Latent Domain Phrase-based Models for Adaptation

Hoang Cuong and **Khalil Sima'an**
Institute for Logic, Language and Computation
University of Amsterdam
Science Park 107, 1098 XG Amsterdam, The Netherlands
{c.hoang,k.simaan}@uva.nl

Abstract

Phrase-based models directly trained on *mix-of-domain corpora* can be sub-optimal. In this paper we equip phrase-based models with a *latent domain* variable and present a novel method for adapting them to an in-domain task represented by a seed corpus. We derive an EM algorithm which alternates between inducing domain-focused phrase pair estimates, and weights for mix-domain sentence pairs reflecting their relevance for the in-domain task. By embedding our latent domain phrase model in a sentence-level model and training the two in tandem, we are able to adapt *all core* translation components together – phrase, lexical and reordering. We show experiments on weighing sentence pairs for relevance as well as adapting phrase-based models, showing significant performance improvement in both tasks.

1 Mix vs. Latent Domain Models

Domain adaptation is usually perceived as utilizing a small seed in-domain corpus to adapt an existing system trained on an *out-of-domain* corpus. Here we are interested in adapting an SMT system trained on a large *mix-domain* corpus C_{mix} to an *in-domain task* represented by a seed parallel corpus C_{in}. The mix-domain scenario is interesting because often a large corpus consists of sentence pairs representing diverse domains, e.g., news, politics, finance, sports, etc.

At the core of a standard state-of-the-art phrase-based system (Och and Ney, 2004) is a phrase table $\{\langle \tilde{e}, \tilde{f} \rangle\}$ extracted from the word-aligned training data together with estimates for $P_t(\tilde{e} \mid \tilde{f})$ and $P_t(\tilde{f} \mid \tilde{e})$. Because the translations of words often vary across domains, it is likely that in a mix-domain corpus C_{mix} the translation ambiguity will increase with the *domain diversity*. Furthermore, the statistics in C_{mix} will reflect translation preferences *averaged* over the diverse domains. In this sense, phrase-based models trained on C_{mix} can be considered *domain-confused*. This often leads to suboptimal performance (Gascó et al., 2012; Irvine et al., 2013).

Recent adaptation techniques can be seen as mixture models, where two or more phrase tables, estimated from in- and mix-domain corpora, are combined together by interpolation, fill-up, or multiple-decoding paths (Koehn and Schroeder, 2007; Bisazza et al., 2011; Sennrich, 2012; Razmara et al., 2012; Sennrich et al., 2013). Here we are interested in the specific question *how to induce* a phrase-based model *from* C_{mix} for *in-domain translation*? We view this as *in-domain focused training* on C_{mix}, a complementary adaptation step which might precede any further combination with other models, e.g., in-, mix- or general-domain.

The main challenge is how to induce from C_{mix} a phrase-based model for the in-domain task, given only C_{in} as evidence? We present an approach whereby the contrast between in-domain prior distributions and "out-domain" distributions is exploited for softly inviting (or recruiting) C_{mix} phrase pairs to either camp. To this end we in-

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 566–576,
October 25-29, 2014, Doha, Qatar.

troduce a *latent domain variable* D to signify in- (D_1) and out-domain (D_0) respectively.[1]

With the introduction of the latent variables, we extend the translation tables in phrase-based models from generic $P_t(\tilde{e} \mid \tilde{f})$ to domain-focused by conditioning them on D, i.e., $P_t(\tilde{e} \mid \tilde{f}, D)$ and decomposing them as follows:

$$P_t(\tilde{e} \mid \tilde{f}, D) = \frac{P_t(\tilde{e} \mid \tilde{f})P(D \mid \tilde{e}, \tilde{f})}{\sum_{\tilde{e}} P_t(\tilde{e} \mid \tilde{f})P(D \mid \tilde{e}, \tilde{f})}. \quad (1)$$

Where $P(D \mid \tilde{e}, \tilde{f})$ is viewed as the *latent phrase-relevance models*, i.e., the probability that a phrase pair is in- (D_1) or out-domain (D_0). In the end, our goal is to replace the domain-confused tables, $P_t(\tilde{e} \mid \tilde{f})$ and $P_t(\tilde{f} \mid \tilde{e})$, with the in-domain focused ones, $P_t(\tilde{e} \mid \tilde{f}, D_1)$ and $P_t(\tilde{f} \mid \tilde{e}, D_1)$.[2] Note how $P_t(\tilde{e} \mid \tilde{f}, D_1)$ and $P_t(\tilde{f} \mid \tilde{e}, D_1)$ contains $P_t(\tilde{e} \mid \tilde{f})$ and $P_t(\tilde{f} \mid \tilde{e})$ as special case.

Eq. 1 shows that the key to training the latent phrase-based translation models is to train the latent phrase-relevance models, $P(D \mid \tilde{e}, \tilde{f})$. Our approach is to embed $P(D \mid \tilde{e}, \tilde{f})$ in asymmetric sentence-level models $P(D \mid \mathbf{e}, \mathbf{f})$ and train them on C_{mix}. We devise an EM algorithm where at every iteration, in- or out-domain estimates provide full sentence pairs $\langle \mathbf{e}, \mathbf{f} \rangle$ with expectations $\{P(D \mid \mathbf{e}, \mathbf{f}) \mid D \in \{0, 1\}\}$. Once these expectation are in C_{mix}, we induce re-estimates for the latent phrase-relevance models, $P(D \mid \tilde{e}, \tilde{f})$. Metaphorically, during each EM iteration the current in- or out-domain phrase pairs compete on *inviting* C_{mix} sentence pairs to be in- or out-domain, which bring in new (weights for) in- and out-domain phrases. Using the same algorithm we also show how to adapt all core translation components in tandem, including also lexical weights and lexicalized reordering models.

Next we detail our model, the EM-based invitation training algorithm and provide technical solutions to a range of difficulties. We report experiments showing good instance weighting performance as well as significantly improved phrase-based translation performance.

