Latent Domain Word Alignment for Heterogeneous Corpora

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

This work focuses on the insensitivity of existing word alignment models to domain differences, which often yields suboptimal results on large heterogeneous data. A novel latent domain word alignment model is proposed, which induces domain-conditioned lexical and alignment statistics. We propose to train the model on a heterogeneous corpus under partial supervision, using a small number of seed samples from different domains. The seed samples allow estimating sharper, domain-conditioned word alignment statistics for sentence pairs. Our experiments show that the derived domain-conditioned statistics, once combined together, produce notable improvements both in word alignment accuracy and in translation accuracy of their resulting SMT systems.

1 Introduction

Word alignment currently constitutes the basis for phrase extraction and reordering in phrase-based systems, and its statistics provide lexical parameters used for smoothing the phrase pair estimates. For over two decades since IBM models (Brown et al., 1993) and the HMM alignment model (Vogel et al., 1996), word alignment remains an active research line, e.g., see recent work (Simion et al., 2013; Tamura et al., 2014; Chang et al., 2014).

During the past years we witnessed an increasing need to collect and use large *heterogeneous* parallel corpora from different domains and sources, e.g., News, Wikipedia, Parliament Proceedings. It is tacitly assumed that assembling a larger corpus should improve a phrase-based system coverage and performance. Recent work (Sennrich et al., 2013; Carpuat et al., 2014; Cuong and Sima'an, 2014b; Kirchhoff and Bilmes, 2014; Cuong and Sima'an, 2014a) shows that this is not necessarily true as phrase translations as well as (bi- and monolingual) word co-occurrence statistics could differ across domains. This suggests that the word alignment quality obtained from IBM and HMM alignment models might also be affected in heterogeneous corpora.

Intuitively, in heterogeneous data certain words are present across many domains, whereas others are more specific to few domains. This suggests that the translation probabilities for words will be as fractioned as the diversity of its translations across the domains. Furthermore, because the IBM and HMM alignment models use *context-insensitive* conditional probabilities, in heterogeneous corpora the estimates of these probabilities will be aggregated over different domains. Both issues could lead to suboptimal word alignment quality.

Surprisingly, the *insensitivity* of the existing IBM and HMM alignment models to domain differences has not received much attention thus far (see the study of Bach et al. (2008) and Gao et al. (2011) for reference in the literature). We conjecture that this is because it is not fully clear how to define what constitutes a *(sub)-domain*. In this paper we propose to exploit the contrast between the alignment statistics in a handful of *seed samples from different domains* in order to induce domain-conditioned probabilities for each sentence pair in the heterogeneous corpus. Crucially, some sentence pairs will be more similar to a seed domain than others, whereas some sentence

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 398–408,
Denver, Colorado, May 31 – June 5, 2015.

pairs might be dissimilar to all seed domains. The number and choice of seed domains depends largely on the available resources but intuitively these seed domains are chosen to be relevant to parts of the heterogeneous corpus. A small number of such seeds can be expected to notably improve word alignment accuracy. In fact, a single seed sample already allows us to exploit the contrast between two parts in the corpus: similar or dissimilar to the seed data.

Considering the small seed samples as *partial supervision*, in this paper we explore the question: *how to obtain better word alignment in a heterogeneous, mix-of-domains corpus?* We present a novel *latent domain HMM alignment model*, which aims to *tighten* the probability estimates of the generative alignment process of a sentence pair, and of the probability estimates of the sentence pair itself for a specific domain. We also present an accompanying training regime *guided* by partial supervision using the seed samples, exploiting the contrast between the domain-conditioned alignment statistics in these samples. This way we aim for an alignment model that is more domain-sensitive than the original HMM alignment model. Once the domain-conditioned statistics are induced, we discuss how to combine them together to express the probability of a sentence pair as a mixture over specific domains.

Finally, we report experimental results over heterogeneous corpora of 1M, 2M and $4M$ sentence pairs, where we are provided domain information for different samples of 10%, 5% and 2.5% of the heterogeneous data respectively. A large number of experiments are reported, showing that the latent domain HMM model produces notable improvements in word alignment accuracy over the original HMM alignment model. Furthermore, the translation accuracy of the resulting SMT systems is significantly improved across *four* different translation tasks.

2 HMM Alignment Model

In this section, we briefly review the HMM alignment model (Vogel et al., 1996). The generative story of the model is shown in Figure 1. The latent states take values from the target language words and generate source language words.

Formally, we use $\mathbf{e} = (e_1, \ldots, e_I)$ to denote the target sentence with length I and $\mathbf{f} = (f_1, \ldots, f_J)$ to denote the source sentence with length J. For an alignment $\mathbf{a} = (a_1, \ldots, a_J)$ of a sentence pair $\langle \mathbf{e}, \mathbf{f} \rangle$, the model factors $P(\mathbf{f}, \mathbf{a}|\ \mathbf{e})$ into the word translation and transition probabilities:

$$P(\mathbf{f}, \mathbf{a}|\ \mathbf{e}) = \prod_{j=1}^{J} P(f_j|\ e_{a_j}) P(a_j|\ a_{j-1}). \quad (1)$$

Here, $P(f_j|\ e_{a_j})$ represents the word translation probabilities and $P(a_j|\ a_{j-1})$[1] represents the transition probabilities between positions. Note that $P(a_j|\ a_{j-1})$ depends only on the distance $(a_j - a_{j-1})$. Note also that the first-order dependency model is an extension of the uniform dependency model and zero-order dependency model of IBM models 1 and 2, respectively.

In this work, we model explicitly distances in the range ± 5. Note that *null*-links are also explicitly added in our implementation, following Och and Ney (2003) and Graca et al. (2010).

