Domain Adaptation for MT: A Study with Unknown and Out-of-Domain Tasks

Hoang Cuong
Agolo, Inc.
cuong.hoang@agolo.com

Abstract

Translation quality could degrade non-gracefully outside the desired domain for MT. Meanwhile, translation requests are often unknown and potentially out-of-domain in practice. This paper shows that having an ecosystem with a range of pre-trained domain-specific MT systems can reduce the effect: a translation task can be out of scope of most pre-trained MT systems, but a few others can be capable of handling the task. But how to obtain the best translation from an ecosystem for such translation requests? We contribute two frameworks to address the problem. Experiments show that our frameworks give the performance in the middle between top rank MT systems with reasonably large-scale ecosystems.

1 Introduction

Translation models have been developed under the assumption that we know the domain at test time in advance, and the domain is strictly relevant to our training data. However, we inevitably will come across test data that is sampled from a different distribution to our training data when using the models in the wild. Another critical thing is that the domain of test data is often unknown in practice (e.g. Google Translate and Microsoft Translators receive translation requests from their users without knowing in advance their interests).

We have not had a solution for this well-known problem yet. Machine Translation (MT) has been advanced by new models, including using Neural Machine Translation (NMT) instead of Statistical Machine Translation (SMT). The hope is that a better translation model would improve the translation in all settings/situations. This, however, is not true. Translation quality could degrade nongracefully outside the desired domain for both NMT and SMT. In fact, it has been known that NMT suffers even harder than SMT when the test data is out-of-domain (Koehn and Knowles, 2017; Chu and Wang, 2018). We also improve MT by using domain adaptation methods (i.e. improving translation system from having a small seed in-domain data such as system interpolation, instance weighting and data selection). In practice, this is not a thorough solution because we do not know the domain of user translation requests in advance.

The contribution of this work is to provide a simple, easy-and-fast-to-deploy, translation model-agnostic[1] solution to the challenging problem. Our approach is to construct an "ecosystem" with a range of pre-trained domain-specific MT systems, each specialized in a certain domain (e.g. Speech, Financial, Food, etc.). Our intuition is that having such an ecosystem could reduce the decrease in translation quality for an outside domain. That is, an out-of-domain translation task can be out of scope of most pre-trained MT systems in the ecosystem. However, with the diversity of domains in a reasonably large ecosystem, we hope there is a chance to have certain pre-trained systems in the ecosystem that can be capable of handling the task well. The larger our ecosystem is, the more likely we have more capable pre-trained MT systems to an out-of-domain task.

The next step is to work on an unsupervised

[1] We aim for a solution that works with both NMT, SMT or other translation models.

method that automatically finds the best translations from an ecosystem for every translation request from an unknown and out-of-domain translation task. This is surprisingly difficult. Creating a domain classifier for translation requests provides suboptimal performance, because the target domain is unknown and out-of-domain. System combination could degrade translation quality substantially, as the majority of pre-trained MT systems in the ecosystem are incapable of handling the task. We propose two frameworks to address the problem.

VOTING I involves two separate steps for handling each translation request: First, the request is translated by *all* pre-trained MT systems. Second, the translation output that is most similar to others is returned to the user. An agreement measure is proposed to calculate how similar translation outputs are. The intuition behind VOTING I is that good translations may be similar to the others. That is, because they are good translations, they must be similar to translation references, and therefore it is likely that they are similar to the others as well.

VOTING II selects only a *limited* number of MT systems for decoding. Decoding cost is thus substantially cheaper in VOTING II. The intuition behind VOTING II is that MT systems that are good in a domain tend to agree with each other. However, the expertise parameters of MT systems regarding to an unknown domain are hidden and we thus do not know which MT systems we should select. In VOTING II expertise parameters are initialized randomly and our heuristic learning algorithm consequently updates the parameters during translation. We note that VOTING II works with the assumption that the translation requests would be handled in sequential (not parallel). While this is not true for all cases, it is true when we translate request translations of large documents as one task.

We conduct extensive experiments with *Spanish-English*, *French-English* and *German-English* to support our intuition. Experiments show that VOTING I gives the performance in between the top two systems for medium-scale ecosystem, and in between the top three systems for a large-scale ecosystem. VOTING II performs substantially better than VOTING I and occasionally reaches close to the top Rank 1 MT system for medium-scale ecosystems. Our framework is scalable and has promising applications to large-scale online translation services.[2]

2 Related Work

This paper discusses a complementary problem to domain adaptation: How to handle unknown and out-of-domain translation tasks. Domain adaptation has been an active topic of research for many years. A survey of domain adaptation for MT can be referred to (Chu and Wang, 2018; Cuong and Sima'an, 2017). Within MT, but the domain of the request is typically known in advance. domain adaptation can be regarded as injecting prior knowledge about the target translation task into learning.

Combination of in-domain data with a general-domain system A common is approach is to combine a system trained on the in-domain data with a general-domain system (Koehn and Schroeder, 2007; Farajian et al., 2017; Kobus et al., 2017; Foster et al., 2010; Shah et al., 2010; Bisazza et al., 2011; Sennrich, 2012b; Razmara et al., 2012; Cuong and Sima'an, 2014a; Cuong and Sima'an, 2015; Sennrich et al., 2013; Haddow, 2013; Hildebrand and Vogel, 2008; Joty et al., 2015; Wang et al., 2018; Khayrallah et al., 2017; Chen et al., 2017; Tars and Fishel, 2018) or to combine the in-domain system with a system trained on a selected subset (Axelrod et al., 2011; Duh et al., 2013; Kirchhoff and Bilmes, 2014; Eetemadi et al., 2015; Chen and Huang, 2016; Wang et al., 2018; van der Wees et al., 2017; Cuong and Sima'an, 2014b).

Meta-information Prior knowledge may also lie in meta-information about training data. This could be document-annotated information (Eidelman et al., 2012; Hu et al., 2014; Hasler et al., 2014; Zhang et al., 2014; Su et al., 2015), and domain-annotated sub-corpora (Chiang et al., 2011; Sennrich, 2012b; Chen et al., 2013; Kothur et al., 2018; Michel and Neubig, 2018; Bapna and Firat, 2019).

