Noname manuscript No.
(will be inserted by the editor)

Induction of Latent Domains in Heterogeneous Corpora: A Case Study of Word Alignment

Hoang Cuong · Khalil Sima'an

Received: date / Accepted: date

Abstract This paper 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-focused 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-focused word alignment statistics for sentence pairs. Our experiments show that the derived domain-focused statistics, once combined together, produce significant improvements both in word alignment accuracy and in translation accuracy of their resulting SMT systems. Going beyond the findings, we surmise that virtually any large corpus (e.g. *Europarl*, *Hansards*, *Common Crawl*) harbors an arbitrary diversity of hidden domains, unknown in advance. We address the novel challenge of unsupervised induction of hidden domains in parallel corpora, applied within a domain-focused word-alignment modeling framework. On the technical side, we contrast flat estimation for the unsupervised induction of domains to a simple form of hierarchical estimation, consisting of two steps aiming at avoiding bad local maxima. Extensive experiments, conducted over seven different language pairs with fully unsupervised induction of domains for word alignment, demonstrate significant improvements in alignment accuracy.

1 Introduction

Word alignment is a fundamental component that automatically learns the correspondence between words. Serving a vital role in SMT systems, it seeds the input to learn their translation of phrases/rules, and the input for advanced neural network translation models (e.g. see (Devlin et al, 2014)). The

ILLC, University of Amsterdam
E-mail: {c.hoang,k.simaan}@uva.nl

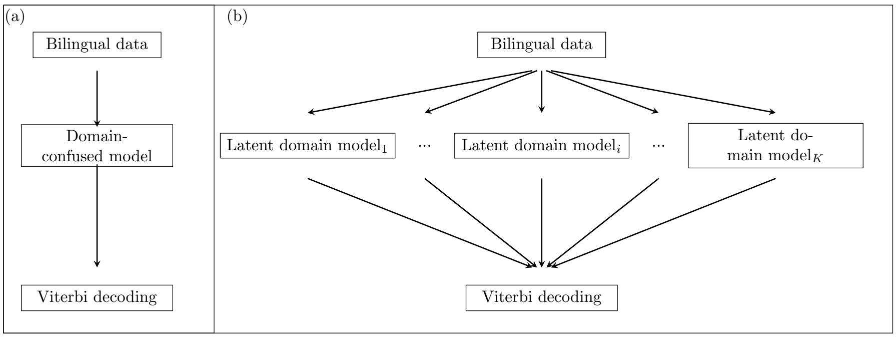

Fig. 1 Statistical alignment framework (a) vs. Statistical alignment framework with K little "hidden" domains (b).

lexical statistics at word level also provide a reliable smooth estimate for the translation (e.g. see (Och et al, 2004; Huck et al, 2012)). After twenty years since the IBM Models (Brown et al, 1993) and HMM-based alignment model (Vogel et al, 1996), word alignment is still an active research line in the literature, e.g. See some recent works (Simion et al, 2013; Tamura et al, 2014; Chang et al, 2014; Shen et al, 2015; Wang et al, 2015; Liu et al, 2015).

Generally, the coverage of domains an SMT system can translate fully depends on the bilingual data used to train. We usually have access to a large mix of diverse domains corpora that consist of sentence pairs representing diverse domains, e.g. *News*, *Politics*, *Financial*, *Sports*, etc. Such mixed corpora sometimes can be also called "heterogeneous" corpora (Carpuat et al, 2014). It is clear that a word could be translated in very different ways when it comes to different domains. Nevertheless, the word alignment statistics induced from word alignment models reflect translation preferences *aggregated* over the diverse domains. In this sense, they can be considered domain-*confused* statistics. This may lead to sub-optimal performance once trained on a large heterogeneous parallel training data.

In this paper, we thus focus on more representative statistics: the domain-focused word alignment statistics, i.e. the word alignment statistics with respect to each of the diverse domains. To this end, our idea is to refine the coarse, domain-confused alignment models to a mixture of different types of domain-focused models. A distribution weighted combination approach (Mansour et al, 2009a,b) is used: the posterior distribution of domains in a source sentence is estimated by the model, and then used to combine the predictions associated with each domain for the given sentence. Figure 1 illustrates our statistical alignment framework with little "hidden" domains and how it is relative to the original statistical alignment framework.