Once the HMM alignment model is trained, the most probable alignment, $\hat{\mathbf{a}}$ for each sentence pair can be computed by: $\hat{\mathbf{a}} = \text{argmax}_{\mathbf{a}} P(\mathbf{f}, \mathbf{a}|\ \mathbf{e})$. Here, the search problem can be solved by the Viterbi algorithm.

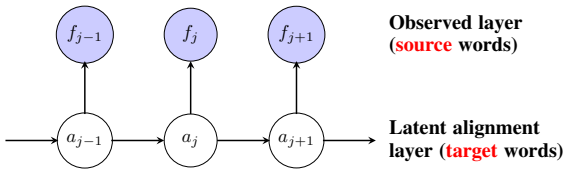

Figure 1: HMM alignment model with observed and latent alignment layers.

3 Latent Domain HMM Alignment Model

Because the heterogeneous data contains a mix of diverse domains, the induced statistics derived from word alignment models reflect translation preferences aggregated over these domains. In this sense, they can be considered domain-*confused* statistics (Cuong and Sima'an, 2014a). This work thus focuses on more **representative** statistics: the domain-*conditioned* word alignment statistics, i.e., the statistics with respect to each of the diverse domains.

By introducing a latent variable D representing domains of the heterogeneous data, we aim

[1] The "full" formula for transition probabilities would be $P(a_j|\ a_{j-1}, I)$. For convenience, we ignore I in our presentation.

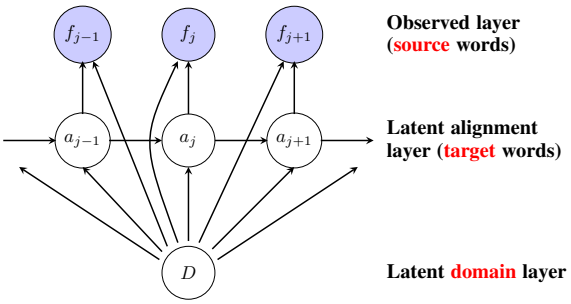

Figure 2: Latent domain HMM alignment model. An additional latent layer representing domains has been conditioned on by both the rest two layers.

to learn the D-conditioned word alignment model $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D)$.[2] Relying on the HMM alignment model, our latent domain HMM alignment model factors $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D)$ into the domain-conditioned word translation and transition probabilities:

$$P(\mathbf{f}, \mathbf{a}|\mathbf{e}, D) = \prod_{j=1}^{J} P(f_j|e_{a_j}, D)P(a_j|a_{j-1}, D). \tag{2}$$

The generative story of the model is shown in Figure 2. Note how domain-conditioned alignment statistics, $P(\cdot|\ \cdot, D)$ contain their former domain-confused alignment statistics, $P(\cdot|\ \cdot)$ as special case

$$P(f_j|e_{a_j}, D) = \frac{P(f_j|e_{a_j})P(D|f_j, e_{a_j})}{\sum_f P(f_j|e_{a_j})P(D|f_j, e_{a_j})}, \tag{3}$$

$$P(a_j|a_{j-1}, D) = \frac{P(a_j|a_{j-1})P(D|a_j, a_{j-1})}{\sum_{a_j} P(a_j|a_{j-1})P(D|a_j, a_{j-1})}. \tag{4}$$

With an additional latent domain layer, it becomes crucial to train the model in an efficient way. As suggested by Eq. 3 and 4, we could simplify training by breaking up the estimation process into two steps. That is, we train alignment parameters, $P(\cdot|\ \cdot)$ or domain parameters, $P(D|\ \cdot,\ \cdot)$ first, hold them fixed before training the other kind of the parameters.[3] Instead, in this work we design an algorithm that trains both of them simultaneously via training domain-conditioned parameters $P(\cdot|\ ,\cdot,\ D)$ directly.

[2]Note that $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D)$ contains their former $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e})$ as special case, i.e., $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D) = \frac{P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e})P(D|\ \mathbf{f},\ \mathbf{a},\ \mathbf{e})}{\sum_{\mathbf{f}}\sum_{\mathbf{a}} P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e})P(D|\ \mathbf{f},\ \mathbf{a},\ \mathbf{e})}$.

[3]This training scheme is in fact applied in the work of Cuong and Sima'an (2014a), however, for a different purpose.

3.1 Training

Basically, our model can be viewed as having a set, Θ of N subsets of domain-conditioned parameters, Θ_D for N different domains, i.e., $\Theta = \{\Theta_{D_1},\ \ldots,\ \Theta_{D_N}\}$. In this work, to simplify the learning problem we assume that the domains are very *different* from each other. If this assumption does not hold, the learning problem would shift from *single-label* learning to *multiple-label* learning. We leave this extension for future work.

Our training procedure seeks the parameters Θ that maximize the log-likelihood, $\mathcal{L}$ of the data: $\mathcal{L} = \sum_{\langle \mathbf{f},\ \mathbf{e}\rangle} \log \sum_D \sum_{\mathbf{a}} P_{\Theta_D}(\mathbf{f},\ \mathbf{e},\ D,\ \mathbf{a})$. There, however, does not exist a closed-form solution for maximizing $\mathcal{L}$, and EM comes as an alternative solution to fit the model. EM maximizes $\mathcal{L}$ via block-coordinate ascent on a "free energy" lower bound $\mathcal{F}(q,\ \Theta)$ (Neal and Hinton, 1999), using an auxiliary distribution q over both the latent variables: $\mathcal{F}(q,\ \Theta) = \sum_{\langle \mathbf{f},\ \mathbf{e}\rangle} \sum_D \sum_{\mathbf{a}} q \log \frac{P_{\Theta_D}(\mathbf{a},\ D,\ \mathbf{f},\ \mathbf{e})}{q}$.