Other DA Topics Recent work also performs adaptation by exploiting separate in-domain development sets (Sennrich, 2012a; Carpuat et al., 2013; Mansour and Ney, 2014; Clark et al., 2012; Wang et al., 2012). Rewarding domain invariance is also

[2]The code can be downloaded at: github.com/hoangcuong2011/UnsupervisedDomainAdaptation.

another approach to perform unsupervised adaptation (Cuong et al., 2016). Combining several different Machine Translation outputs operating on the same input is also a promising DA approach (Jayaraman and Lavie, 2005; Hildebrand and Vogel, 2008).

Using online methods for adapting MT systems in a scenario where human feedback (e.g. post-edited MT output) is constantly returned has been gaining interest recently (Ortiz-Martínez et al., 2010; Koehn et al., 2014; Denkowski et al., 2014; Bertoldi et al., 2014; Blain et al., 2015; Ortiz-Martínez, 2016; Wuebker et al., 2016; Karimova et al., 2018). Using Bayesian models provides promising results for adapting MT systems (e.g. see (Denkowski et al., 2014; Bertoldi et al., 2014; Blain et al., 2015; Peris and Casacuberta, 2018)). Recently, deploying bandit learning algorithms shows promising results for minimizing the cost of human feedback for improving system performance (e.g. see (Sokolov et al., 2015; Sokolov et al., 2016; Sokolov et al., 2017; Nguyen et al., 2017)).

3 Our Framework

Assume we are given a set of N pre-trained MT systems $\mathbf{m}_1^N = \{m_1,\ m_2, \ldots,\ m_N\}$. At test time, our goal is to handle an *unknown* and out-of-domain translation task: $\mathbf{f}_1^K = \{f_1,\ f_2, \ldots,\ f_K\}$. Note that the requests may be submitted intermittently by the user, which is common in practice (e.g. as in web-based translation services).

3.1 Voting I

Our first proposed framework is VOTING I. It involves two separate steps. First, each translation request f is translated by *all* pre-trained MT systems. Second, the translation output produced by an MT system that is most similar to others is returned to the user. Note that this approach is quite similar to (Macherey and Och, 2007), only that the approach here is made to be symmetrical.

Technically, the agreement between two translation outputs e_m and $e_{m'}$ produced by two different MT systems m and m' is calculated as the *arithmetic mean* between BLEU+1(e, e') and BLEU+1(e',e):

$$a(e_m, e_{m'}) = \frac{\text{BLEU+1}(e_m, e_{m'}) + \text{BLEU+1}(e_{m'}, e_m).}{2}$$

Here, BLEU+1 (Lin and Och, 2004) is a variant of BLEU for sentence-level assessment (Papineni et al., 2002). Given that all N MT systems are used to decode each translation request, the average agreement score between one translation output e_m produced by an MT system m and all the others produced by other MT systems m′ is calculated as:

$$a(e_m) = \sum\nolimits_{m' \neq m} \frac{1}{N-1} a(e_m, e_{m'}). \quad (1)$$

VOTING I simply uses the proposed agreement measure to rank translation outputs. As discussed, our assumption is that good translations (e.g. Book, Wikipedia) is likely to be similar to the others. See Table 1 for a positive example we obtain from our experiments with VOTING I .

3.2 Voting II

MT systems can generate similar translations by chance. We show such an example we obtain from our experiments with VOTING I in Table 2 (on the left). There are also cases of "black sheep": a very good translation may be too different from the others. Table 2 (on the right) shows such an example. VOTING I is not able to handle these issues. Applying VOTING I is expensive regarding the decoding cost.

How to address these issues? In our refined framework – VOTING II, we introduce a set of *expertise parameters* of all MT systems: $\Theta_1^N = \{\theta_{m_1}, \theta_{m_2}, \ldots, \theta_{m_N}\}$. Here, expertise parameter θ_m represents how suitable a system m to a certain domain. VOTING II simply selects only the top M MT systems with the highest expertise parameters, instead of using all N MT systems for decoding each translation request. In our experiments, we set $M = 3$.

VOTING II addresses the shortcomings of VOTING I as follows:

- (1) VOTING II explicitly filters bad MT systems for a certain domain;
- (2) VOTING II ranks translation outputs according to a sum of $a(e_m) + \theta_m$ instead of only $a(e_m)$ as in VOTING I;
- and (3) the decoding cost is substantially reduced (with a ratio of $(N - M)/N$). As discussed, VOTING II works with the assumption that the translation requests would be handled in sequential and not parallel (e.g. we translate request translations of large documents as one task).

Medicine	**Input**: aliments et boissons abilify peut se prendre pendant ou en dehors des repas . **Reference**: taking abilify with food and drink abilify can be taken regardless of meals .	
MT System	**Score**	**Translation Output**
Book	**0.70**	**food and drink abilify can take during or outside meals .**
Speech	0.64	food and drink abilify can take yourself for or outside meals .
IT	0.45	aliments and boissons abilify might take in or out of meal .
Bank	0.58	foods and beverages abilify may take during or outside the repas .
News	0.65	foods and drinks abilify can take during or outside the meal .
Wikipedia	0.69	food and drink abilify can be take during or outside the meal .
Legal	0.52	feedingstuffs and beverages abilify may be taken during or outside the meals .
Europarl	0.65	food and drink abilify can take over or outside meals .
Subtitles	0.58	aliments and drinks abilify can take for or out the food .

Table 1: Positive example with VOTING I: Good translations (e.g. Book, Wikipedia) tend to be similar to the others.