[1] Crucially, the lack of explicit out-domain data in C_{mix} is a major technical difficulty. We follow (Cuong and Sima'an, 2014) and in the sequel present a relatively efficient solution based on a kind of "burn-in" procedure.

[2] It is common to use these domain-focused models as additional features besides the domain-confused features. However, here we are more interested in *replacing* the domain-confused features rather than complementing them. This distinguishes this work from other domain adaptation literature for MT.

2 Model and training by invitation

Eq. 1 shows that the key to training the latent phrase-based translation models is to train the latent phrase-relevance models, $P(D \mid \tilde{e}, \tilde{f})$. As mentioned, for training $P(D \mid \tilde{e}, \tilde{f})$ on parallel sentences in C_{mix} we embed them in two asymmetric sentence-level models $\{P(D \mid \mathbf{e}, \mathbf{f}) \mid D \in \{0, 1\}\}$.

2.1 Domain relevance sentence models

Intuitively, sentence models for domain relevance $P(D \mid \mathbf{e}, \mathbf{f})$ are somewhat related to *data selection* approaches (Moore and Lewis, 2010; Axelrod et al., 2011). The dominant approach to data selection uses the contrast between perplexities of in- and mix-domain language models.[3] In the translation context, however, often a source phrase has different senses/translations in different domains, which cannot be distinguished with monolingual language models (Cuong and Sima'an, 2014). Therefore, our proposed latent sentence-relevance model includes two major latent components - *monolingual domain-focused relevance models* and *domain-focused translation models* derives as follows:

$$P(D \mid \mathbf{e},\ \mathbf{f}) = \frac{P(\mathbf{e}, \mathbf{f}, D)}{\sum_{D \in \{D_1, D_0\}} P(\mathbf{e}, \mathbf{f}, D)}, \quad (2)$$

where $P(\mathbf{e}, \mathbf{f}, D)$ can be decomposed as:

$$P(\mathbf{f}, \mathbf{e}, D) = \frac{1}{2}\Big(P(D)P_{lm}(\mathbf{e} \mid D)P_t(\mathbf{f} \mid \mathbf{e},\ D) + P(D)P_{lm}(\mathbf{f} \mid D)P_t(\mathbf{e} \mid \mathbf{f},\ D)\Big). \quad (3)$$

Here

- $P_t(\mathbf{e}|\mathbf{f}, D)$ and similarly $P_t(\mathbf{f}|\mathbf{e}, D)$: the latent domain-focused translation models aim at capturing the faithfulness of translation with respect to different domains. We simplify this as

[3] Note that earlier work on data selection exploits the contrast between in- and mix-domain. In (Cuong and Sima'an, 2014), we present the idea of using the language and translation models derived separately from in- and out-domain data, and show how it helps for data selection.

"bag-of-possible-phrases" translation models:[4]

$$P_t(\mathbf{e}|\mathbf{f}, D) := \prod_{\langle\tilde{e},\tilde{f}\rangle \in \mathcal{A}(\mathbf{e},\mathbf{f})} P_t(\tilde{e}|\tilde{f}, D)^{c(\tilde{e},\tilde{f})}, \quad (4)$$

where $\mathcal{A}(\mathbf{e}, \mathbf{f})$ is the multiset of phrases in $\langle \mathbf{e}, \mathbf{f} \rangle$ and $c(\cdot)$ denotes their count. Sub-model $P_t(\tilde{e}|\tilde{f}, D)$ is given by Eq. 1.

- $P_{lm}(\mathbf{e}|D)$, $P_{lm}(\mathbf{f}|D)$: the latent monolingual domain-focused relevance models aim at capturing the relevance of $\mathbf{e}$ and $\mathbf{f}$ for identifying domain D but here we consider them language models (LMs).[5] As mentioned, the out-domain LMs differ from previous works, e.g., (Axelrod et al., 2011), which employ mix-domain LMs. Here, we stress the difficulty in finding data to train *out-domain LMs* and present a solution based on *identifying pseudo out-domain data*.

- $P(D)$: the domain priors aim at modeling the percentage of relevant data that the learning framework induces. It can be estimated via phrase-level parameters but here we prefer sentence-level parameters:[6]

$$P(D) := \frac{\sum_{\langle \mathbf{e},\mathbf{f}\rangle \in C_{mix}} P(D \mid \mathbf{e}, \mathbf{f})}{\sum_D \sum_{\langle \mathbf{e},\mathbf{f}\rangle \in C_{mix}} P(D \mid \mathbf{e}, \mathbf{f})} \quad (5)$$

2.2 Training by invitation

Generally, our model can be viewed to have latent parameters $\Theta = \{\Theta_{D_0}, \Theta_{D_1}\}$. The training procedure seeks Θ that maximize the log-likelihood of the observed sentence pairs $\langle \mathbf{e}, \mathbf{f} \rangle \in C_{mix}$:

$$\mathcal{L} = \sum_{\langle \mathbf{e},\mathbf{f}\rangle \in C_{mix}} \log \sum_D P_{\Theta_D}(D, \mathbf{e}, \mathbf{f}). \quad (6)$$

