domain models. The feature functions are expected to return appropriate values for these phrases. We refer to these as empty values. Even though an empty value is a type of unknown probability and should be computed from the probability distribution of the phrases, we treat it as a hyper-parameter. In other words, empty values are set experimentally to maximize the BLEU score of a development corpus[2].

3.3 Optimization

3.3.1 Joint Optimization

One merit of feature augmentation in machine learning is that conventional algorithms can be used for optimization because feature augmentation operates only in the feature space.

Machine translation uses optimization algorithms such as MERT (Och, 2003), pairwise ranking optimization (PRO) (Hopkins and May, 2011), and k-best batch MIRA (KBMIRA) (Cherry and Foster, 2012). We employ KBMIRA in this paper because it is appropriate for high-dimensional optimization[3].

[2] Moses assigns -100 as the empty value (Koehn and Schroeder, 2007; Birch et al., 2007). As we describe in Section 4.2, this is extremely small and produces low BLEU scores.

[3] Another reason is that the BLEU score of a baseline system was the highest in our preliminary experiments.

A major difference between general machine learning and optimization in machine translation is in the loss functions. The loss functions of machine learning algorithms use likelihood output by decoders. In contrast, the optimization algorithms employed in machine translation use both likelihood and automatic evaluation scores, such as BLEU (Papineni et al., 2002). Automatic evaluation scores are computed by comparing system outputs with their reference translations over *the entire document*. In fact, MERT and KBMIRA contain BLEU scores of the development set in their loss functions[4]. This means that BLEU scores must be computed for each domain to optimize multiple domains.

To solve this problem, we modify the KBMIRA algorithm. The modifications to Algorithm 1 proposed by Cherry and Foster (2012) are as follows.

1. The variable BG that maintains BLEU statistics (such as the number of n-gram matches) is extended to the D-dimensional array, where D denotes the number of domains.
2. The BLEU score of each translation is computed from BG_i, where i is the domain of the input sentence.
3. After the weights are updated, the BLEU statistics of the best translation are added to BG_i.

These modifications optimize the feature weights of each domain space to the development set of each domain.

3.3.2 Independent Optimization

Joint optimization is sometimes redundant because it optimizes all domains even when only one domain is to be adapted. To solve this problem, we restrict and optimize the feature space only to subspaces related to the domain that we want to adapt. We refer to this as independent optimization.

Independent optimization restricts the feature space to the common and the domain i spaces, and only the tuning data of domain i are used for optimization. Namely, Equation 2 is replaced with Equation 7.

$$\mathbf{h}(f,e) = \langle \mathbf{h}_c, \mathbf{h}_i \rangle \quad (7)$$

$$\mathbf{h}_c = \mathbf{\Phi}_c(f,e) \quad (8)$$

$$\mathbf{h}_i = \mathbf{\Phi}_i(f,e) \quad (9)$$

This is the same as a standard log-linear model, which can be optimized without joint optimization by using existing optimizers. Furthermore, we can use multiple decoders with slight modifications because they only have to allow for 1) multiple models to be jointly used and 2) setting the empty value.

The common space might not be strictly optimized compared to joint optimization. However, in machine translation, feature functions affect translation quality to a greater extent than feature weights. Therefore, we assume that, in practice, partially rough optimization is not problematic. Note that the following two points are common to joint optimization.

1. Features of the common space are obtained from the corpus-concatenated model.
2. An empty value is set appropriately.

4 Experiments

4.1 Experimental Settings

Domains/Corpora Four domains were used in this paper. The language pairs were English-to-Japanese (En-Ja) and Japanese-to-English (Ja-En). The size of the corpora of each domain

[4]PRO, which was used by Clark et al. (2012) for feature augmentation, employs BLEU scores approximated by sentences.

Domain	# of Sentences			# of Training Words	
	Training	Dev.	Test	En	Ja
MED	222,945	1,000	1,000	3.1M	3.3M
LIVING	986,946	1,800	1,800	14.3M	16.5M
NTCIR	1,387,713	2,000	2,000	48.7M	52.3M
ASPEC	1,000,000	1,790	1,784	25.9M	28.7M

Table 1: Corpus Statistics

is listed in Table 1. The MED corpus is relatively small, whereas the other corpora comprise nearly a million sentences each. Sentences with fewer than 80 words were used for training.