In that spirit, we first propose the latent domain word alignment model. Specifically, we introduce a latent domain variable z to signify the domains of the heterogeneous parallel training data. Thus we extend the concept of the alignment, $\mathbf{a}$ over a sentence pair, $\langle \mathbf{e},\ \mathbf{f} \rangle$ from $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e})$ towards $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ z)$, i.e. the translation of the alignment with respect to a specific domain. To model

$P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ z)$ we extend the HMM-based alignment model (Vogel et al, 1996),[1] representing an alignment by word translation, word transition probabilities, plus an additional latent domain layer that is conditioned on by the rest of parameters. Our ultimate goal is to tighten the generative process of the alignment over a sentence pair and of the sentence pair itself over a domain.

Given our proposed model, we first investigate the specific question: Given domain information for some small subsets of a large heterogeneous parallel training data, how can we use the information as *priors* and learn the corresponding domain-focused word alignment statistics for the pool of the rest of the sentence pairs in the mixed data? Having domain information for such small subsets is a common scenario in practice. This is especially for popular language pairs (e.g. English-German, English-French and English-Spanish), where one can easily access this kind of resource from DGT-Translation Memory (Steinberger et al, 2012), JRC-ACQUIS (Steinberger et al, 2006) and Translation Automation Society (TAUS.com). We view this challenge as learning domain-focused word alignment statistics with partial supervision.

We derive Expectation Maximization (EM) estimation algorithm (Dempster et al, 1977), and present how to guide the training with the partial supervision, i.e. through the way we initialize the parameters and also the way we keep some of them fixed as constraints according to the given "domain knowledge". Once the domain-focused statistics have been induced we present how to combine them together, conveying a mix of domain-specific and general-domain for each sentence pair. Such a combination, however, is intractable to compute. We address the problem by proposing an approximate objective function for search, which we find it yields good results in practice.

We report experimental results over heterogeneous parallel training corpora of $1M$, $2M$ and $4M$ sentence pairs, where we are provided domain information for different subsets of 10%, 5% and 2.5% of the mixed data respectively. Learning with the domain knowledge from each subset, we show that the latent HMM-based alignment model produces significant improvements in the word alignment accuracy compared to its former HMM-based alignment model. We also show that learning the model with the combining domain knowledge we are given in advance brings out the best performance of the model. We proceed further to exemplify the translation accuracy yielded by derived SMT systems, showing significant improvements across *four* translation tasks.

Going beyond the findings, we surmise that virtually any large corpus harbors an arbitrary diversity of hidden domains, unknown in advance. Our goal is thus explicitly to abandon the assumption of known target domains, and with it the need for seed data exemplifying these domains. In other words, we apply the model to arbitrary parallel corpora, i.e. to not only heterogeneous ones but also standard corpora often used to train SMT systems, e.g. *Hansards*, *Europarl* (Koehn, 2005), *JRC-Acquis* (Steinberger et al, 2012), *Common Crawl*, *United Nations* and *News Commentary*. The problem is of directly inducing

[1] Although our work focuses on the HMM-based alignment model, the approach can be also straightforwardly applied to fertility-based alignment models (Brown et al, 1993).

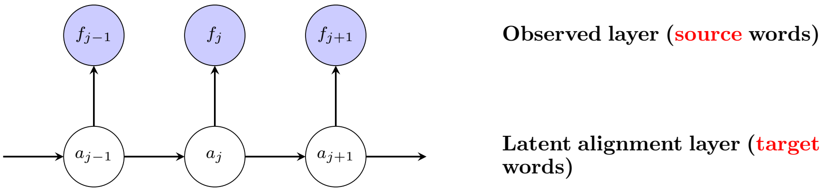
f_{j-1}
f_j
f_{j+1}
a_{j-1}
a_j
a_{j+1}
Observed layer (source words)
Latent alignment layer (target words)

Fig. 2 HMM alignment model with observed and latent alignment layers.

hidden domains of arbitrary granularity. For the unsupervised induction of domains, an estimation using the EM algorithm turns out to provide a sub-optimal performance, specifically with a large number of latent domains (e.g. 128 domains). We address the problem by contributing a simple form of hierarchical estimation involving a two-step procedure that prevents EM from getting stuck in bad local maxima.