It is obvious that there does not exist a closed-form solution for Equation 6 because of the existence of the log-term $\log \sum$. The EM algorithm (Dempster et al., 1977) comes as an alternative solution to fit the model. It can be seen to maximize $\mathcal{L}$ via block-coordinate ascent on a lower bound $\mathcal{F}(q, \Theta)$ using an auxiliary distribution $q(D \mid \mathbf{e}, \mathbf{f})$

$$\mathcal{F}(q, \Theta) = \sum_{\langle \mathbf{e},\mathbf{f}\rangle} \sum_D q(D \mid \mathbf{e}, \mathbf{f}) \log \frac{P_{\Theta_D}(D, \mathbf{e}, \mathbf{f})}{q(D \mid \mathbf{e}, \mathbf{f})} \quad (7)$$

where the inequality results, i.e., $\mathcal{L} \geq \mathcal{F}(q, \Theta)$, derived from $\log$ being concave and Jensen's inequality. We rewrite the Free Energy $\mathcal{F}(q, \Theta)$ (Neal and Hinton, 1999) as follows:

$$\begin{aligned} \mathcal{F} = & \sum_{\langle \mathbf{e},\mathbf{f}\rangle} \sum_D q(D \mid \mathbf{e}, \mathbf{f}) \log \frac{P_{\Theta_D}(D \mid \mathbf{e}, \mathbf{f})}{q(D \mid \mathbf{e}, \mathbf{f})} \\ & + \sum_{\langle \mathbf{e},\mathbf{f}\rangle} \sum_D q(D \mid \mathbf{e}, \mathbf{f}) \log P_\Theta(\mathbf{e}, \mathbf{f}) \\ & = \sum_{\langle \mathbf{e},\mathbf{f}\rangle} \log P_\Theta(\mathbf{e}, \mathbf{f}) \\ & - KL[q(D \mid \mathbf{e}, \mathbf{f}) \mid\mid P_{\Theta_D}(D \mid \mathbf{e}, \mathbf{f})], \end{aligned} \quad (8)$$

where $KL[\cdot \mid\mid \cdot]$ is the KL-divergence.

With the introduction of the KL-divergence, the alternating E and M steps for our EM algorithm are easily derived as

$$\begin{aligned} & \textbf{E-}step : q^{t+1} \\ & \text{argmax}_{q(D \mid \mathbf{e},\mathbf{f})} \mathcal{F}(q, \Theta^t) \; = \\ & \text{argmin}_{q(D \mid \mathbf{e},\mathbf{f})} KL[q(D|\mathbf{e}, \mathbf{f}) \mid\mid P_{\Theta_D^t}(D|\mathbf{e}, \mathbf{f})] \\ & = P_{\Theta_D^t}(D \mid \mathbf{e}, \mathbf{f}) \end{aligned} \quad (9)$$

$$\begin{aligned} & \textbf{M-}step : \Theta^{t+1} \\ & \text{argmax}_\Theta \mathcal{F}(q^{t+1}, \Theta) \; = \\ & \text{argmax}_\Theta \sum_{\langle \mathbf{e},\mathbf{f}\rangle} \sum_D q(D \mid \mathbf{e}, \mathbf{f}) \log P_{\Theta_D}(D, \mathbf{e}, \mathbf{f}) \end{aligned} \quad (10)$$

The iterative procedure is illustrated in Figure 1.[7] At the E-step, a guess for $P(D \mid \tilde{e}, \tilde{f})$ can be used to update $P_t(\tilde{f} \mid \tilde{e}, D)$ and $P_t(\tilde{e} \mid \tilde{f}, D)$ (i.e., using Eq. 1) and consequently $P_t(\mathbf{f} \mid \mathbf{e}, D)$ and $P_t(\mathbf{e} \mid \mathbf{f}, D)$ (i.e., using Eq. 4). These resulting table estimates, together with the domain-focused LMs and the domain priors are served as *expected counts* to update $P(D \mid \mathbf{e}, \mathbf{f})$.[8] At the M-step,

[4] We design our latent domain translation models with efficiency as our main concern. Future extensions could include the lexical and reordering sub-models (as suggested by an anonymous reviewer.)

[5] Relevance for identification or retrieval could be different from frequency or fluency. We leave this extension for future work.

[6] It should be noted that in most phrase-based SMT systems bilingual phrase probabilities are estimated heuristically from word alignmened data which often leads to overfitting. Estimating $P(D)$ from sentence-level parameters rather than from phrase-level parameters helps us avoid the overfitting which often accompanies phrase extraction.

[7] For simplicity, we ignore the LMs and prior models in the illustration in Fig. 1.

[8] Since we only use the in-domain corpus as priors to initilize the EM parameters, in technical perspective we do not want $P(D \mid \mathbf{e}, \mathbf{f})$ parameters to go too far off from the initialization. We therefore prefer the averaged style in practice, i.e., at the iteration n we update the $P(D \mid \mathbf{e}, \mathbf{f})$ parameters, $P^{(n)}(D|\mathbf{e}, \mathbf{f})$ as $\frac{1}{n}(P^{(n)}(D \mid \mathbf{e}, \mathbf{f}) + \sum_{i=1}^{n-1} P^{(i)}(D \mid \mathbf{e}, \mathbf{f}))$.

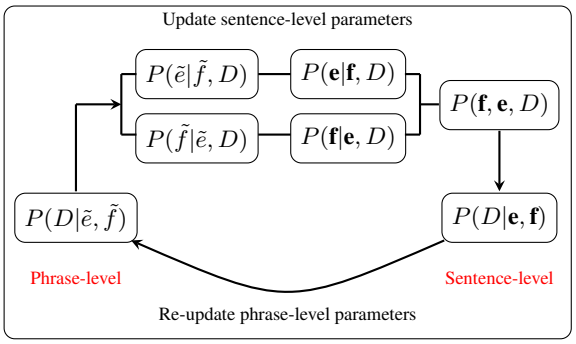

Figure 1: Our probabilistic invitation framework.

the new estimates for $P(D \mid \mathbf{e}, \mathbf{f})$ can be used to (softly) fill in the values of hidden variable D and estimate parameters $P(D \mid \tilde{e}, \tilde{f})$ and $P(D)$. The EM is guaranteed to converge to a local maximum of the likelihood under mild conditions (Neal and Hinton, 1999).