Medicine	**Input**: resume des caracteristiques du produit **Reference**: summary of product characteristics		**Input**: étiquetage et notice **Reference**: labelling and package leaflet	
MT System	**Score**	**Translation Output**	**Score**	**Translation Output**
Book	0.62	resume of product characteristics	**0.30**	**labelling and package leaflet**
Speech	**0.78**	**resume of caracteristiques of the product**	0.53	étiquetage and warning
IT	0.77	resume of caracteristiques the product	0.53	tag and notice
Bank	0.46	**summary of characteristics of product**	**0.74**	**étiquetage and notice**
News	**0.78**	**resume of caracteristiques of the product**	**0.74**	**étiquetage and notice**
Wikipedia	0.74	resume the caracteristiques of the product	**0.74**	**étiquetage and notice**
Legal	0.69	resume of the characteristics of the product	0.36	labelling and document
Europarl	0.70	resume the caracteristiques product	**0.74**	**étiquetage and notice**
Subtitles	0.63	resume some caracteristiques the product	**0.74**	**étiquetage and notice**

Table 2: Two negative examples with VOTING I. On the left: bad translations (e.g. IT, Wikipedia, Speech) are also similar to the others by chance. On the right: a case of "black sheep": a very good translation (Book) is too different from the others.

Of course the expertise parameters of MT systems are hidden. The question is how to learn them? The intuition behind VOTING II is that MT systems that are good in a certain domain are likely to agree with each other.

Two models are proposed in this paper to implement the idea. They are in the same spirit: the expertise parameter of each system m is sampled from a posterior distribution $\pi_m(\theta)$: $\theta_m \sim \pi_m(\theta)$. Our heuristic learning algorithm starts in a naive state, and we do not have any a-priori preference for one system over another. The algorithm consequently updates the parameters of the posterior distribution $\pi_m(\theta)$ based on agreement scores for translation outputs produced by system m. The proposed models use different posterior distributions $\pi(\theta)$ for sampling θ. Our goal of proposing different models is to investigate which one that addresses the problem best.

Figure 1 illustrates the framework.

3.2.1 Voting II Real

Our first model (VOTING II - REAL) uses normal distribution to sample expertise parameters. Let us assume a sample of agreement scores from all translation outputs produced by an MT system m as $\mathcal{A}_m = \{a_1, a_2, \ldots, a_{|\mathcal{A}_m|}\}$. Here, $|\mathcal{A}_m|$ denotes the sample size. Let us denote the sample *mean* and sample *variance* as $\bar{\mu}_m$ and δ_m^2.

In VOTING II - REAL, we assume (by way of the Central Limit Theorem) that the expertise parameter of system m is approximately normal with mean $\bar{\mu}_m$ and variance $\delta_m^2/|\mathcal{A}_m|$:

$$\theta_m \sim \mathcal{N}(\bar{\mu}_m, \delta_m^2/|\mathcal{A}_m|). \quad (2)$$

We propose a heuristic algorithm for learning expertise parameters in VOTING II - REAL:

- Given each translation request f, expertise parameter is first drawn from the posterior distribution for each MT system.

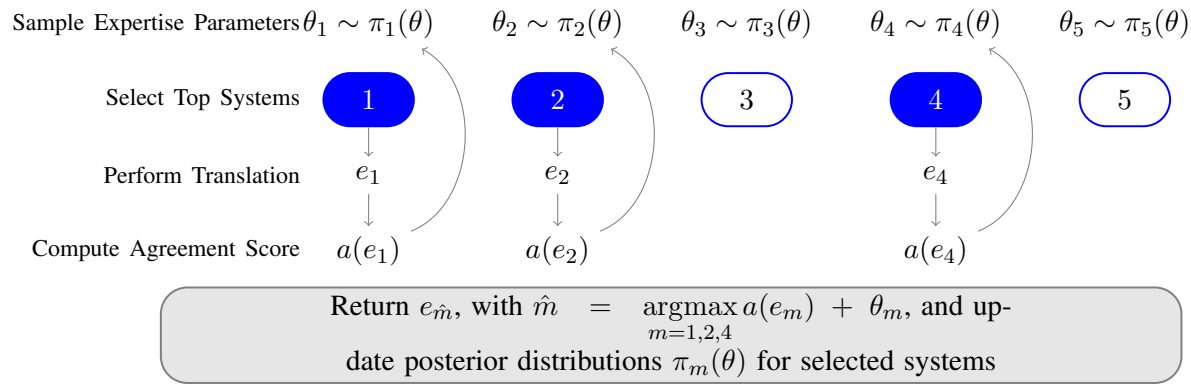

Figure 1: The setup of VOTING II for $N = 5$. Expertise θ is sampled from posterior distribution $\pi(\theta)$ for each system, and three systems are selected. Here we assume the top 3 systems are m_1, m_2 and m_4. Then, agreement scores for their translations are calculated. The translation with the highest agreement is returned. Finally, the posterior distributions are updated for all three selected systems.

- Select top three MT systems m, m' and m'' with the highest expertise parameters and decode translation request f. Let us assume translation outputs are as e_m, $e_{m'}$ and $e_{m''}$ respectively.
- Compute $a(e_m)$, $a(e_{m'})$ and $a(e_{m''})$.
- Add $a(e_m)$ to $\mathcal{A}_m$, $a(e_{m'})$ to $\mathcal{A}_{m'}$ and $a(e_{m''})$ to $\mathcal{A}_{m''}$. Update sample mean $\bar{\mu}_m$ and sample variance δ_m^2 for $\mathcal{A}_m$, $\mathcal{A}_{m'}$ and $\mathcal{A}_{m''}$.

Analysis: MT systems are promoted/demoted explicitly during learning. A high agreement score increases the sample mean for a promoted system, while a low agreement score decreases the sample mean for a demoted system. A promoted system becomes more likely to be selected in later rounds, but it is not the case for a demoted system.

The chance of being selected for MT systems also depends on variance for sampling expertise parameters. The variance effect decreases with sample size $|\mathcal{A}|$. This reflects that the learning becomes gradually more confident about its estimate of expertise parameters.

3.2.2 Voting II Binary

Our second model (VOTING II - BINARY) uses Beta distribution to sample expertise parameters. The parameters of the posterior distribution is updated based on a simplified outcome of agreement scores, which has only two values: [0, 1] (i.e. SUCCESS/FAILURE). This is done by performing a *Bernoulli* trial with success probability exactly as the agreement score.

Let us assume a sample of simplified agreement scores from all translation outputs produced by an MT system m as $\bar{\mathcal{A}}_m = \{\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_{|\bar{\mathcal{A}}_m|}\}$. For this sample, we focus on the numbers of SUCCESSes/FAILUREs instead of the sample *mean* and sample *variance*. Let us denote the numbers as S_m and F_m.