Finally, we provide a systematic evaluation on *seven* different language pairs - *English-Spanish*, *English-Portuguese*, *English-Romanian*, *English-Swedish*, *English-Italian*, *English-German* and *English-French* to validate the unsupervised induction of domains for word alignment. Our experiments are on datasets of vastly different sizes. We show that our approach shows consistent improvements over a wide range of standard bilingual corpora - *Hansards*, *Europarl*, *JRC-Acquis*, *Common Crawl*, *United Nations* and *News Commentary* - while varying the number of the hidden domains (8, 16, 32, 48, 64 and 128) and, consequently, regulating their granularity. Further experiments reveal that our hierarchical estimation plays a key role to the success of the proposed model.

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 2. The latent states take values from the target language words and generate source language words. Formally, we use $\mathbf{e} = (e_1, \dots, e_I)$ to denote the target sentence with length I and $\mathbf{f} = (f_1, \dots, f_J)$ to denote the source sentence with length J. For an alignment $\mathbf{a} = (a_1, \dots, a_J)$ 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:

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

Here, $P(f_j|\, e_{a_j})$ represents the word translation probabilities, $P(a_j|\, a_{j-1})$ 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.[2] *Null*-links are also

[2] We model explicitly distances in the range ±5 in this work.

E-step

$$c(f|\ e;\ \mathbf{f},\ \mathbf{e})\ =\ \sum\nolimits_{\mathbf{a}} \frac{P^{(c)}(\mathbf{f},\ \mathbf{a}|\ \mathbf{e})}{P^{(c)}(\mathbf{f}|\ \mathbf{e})} \sum\nolimits_{j=1}^{J} \delta(f,\ f_j) \sum\nolimits_{i=0}^{I} \delta(e,\ e_i) \tag{2}$$

$$c(i|\ i';\ \mathbf{f},\ \mathbf{e})\ =\ \sum\nolimits_{\mathbf{a}} \frac{P^{(c)}(\mathbf{f},\ \mathbf{a}|\ \mathbf{e})}{P^{(c)}(\mathbf{f}|\ \mathbf{e})} \sum\nolimits_{j=1}^{J} \delta(a_j,\ i)\delta(a_{j-1},\ i') \tag{3}$$

M-step

$$P^{(+)}(f|e) = \frac{\sum_{\langle \mathbf{f},\ \mathbf{e}\rangle} c(f|e;\mathbf{f},\ \mathbf{e})}{\sum_f \sum_{\langle \mathbf{f},\ \mathbf{e}\rangle} c(f|\ e;\ \mathbf{f},\ \mathbf{e})},\ P^{(+)}(i|i') = \frac{\sum_{\langle \mathbf{f},\ \mathbf{e}\rangle} c(i|i';\mathbf{f},\ \mathbf{e})}{\sum_i \sum_{\langle \mathbf{f},\ \mathbf{e}\rangle} c(i|\ i';\ \mathbf{f},\ \mathbf{e})} \tag{4}$$

Fig. 3 Pseudocode for the training algorithm for the HMM alignment model. Note that notation $P^{(c)}$ denotes current iteration estimates, and $P^{(+)}$ denotes the re-estimates. Meanwhile, δ is the Kronecker delta function. Note that $P(\cdot|\ \cdot) = \sum_{\mathbf{a}} P(\cdot,\ \mathbf{a}|\ \cdot)$ and it can be thus computed efficiently using dynamic programming.

explicitly added in our implementation, following (Och and Ney, 2003) and (Graça et al, 2010).