In the **E**-step of the EM algorithm, we fix Θ and aim to find the distribution q^* that maximizes $\mathcal{F}(q, \Theta)$ over the heterogeneous data. Simple mathematics lead to $\mathcal{F}(q,\ \Theta) = \sum_{\langle \mathbf{f},\ \mathbf{e}\rangle} \log P_\Theta(\mathbf{f},\ \mathbf{e}) - KL[q\ ||\ P_{\Theta_D}(\mathbf{a},\ D|\ \mathbf{f},\ \mathbf{e})]$, where $KL[\cdot\ ||\ \cdot]$ is the Kullback-Leiber divergence between two distributions. The distribution q^* can be thus derived as

$$\begin{aligned} q^* &= \text{argmax}_q\, \mathcal{F}(q,\ \Theta) \\ &= \text{argmin}_q\, KL[q\ ||\ P_{\Theta_D}(\mathbf{a},\ D|\ \mathbf{f},\ \mathbf{e})] \\ &= \frac{P_{\Theta_D}(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D)}{\sum_{\mathbf{a}} P_{\Theta_D}(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D)} P_{\Theta_D}(D|\ \mathbf{f},\ \mathbf{e}). \end{aligned}$$

Here, $P_{\Theta_D}(D|\ \mathbf{f},\ \mathbf{e})$ aims to exploit the contrast between the domain-sensitive alignment statistics. Assigning higher probability to one domain forces lower probability assignment to other domains.

Note that $P_{\Theta_D}(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D)$ is given in Eq. 2 and $\sum_{\mathbf{a}} P_{\Theta_D}(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ D)$ can be computed efficiently using dynamic programming.[4] Meanwhile, $P_{\Theta_D}(D|\ \mathbf{f},\ \mathbf{e})$ can be derived by Bayes' rule, i.e.,

$$P_{\Theta_D}(D|\ \mathbf{f},\ \mathbf{e}) \propto P_{\Theta_D}(\mathbf{f},\ \mathbf{e}|\ D) P_{\Theta_D}(D).$$

Here, the estimation of the domain prior parameters is easy, $P_{\Theta_D}(D) \propto \sum_{\langle \mathbf{f},\ \mathbf{e}\rangle} P_{\Theta_D}(D|\ \mathbf{f},\ \mathbf{e})$. The estimation of $P_{\Theta_D}(\mathbf{f},\ \mathbf{e}|\ D)$ raises a task of defining a

[4]Its time complexity is $\mathcal{O}(J \times I^2)$ for each sentence pair $\langle \mathbf{f},\ \mathbf{e}\rangle$ with their length J and I respectively.

E-step $\forall D \in \{D_1, \ldots, D_N\}$ do

$$c(D;\, \mathbf{f},\, \mathbf{e}) = P^{(c)}(D|\, \mathbf{f},\, \mathbf{e})$$

$$c(f|\, e;\, \mathbf{f},\, \mathbf{e},\, D) = P^{(c)}(D|\, \mathbf{f},\, \mathbf{e}) \sum\nolimits_{\mathbf{a}} P^{(c)}(\mathbf{a}|\, \mathbf{f},\, \mathbf{e},\, D) \sum\nolimits_{j=1}^{J} \delta(f, f_j) \sum\nolimits_{i=0}^{I} \delta(e,\, e_i)$$

$$c(i|\, i';\, \mathbf{f},\, \mathbf{e},\, D) = P^{(c)}(D|\, \mathbf{f},\, \mathbf{e}) \sum\nolimits_{\mathbf{a}} P^{(c)}(\mathbf{a}|\, \mathbf{f},\, \mathbf{e},\, D) \sum\nolimits_{j=1}^{J} \delta(a_j,\, i) \delta(a_{j-1},\, i')$$

M-step $\forall D \in \{D_1, \ldots, D_N\}$ do

$$P^{(+)}(f|e, D) = \frac{\sum_{\langle \mathbf{f},\mathbf{e} \rangle} c(f|e; \mathbf{f}, \mathbf{e}, D)}{\sum_f \sum_{\langle \mathbf{f},\mathbf{e} \rangle} c(f|e; \mathbf{f}, \mathbf{e}, D)} \; P^{(+)}(i|i', D) = \frac{\sum_{\langle \mathbf{f},\mathbf{e} \rangle} c(i|i'; \mathbf{f}, \mathbf{e}, D)}{\sum_i \sum_{\langle \mathbf{f},\mathbf{e} \rangle} c(i|i'; \mathbf{f}, \mathbf{e}, D)} \; P^{(+)}(D) \; = \frac{\sum_{\langle \mathbf{f},\mathbf{e} \rangle} c(D; \mathbf{f}, \mathbf{e})}{\sum_D \sum_{\langle \mathbf{f},\mathbf{e} \rangle} c(D; \mathbf{f}, \mathbf{e})}$$

Figure 3: Pseudocode for the training algorithm for the latent domain HMM alignment model. Note that notation $P^{(c)}$ denotes current iteration estimates, and $P^{(+)}$ denotes the re-estimates.

generative process for every sentence pair in the heterogeneous data with respect to a specific domain. Following (Cuong and Sima'an, 2014b), we factor it into two kinds of models in a symmetrized strategy: $P_{\Theta_D}(\mathbf{f},\, \mathbf{e}|\, D) \propto \big(P_{\Theta_D}(\mathbf{e}|\, D) P_{\Theta_D}(\mathbf{f}|\, \mathbf{e},\, D) + P_{\Theta_D}(\mathbf{f}|\, D) P_{\Theta_D}(\mathbf{e}|\, \mathbf{f},\, D)\big)$.