In VOTING II - BINARY, we assume that for a sample of simplified agreement scores $\bar{\mathcal{A}}$, the number of SUCCESSes is the output of a Binomial probability distribution with $|\bar{\mathcal{A}}|$ Bernoulli trials with success probability exactly as expertise parameter θ. We also use the Beta distribution with two hyper-parameters α and β for priors for the expertise parameter θ in VOTING II - BINARY, maintaining uncertainty over their values.

This results in a Beta-Binomial model for VOTING II - BINARY: the expertise parameter θ_m of each MT system m is sample from a Beta distribution with hyper-parameters $S_m + \alpha$ and $F_m + \beta$:

$$\theta_m \sim \text{Beta}(S_m + \alpha, F_m + \beta). \quad (3)$$

In our experiments we set $\alpha = \beta = 1$ for every MT system.

Our heuristic algorithm for learning expertise parameters in VOTING II - BINARY is in the same spirit as in VOTING II - REAL. Given a translation request f, expertise parameters are drawn from the posterior distributions, and top three MT systems m, m' and m'' with the highest expertise parameters are selected to decode f. This results in different translation outputs e_m, $e_{m'}$ and $e_{m''}$ respectively. The update is as follows:

- Compute $a(e_m)$, $a(e_{m'})$ and $a(e_{m''})$.
- Sample $\bar{a}(e_m)$, $\bar{a}(e_{m'})$ and $\bar{a}(e_{m''})$ from

Bernoulli trials with success probability exactly as $a(e_m)$, $a(e_{m'})$ and $a(e_{m''})$ respectively.

- Add $\bar{a}(e_m)$ to $\bar{\mathcal{A}}_m$, $\bar{a}(e_{m'})$ to $\bar{\mathcal{A}}_{m'}$ and $\bar{a}(e_{m''})$ to $\bar{\mathcal{A}}_{m''}$. Update for S_m and F_m for $\bar{\mathcal{A}}_m$, $S_{m'}$ and $F_{m'}$ for $\bar{\mathcal{A}}_{m'}$, $S_{m''}$ and $F_{m''}$ for $\bar{\mathcal{A}}_{m''}$.

Analysis: MT systems are promoted/demoted explicitly during learning: the posterior Beta($S+1+\alpha, F+\beta$) has a higher mean than Beta($S+\alpha, F+\beta$) and the posterior Beta($S+\alpha, F+1+\beta$) has a lower mean than distribution Beta($S+\alpha, F+\beta$).

Both Beta($S+1+\alpha, F+\beta$) and Beta($S+\alpha, F+1+\beta$) have a lower variance than distribution Beta($S+\alpha, F+\beta$). The variance effect thus also decreases with sample size $|\bar{\mathcal{A}}|$.

4 Experiment Design

We conduct experiments with three language pairs: *Spanish-English*, *French-English* and *German-English*. We create different translation ecosystems with a large number (from 6 to 10) of domain-specific MT systems for experiments. Our experiments are extensive with 23 translation tasks in total, which are unknown and out-of-domain. Note that we use NMT for one language pair and SMT for the rest, and the motivation behind this decision is simply that training SMT is somewhat easier than NMT for us.

4.1 Domain-specific MT system

Spanish-English: Our MT system is an attention-based Neural MT system (Bahdanau et al., 2015) for English-Spanish. We use `Nematus` (Sennrich et al., 2016; Sennrich and Haddow, 2016) with 512-dimensional word embeddings and layers. We use a vocab size of $50K$ for both the source and target languages. The vocabulary contains the top word types from all domains combined, and we train on sentences up to length 50. Pervasive dropout (Gal and Ghahramani, 2015) is applied to all vertical and recurrent connections, but not on word types. We optimize MT systems using Adam (Kingma and Ba, 2014) with a learning rate of 0.0001 and use early-stopping to prevent over-fitting. Translations are obtained using beam search with a beam of size 12.

We create a medium scale translation ecosystem with 6 different domain-specific Neural MT systems for Spanish-English. Each MT system is trained on a domain-specific dataset consisting of $250K$ sentence pairs, which is taken from OPUS. The system is tuned on an in-domain devset with $3K$ sentence pairs. The domains are: *Subtitles* (Domain 1), *Wikipedia* (Domain 2), *Medicine* (Domain 3), *Legal* (Domain 4), *News* (Domain 5), and *Speech* (Domain 6). Each domain has an in-domain test set with $3K$ sentence pairs as translation task.

French-English: The scale of our ecosystem is increased to 10 instead of 6 for experiments with French-English, Our MT systems are with SMT instead of Neural MT systems. Each SMT system is a standard phrase-based approach (Koehn et al., 2003). The language model is a 4-gram model with Kneser-Ney smoothing, estimated by KenLM (Heafield et al., 2013) from in-domain monolingual corpus. We use the k-best batch MIRA to tune MT systems (Cherry and Foster, 2012). Finally, the decoder is MOSES (Koehn et al., 2007).

Each domain-specific SMT system is trained on a domain-specific dataset consisting of $250K$ sentence pairs, and tuned on an in-domain devset with $3K$ sentence pairs taken from OPUS. The domains are: *Book* (Domain 1), *Speech* (Domain 2), *IT* (Domain 3), *Bank* (Domain 4), *News* (Domain 5), *Medicine* (Domain 6), *Wikipedia* (Domain 7), *Legal* (Domain 8), *European Parliament* (Domain 9), *Subtitles* (Domain 10). Similarly, each domain has an in-domain test set with $3K$ sentence pairs as translation task.

German-English: Domain-specific MT systems are constructed differently for German-English. We first train an SMT system on a dataset consisting of $4.1M$ sentence pairs released for WMT 2015 Shared Task. We then optimize the system over 7 different domain-specific devsets with different domains taken from TAUS. The domains are: Consumer Electronics (Domain 1), Hardware (Domain 2), Industrial Electronics (Domain 3), Legal (Domain 4), Professional & Business (Domain 5), Software (Domain 6), Retail Distribution (Domain 7).