The HMM alignment model has two kinds of parameters - word translation and word transition. Designing the EM algorithm for training the model is straightforward (e.g. see (Vogel et al, 1996)). First, we present expected count notations with respect to domains for the parameters. We use $c(f|\ e;\ \mathbf{f},\ \mathbf{e})$ to denote the expected counts that word e aligns to word f. We use $c(i|\ i';\ \mathbf{f},\ \mathbf{e})$ 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'. Note that all the expected counts are in the translation $(\mathbf{f}|\ \mathbf{e})$. Figure 3 presents the algorithm.

Once the HMM alignment model is trained, the most probable alignment, $\hat{\mathbf{a}}$ for each sentence pair can be computed by

$$\hat{\mathbf{a}} = \operatorname{argmax}_{\mathbf{a}} P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e}). \tag{5}$$

The search problem can be solved by the Viterbi algorithm.

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-focused word alignment statistics, i.e. the statistics with respect to each of the diverse domains.

By introducing a latent variable z representing domains of the heterogeneous data, we aim to learn the z-conditioned word alignment model $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ z)$. Note that $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ z)$ contains their former $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e})$ as special case, i.e.

$$P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ z) = \frac{P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e})P(z|\ \mathbf{f},\ \mathbf{a},\ \mathbf{e})}{\sum_{\mathbf{f}} \sum_{\mathbf{a}} P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e})P(z|\ \mathbf{f},\ \mathbf{a},\ \mathbf{e})}. \tag{6}$$

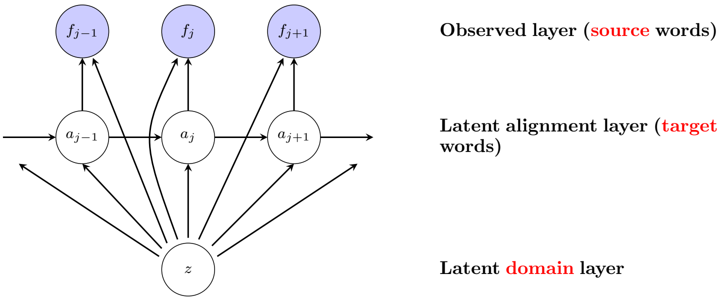

Fig. 4 Latent domain HMM alignment model. An additional latent layer representing domains has been conditioned on by both the rest two layers.

Relying on the HMM alignment model, our latent domain HMM alignment model factors $P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ z)$ into the domain-focused word translation and transition probabilities:

$$P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ z) = \prod_{j=1}^{J} P(f_j|\ e_{a_j},\ z)P(a_j|\ a_{j-1},\ z). \tag{7}$$

The generative story of the model is shown in Figure 4. Note how domain-focused alignment statistics, $P(\cdot|\ \cdot,\ z)$ contain their former domain-confused alignment statistics, $P(\cdot|\ \cdot)$ as special case

$$P(f_j|\ e_{a_j},\ z) = \frac{P(f_j|\ e_{a_j})P(z|\ f_j,\ e_{a_j})}{\sum_f P(f_j|\ e_{a_j})P(z|\ f_j,\ e_{a_j})}, \tag{8}$$

$$P(a_j|\ a_{j-1},\ z) = \frac{P(a_j|\ a_{j-1})P(z|\ a_j,\ a_{j-1})}{\sum_{a_j} P(a_j|\ a_{j-1})P(z|\ a_j,\ a_{j-1})}. \tag{9}$$

With an additional latent domain layer, it becomes crucial to train the model in an efficient way. As suggested by Equations 8 and 9, 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, $P(z|\ \cdot,\ \cdot)$ first, hold them fixed before training the other kind of the parameters. This training scheme is applied in the work of (Cuong and Sima'an, 2014a), however, for a different purpose of data selection for SMT. Instead, in this work we design an algorithm that trains both of them simultaneously via training domain-focused parameters $P(\cdot|\ ,\cdot,\ z)$ directly.