- **MED**: A pseudo-dialogue corpus between patients and medical doctors (or staff) in hospitals. This was developed in-house.
- **LIVING**: A pseudo-dialogue corpus wherein visitors (or residents) from foreign countries talk to local people. This was also developed in-house.
- **NTCIR**: A patent corpus. The training and development sets were provided by the international conference NTCIR-8, and the test set was provided by NTCIR-9[5].
- **ASPEC**: An Asian scientific paper excerpt corpus (Nakazawa et al., 2016)[6]. We used a million sentences of high-confidence translation from ASPEC-JE.

Translation System Each source sentence was preordered using an in-house preordering system (Section 4.5 of Goto et al. (2015)) trained for general-purpose. The same preordering system was applied to all domains. In addition, all Japanese sentences, including the test sets, were segmented into words in advance using the MeCab morphological analyzer (Kudo et al., 2004).

The phrase tables and lexicalized reordering models were trained using the default settings in the Moses toolkit. The 5-gram language models were learned from the target side of the training sentences using KenLM (Heafield et al., 2013). Multi-domain KBMIRA, described in Section 3.3.1, was used for optimization.

A clone of the Moses decoder was used for decoding. The settings were the same as the default values in Moses, i.e., `phrase_table_limit` $= 20$, `distortion_limit` $= 6$, and the beam width was 200. When the decoder selected phrase pair candidates, 1) phrase pairs were first obtained from all phrase tables, 2) a likelihood of each phrase was computed in accordance with the augmented feature space, and 3) the highest 20 pairs were selected.

Empty Value The empty value described in Section 3.2.1 was set empirically. From integer values of -3 to -20, we selected the empty value that achieved the highest BLEU score in the set in which all development sets were concatenated. The resulting empty values were -7 for En-Ja translation and -6 for Ja-En translation. If we treat these values as probabilities, they are $\exp(-7) \approx 0.0009$ and $\exp(-6) \approx 0.0025$, respectively.

Evaluation Metrics We used word BLEU, the translation edit rate (TER) (Snover et al., 2006), Meteor (English only) (Denkowski and Lavie, 2014), and the rank-based intuitive bilingual evaluation score (RIBES; Japanese only) (Isozaki et al., 2010) as the evaluation metrics.

[5] `http://research.nii.ac.jp/ntcir/index-ja.html`
[6] `http://lotus.kuee.kyoto-u.ac.jp/ASPEC/`

The MultEval tool (Clark et al., 2011)[7] was used for statistical testing[8] with the significance level set to $p < 0.05$. The mean scores of five runs were used to reduce instability in optimization. Although we used multiple metrics, for simplicity we will describe the results using BLEU.

Comparison Methods We compared various methods using the single-domain model as the baseline. We used the following conventional methods, which have been described in Section 2.

- **Corpus Concatenation**: A corpus-concatenated model was constructed using all domain data. Optimization and testing were performed using the development and test sets of each domain.
- **Feature Augmentation (Clark)**: Feature augmentation was applied to the adaptation; however, all features of the common and domain-specific spaces were obtained from the corpus-concatenated model. This is the same setting as that used by Clark et al. (2012), except that we used multi-domain KBMIRA for optimization.
- **Fill-Up**: The fill-up method (Bisazza et al., 2011) was used for domain adaptation.
- **TM Combination**: Translation model combination (Sennrich, 2012) was applied for adaptation. We used the `tmcombine` program in the Moses toolkit.
- **Corpus Filtering**: The modified Moore-Lewis filtering scheme proposed by Axelrod et al. (2011) was applied. All corpora, except for the target domain, were used as out-domain corpora. The number of additional sentences was determined to maximize the BLEU score of the development set.

As variations of the proposed methods, we tested the following settings.