Basically, $P_{\Theta_D}(\cdot|\, \cdot,\, D)$ can be thought of as the domain-conditioned translation models, aiming to model how well a target/source sentence is generated over a source/target sentence with respect to a domain.[5] Meanwhile, $P_{\Theta_D}(\cdot|\, D)$ can be thought of as the domain-conditioned language models (LMs), aiming to model how fluent a source/target sentence with respect to a domain. For simplicity, once the domain-conditioned LMs are trained, they will stay *fixed* during training, i.e., LM probabilities are not parameters in our model.

In the **M**-step of the EM algorithm, we fix the derived q^* and aim to find the parameter set Θ^* that maximizes $\mathcal{F}(q, \Theta)$ over the data. This can be (easily) done by using q^* to softly fill in the values of $\mathbf{a}$ and D to estimate model parameters.

Pseudocode

In summary, the model has three kinds of parameters - word translation, word transition, and domain prior parameters. We now summarize the training via presenting the pseudocode.

First, we present expected count notations with respect to domains for the parameters. We use $c(f|\, e;\, \mathbf{f},\, \mathbf{e},\, D)$ to denote the expected counts that word e aligns to word f. We use $c(i|\, i';\, \mathbf{f},\, \mathbf{e},\, D)$ to denote the expected counts that two certain consecutive source words j and $j-1$ align to two target words i and i' respectively, i.e., j aligns to i and $j-1$ aligns to i'. Finally, we also use $c(D;\, \mathbf{f},\, \mathbf{e})$ to denote the expected count of domain priors. Note that all the expected counts are in the translation $(\mathbf{f}|\, \mathbf{e})$.

Figure 3 represents the pseudocode.

4 Learning with Partial Supervision

We now discuss remaining issues on how to guide the learning with partial supervision, i.e., how to use the given domain information of seed samples to guide the learning.

Number of Domains The values of $D \in [1..(N+1)]$ depends on the N available seed samples plus the so-called "out-domain," i.e., the part of the heterogeneous data that is dissimilar to all of the N sample domains.

Parameter Initialization We first discuss how to initialize the domain prior parameters. If a sentence pair $\langle \mathbf{f},\, \mathbf{e} \rangle$ belongs to a sample with a pre-specified domain D_i, we initialize $P(D_i|\, \mathbf{f},\, \mathbf{e})$ close to 1, and, $P(D_{i'}|\, \mathbf{f},\, \mathbf{e})$ close to 0 for other domains i', $i' \neq i$. Furthermore, we uniformly create the domain prior parameters for the rest of sentence pairs.

Uniform initialization for the domain-conditioned alignment parameters is also a reasonable option. Nevertheless, a more effective way is to make use of the domain-specific seed samples and the pool of the rest sentence pairs in the heterogeneous data.[6] That is, we train the model on each of the samples, assign-

[5] Note that $P_{\Theta_D}(\cdot|\, \cdot,\, D) = \sum_{\mathbf{a}} P_{\Theta_D}(\cdot,\, \mathbf{a}|\, \cdot,\, D)$ and it can be thus computed efficiently using dynamic programming.

[6] During the initialization, we assume that the pool of the rest sentence pairs in the heterogeneous data is the exemplifying sample of the out-domain.

ing the derived probabilities as the initialization for their corresponding domain-conditioned alignment parameters. In our implementation, one EM iteration is usually dedicated for this. It should be noted that we ignore the domain prior parameters in the model during the period.

Parameter Constraints During training, it would be also necessary to keep the domain prior parameters fixed for all sentence pairs that belong to seed samples. This can be thought of as the constraints derived from the partial knowledge, guiding the learning to a desirable parameter space.

Domain-conditioned LMs training We now discuss how to train the domain-conditioned LMs with partial supervision. It would be reasonable to use the domain-specific seed samples to train their exemplifying domain-conditioned LMs, and the pool of the rest sentence pairs to train the out-domain LMs. Nevertheless, the out-domain LMs trained on such a big corpus could dominate the other domain-conditioned LMs. Following Cuong and Sima'an (2014b), we rather create a "pseudo" out-domain sample to train the out-domain LMs, i.e., the creation is via an inspired burn-in period. In brief, an EM iteration is dedicated just to compute $P(D_{OUT}|\ \mathbf{f},\ \mathbf{e})$ for all sentences, ranking them and select a small subset with highest score as the (on the fly) pseudo out-domain sample.

Note that our partial learning framework is very simple. There are various advanced learning framework that are also applicable with the partial supervision, e.g., Posterior Regularization (Ganchev et al., 2010). This leaves much space for future work.

5 Domain-conditioned Decoding

At test time, assigning each sentence pair to a single most likely domain (hard decision) is likely to result in sub-optimal performance.[7] Instead we average over domains (soft decision) while predicting the translation. Formally for each sentence pair, $\langle \mathbf{e},\ \mathbf{f} \rangle$, we can find their best Viterbi alignment, $\hat{\mathbf{a}}$ as follows:

$$\begin{aligned}\hat{\mathbf{a}} &= \text{argmax}_{\mathbf{a}} \sum\nolimits_D P(\mathbf{f}, \mathbf{a}, D|\mathbf{e}) \\ &= \text{argmax}_{\mathbf{a}} \sum\nolimits_D P(\mathbf{f}, \mathbf{a}|\mathbf{e}, D) P(D|\mathbf{e}) \\ &= \text{argmax}_{\mathbf{a}} \sum\nolimits_D P(\mathbf{f}, \mathbf{a}|\mathbf{e}, D) P(\mathbf{e}|D) P(D).\end{aligned}$$