3.1 Training

Our model can be viewed as having a set, Θ of N subsets of domain-focused parameters, Θ_z for N different domains, i.e.

$$\Theta = \{\Theta_{z_1},\ \ldots,\ \Theta_{z_N}\}.$$

```
E-step $\forall z \in \{z_1,\ \ldots,\ z_N\}$ do
    $c(D;\ \mathbf{f},\ \mathbf{e}) = P^{(c)}(z|\ \mathbf{f},\ \mathbf{e})$    (10)
$c(f|\ e;\ \mathbf{f},\ \mathbf{e},\ z) = P^{(c)}(z|\ \mathbf{f},\ \mathbf{e}) \sum_{\mathbf{a}} \frac{P^{(c)}(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ z)}{P^{(c)}(\mathbf{f}|\ \mathbf{e},\ z)} \sum_{j=1}^{J} \delta(f, f_j) \sum_{i=0}^{I} \delta(e,\ e_i)$    (11)
$c(i|\ i';\ \mathbf{f},\ \mathbf{e},\ z) = P^{(c)}(z|\ \mathbf{f},\ \mathbf{e}) \sum_{\mathbf{a}} \frac{P^{(c)}(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ z)}{P^{(c)}(\mathbf{f}|\ \mathbf{e},\ z)} \sum_{j=1}^{J} \delta(a_j,\ i)\delta(a_{j-1},\ i')$    (12)
    M-step $\forall z \in \{z_1,\ \ldots,\ z_N\}$ do
    $P^{(+)}(f|e,z) = \frac{\sum_{\langle \mathbf{f},\ \mathbf{e} \rangle} c(f|e; \mathbf{f},\ \mathbf{e}, z)}{\sum_f \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} c(f|e; \mathbf{f},\ \mathbf{e}, z)},\ P^{(+)}(i|i',z) = \frac{\sum_{\langle \mathbf{f}, \mathbf{e} \rangle} c(i|i'; \mathbf{f}, \mathbf{e}, z)}{\sum_i \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} c(i|i'; \mathbf{f}, \mathbf{e}, z)}$
                    $P^{(+)}(z) = \frac{\sum_{\langle \mathbf{f},\ \mathbf{e} \rangle} c(z; \mathbf{f}, \mathbf{e})}{\sum_z \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} c(z;\ \mathbf{f},\ \mathbf{e})}$    (13)
```

Fig. 5 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. Meanwhile, δ is the Kronecker delta function. Note that $P(\cdot|\ \cdot,\ \cdot,\ z) = \sum_{\mathbf{a}} P(\cdot,\ \mathbf{a}|\ \cdot,\ \cdot,\ z)$ and it can be thus computed efficiently using dynamic programming.

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.

The model has three kinds of parameters - word translation, word transition, and domain prior. Similarly, we use $c(f|\ e;\ \mathbf{f},\ \mathbf{e},\ z)$ to denote the expected counts that word e aligns to word f with respect to latent domain z. We use $c(i|\ i';\ \mathbf{f},\ \mathbf{e},\ z)$ to denote the expected counts that two certain consecutive source words j and $j-1$ align to two target words i and i' respectively with respect to latent domain z. Finally, we also use $c(z;\ \mathbf{f},\ \mathbf{e})$ to denote the expected count of domain priors.

Figure 5 presents the EM algorithm for training our latent domain HMM alignment model. The key difference to the training algorithm for the HMM alignment model is that the new algorithm involves a larger number of parameters (z-conditioned parameters) during training. It also involves new parameters $P(z|\ \mathbf{f},\ \mathbf{e})$. Using Bayes' theorem, $P(z|\ \mathbf{f},\ \mathbf{e})$ can be computed as[3]

$$P(z|\ \mathbf{f},\ \mathbf{e}) \propto P(\mathbf{f}|\ \mathbf{e},\ z)P(\mathbf{e}|\ z)P(z). \tag{14}$$