- **Proposed (Joint)**: The best setting of the proposed method using joint optimization (cf., Section 3.3.1).
- **Proposed (Independent)**: The best setting of the proposed method using independent optimization (cf., Section 3.3.2).
- **Proposed (empty = -100)**: The empty value was set to -100, which is equivalent to the Moses value. We used independent optimization in this setting; however, the same tendency was observed when using joint optimization.
- **Proposed (Out-Domain)**: The common space model was changed from the corpus-concatenated model to the out-domain model learned from the other three domain corpora. In addition, independent optimization was used.

4.2 Translation Quality

Tables 2 and 3 show the BLEU scores of the abovementioned methods for En-Ja and Ja-En translations, respectively. Bold values represent the highest scores. The symbols (+) and (-) denote whether the score was significantly improved or degraded compared to that of the single-domain model ($p < 0.05$).

Because the domains used in the experiments are not closely related, many BLEU scores were degraded by conventional adaptations. For instance, the corpus concatenation approach

[7] `https://github.com/jhclark/multeval`.

[8] We incorporated the RIBES script (`http://www.kecl.ntt.co.jp/icl/lirg/ribes/index.html`) into the MultEval tool.

Method	Domain					
	MED			LIVING		
	BLEU	TER	RIBES	BLEU	TER	RIBES
Single Domain Model	23.23	62.46	78.34	**24.56**	61.33	78.78
Corpus Concatenation	22.65(-)	64.08(-)	77.53(-)	22.99(-)	63.41(-)	77.52(-)
Feature Augmentation (Clark)	22.49(-)	63.78(-)	77.70(-)	22.97(-)	63.40(-)	77.41(-)
Fill-Up	22.42(-)	63.63(-)	77.46(-)	23.38(-)	63.16(-)	77.80(-)
TM Combination	23.81(+)	61.94(+)	78.37	24.05(-)	62.05(-)	78.62(-)
Corpus Filtering	**24.02**(+)	**61.61**(+)	78.43	24.50	62.08(-)	78.51(-)
Proposed (Joint)	23.69(+)	61.79(+)	**78.67**(+)	24.43	61.51	78.72
Proposed (Independent)	23.75(+)	61.78(+)	78.56	24.43	**61.09**(+)	**78.80**
Proposed (empty=-100)	23.66(+)	62.37	78.12	23.91(-)	61.84(-)	78.39(-)
Proposed (Out-Domain)	23.79(+)	62.19	78.46	24.29(-)	61.77(-)	78.75

Method	Domain					
	NTCIR			ASPEC		
	BLEU	TER	RIBES	BLEU	TER	RIBES
Single Domain Model	38.62	47.85	80.16	32.69	54.12	78.16
Corpus Concatenation	38.09(-)	48.54(-)	79.88(-)	30.59(-)	55.95(-)	77.22(-)
Feature Augmentation (Clark)	38.09(-)	48.54(-)	79.90(-)	30.65(-)	55.91(-)	77.28(-)
Fill-Up	38.37(-)	48.12(-)	80.03(-)	31.50(-)	55.06(-)	77.62(-)
TM Combination	38.32(-)	48.11(-)	80.09(-)	31.97(-)	54.77(-)	77.84(-)
Corpus Filtering	38.77(+)	47.82	80.17	32.57(-)	54.14	78.14
Proposed (Joint)	38.72(+)	47.77	80.22(+)	32.69	54.20	78.10
Proposed (Independent)	**38.83**(+)	**47.64**(+)	80.22	**32.76**	54.13	78.15
Proposed (empty=-100)	38.56	47.90	80.10(-)	32.62	54.17	78.13
Proposed (Out-Domain)	38.65	47.78	**80.28**(+)	32.72	**54.02**	**78.19**

Table 2: Automatic Evaluation Scores of Various Methods (En-Ja Translation)

and feature augmentation (Clark) were degraded for most domains. This is because the corpus-concatenated models are an average of all the domain models and the precision of the model parameters was degraded for each specific domain.

The BLEU scores of Fill-Up were superior to those of corpus concatenation in several cases but inferior to those of the single-domain models. The TM Combination scores were both improved and degraded depending on the domain, and we could not confirm the effects. In corpus filtering, the scores were improved or were at the same level compared to the single-domain model, except for En-Ja translation of the ASPEC domain. Although only 100k sentences were added in the ASPEC En-Ja domain, they affected translation quality. This shows that corpus filtering is effective; however, determining the ideal number of additional sentences is difficult.