Here, we derive the last equation by applying Bayes' rule to $P(D|\ \mathbf{e})$, i.e., $P(D|\ \mathbf{e}) \propto P(\mathbf{e}|\ D)P(D)$. Interestingly, our Viterbi decoding now relies on a mix of domain-conditioned statistics for each sentence pair. The computing of term $\sum_D(\mathbf{a})$ for all possible alignments, $\mathbf{a}$, however, is intractable, making the search problem difficult. Inspired by Liang et al. (2006), we opt instead for a heuristic objective function as follows[8]:

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\text{argmax}} \prod\nolimits_D P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e}, D)^{P(\mathbf{e}|\ D)P(D)}. \quad (5)$$

Here, note that $\prod p$ is a lower bound for $\sum p$, when $0 \leq p \leq 1$, according to Jensen's inequality. With Eq. 5, it is straightforward to design a dynamic programming algorithm to decoding, e.g., the Viterbi algorithm. In practice, we observe that the approximation yields good results. Later experiments on word alignment will present this in detail.

6 Experimental Setup

In the following experiments, we use three heterogeneous English-Spanish corpora consisting of $1M$, $2M$ and $4M$ sentence pairs respectively. These corpora combine two parts. The first part respectively $0.7M$, $1.7M$ and $3.7M$ is collected from multiple domains and resources including EuroParl (Koehn, 2005), Common Crawl, United Nation, News Commentary. The second part consists of three domain-exemplifying samples consisting of roughly $100K$ sentence pairs for each one (total $300K$). Each of these three samples (manually collected by a commercial partner) exemplifies a specific domain related to **Legal**, **Hardware** and **Pharmacy**.

Outlook In Section 7 we examine the word alignment yielded by the HMM alignment model and our latent domain HMM alignment model. In Section 8 we proceed further to examine the translation produced by derived SMT systems.

[7] Later experiments on word alignment will confirm this.

[8] Alternative solutions could be Lagrangian relaxation-based decoder (DeNero and Macherey, 2011; Chang et al., 2014).

Model	Domain Prior	Prec.↑	Δ	Rec.↑	Δ	AER↓	Δ
		1 **Million**					
Model 4 (ref.)	-	71.56	-	64.59	-	32.10	-
Baseline	-	66.95	-	61.29	-	36.00	-
Latent	Pharmacy	67.85	**+0.90**	61.72	**+0.43**	35.36	**-0.64**
	Legal	67.57	**+0.62**	62.29	**+1.00**	35.17	**-0.83**
	Hardware	69.41	**+2.46**	63.58	**+2.29**	33.63	**-2.37**
	Legal + Hardware + Software	**69.64**	**+2.69**	**63.30**	**+2.01**	**33.68**	**-2.32**
		2 **Million**					
Model 4 (ref.)	-	74.13	-	65.30	-	30.56	-
Baseline	-	68.34	-	61.58	-	35.22	-
Latent	Pharmacy	68.85	**+0.51**	62.58	**+1.00**	34.43	**-0.79**
	Legal	69.98	**+1.64**	64.01	**+2.43**	33.13	**-2.09**
	Hardware	69.45	**+1.11**	63.23	**+1.65**	33.81	**-1.41**
	Legal + Hardware + Software	**71.51**	**+3.17**	**63.87**	**+2.29**	**32.53**	**-2.69**
		4 **Million**					
Model 4 (ref.)	-	75.53	-	65.95	-	29.58	-
Baseline	-	69.37	-	64.30	-	33.26	-
Latent	Pharmacy	69.69	**+0.32**	62.80	-1.50	33.94	+0.68
	Legal	70.51	**+1.14**	63.94	-0.36	32.93	**-0.33**
	Hardware	71.75	**+2.38**	64.44	**+0.14**	32.10	**-1.16**
	Legal + Hardware + Software	**72.16**	**+2.79**	**64.30**	±0.0	**31.99**	**-1.27**

Table 1: Alignment accuracy over heterogeneous corpora.

7 Word Alignment Experiment

For alignment accuracy evaluation, we use a data set of 100 sentence pairs with their "golden" alignment from Graca et al. (2008). Here, the golden alignment consists of *sure* links (S) and *possible* links (P) for each sentence pair. Counting the set of generating *alignment* links (A), we report the word alignment accuracy by *precision* ($\frac{|A \cap P|}{|P|}$), *recall* ($\frac{|A \cap S|}{|S|}$), *alignment error rate* (AER) ($1 - \frac{|A \cap P| + |A \cap S|}{|A| + |S|}$) (Och and Ney, 2003).[9]

For all experiments, we use the same training configuration for both the baseline/the latent domain alignment model: 5 iterations for IBM model 1/the latent domain model; 3 iterations for HMM alignment model/the latent domain model. For evaluation, we first align the sentence pairs in both directions and then symmetrize them using the *grow-diag-final* heuristic (Koehn et al., 2003).

For reference we also report the performance of a considerably more expressive Model 4, capable of capturing more structure, but at the expense of intractable inference. Using MGIZA++ (Gao and Vogel, 2008), we run 5 iterations for training Model 1, 3 iterations for training the HMM alignment model, Model 3 and Model 4.

7.1 Learning with Single Domain

We first examine the binary case, where we are given domain information in advance for each kind of samples **only**, e.g., Legal, or Pharmacy, or Hardware. For the different sizes of the heterogeneous data ($1M$, $2M$ and $4M$) the seed sample size is thus 10%, 5% and 2.5% respectively. Note that in such cases, training the latent domain alignment model induces two domain-conditioned statistics: in-domain vs. out-domain (D_1 and D_2 respectively). Once the model is trained, we combine the induced domain-conditioned statistics together (Eq. 5) and examine the produced word alignment output.