Before EM training starts we must provide a "reasonable" initial guess for $P(D \mid \tilde{e}, \tilde{f})$. We must also train the out-domain LMs, which needs the construction of *pseudo out-domain* data.[9] One simple way to do that is inspired by burn-in in sampling, under the guidance of an in-domain data set, C_{in} as prior. At the beginning, we train $P_t(\tilde{e} \mid \tilde{f}, D_1)$ and $P_t(\tilde{f} \mid \tilde{e}, D_1)$ for all phrases learned from C_{in}. We also train $P_t(\tilde{e} \mid \tilde{f})$ and $P_t(\tilde{f} \mid \tilde{e})$ for all phrases learned from C_{mix}. During burn-in we assume that the out-domain phrase-based models are the domain-confused phrase-based models, i.e., $P_t(\tilde{e} \mid \tilde{f}, D_0) \approx P_t(\tilde{e} \mid \tilde{f})$ and $P_t(\tilde{f} \mid \tilde{e}, D_0) \approx P_t(\tilde{f} \mid \tilde{e})$. We isolate all the LMs and the prior models from our model, and apply a single EM iteration to update $P(D \mid \mathbf{e}, \mathbf{f})$ based on those domain-focused models $P_t(\tilde{e} \mid \tilde{f}, D)$ and $P_t(\tilde{f} \mid \tilde{e}, D)$.

In the end, we use $P(D \mid \mathbf{e}, \mathbf{f})$ to fill in the values of hidden variable D in C_{mix}, so it provides us with an initialization for $P(D \mid \tilde{e}, \tilde{f})$. Subsequently, we also rank sentence pairs in $\mathcal{C}_{mix}$ with $P(D_1 \mid \mathbf{e}, \mathbf{f})$ and select a subset of smallest scoring pairs as a *pseudo out-domain subset* to train $P_{lm}(\mathbf{e} \mid D_0)$ and $P_{lm}(\mathbf{f} \mid D_0)$. Once the latent domain-focused LMs have been trained, the LM probabilities stay *fixed* during EM. Crucially, it is important to scale the probabilities of the four LMs to make them comparable: we normalize the probability that a LM assigns to a sentence by the total probability this LM assigns to all sentences in $\mathcal{C}_{mix}$.

[9]The in-domain LMs $P_{lm}(\mathbf{e} \mid D_1)$ and $P_{lm}(\mathbf{f} \mid D_1)$ can be simply trained on the source and target sides of $\mathcal{C}_{in}$ respectively.

3 Intrinsic evaluation

We evaluate the ability of our model to retrieve "hidden" in-domain data in a large mix-domain corpus, i.e., we hide some in-domain data in a large mix-domain corpus. We weigh sentence pairs under our model with $P(D_1 \mid \tilde{e}, \tilde{f})$ and $P(D_1 \mid \mathbf{e}, \mathbf{f})$ respectively. We report *pseudo*-precision/recall at the *sentence-level* using a range of cut-off criteria for selecting the top scoring instances in the mix-domain corpus. A good relevance model expects to score higher for the hidden in-domain data.

Baselines Two standard perplexity-based selection models in the literature have been implemented as the baselines: cross-entropy difference (Moore and Lewis, 2010) and bilingual cross-entropy difference (Axelrod et al., 2011), investigating their ability to retrieve the hiding data as well. Training them over the data to learn the sentences with their relevance, we then rank the sentences to select top of pairs to evaluate the *pseudo*-precision/recall at the *sentence-level*.

Results We use a mix-domain corpus $\mathcal{C}_g$ of 770K sentence pairs of different genres.[10] There is also a Legal corpus of 183K pairs that serves as the in-domain data. We create $\mathcal{C}_{mix}$ by selecting an arbitrary 83K pairs of in-domain pairs and adding them to $\mathcal{C}_g$ (the hidden in-domain data); we use the remaining 100k in-domain pairs as $\mathcal{C}_{in}$.

To train the baselines, we construct interpolated 4-gram Kneser-Ney LMs using BerkeleyLM (Pauls and Klein, 2011). Training our model on the data takes six EM-iterations to converge.[11]

[10]Count of *sentence pairs*: European Parliament (Koehn, 2005): 183,793; Pharmaceuticals: 190,443, Software: 196,168, Hardware: 196,501.

[11]After the fifth EM iteration we do not observe any significant increase in the likelihood of the data. Note that we use the same setting as for the baselines to train the latent domain-focused LMs for use in our model – interpolated 4-gram Kneser-Ney LMs using BerkeleyLM. This training setting is used for all experiments in this work.

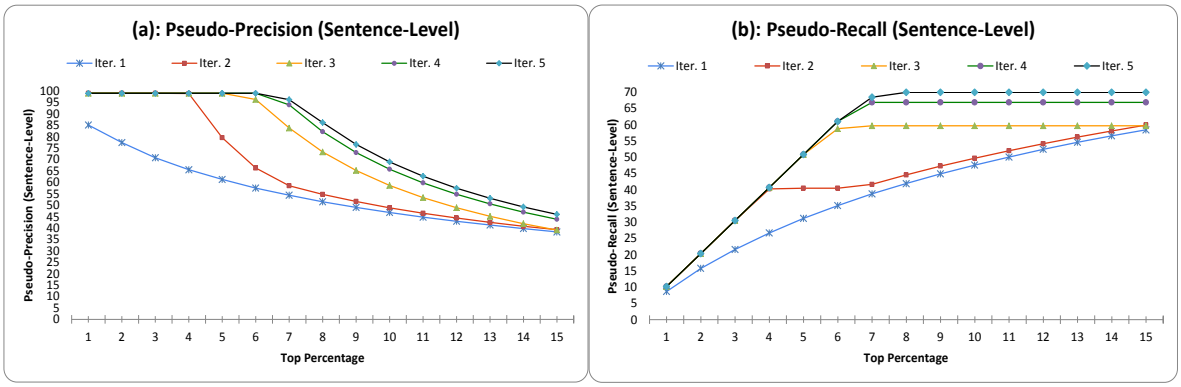

Figure 2: Intrinsic evaluation.

Fig. 2 helps us examine how the pseudo sentence invitation are done during each EM iteration. For later iterations we observe a better pseudo-precision and pseudo-recall at sentence-level (Fig. 2(a), Fig. 2(b)). Fig. 2 also reveals a good learning capacity of our learning framework. Nevertheless, we observe that the baselines do not work well for this task. This is not new, as pointed out in our previous work (Cuong and Sima'an, 2014).