Conversely, all BLEU scores of the proposed methods (joint and independent) were significantly improved or were at the same level as those of the single-domain models. The independent optimization scores tended to be better than those of joint optimization. When the empty value was set to -100, the weight vector could not be optimized because the BLEU score did not converge in some cases (N/A in Table 3). Focusing on the proposed method (out-domain), the scores were degraded from those of the proposed methods (joint and independent) in most cases. This indicates that the corpus-concatenated model is better than the out-domain model for the common space.

Method	Domain					
	MED			LIVING		
	BLEU	TER	Meteor	BLEU	TER	Meteor
Single Domain Model	17.38	71.14	25.49	19.71	67.08	27.58
Corpus Concatenation	17.07	70.72	25.26(-)	18.80(-)	68.50(-)	26.74(-)
Feature Augmentation (Clark)	16.75(-)	71.39	25.13(-)	18.95(-)	68.31(-)	26.75(-)
Fill-Up	16.56(-)	71.98(-)	25.11(-)	19.06(-)	68.45(-)	26.65(-)
TM Combination	17.55	71.17	25.65(+)	19.99(+)	67.22	27.31(-)
Corpus Filtering	18.14(+)	70.14(+)	**26.16**(+)	19.76	66.81(+)	27.48(-)
Proposed (Joint)	18.14(+)	**69.29**(+)	25.90(+)	20.16(+)	**66.61**(+)	27.45(-)
Proposed (Independent)	**18.43**(+)	69.85(+)	26.00(+)	**20.17**(+)	66.94	27.53
Proposed (empty=-100)	17.13	71.22	25.78(+)	19.86	67.24	27.67(+)
Proposed (Out-Domain)	17.32	70.93	25.34	19.66	67.31(-)	27.02(-)

Method	Domain					
	NTCIR			ASPEC		
	BLEU	TER	Meteor	BLEU	TER	Meteor
Single Domain Model	33.63	52.67	35.68	21.75	64.95	31.01
Corpus Concatenation	33.21(-)	52.94(-)	35.33(-)	20.41(-)	66.00(-)	30.36(-)
Feature Augmentation (Clark)	33.24(-)	53.00(-)	35.38(-)	20.39(-)	66.18(-)	30.33(-)
Fill-Up	33.14(-)	53.06(-)	35.48(-)	20.98(-)	65.41(-)	30.58(-)
TM Combination	33.32(-)	52.78(-)	35.54(-)	21.16(-)	65.17(-)	30.77(-)
Corpus Filtering	**33.73**	52.45(+)	**35.71**	21.72	64.71(+)	31.03(+)
Proposed (Joint)	33.68	52.42(+)	35.70	21.75	64.79(+)	**31.20**(+)
Proposed (Independent)	33.70	**52.33**(+)	35.67	**21.81**	64.76(+)	31.19(+)
Proposed (empty=-100)	N/A	N/A	N/A	N/A	N/A	N/A
Proposed (Out-Domain)	33.52(-)	52.70	35.62	21.73	**64.72**(+)	31.06

Table 3: Automatic Evaluation Scores of Various Methods (Ja-En Translation)

In summary, compared to the other methods, the proposed methods achieved the best translation quality. Therefore, state-of-the-art domain adaptation can be realized with the appropriate settings even when using a standard log-linear model such as independent optimization.

4.3 Effects as Single-Domain Adaptation

A typical situation wherein domain adaptation is needed would be one in which sufficient training data cannot be collected and new domain data must be translated. In this section, we investigate translation quality when changing training corpus size, focusing only on the En-Ja translation in the MED domain. Note that the other domains are not changed.

Table 4 compares the results obtained using the single-domain model, corpus concatenation, and the proposed method (independent). The symbols (+) and (-) denote scores that were significantly improved or degraded compared to those of the single-domain model. The symbol (†) denotes a score that was significantly improved compared to that of corpus concatenation.