Table 1 presents the results. Most importantly, it shows that as long as providing domain information for reasonably large enough data, learning the latent domain alignment model notably improves the word alignment accuracy. For instance, given in advance the domain information for a sample of 10%, and 5% of the heterogeneous corpora, our model consistently improves the word alignment accuracy in

[9] Note that better results correspond to larger Precision, Recall and to smaller AER.

all cases. Meanwhile, given in advance the domain information for a relatively small sample of 2.5% of the heterogeneous data, the results are mixed. We obtain a good performance/slightly better performance/worse performance with the case of Hardware/Legal/Pharmacy respectively.

What do domain-conditioned statistics look like?

To have an idea what the induced statistics look like, we investigate their **conditional entropy**. Here, we present the conditional entropy for the domain-confused/-conditioned word translation statistics induced from the HMM alignment model/its latent domain model. Note that similar results are observed for transition tables.

Model	Prior	Statistics	H(F| E)
Baseline	-	Domain-confused	**1348.53**
Latent	Hardware	D_1-conditioned	**1124.43**
		D_2-conditioned	**1354.58**
	Legal	D_1-conditioned	**1104.58**
		D_2-conditioned	**1385.35**
	Pharmacy	D_1-conditioned	**1115.52**
		D_2-conditioned	**1342.54**

Table 2: Conditional entropy of the statistics.

Formally, for a translation table, $\langle F,\ E\rangle$, its conditional entropy, $H(F|\ E)$ can be estimated from its possible word pairs, $\langle e,\ f\rangle$: $H(F|\ E) = -\sum_e P(e) \sum_f P(f|\ e) \log P(f|\ e)$. Table 2 reveals that the induced D_1-conditioned statistics need much less *bits* to represent than the induced domain-confused statistics, e.g., 1124.43, 1104.58, 1115.52 vs. 1348.53. This implies the induced D_1-conditioned statistics are much more **predictable** compared to the domain-confused statistics. Meanwhile, the induced D_2-conditioned statistics are similar to the domain-confused statistics in terms of the conditional entropy, e.g., 1354.58, 1385.35, 1342.54 vs. 1348.53.

7.2 Learning with Multiple Domains

It would be more interesting to learn the latent domain alignment model for multiple domains, rather than learning with each of them separately. In detail, using **all** the seed samples from different domains, we aim to learn four different domain-conditioned statistics simultaneously. Under this setting, we obtain good results, as described in Table 1. For the two cases with the training corpora of 2M and 4M sentence pairs respectively, learning with the combining domain prior knowledge produces the best word alignment accuracy compared to the rest. In the last case with the training corpus of 1M sentence pairs, learning with the combining domain prior knowledge produces compatible with the case of Hardware, i.e., the best binary domain case.

Table 1 also reveals that the performance of our model approaches Model 4, even though Model 4 is much more complex and computationally expensive.

Domain-conditioned statistics combination

We also investigate the relation between the number of domain-conditioned statistics "involved" in the Viterbi decoding (Eq. 5) and the word alignment accuracy. Table 3 presents the results in case of using only the induced D_1-/, D_2-/, D_3-/, D_4-conditioned statistics separately, and also using their different combinations. Interestingly, we observe that using more domain-conditioned statistics for decoding incrementally improves the word alignment accuracy over the heterogeneous data. While the domain-conditioned statistics are very different in their characteristics from each other, the results reveal how they are *complementary* to the others, conveying a mix of domains for each sentence pair.

Decoding's Statistics	Prec.↑	Rec.↑	AER↓
Hard Decision (ref.)	68.49	62.80	34.48
D_1 (Pharmacy)	64.78	59.86	37.78
D_2 (Legal)	66.54	61.15	36.27
D_3 (Hardware)	66.98	61.36	35.95
D_4 (OUT)	68.46	63.01	34.38
D_1 + D_2	66.80	61.72	35.84
D_1 + D_2 + D_3	68.54	62.80	34.46
D_1 + D_2 + D_3 + D_4	**69.64**	**63.30**	**33.68**

Table 3: Domain-conditioned statistics combination for Viterbi decoding. The reported results are for the heterogeneous corpus of 1M sentence pairs. Similar results are observed for other training data.

Finally, it is also tempting to make a comparison between the *hard* vs. *soft* domain assignment in Viterbi decoding. Here, for hard domain decision we simply do decoding with the following objec-

tive function: $\hat{\mathbf{a}} = \text{argmax}_{\mathbf{a}} P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e}, \hat{D})$, where $\hat{D} = \text{argmax}_D P(D|\ \mathbf{e})$. Table 3 presents the results. It reveals that a soft domain assignment on the domain of sentence pairs results in a better alignment accuracy than a hard domain assignment.[10]

8 Translation Experiment

In this section, we investigate the contribution of our model in terms of the translation accuracy. Here, we run experiments on the heterogeneous corpora of 1M, 2M, and 4M sentence pairs, testing the translation accuracy over four different domain-specific test sets related to News, Pharmacy, Legal, and Hardware.

We use a standard state-of-the-art phrase-based system as the baseline. Our dense features include MOSES (Koehn et al., 2007) baseline features, plus hierarchical lexicalized reordering model features (Galley and Manning, 2008), and the word-level feature derived from IBM model 1 score, c.f., (Och et al., 2004).[11] The interpolated 5-grams LMs with Kneser-Ney are trained on a very large monolingual corpus of 2B words. We tune the systems using k-best batch MIRA (Cherry and Foster, 2012). Finally, we use MOSES (Koehn et al., 2007) as decoder.