Which component type contributes more to the performance, the latent domain language models or the latent domain translation models? Further experiments have been carried on to neutralize each component type in turn and build a selection system with the rest of our model parameters. It turns out that the latent domain translation models are crucial for performance for the learning framework, while the latent domain LMs make a far smaller yet substantial contribution. We refer readers to our previous work (Cuong and Sima'an, 2014), which provides detail analysis of the data selection problem.

4 Translation experiments: Setting

Data We use a mix-domain corpus consisting of $4M$ sentence pairs, collected from multiple resources including EuroParl (Koehn, 2005), Common Crawl Corpus, UN Corpus, News Commentary. As in-domain corpus we use "Consumer and Industrial Electronics" manually collected by Translation Automation Society (TAUS.com). The corpus statistics are summarized in Table 1.

System We train a standard state-of-the-art phrase-based system, using it as the baseline.[12] There are three main kinds of features for the translation model in the baseline - phrase-based translation features, lexical weights (Koehn et al., 2003) and lexicalized reordering features (Koehn et al., 2005).[13] Other features include the penalties for word, phrase and distance-based reordering.

			English	Spanish
	$\mathbf{C_{mix}}$	Sents	**4M**	
		Words	$113.7M$	127.1M
Domain: Electronics	$\mathbf{C_{in}}$	Sents	**109K**	
		Words	$1,485,558$	$1,685,716$
	Dev	Sents	984	
		Words	13130	$14,955$
	Test	Sents	982	
		Words	$13,493$	$15,392$

Table 1: The data preparation.

The mix-domain corpus is word-aligned using GIZA++ (Och and Ney, 2003) and symmetrized with *grow(-diag)-final-and* (Koehn et al., 2003). We limit phrase length to a maximum of seven words. The LMs are interpolated 4-grams with Kneser-Ney, trained on $2.2M$ English sentences from Europarl augmented with $248.8K$ sentences from News Commentary Corpus (WMT 2013). We tune the system using k-best batch MIRA (Cherry and Foster, 2012). Finally, we use Moses

[12] We use Stanford Phrasal - a standard state-of-the-art phrase-based translation system developed by Cer et al. (2010).

[13] The lexical weights and the lexical reordering features will be described in more detail in Section 6.

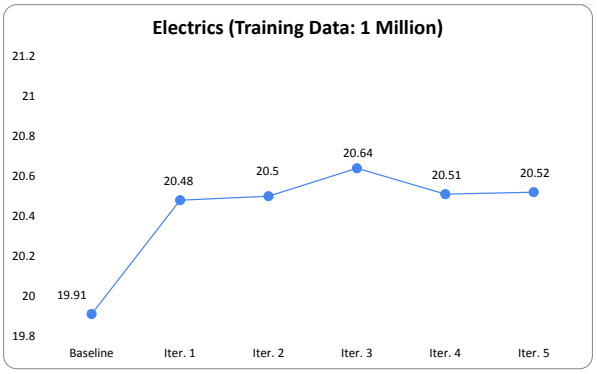

Figure 3: BLEU averaged over multiple runs.

(Koehn et al., 2007) as decoder.[14]

We report BLEU (Papineni et al., 2002), METEOR 1.4 (Denkowski and Lavie, 2011) and TER (Snover et al., 2006), with statistical significance at 95% confidence interval under paired bootstrap re-sampling (Press et al., 1992). For every system reported, we run the optimizer at least three times, before running MultEval (Clark et al., 2011) for resampling and significance testing.

Outlook In Section 5 we examine the effect of training only the latent domain-focused phrase table using our model. In Section 6 we proceed further to estimate also latent domain-focused lexical weights and lexicalized reordering models, examining how they incrementally improve the translation as well.

5 Adapting phrase table only

Here we investigate the effect of adapting the phrase table only; we will delay adapting the lexical weights and lexicalized reordering features to Section 6. We build a phrase-based system with the usual features as the baseline, including two bi-directional phrase-based models, plus the penalties for word, phrase and distortion. We also build a latent domain-focused phrase-based system with the two bi-directional latent phrase-based models, and the standard penalties described above.

We explore training data sizes $1M$, $2M$ and $4M$ sentence pairs. Three baselines are trained yielding $95.77M$, $176.29M$ and $323.88M$ phrases respectively. We run 5 EM iterations to train our learning framework. We use the parameter estimates for $P(D \mid \tilde{e}, \tilde{f})$ derived at each EM iteration to train our latent domain-focused phrase-based systems. Fig. 3 presents the results (in BLEU) at each iteration in detail for the case of $1M$ sentence pairs. Similar improvements are observed for METEOR and TER. Here, we consistently observe improvements at p-value $= 0.0001$ for all cases.

It should be noted that when doubling the training data to $2M$ and $4M$, we observe the similar results.

Finally, for all cases we report their best result in Table 2. Here, note how the improvement could be gained when doubling the training data.

Data	System	Avg	Δ	p-value
$1M$	Baseline	19.91	–	–
	Our System	20.64	**+0.73**	**0.0001**
$2M$	Baseline	20.54	–	–
	Our System	21.41	**+0.87**	**0.0001**
$4M$	Baseline	21.44	–	–
	Our System	22.62	**+1.18**	**0.0001**

Table 2: BLEU averaged over multiple runs.