The agreement degree between domain-specific MT systems for our German-English translation ecosystem for the pair is expected to be significantly higher than for the other cases.

4.2 Translation Task

Given each translation ecosystem, we are given one task out of the N translation tasks at test time. We evaluate how do we obtain translation quality from an ecosystem with range of remaining $N-1$

Spanish-English												
Tasks	Reference						Avg.	Rank 2	Rank 1	VOTE I	VOTE II	
	MT1	MT2	MT3	MT4	MT5	MT6					REAL	BIN.
Task 1	–	14.3	2.2	2.8	22.5	19.4	15.1	19.4	22.5	20.4	*22.4*	*22.1*
Task 2	7.0	–	6.2	13.6	31.0	20.3	14.1	20.3	31.0	26.6	*29.9*	*29.5*
Task 3	2.3	21.0	–	20.7	17.8	11.3	14.6	20.7	21.0	22.8	*23.1*	*23.0*
Task 4	2.7	25.6	8.0	–	22.1	15.1	14.7	22.1	25.6	23.9	*24.7*	*24.4*
Task 5	7.6	27.6	4.9	10.9	–	22.6	14.7	22.6	27.6	25.6	*26.6*	*26.6*
Task 6	16.1	24.8	4.8	7.2	29.0	–	16.4	24.8	29.0	26.8	*26.7*	*28.2*

Table 3: Results for Spanish-English experiments.

French-English																	
Tasks	Reference										Avg.	Rank 3	Rank 2	Rank 1	VOTE I	VOTE II	
	MT1	MT2	MT3	MT4	MT5	MT6	MT7	MT8	MT9	MT10						REAL	BIN.
Task 1	–	9.6	6.3	9.8	12.2	8.7	11.5	11.7	13.9	5.7	9.9	11.7	12.2	13.9	12.7	*13.0*	*12.5*
Task 2	18.3	–	14.8	13.3	27.3	11.4	21.4	10.7	22.3	20.2	17.7	21.4	22.3	27.3	22.5	*23.1*	*23.0*
Task 3	16.9	22.9	–	15.6	19.6	14.1	19.9	12.5	17.5	16.1	17.2	19.6	19.9	22.9	19.2	*19.2*	*20.5*
Task 4	33.9	21.9	21.5	–	29.0	22.9	26.2	35.0	34.2	11.3	26.2	33.9	34.2	35.0	30.2	*29.0*	*30.3*
Task 5	16.0	20.7	11.2	13.2	–	10.7	18.4	12.0	17.9	12.9	14.8	17.9	18.4	20.7	17.5	*17.8*	*16.9*
Task 6	26.7	22.5	21.6	24.7	26.9	–	25.0	21.8	22.3	16.8	23.1	25.0	26.7	26.9	25.9	*26.2*	*25.9*
Task 7	15.8	18.8	14.1	15.6	20.8	14.9	–	14.8	17.8	14.9	16.4	17.8	18.8	20.8	18.6	*19.4*	*18.1*
Task 8	31.4	15.8	11.0	27.3	22.3	15.2	23.6	–	29.4	18.8	20.8	27.3	29.4	31.4	26.6	*24.9*	*27.9*
Task 9	21.4	15.1	7.6	15.7	19.5	8.4	16.4	14.8	–	8.6	14.2	16.4	19.5	21.4	19.3	*18.9*	*19.5*
Task 10	12.0	23.3	10.7	9.6	22.8	8.3	16.9	8.4	17.3	–	14.4	17.3	22.8	23.3	17.5	*17.6*	*15.5*

Table 4: Results for French-English experiments.

pre-trained domain-specific systems.

5 Results

5.1 Ecosystem Performance

We first investigate how well the ecosystems handle unknown and out-of-domain translation tasks. Tables 3, 4 and 5 present the results (in BLEU). Note that:

- AVG: average of BLEU score of MT systems
- Rank 3, Rank 2, Rank 1: top 3 MT systems
- Vote I: VOTING I method
- Vote II Real: VOTING II method with real reward
- Vote II Bin: VOTING II method with binary reward

As expected, translation quality degrades substantially for most pre-trained MT systems given such a translation task. The *Subtitle*-adapted MT system for Spanish-English (MT 1 - Tables 3) is a notable example to raise the issue: the translation accuracy substantially drops for the other out-of-domain translation tasks (i.e. Task 2 (Wikipedia): 7.0 BLEU score, Task 3 (Medicine): 2.3 BLEU score, Task 4 (Legal): 2.7 BLEU score, Task 5 (News): 7.6 BLEU score, Task 6 (Speech): 16.1 BLEU score).

However, the degradation of each pre-trained MT system is different from the others. For example, the Speech-adapted MT system for Spanish-English (MT 6 - Tables 3) drops their performance significantly for only Task 3 (Medicine) (11.3 BLEU score) and Task 4 (Legal) (15.1 BLEU score). The Speech-adapted MT system is capable of handling other out-of-domain translation tasks (i.e. Task 1 (Subtitles): 19.4 BLEU score, Task 2 (Wikipedia): 20.3 BLEU score, Task 5 (News): 22.6 BLEU score).

For 23 out-of-domain translation tasks in total, our results show that despite the translation quality substantially drops for most pre-trained MT systems, a few pre-trained MT systems are still competitive to handle the tasks. In $21/23$ cases, top MT systems with respect to a certain translation task are still able to handle the task well.[3]

This supports our claim: Having a large-scale ecosystem of pre-trained MT systems is very useful for handling out-of-domain tasks in practice. But is it possible to gain competitive performance to top rank MT systems from ecosystem of pre-trained domain-specific systems for unknown and out-of-domain translation tasks? Our experiments show that it is possible with our proposed frameworks.

[3]For convenience, we set a BLEU threshold (20) to decide if the MT quality is good or not. In practice, it should not be a good idea to have such a fixed threshold for any domain.