Our system has exactly the same setting with the baseline, except: (1) To learn the translation, we use the alignment result derived from our latent domain HMM alignment model, rather than the HMM alignment model; and (2) We replace the word-level feature with our four domain-conditioned word-level features derived from the latent domain IBM model 1. Here, note that our latent model is learned with the supervision from the combining domain knowledge of all three domain-specific seed samples.

[10]Note that similar results are also observed for training, in which a soft domain assignment using soft EM produces better alignment accuracy than a hard domain assignment using hard EM. (See (Gao et al., 2011) for reference to hard domain assignment to training data.) This is perhaps due to the characteristics of the data we use. For instance, News sentence pairs are useful for translating Legal, Financial or EuroParl to varying degrees.

[11]For every phrase pair $\langle \tilde{f},\ \tilde{e} \rangle$ with their length of $m_{\tilde{f}}$ and $l_{\tilde{e}}$ respectively, the lexical feature estimates a probability in Model 1 style between their word pairs $\langle f_j,\ e_i \rangle$ (i.e. $P(\tilde{f}|\ \tilde{e}) = \frac{\epsilon}{l_{\tilde{e}}} \prod_{j=1}^{m_{\tilde{f}}} \sum_{i=1}^{l_{\tilde{e}}} P(f_j|e_i)$). Note that adding word-level features from both translation sides does not help much, as observed by (Och et al., 2004). We thus add only an one from a translation side.

Data	System	BLEU↑	METEOR↑	TER↓
News test				
1M	Model 4 (ref.)	23.6	30.8	58.3
	Baseline	23.2	30.6	58.9
	Our System	23.5/**+0.3**	30.8/+0.2	58.7/-0.2
2M	Baseline	25.9	32.4	56.1
	Our System	26.3/**+0.4**	32.6/**+0.2**	55.6/**-0.5**
4M	Baseline	26.8	33.0	55.0
	Our System	27.0/**+0.2**	33.1/+0.1	54.7/**-0.3**
Pharmacy				
1M	Model 4 (ref.)	54.7	43.8	33.4
	Baseline	53.9	43.4	34.6
	Our System	54.4/**+0.5**	43.8/**+0.4**	34.0/**-0.6**
2M	Baseline	54.5	43.7	34.4
	Our System	55.3/**+0.8**	44.3/**+0.6**	33.5/**-0.9**
4M	Baseline	54.8	43.9	33.8
	Our System	55.0/**+0.2**	44.0/+0.1	33.7/-0.1
Legal				
1M	Model 4 (ref.)	56.6	44.7	34.1
	Baseline	56.0	44.2	35.0
	Our System	57.2/**+1.2**	44.4/+0.2	34.0/**-1.0**
2M	Baseline	55.8	43.9	35.4
	Our System	58.3/**+2.5**	44.7/**+0.8**	**33.4**/**-2.0**
4M	Baseline	55.9	43.9	34.3
	Our System	57.3/**+1.4**	44.4/**+0.5**	33.4/**-0.9**
Hardware				
1M	Model 4 (ref.)	75.4	53.6	17.7
	Baseline	74.9	53.1	19.0
	Our System	76.8/**+1.9**	53.9/**+0.8**	17.3/**-1.7**
2M	Baseline	75.7	53.5	18.6
	Our System	77.4/**+1.7**	54.3/**+0.8**	17.0/**-1.6**
4M	Baseline	77.1	54.2	17.3
	Our System	77.9/**+0.8**	54.5/**+0.3**	16.7/**-0.6**

Table 4: Metric scores for the systems, which are averages over multiple runs. Bold results indicate that the comparison is significant over the baseline.

For the News translation task, we tune systems on the News-test 2008 of $2,051$ sentence pairs and test them on the News-test 2013 of $3,000$ sentence pairs from the WMT 2013 shared task (Bojar et al., 2013). For the Pharmacy, Legal, and Hardware translation tasks, we tune systems on three domain-specific dev sets of $1,000$ sentence pairs and test them on three domain-specific test sets of $1,016$, $1,326$ and $1,721$ sentence pairs. We report three metrics - BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2011) and TER (Snover et al., 2006), with statistical significance at 95% confidence interval

under paired bootstrap re-sampling.[12] For every system reported, we run the optimizer three times, before running MultEval (Clark et al., 2011) for resampling and significance testing.

Data	BLEU↑	METEOR↑	TER↓
1M	**+1.0**	**+0.4**	**-0.9**
2M	**+1.4**	**+0.6**	**-1.3**
4M	**+0.7**	**+0.3**	**-0.5**

Table 5: Averaged improvements across the tasks.

Results are in Table 4, showing significant improvements across four different test sets over different heterogeneous corpora sizes. Table 5 gives a summary of the improvements. On average, over heterogeneous corpora of 1M, 2M and 4M sentence pairs, our system outperforms the baseline by 1.0 BLEU, 1.4 BLEU and 0.7 BLEU, respectively. Finally, we observe that our system produces comparably good performance to the MGIZA++-based system. When 1M data is considered, on *three* of *four* tasks, our system produces at least compatible translation accuracy to the corresponding MGIZA++-based system.

Further analysis reveals that the improvement is due to not only the reduction in alignment error rate, but also the use of the domain-sensitive lexical features. Moreover, the domain-sensitive lexical features is particularly useful when the domain of the test data matches with the domain of seed samplers. This is also widely observed in the literature, e.g., see (Eidelman et al., 2012; Hasler et al., 2014; Hu et al., 2014).