It is also interesting to consider the average entropy of phrase table entries in the domain-confused systems, i.e.,

$$\frac{-\sum_{\langle\tilde{e},\tilde{f}\rangle} p_t(\tilde{e}|\tilde{f}) \log p_t(\tilde{e}|\tilde{f})}{\text{number of phrases}\langle\tilde{e},\tilde{f}\rangle}$$

against that in the domain-focused systems

$$\frac{-\sum_{\langle\tilde{e},\tilde{f}\rangle} p_t(\tilde{e}|\tilde{f}, D_1) \log p_t(\tilde{e}|\tilde{f}, D_1)}{\text{number of phrases}\langle\tilde{e},\tilde{f}\rangle}.$$

Following (Hasler et al., 2014) in Table 3 we also show that the entropy decreases significantly in

[14] While we implement the latent domain phrase-based models using Phrasal for some advantages, we prefer to use Moses for decoding.

the adapted tables in all cases, which indicates that the distributions over translations of phrases have become sharper.

Baseline	Iter. 1	Iter. 2	Iter. 3	Iter. 4	Iter. 5
0.210	0.187	0.186	0.185	0.185	0.184

Table 3: Average entropy of distributions.

In practice, the third iteration systems usually produce best translations. This is somewhat expected because as EM invites more pseudo in-domain pairs in later iterations, it sharpens the estimates of $P(D_1 \mid \tilde{e}, \tilde{f})$, making pseudo out-domain pairs tend to 0.0. Table 4 shows the percentage of entries with $P(D_1 \mid \tilde{e}, \tilde{f}) < 0.01$ at every iteration, e.g., 34.52% at the fifth iteration. This induced schism in C_{mix} diminishes the difference between the relevance scores for certain sentence pairs, limiting the ability of the latent phrase-based models to further discriminate in the gray zone.

Entries	$P(D_1 \mid \tilde{f}, \tilde{e}) < 0.01$
Iter. 1	22.82%
Iter. 2	27.06%
Iter. 3	30.07%
Iter. 4	32.47%
Iter. 5	34.52%

Table 4: Phrase analyses.

Finally, to give a sense of the improvement in translation, we (randomly) select cases where the systems produce different translations and present some of them in Table 5. These examples are indeed illuminating, e.g., "*can reproduce signs of audio*"/"*can play signals audio*", "*password teacher*"/"*password master*", revealing thoroughly the benefit derived from adapting the phrase models from being domain-confused to being domain-focused. Table 6 presents phrase table entries, i.e., $p_t(e \mid f)$ and $p_t(e \mid f, D_1)$, for the "*can reproduce signs of audio*"/"*can play signals audio*" example.

6 Fully adapted translation model

The preceding experiments reveal that adapting the phrase tables significantly improves translation performance. Now we also adapt the lexical and reordering components. The result is a fully adapted, domain-focused, phrase-based system.

	señales		**reproducir**	
Entries	**signals**	**signs**	**play**	**reproduce**
Baseline	0.29	0.36	0.15	0.20
Iter. 1	0.36	0.23	0.29	0.16
Iter. 2	0.37	0.19	0.32	0.17
Iter. 3	0.37	0.17	0.34	0.16
Iter. 4	0.37	0.16	0.36	0.16
Iter. 5	0.37	0.15	0.37	0.16

Table 6: Phrase entry examples.

Briefly, the lexical weights provide smooth estimates for the phrase pair based on word translation scores $P(e \mid f)$ between pairs of words $\langle e, f \rangle$, i.e., $P(e \mid f) = \frac{c(e,f)}{\sum_e c(e,f)}$ (Koehn et al., 2003). Our latent domain-focused lexical weights, on the other hand, are estimated according to $P(e \mid f,\ D_1)$, i.e., $P(e \mid f,\ D_1) = \frac{P(e \mid f)P(D_1 \mid e,\ f)}{\sum_f P(e \mid f)P(D_1 \mid e,\ f)}$.

The lexicalized reordering models with orientation variable O, $P(O \mid \tilde{e},\ \tilde{f})$, model how likely a phrase $\langle \tilde{e}, \tilde{f} \rangle$ directly follows a previous phrase ($monotone$), swaps positions with it ($swap$), or is not adjacent to it ($discontinous$) (Koehn et al., 2005). We make these domain-focused:

$$P(O \mid \tilde{e},\ \tilde{f}, D_1) = \frac{P(O \mid \tilde{e},\ \tilde{f})P(D_1 \mid O, \tilde{e},\ \tilde{f})}{\sum_O P(O \mid \tilde{e},\ \tilde{f})P(D_1 \mid O, \tilde{e},\ \tilde{f})} \quad (11)$$

Estimating $P(D_1 \mid O,\ \tilde{e},\ \tilde{f})$ and $P(D_1 \mid e,\ f)$ is similar to estimating $P(D_1 \mid \tilde{e},\ \tilde{f})$ and hinges on the estimates of $P(D_1 \mid \mathbf{e},\ \mathbf{f})$ during EM.

The baseline for the following experiments is a standard state-of-the-art phrase-based system, including two bi-directional phrase-based translation features, two bi-directional lexical weights, six lexicalized reordering features, as well as the penalties for word, phrase and distortion. We develop three kinds of domain-adapted systems that are different at their adaptation level to fit the task. The first (**Sys. 1**) adapts only the phrase-based models, using the same lexical weights, lexicalized reordering models and other penalties as the baseline. The second (**Sys. 2**) adapts also the lexical weights, fixing all other features as the baseline. The third (**Sys. 3**) adapts both the phrase-based models, lexical weights and lexicalized re-

	Translation Examples
Input	*El reproductor puede reproducir señales de audio grabadas en mix-mode cd, cd-g, cd-extra y cd text.*
Reference	*The player **can play back audio signals** recorded in mix-mode cd, cd-g, cd-extra and cd text.*
Baseline	*The player **can reproduce signs of audio** recorded in mix-mode cd, cd-g, cd-extra and cd text.*
Our System	*The player **can play signals audio** recorded in mix-mode cd, cd-g, cd-extra and cd text.*
Input	*Se puede crear un archivo autodescodificable cuando el archivo codificado se abre con la contraseña maestra.*
Reference	*A self-decrypting file can be created when the encrypted file is **opened with** the **master password**.*
Baseline	*To create an file autodescodificable when the file codified **commenced with** the **password teacher**.*
Our System	*You can create an archive autodescodificable when the file codified **opens with** the **password master**.*
Input	*Repite todas las pistas (únicamente cds de vídeo sin pbc)*
Reference	*Repeat **all tracks** (non-pbc video cds only)*
Baseline	*Repeated **all avenues** (only cds video without pbc)*
Our System	*Repeated **all the tracks** (only cds video without pbc)*

Table 5: Translation examples yielded by a domain-confused phrase-based system (**the baseline**) and a domain-focused phrase-based system (**our system**).

ordering models[15], fixing other penalties as the baseline.