German-English													
Tasks	Reference							Avg.	Rank 2	Rank 1	VOTE I	VOTE II	
	MT1	MT2	MT3	MT4	MT5	MT6	MT7	All				REAL	BIN.
Task 1	–	22.9	23.1	19.8	18.9	23.2	23.0	21.8	23.1	23.2	23.0	23.0	23.0
Task 2	20.2	–	20.5	19.7	19.0	20.8	20.7	20.2	20.7	20.8	20.7	20.7	20.7
Task 3	20.7	20.9	–	18.1	17.4	21.1	20.7	19.8	20.9	21.1	21.0	20.2	20.9
Task 4	28.5	29.0	28.9	–	28.5	29.5	29.4	29.0	29.4	29.5	29.4	29.4	29.3
Task 5	12.6	13.8	13.7	14.8	–	13.4	13.4	13.6	13.8	14.8	13.6	13.6	13.6
Task 6	21.8	23.3	23.2	20.8	20.8	–	22.8	22.1	23.2	23.3	23.0	23.1	23.0
Task 7	32.3	33.5	33.5	28.2	28.2	33.0	–	31.5	33.5	33.5	33.2	33.4	33.3

Table 5: Results for German-English experiments.

Spanish-English								
Tasks	MIN	SC	Avg. All	DC	Avg. TRs	VOTE I	VOTE II REAL	VOTE II BIN.
Task 1	2.2	10.9	15.1	18.6	21.0	*20.4*	*22.4*	*22.1*
Task 2	6.2	15.2	14.1	15.7	25.7	*26.6*	*29.9*	*29.5*
Task 3	2.3	18.4	14.6	15.2	20.9	*22.8*	*23.1*	*23.0*
Task 4	2.7	13.9	14.7	13.5	23.9	*23.9*	*24.7*	*24.4*
Task 5	4.9	14.0	14.7	17.3	25, 1	*25.6*	*26.6*	*26.6*
Task 6	4.8	17.0	16.4	18.2	26, 9	*26.8*	*26.7*	*28.2*

Table 6: A detailed comparison for other baselines (SC: System Combination, DC: Domain Classification, Avg. TR: Average baseline between top rank MT systems (Rank 1 and Rank 2) for Spanish-English.

5.2 Our Framework Performance

Tables 3, 4 and 5 present the results. Note that our models are stochastic, and results for our experiments are averaged among 20 runs. The main findings are:

VOTING I substantially outperforms Rank 2 for all cases for Spanish-English. It outperforms Rank 3 for 6/10 tasks for French-English. We would like to emphasize that: (1) this performance is obtained without any knowledge about translation task; and (2) the gap between the best and the worst MT systems for each task in ecosystems is huge (i.e. usually around $+20$ BLEU score). This validates the idea behind VOTING I: Good translations are likely to be similar to the others.

We perform System Combination (SC) by ensembling all NMT systems for the tasks. SC rather gives a poor performance in our setting (Table 6). We should emphasize that the result is rather expected: SC degrades translation quality substantially because most pre-trained MT systems in the ecosystem are incapable of handling the task.[4]

We also create a simple domain classifier (DC) for translation requests: We train different in-domain language models from in-domain monolingual corpora, and perform a search to select an MT system from the ecosystem based on their language model probability of each translation request: $\hat{m} = \underset{m=1,...,N}{\operatorname{argmax}} P_m(f)$. DC also rather gives a poor performance in our setting (Table 6). It outperforms the average baseline (Avg. All) in most cases, but its performance is far behind the middle of top rank MT systems (Avg. TRs). The result is unsurprising: it is hard to expect a domain classifier for translation requests provides robust performance for target domain that is not only unknown but also out-of-domain.

Interestingly, VOTING I gives the performance at least in the middle between Rank 1 and Rank 2 in $5/6$ tasks for Spanish-English, except only Task 1. Meanwhile, the performance is at least in the middle between Rank 1, Rank 2 and Rank 3 in $3/10$ tasks for French-English.

VOTING II - REAL and VOTING II - BINARY perform better than VOTING I for $5/6$ tasks for Spanish-English. All these frameworks perform substantially better (at least $+1.0$ BLEU score) than VOTING I in 4 cases (Tasks 1, 2, 3 and 5). For French-English, VOTING II - REAL and VOTING II - BINARY perform at least compatible to VOTING I for $6/10$ tasks. Each of these frameworks performs better than VOTING I for $4/10$ tasks.

The results validate the idea behind VOTING II: MT systems that are good in a domain tend to agree with each other.

VOTING II - REAL usually performs better than VOTING II - BINARY. This is reasonable as in VOTING II - BINARY, model parameters are updated based on simplified outcome of the agreement scores instead of the agreement scores.

Despite having a different set up for constructing domain-specific MT systems, all our observations are also confirmed for German-English as in Table 5. VOTING I gives the performance in the middle between Rank 1 and Rank 2 in $6/7$ tasks,

[4]We should also note that interpolating all SMT systems gives a rather poor performance as well. This is because of the same reason: most pre-trained MT systems in the ecosystem are incapable of handling the task. We did not report the results here due to space constraints.

except only Task 5. VOTING II provides compatible performance to VOTING I. This is reasonable as when MT systems are close to the others regarding their translation quality, the benefits of reducing the decoding cost is what VOTING II is expected to provide. It is worthy to emphasize that our VOTING frameworks still outperform the average baseline significantly.

5.3 Disadvantage of our method

While the result from our method is impressive, we should be clear about its disadvantage. We found that:

- A generic system trained with all the training data of the different domains normally produces significantly better performance than what our framework provides.
- An indomain MT system trained on in-domain training data normally produces significantly better performance than what our framework provides as well.

Improving our framework to make it work compatible to those stronger baselines is a goal of future research.

6 Conclusion

This work shows that having an ecosystem of pre-trained domain-specific MT systems is not only efficient for in-domain translation tasks, but could be also very useful for out-of-domain translation tasks. More specifically, we show that an out-of-domain translation task can be out-of-scope of most pre-trained adapted MT systems in the ecosystem, but a few others can be still very capable of handling the task. We conduct extensive experiments with different scale (from 6 to 10) ecosystems of pre-trained MT systems to support our claim. We also contribute two frameworks that gain competitive performance to top rank MT systems from ecosystem of pre-trained domain-specific systems for unknown and potentially out-of-domain translation tasks. We hope our study fills an important gap in the domain adaptation literature: making translation ecosystems with domain-adapted MT systems capable of handling unknown and out-of-domain tasks.