When using one thousand training sentences (1k), the score achieved by the proposed method was significantly higher than that achieved by the single-domain model and equal to that achieved by corpus concatenation. When the size of the training corpus was increased, BLEU scores increased for all methods. However, the improvement in the corpus concatenation score was less than that in the single-domain model score. The single-domain model surpassed corpus

# of Sentences	Single Domain Model	Corpus Concatenation	Proposed (Independent)
1k	6.42	17.51 (+)	**17.59** (+)
3k	8.99	17.52 (+)	**17.95** (+)(†)
10k	12.54	18.19 (+)	**19.02** (+)(†)
30k	16.49	19.18 (+)	**20.28** (+)(†)
100k	20.63	20.92	**22.53** (+)(†)
223k (All)	23.23	22.65 (-)	**23.75** (+)(†)

Table 4: BLEU Scores for Different Training Sizes (MED Domain, En-Ja Translation)

concatenation when more than 100 thousand (100k) training sentences were used. BLEU scores for the proposed method exceeded those of the single-domain model and corpus concatenation when more than three thousand (3k) training sentences were used. These results demonstrate that the proposed method successfully integrated the merits of the single-domain and corpus-concatenated models.

Here, we refer back to Tables 2 and 3. From these tables, it can be seen that the BLEU scores of the proposed method (joint and independent) in the MED domain improved for both En-Ja and Ja-En. We assume that it was possible to improve the translation quality because the MED corpus was relatively small. In contrast, the translation quality was not necessarily improved in other domains because these were trained using approximately a million sentences. However, it should be noted that the translation quality was not degraded and, in some cases, it was improved, even when the proposed method was applied to domains of very large corpora. The proposed method is, therefore, robust to corpus size.

4.4 Unknown Words

The proposed methods take the advantage of the corpus-concatenated and single-domain models. Finally, we analyze the characteristics of the proposed methods from the viewpoint of unknown words.

In this paper, we distinguish between source unknown words (source UNK) and target unknown words (target UNK) by referring to the categories suggested by Irvine et al. (2013)[9]. Source unknown words occur when the source words (or phrases) do not exist in the phrase tables. Target unknown words occur when a reference translation cannot be generated because the target words (or phrases) are not in the phrase tables even though the source words exist. This can be determined by forced decoding (Yu et al., 2013).

Table 5 shows the sentence rates that include unknown words in the En-Ja translation. Although there were differences depending on the domains, the rates of source and target UNK both decreased in corpus concatenation compared to those in the single-domain models. For example, in the MED domain, the rate of source UNK decreased from 9.1% to 1.0%, and that of target UNK decreased from 38.5% to 18.1%. This result proved that words in the other domains were used for translation. However, this result also proved that reduction of unknown words does not directly contribute to translation quality because the quality of the single-domain models was better than that of corpus concatenation. Hence, optimization is important.

The proposed methods further reduced unknown words, except for the NTCIR domain. The proposed methods use phrase tables of the corpus-concatenated and single-domain models. Even though these phrase tables were trained from corpora that partially overlap, the acquired

[9]The source unknown and target unknown words correspond to the SEEN and SENSE errors, respectively, used by Irvine et al. (2013).

UNK Type	Method	Domain MED	LIVING	NTCIR	ASPEC
Source UNK	Single-Domain Model	9.1%	4.1%	7.9%	22.5%
	Corpus Concatenation	1.0%	2.8%	6.5%	21.6%
	Proposed (Independent)	0.9%	2.4%	6.4%	21.2%
Target UNK	Single-Domain Model	38.5%	26.4%	21.3%	26.5%
	Corpus Concatenation	18.1%	20.1%	17.1%	21.6%
	Proposed (Independent)	16.1%	17.0%	17.2%	20.0%

Table 5: Sentence Rates that Include Unknown Words (En-Ja Translation)

phrases were slightly different. Hence, the coverage of the proposed methods increased.

5 Conclusions

In this paper, we presented multi-domain adaptation methods for statistical machine translation. The proposed methods combined the corpus-concatenated model, which has high coverage and few unknown words, with single-domain models, which ensure precision of the feature functions. To take advantage of the benefits of both models, we applied feature augmentation to machine translation. In addition, we tuned the empty value to balance the two models.