Metric	System	Avg	Δ	p-value
Consumer and Industrial Electronics				
(In-domain: $109K$ pairs; Dev: 982 pairs; Test: 984 pairs)				
BLEU	Baseline	22.9	–	–
	Sys. 1	23.4	+0.5	0.008
	Sys. 2	23.9	+1.0	0.0001
	Sys. 3	**24.0**	**+1.1**	**0.0001**
METEOR	Baseline	30.0	–	–
	Sys. 1	30.4	+0.4	0.0001
	Sys. 2	30.8	+0.8	0.0001
	Sys. 3	**30.9**	**+0.9**	**0.0001**
TER	Baseline	59.5	–	–
	Sys. 1	58.8	-0.7	0.0001
	Sys. 2	58.0	-1.5	0.0001
	Sys. 3	**57.9**	**-1.6**	**0.0001**

Table 7: Metric scores for the systems, which are averages over multiple runs.

Table 7 presents results for training data size of $4M$ parallel sentences. It shows that the fully domain-focused system (**Sys. 3**) significantly improves over the baseline. The table also shows that the latent domain-focused phrase-based models and lexical weights are crucial for the improved performance, whereas adapting the reordering models makes a far smaller contribution.

Finally we also apply our approach to other tasks where the relation between their in-domain data and the mix-domain data varies substantially. Table 8 presents their in-domain, tuning and test data in detail, as well as the translation results over them. It shows that the fully domain-focused systems consistently and significantly improve the translation accuracy for all the tasks.

7 Combining multiple models

Finally, we proceed further to test our latent domain-focused phrase-based translation model on standard domain adaptation. We conduct experiments on the task "Professional & Business Services" as an example.[16] For standard adaptation we follow (Koehn and Schroeder, 2007) where we pass multiple phrase tables directly to the Moses decoder and tune them together. For baseline we combine the standard phrase-based system trained on C_{mix} with the one trained on the in-domain data C_{in}. We also combine our latent domain-focused phrase-based system with the one trained on C_{in}. Table 9 presents the results showing that combining our domain-focused system adapted from C_{mix} with the in-domain model outperforms the baseline.

[15]We run three EM iterations to train our invitation framework, and then use the parameter estimates for $P(D_1 \mid \tilde{e}, \tilde{f})$, $P(D_1 \mid e, f)$ and $P(D_1 \mid O, \tilde{e}, \tilde{f})$ to train these domain-focused features. We adopt this training setting for all other different tasks in the sequel.

[16]We choose this task for additional experiments because it has very small in-domain data ($23K$). This is supposed to make adaptation difficult because of the robust large-scale systems trained on C_{mix}.

Metric	System	Avg	Δ	p-value
Professional & Business Services				
(In-domain: $23K$ pairs; Dev: $1,000$ pairs; Test: 998 pairs)				
BLEU	Baseline	22.0	–	–
	Our System	23.1	**+1.1**	**0.0001**
METEOR	Baseline	30.8	–	–
	Our System	31.4	**+0.6**	**0.0001**
TER	Baseline	58.0	–	–
	Our System	56.6	**-1.4**	**0.0001**
Financials				
(In-domain: $31K$ pairs; Dev: $1,000$ pairs; Test: $1,000$ pairs)				
BLEU	Baseline	31.1	–	–
	Our System	31.8	**+0.7**	**0.0001**
METEOR	Baseline	36.3	–	–
	Our System	36.6	**+0.3**	**0.0001**
TER	Baseline	48.8	–	–
	Our System	48.3	**-0.5**	**0.0001**
Computer Hardware				
(In-domain: $52K$ pairs; Dev: $1,021$ pairs; Test: $1,054$ pairs)				
BLEU	Baseline	24.6	–	–
	Our System	25.3	**+0.7**	**0.0001**
METEOR	Baseline	32.4	–	–
	Our System	33.1	**+0.7**	**0.0001**
TER	Baseline	56.4	–	–
	Our System	55.0	**-1.4**	**0.0001**
Computer Software				
(In-domain: $65K$ pairs; Dev: $1,100$ pairs; Test: $1,000$ pairs)				
BLEU	Baseline	27.4	–	–
	Our System	28.3	**+0.9**	**0.0001**
METEOR	Baseline	34.0	–	–
	Our System	34.7	**+0.7**	**0.0001**
TER	Baseline	51.7	–	–
	Our System	50.6	**-1.1**	**0.0001**
Pharmaceuticals & Biotechnology				
(In-domain: $85K$ pairs; Dev: 920 pairs; Test: $1,000$ pairs)				
BLEU	Baseline	31.6	–	–
	Our System	32.4	**+0.8**	**0.0001**
METEOR	Baseline	34.0	–	–
	Our System	34.4	**+0.4**	**0.0001**
TER	Baseline	51.4	–	–
	Our System	50.6	**-0.8**	**0.0001**

Table 8: Metric scores for the systems, which are averages over multiple runs.

Metric	System	Avg	Δ	p-value
Professional & Business Services				
(In-domain: $23K$ pairs; Dev: $1,000$ pairs; Test: 998 pairs)				
BLEU	In-domain	46.5	–	–
	+ Mix-domain	46.6	–	–
	+ Our system	47.9	**+1.3**	**0.0001**
METEOR	In-domain	39.8	–	–
	+ Mix-domain	40.1	–	–
	+ Our System	41.1	**+1.0**	**0.0001**
TER	In-domain	38.2	–	–
	+ Mix-domain	38.0	–	–
	+ Our System	36.9	**-1.1**	**0.0001**

Table 9: Domain adaptation experiments. Metric scores for the systems, which are averages over multiple runs.