Acknowledgement

This work was conducted when the author was at University of Amsterdam, as well as when he was in his visiting research to University of Sheffield. The author would like to thanks many people since then, including: Kashif Shah, Lucia Specia, Joost Bastings, Khalil Simaa'n, Ivan Titov. The author also thanks reviewers for their constructive comments.

References

[Axelrod et al.2011] Axelrod, Amittai, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *EMNLP*.

[Bahdanau et al.2015] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[Bapna and Firat2019] Bapna, Ankur and Orhan Firat. 2019. Non-parametric adaptation for neural machine translation. *CoRR*, abs/1903.00058.

[Bertoldi et al.2014] Bertoldi, Nicola, Patrick Simianer, Mauro Cettolo, Katharina Wäschle, Marcello Federico, and Stefan Riezler. 2014. Online adaptation to post-edits for phrase-based statistical machine translation. *Machine Translation*.

[Bisazza et al.2011] Bisazza, Arianna, Nick Ruiz, and Marcello Federico. 2011. Fill-up versus interpolation methods for phrase-based smt adaptation. In *IWSLT*.

[Blain et al.2015] Blain, F., F. Bougares, A. Hazem, L. Barrault, and H. Schwenk. 2015. Continuous adaptation to user feedback for statistical machine translation. In *NAACL-HLT (Short Papers)*.

[Carpuat et al.2013] Carpuat, Marine, Hal Daume III, Katharine Henry, Ann Irvine, Jagadeesh Jagarlamudi, and Rachel Rudinger. 2013. Sensespotting: Never let your parallel data tie you to an old domain. In *ACL*.

[Chen and Huang2016] Chen, Boxing and Fei Huang. 2016. Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data. In *Conll*.

[Chen et al.2013] Chen, Boxing, Roland Kuhn, and George Foster. 2013. Vector space model for adaptation in statistical machine translation. In *ACL*.

[Chen et al.2017] Chen, Boxing, Colin Cherry, George Foster, and Samuel Larkin. 2017. Cost weighting for neural machine translation domain adaptation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 40–46. Association for Computational Linguistics.

[Cherry and Foster2012] Cherry, Colin and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *NAACL HLT*.

[Chiang et al.2011] Chiang, David, Steve DeNeefe, and Michael Pust. 2011. Two easy improvements to lexical weighting. In *ACL HLT (Short Papers)*.

[Chu and Wang2018] Chu, Chenhui and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

[Clark et al.2012] Clark, Jonathan H, Alon Lavie, and Chris Dyer. 2012. One system, many domains: Open-domain statistical machine translation via feature augmentation. *AMTA*.

[Cuong and Sima'an2014a] Cuong, Hoang and Khalil Sima'an. 2014a. Latent domain phrase-based models for adaptation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

[Cuong and Sima'an2014b] Cuong, Hoang and Khalil Sima'an. 2014b. Latent domain translation models in mix-of-domains haystack. In *Proceedings of COLING*.

[Cuong and Sima'an2015] Cuong, Hoang and Khalil Sima'an. 2015. Latent domain word alignment for heterogeneous corpora. In *NAACL-HLT*.

[Cuong and Sima'an2017] Cuong, Hoang and Khalil Sima'an. 2017. A survey of domain adaptation for statistical machine translation. *Machine Translation*, 31(4):187–224, December.

[Cuong et al.2016] Cuong, Hoang, Khalil Sima'an, and Ivan Titov. 2016. Adapting to all domains at once: Rewarding domain invariance in smt. In *TACL*.

[Denkowski et al.2014] Denkowski, Michael, Chris Dyer, and Alon Lavie. 2014. Learning from post-editing: Online model adaptation for statistical machine translation. In *EACL*.

[Duh et al.2013] Duh, Kevin, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *ACL (Short Papers)*.

[Eetemadi et al.2015] Eetemadi, Sauleh, William Lewis, Kristina Toutanova, and Hayder Radha. 2015. Survey of data-selection methods in statistical machine translation. *Machine Translation*.

[Eidelman et al.2012] Eidelman, Vladimir, Jordan Boyd-Graber, and Philip Resnik. 2012. Topic models for dynamic translation model adaptation. In *ACL (Short Papers)*.

[Farajian et al.2017] Farajian, M. Amin, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark, September. Association for Computational Linguistics.

[Foster et al.2010] Foster, George, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *EMNLP*.

[Gal and Ghahramani2015] Gal, Y. and Z. Ghahramani. 2015. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. *ArXiv e-prints*, December.

[Haddow2013] Haddow, Barry. 2013. Applying pairwise ranked optimisation to improve the interpolation of translation models. In *NAACL HLT (Short Papers)*.

[Hasler et al.2014] Hasler, Eva, Phil Blunsom, Philipp Koehn, and Barry Haddow. 2014. Dynamic topic adaptation for phrase-based mt. In *EACL*.

[Heafield et al.2013] Heafield, Kenneth, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified kneser-ney language model estimation. In *ACL (Short Papers)*.

[Hildebrand and Vogel2008] Hildebrand, Almut Silja and Stephan Vogel. 2008. Combination of Machine Translation Systems via Hypothesis Selection from Combined N-Best Lists. In *Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*, pages 254–261.

[Hu et al.2014] Hu, Yuening, Ke Zhai, Vladimir Eidelman, and Jordan Boyd-Graber. 2014. Polylingual tree-based topic models for translation domain adaptation. In *ACL*.

[Jayaraman and Lavie2005] Jayaraman, Shyamsundar and Alon Lavie. 2005. Multi-engine machine translation guided by explicit word matching. In *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, ACLdemo '05, pages 101–104, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Joty et al.2015] Joty, Shafiq, Hassan Sajjad, Nadir Durrani, Kamla Al-Mannai, Ahmed Abdelali, and Stephan Vogel. 2015. How to avoid unwanted pregnancies: Domain adaptation using neural network models. In *EMNLP*.