In both joint and independent optimization, the translation quality was improved or at the same level compared with the single-domain models in our experiments. The resulting BLEU scores of the proposed method clearly surpassed those of the single-domain models in low-resource domains. Even in domains with a million training sentences, the translation quality remained at least equal, and in some domains, it was improved. These results show that state-of-the-art domain adaptation can be realized with appropriate settings, even when using standard log-linear models.

The proposed methods can be easily applied to other translation strategies, such as tree-to-tree translation. We plan to apply our methods to tree-to-tree translation and evaluate the effects.

Acknowledgments

This work was supported by "Promotion of Global Communications Plan — Research and Development and Social Demonstration of Multilingual Speech Translation Technology," a program of Ministry of Internal Affairs and Communications, Japan.

References

Axelrod, A., He, X., and Gao, J. (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.

Birch, A., Osborne, M., and Koehn, P. (2007). CCG supertags in factored statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 9–16, Prague, Czech Republic.

Bisazza, A., Ruiz, N., and Federico, M. (2011). Fill-up versus interpolation methods for phrase-based SMT adaptation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, California, USA.

Cherry, C. and Foster, G. (2012). Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada.

Clark, J. H., Dyer, C., Lavie, A., and Smith, N. A. (2011). Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA.

Clark, J. H., Lavie, A., and Dyer, C. (2012). One system, many domains: Open-domain statistical machine translation via feature augmentation. In *Proceedings of the 10th biennial conference of the Association for Machine Translation in the Americas (AMTA 2012)*, San Diago, California, USA.

Daumé, III, H. (2007). Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic.

Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the 9th Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA.

Finch, A. and Sumita, E. (2008). Dynamic model interpolation for statistical machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 208–215, Columbus, Ohio, USA.

Foster, G., Goutte, C., and Kuhn, R. (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, Massachusetts, USA.

Foster, G. and Kuhn, R. (2007). Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Prague, Czech Republic.

Goto, I., Utiyama, M., Sumita, E., and Kurohashi, S. (2015). Preordering using a target-language parser via cross-language syntactic projection for statistical machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 14(3):13:1–13:23.

Heafield, K., Pouzyrevsky, I., Clark, J. H., and Koehn, P. (2013). Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696, Sofia, Bulgaria.

Hopkins, M. and May, J. (2011). Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK.

Irvine, A., Morgan, J., Carpuat, M., III, H. D., and Munteanu, D. (2013). Measuring machine translation errors in new domains. *Transactions of the Association for Computational Linguistics*, 1:429–440.

Isozaki, H., Hirao, T., Duh, K., Sudoh, K., and Tsukada, H. (2010). Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, Massachusetts, USA.

Jeblee, S., Feely, W., Bouamor, H., Lavie, A., Habash, N., and Oflazer, K. (2014). Domain and dialect adaptation for machine translation into Egyptian Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 196–206, Doha, Qatar.

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

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *HLT-NAACL 2003: Main Proceedings*, pages 127–133, Edmonton, Alberta, Canada.

Koehn, P. and Schroeder, J. (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.

Kudo, T., Yamamoto, K., and Matsumoto, Y. (2004). Applying conditional random fields to Japanese morphological analysis. In Lin, D. and Wu, D., editors, *Proceedings of EMNLP 2004*, pages 230–237, Barcelona, Spain.

Matsoukas, S., Rosti, A.-V. I., and Zhang, B. (2009). Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 708–717, Singapore.

Nakazawa, T., Yaguchi, M., Uchimoto, K., Utiyama, M., Sumita, E., Kurohashi, S., and Isahara, H. (2016). ASEPC: Asian scientific paper excerpt corpus. In *Proceedings of the Tenth Edition of the Language Resources and Evaluation Conference (LREC-2016)*, Portoroz, Slovenia.

Och, F. J. (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.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, Pennsylvania, USA.

Sennrich, R. (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.

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

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Biennial Conference of the Association for Machine Translation in the Americas (AMTA-2006)*, pages 223–231, Cambridge, Massachusetts, USA.

Yu, H., Huang, L., Mi, H., and Zhao, K. (2013). Max-violation perceptron and forced decoding for scalable MT training. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1112–1123, Seattle, Washington, USA.