8 Related work

A distantly related, but clearly complementary, line of research focuses on the role of document topics (Eidelman et al., 2012; Zhang et al., 2014; Hasler et al., 2014). An off-the-shelf Latent Dirichlet Allocation tool is usually used to infer document-topic distributions. On one hand, this setting may not require in-domain data as prior. On the other hand, it requires meta-information (e.g., document information).

Part of this work (the latent sentence-relevance models) relates to data selection (Moore and Lewis, 2010; Axelrod et al., 2011), where sentence-relevance weights are used for hard-filtering rather than weighting. The idea of using sentence-relevance estimates for phrase-relevance estimates relates to Matsoukas et al. (2009) who estimate the former using meta-information over documents as main features. In contrast, our work overcomes the mutual dependence of sentence and phrase estimates on one another by training both models in tandem.

Adaptation using small in-domain data has a different but complementary goal to another line of research aiming at combining a domain-adapted system with the another trained on the in-domain data (Koehn and Schroeder, 2007; Bisazza et al., 2011; Sennrich, 2012; Razmara et al., 2012; Sennrich et al., 2013). Our work is somewhat related to, but markedly different from, phrase pair weighting (Foster et al., 2010). Finally, our latent domain-focused phrase-based models and invitation training paradigm can be seen to shift attention from adaptation to making explicit the role of domain-focused models in SMT.

9 Conclusion

We present a novel approach for in-domain focused training of a phrase-based system on a mix-of-domain corpus by using prior distributions from a small in-domain corpus. We derive an EM training algorithm for learning latent domain relevance models for the phrase- and sentence-levels in tandem. We also show how to overcome the difficulty of lack of explicit out-domain data by bootstrapping pseudo out-domain data.

In future work, we plan to explore generative Bayesian models as well as discriminative learning approaches with different ways for estimat-

ing the latent domain relevance models. We hypothesize that bilingual, but also monolingual, relevance models can be key to improved performance.

Acknowledgements

We thank Ivan Titov for stimulating discussions, and three anonymous reviewers for their comments on earlier versions. The first author is supported by the EXPERT (EXPloiting Empirical appRoaches to Translation) Initial Training Network (ITN) of the European Union's Seventh Framework Programme. The second author is supported by VICI grant nr. 277-89-002 from the Netherlands Organization for Scientific Research (NWO). We thank TAUS for providing us with suitable data.

References

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

Arianna Bisazza, Nick Ruiz, and Marcello Federico. 2011. Fill-up versus interpolation methods for phrase-based smt adaptation. In *IWSLT*, pages 136–143.

Daniel Cer, Michel Galley, Daniel Jurafsky, and Christopher D. Manning. 2010. Phrasal: A toolkit for statistical machine translation with facilities for extraction and incorporation of arbitrary model features. In *Proceedings of the NAACL HLT 2010 Demonstration Session*, HLT-DEMO '10, pages 9–12, Stroudsburg, PA, USA. Association for Computational Linguistics.

Colin Cherry and George Foster. 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*, NAACL HLT '12, pages 427–436, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 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: Short Papers - Volume 2*, HLT '11, pages 176–181, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hoang Cuong and Khalil Sima'an. 2014. Latent domain translation models in mix-of-domains haystack. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1928–1939, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 85–91, Stroudsburg, PA, USA. Association for Computational Linguistics.

Vladimir Eidelman, Jordan Boyd-Graber, and Philip Resnik. 2012. Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 115–119, Stroudsburg, PA, USA. Association for Computational Linguistics.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 451–459, Stroudsburg, PA, USA. Association for Computational Linguistics.

Guillem Gascó, Martha-Alicia Rocha, Germán Sanchis-Trilles, Jesús Andrés-Ferrer, and Francisco Casacuberta. 2012. Does more data always yield better translations? In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 152–161, Stroudsburg, PA, USA. Association for Computational Linguistics.

Eva Hasler, Phil Blunsom, Philipp Koehn, and Barry Haddow. 2014. Dynamic topic adaptation for phrase-based mt. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 328–337, Gothenburg, Sweden, April. Association for Computational Linguistics.

Ann Irvine, John Morgan, Marine Carpuat, Daume Hal III, and Dragos Munteanu. 2013. Measuring machine translation errors in new domains. pages 429–440.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 224–227, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn, Amittai Axelrod, Alexandra Birch, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *International Workshop on Spoken Language Translation*.

Philipp Koehn, 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 *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT, AAMT.

Spyros Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 708–717, Stroudsburg, PA, USA. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 220–224, Stroudsburg, PA, USA. Association for Computational Linguistics.

Radford M. Neal and Geoffrey E. Hinton. 1999. Learning in graphical models. chapter A View of the EM Algorithm That Justifies Incremental, Sparse, and Other Variants, pages 355–368. MIT Press, Cambridge, MA, USA.

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

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Comput. Linguist.*, 30(4):417–449, December.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Adam Pauls and Dan Klein. 2011. Faster and smaller n-gram language models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 258–267, Stroudsburg, PA, USA. Association for Computational Linguistics.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 1992. *Numerical Recipes in C (2Nd Ed.): The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA.

Majid Razmara, George Foster, Baskaran Sankaran, and Anoop Sarkar. 2012. Mixing multiple translation models in statistical machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 940–949, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rico Sennrich, Holger Schwenk, and Walid Aransa. 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, August. Association for Computational Linguistics.

Rico Sennrich. 2012. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 539–549, Stroudsburg, PA, USA. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Min Zhang, Xinyan Xiao, Deyi Xiong, and Qun Liu. 2014. Topic-based dissimilarity and sensitivity models for translation rule selection. *Journal of Artificial Intelligence Research*, 50(1):1–30.