[Karimova et al.2018] Karimova, Sariya, Patrick Simianer, and Stefan Riezler. 2018. A user-study on online adaptation of neural machine translation to human post-edits. *Machine Translation*, 32(4):309–324.

[Khayrallah et al.2017] Khayrallah, Huda, Gaurav Kumar, Kevin Duh, Matt Post, and Philipp Koehn. 2017. Neural lattice search for domain adaptation in machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 20–25. Asian Federation of Natural Language Processing.

[Kingma and Ba2014] Kingma, Diederik and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kirchhoff and Bilmes2014] Kirchhoff, Katrin and Jeff Bilmes. 2014. Submodularity for data selection in machine translation. In *EMNLP*.

[Kobus et al.2017] Kobus, Catherine, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria, September. INCOMA Ltd.

[Koehn and Knowles2017] Koehn, Philipp and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39. Association for Computational Linguistics.

[Koehn and Schroeder2007] Koehn, Philipp and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *WMT*.

[Koehn et al.2003] Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL HLT*.

[Koehn et al.2007] Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*.

[Koehn et al.2014] Koehn, Philipp, Chara Tsoukala, and Herve Saint-Amand. 2014. Refinements to interactive translation prediction based on search graphs. In *ACL (Short Papers)*.

[Kothur et al.2018] Kothur, Sachith Sri Ram, Rebecca Knowles, and Philipp Koehn. 2018. Document-level adaptation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 64–73, Melbourne, Australia, July. Association for Computational Linguistics.

[Lin and Och2004] Lin, Chin-Yew and Franz Josef Och. 2004. Orange: A method for evaluating automatic evaluation metrics for machine translation. In *COLING*.

[Macherey and Och2007] Macherey, Wolfgang and Franz J. Och. 2007. An empirical study on computing consensus translations from multiple machine translation systems. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 986–995, Prague, Czech Republic, June. Association for Computational Linguistics.

[Mansour and Ney2014] Mansour, Saab and Hermann Ney. 2014. Unsupervised adaptation for statistical machine translation. In *WMT*.

[Michel and Neubig2018] Michel, Paul and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 312–318, Melbourne, Australia, July. Association for Computational Linguistics.

[Nguyen et al.2017] Nguyen, Khanh, Hal Daumé III, and Jordan Boyd-Graber. 2017. Reinforcement learning for bandit neural machine translation with simulated human feedback. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1464–1474, Copenhagen, Denmark, September. Association for Computational Linguistics.

[Ortiz-Martínez et al.2010] Ortiz-Martínez, Daniel, Ismael García-Varea, and Francisco Casacuberta. 2010. Online learning for interactive statistical machine translation. In *NAACL-HLT*.

[Ortiz-Martínez2016] Ortiz-Martínez. 2016. Online learning for statistical machine translation. *Comput. Linguist.*

[Papineni et al.2002] Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *ACL*.

[Peris and Casacuberta2018] Peris, Álvaro and Francisco Casacuberta. 2018. Active learning for interactive neural machine translation of data streams. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 151–160, Brussels, Belgium, October. Association for Computational Linguistics.

[Razmara et al.2012] Razmara, Majid, George Foster, Baskaran Sankaran, and Anoop Sarkar. 2012. Mixing multiple translation models in statistical machine translation. In *ACL*.

[Sennrich and Haddow2016] Sennrich, Rico and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 83–91, Berlin, Germany, August. Association for Computational Linguistics.

[Sennrich et al.2013] Sennrich, Rico, Holger Schwenk, and Walid Aransa. 2013. A multi-domain translation model framework for statistical machine translation. In *ACL*.

[Sennrich et al.2016] Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for wmt 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany, August. Association for Computational Linguistics.

[Sennrich2012a] Sennrich, Rico. 2012a. Mixture-modeling with unsupervised clusters for domain adaptation in statistical machine translation. In *EAMT*.

[Sennrich2012b] Sennrich, Rico. 2012b. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *EACL*.

[Shah et al.2010] Shah, Kashif, Loïc Barrault, and Holger Schwenk. 2010. Translation model adaptation by resampling. In *WMT*.

[Sokolov et al.2015] Sokolov, Artem, Stefan Riezler, and Tanguy Urvoy. 2015. Bandit structured prediction for learning from partial feedback in statistical machine translation. In *MTSummit*.

[Sokolov et al.2016] Sokolov, Artem, Julia Kreutzer, Christopher Lo, and Stefan Riezler. 2016. Learning structured predictors from bandit feedback for interactive nlp. In *ACL*.

[Sokolov et al.2017] Sokolov, Artem, Julia Kreutzer, Kellen Sunderland, Pavel Danchenko, Witold Szymaniak, Hagen Fürstenau, and Stefan Riezler. 2017. A shared task on bandit learning for machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 514–524, Copenhagen, Denmark, September. Association for Computational Linguistics.

[Su et al.2015] Su, Jinsong, Deyi Xiong, Yang Liu, Xianpei Han, Hongyu Lin, Junfeng Yao, and Min Zhang. 2015. A context-aware topic model for statistical machine translation. In *ACL-IJCNLP*.

[Tars and Fishel2018] Tars, Sander and Mark Fishel. 2018. Multi-domain neural machine translation. *CoRR*, abs/1805.02282.

[van der Wees et al.2017] van der Wees, Marlies, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410. Association for Computational Linguistics.

[Wang et al.2012] Wang, Wei, Klaus Macherey, Wolfgang Macherey, Franz Och, and Peng Xu. 2012. Improved domain adaptation for statistical machine translation. In *AMTA*.

[Wang et al.2018] Wang, R., M. Utiyama, A. Finch, L. Liu, K. Chen, and E. Sumita. 2018. Sentence selection and weighting for neural machine translation domain adaptation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pages 1–1.

[Wuebker et al.2016] Wuebker, Joern, Spence Green, John DeNero, Sasa Hasan, and Minh-Thang Luong. 2016. Models and inference for prefix-constrained machine translation. In *ACL*.

[Zhang et al.2014] Zhang, Min, Xinyan Xiao, Deyi Xiong, and Qun Liu. 2014. Topic-based dissimilarity and sensitivity models for translation rule selection. *JAIR*.