9 Related Work and Conclusion

In terms of domain-conditioned statistics for word alignment, a distantly related research line (Tam et al., 2007; Zhao and Xing, 2008) focuses on using document topics to improve the word alignment. In terms of learning word alignment with partial supervision, another distantly related research line focuses on semi-supervised training with partial manual alignments (Fraser and Marcu, 2006; Gao and Vogel, 2010; Gao et al., 2010). Finally, recent work also focuses on data selection (Kirchhoff and Bilmes, 2014; Cuong and Sima'an, 2014b), mixture models (Carpuat et al., 2014), instance weighting (Foster et al., 2010) and latent variable models (Cuong and Sima'an, 2014a) over heterogeneous corpora.

One main contribution of this work is the novelty of exploring the quality of word alignment in heterogeneous corpora. This, surprisingly, has not received much attention thus far (see the study of Bach et al. (2008) and Gao et al. (2011) for reference in the literature). Another major contribution of this work is a learning framework for latent domain word alignment with partial supervision using seed domains. We present its benefits for improving not only the word alignment accuracy, but also the translation accuracy resulting SMT systems produce. We hope this study sparks a new research direction for using domain samples, which is cheap to gather, but has not been exploited before.

One obvious direction for future work might be to integrate the model into fertility-based alignment models (Brown et al., 1993), as well as other recently advanced alignment frameworks, e.g., (Simion et al., 2013; Tamura et al., 2014; Chang et al., 2014). Another interesting direction might be to integrate our model into advanced mixing multiple translation models, improving SMT systems trained on the heterogeneous data (Razmara et al., 2012; Sennrich et al., 2013; Carpuat et al., 2014). Finally, an open question is whether it is possible to learn the latent domain alignment model in a fully unsupervised style. This challenge deserves more attention in future work.

Acknowledgements

We are indebted to Ivan Titov and three anonymous reviewers for their constructive 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).

[12] Note that better results correspond to larger BLEU, METEOR and to smaller TER.

References

Nguyen Bach, Qin Gao, and Stephan Vogel. 2008. Improving word alignment with language model based confidence scores. In *Proceedings of the Third Workshop on Statistical Machine Translation*.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19:263–311, June.

Marine Carpuat, Cyril Goutte, and George Foster. 2014. Linear mixture models for robust machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*.

Yin-Wen Chang, Alexander M. Rush, John DeNero, and Michael Collins. 2014. A constrained viterbi relaxation for bidirectional word alignment. In *Proceedings of ACL*.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of NAACL HLT*.

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 HLT: Short Papers*.

Hoang Cuong and Khalil Sima'an. 2014a. Latent domain phrase-based models for adaptation. In *Proceedings of EMNLP*.

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

John DeNero and Klaus Macherey. 2011. Model-based aligner combination using dual decomposition. In *Proceedings of ACL*.

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.

Vladimir Eidelman, Jordan Boyd-Graber, and Philip Resnik. 2012. Topic models for dynamic translation model adaptation. In *Proceedings of ACL: Short Papers*.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of EMNLP*.

Alexander Fraser and Daniel Marcu. 2006. Semi-supervised training for statistical word alignment. In *Proceedings of ACL*.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of EMNLP*.

Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *J. Mach. Learn. Res.*, 11:2001–2049, August.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, SETQA-NLP '08.

Qin Gao and Stephan Vogel. 2010. Consensus versus expertise: A case study of word alignment with mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10.

Qin Gao, Nguyen Bach, and Stephan Vogel. 2010. A semi-supervised word alignment algorithm with partial manual alignments. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10.

Qin Gao, Will Lewis, Chris Quirk, and Mei-Yuh Hwang. 2011. Incremental training and intentional over-fitting of word alignment. In *Proceedings of MT Summit XIII*.

Joao Graca, Joana Paulo Pardal, Luisa Coheur, and Diamantino Caseiro. 2008. Building a golden collection of parallel multi-language word alignment. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Joao Graca, Kuzman Ganchev, and Ben Taskar. 2010. Learning tractable word alignment models with complex constraints. *Comput. Linguist.*, 36(3):481–504.

Eva Hasler, Phil Blunsom, Philipp Koehn, and Barry Haddow. 2014. Dynamic topic adaptation for phrase-based mt. In *Proceedings of EACL*.

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

Katrin Kirchhoff and Jeff Bilmes. 2014. Submodularity for data selection in machine translation. In *Proceedings of EMNLP*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL*.

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.

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, MMichigan0605 AAMT.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of HLT-NAACL*.

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.

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, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In *HLT-NAACL*.

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

Majid Razmara, George Foster, Baskaran Sankaran, and Anoop Sarkar. 2012. Mixing multiple translation models in statistical machine translation. In *Proceedings of ACL*.

Rico Sennrich, Holger Schwenk, and Walid Aransa. 2013. A multi-domain translation model framework for statistical machine translation. In *Proceedings of ACL*.

Andrei Simion, Michael Collins, and Cliff Stein. 2013. A convex alternative to ibm model 2. Proceedings of EMNLP.

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 AMTA*.

Yik-Cheung Tam, Ian Lane, and Tanja Schultz. 2007. Bilingual lsa-based adaptation for statistical machine translation. *Machine Translation*, 21(4):187–207.

Akihiro Tamura, Taro Watanabe, and Eiichiro Sumita. 2014. Recurrent neural networks for word alignment model. In *Proceedings of ACL*.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of COLING*.

Bing Zhao and Eric P. Xing. 2008. Hm-bitam: Bilingual topic exploration, word alignment, and translation. In *Proceedings of NIPS*.