Here, $P(\mathbf{f}|\ \mathbf{e},\ z)$ can be thought of as the domain-focused translation models, aiming to model how well a source sentence is generated over a target sentence with respect to a domain. Meanwhile, $P(\mathbf{e}|\ z)$ can be thought of as the domain-focused language models (LMs), aiming to model how fluent a target sentence with respect to a domain. We use standard n^{th}-order Markov model for $P(\mathbf{e}|\ z)$, in which

$$P(\mathbf{e}|\ z) = \prod_i P(e_i|\ e_{i-n}^{i-1},\ z). \tag{15}$$

[3] $P(z|\ \mathbf{f},\ \mathbf{e})$ can be also heuristically computed a symmetrized strategy $P(z|\ \mathbf{f},\ \mathbf{e}) \propto P(z)\Big(P(\mathbf{f}|\ \mathbf{e},\ z)P(\mathbf{e}|z) + P(\mathbf{e}|\ \mathbf{f},\ z)P(\mathbf{f}|z)\Big)$. However, we found that this strategy does not provide any significant contribution to the final performance of alignment accuracy.

Here, the notation e_{i-n}^{i-1} denotes the history of length n for the source and target words e_i, respectively.

The final piece of the training is the details of initializing the parameters, as well as training language models. This depends on the induction setting of latent domains we aim for, i.e. supervised or unsupervised setting. Sections 5.1 and 6.1 will discuss them in detail for each specific induction setting.

3.2 Domain-focused decoding

We now present the decoding problem of the Viterbi alignment for each sentence pair, using their mix of diverse domain-focused statistics. For each sentence pair, $\langle \mathbf{e},\ \mathbf{f} \rangle$ we follow a distribution weighted combination approach (Mansour et al, 2009a,b) and find their best Viterbi alignment, $\hat{\mathbf{a}}$ as follows:

$$\begin{aligned} \hat{\mathbf{a}} &= \text{argmax}_{\mathbf{a}} \sum\nolimits_z P(\mathbf{f},\ \mathbf{a},\ z|\ \mathbf{e}) && (16) \\ &= \text{argmax}_{\mathbf{a}} \sum\nolimits_z P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ z) P(z|\ \mathbf{e}) && (17) \\ &= \text{argmax}_{\mathbf{a}} \sum\nolimits_z P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ z) P(\mathbf{e}|\ z) P(z). && (18) \end{aligned}$$

Here, we derive the last equation by applying Bayes' rule to $P(z|\ \mathbf{e})$:

$$P(z|\ \mathbf{e}) \propto P(\mathbf{e}|\ z) P(z) \tag{19}$$

Interestingly, our Viterbi decoding now relies on a mix of domain-focused statistics for each sentence pair. The computing of the term $\sum_z(\mathbf{a})$ 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

$$\hat{\mathbf{a}} = \text{argmax}_{\mathbf{a}} \prod\nolimits_z P(\mathbf{f},\ \mathbf{a}|\ \mathbf{e},\ z)^{P(\mathbf{e}|\ z)P(z)}. \tag{20}$$

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. 20, 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.

4 Experimental setup

This paper is largely empirical. We report results for *seven* different language pairs (*English-Spanish*, *English-Portuguese*, *English-Romanian*, *English-Swedish*, *English-Italian*, *English-German* and *English-French*).

We use standard word-aligned data set for each language pair for evaluation as below:

- For *English-Spanish*, we use a test set of 100 sentence pairs from Europarl with "gold" alignments taken from (Graca et al, 2008).
- Another word-aligned data set of 100 sentence pairs from Europarl, with gold alignments taken from (Graca et al, 2008) is used for the pair of *English-Portuguese.*
- The Gold Standard Word Aligned corpus of 508 Europarl sentence pairs supplied by RWTH is used for the pair of *German-English.*
- The Hansards corpus of gold alignment consisting of 447 sentence pairs from (Och and Ney, 2003) is used for the pair of *English-French.*
- We use the Gold Word-Aligned Data of 248 sentence pairs from (Mihalcea and Pedersen, 2003) for *English-Romanian.*
- We use the Gold Standard Word Aligned Data of 192 sentence pairs from (Holmqvist and Ahrenberg, 2011) for *English-Swedish.*
- Finally, for *English-Italian* we use a test set of 200 English-Italian sentence pairs extracted from JRC-Acquis Corpus with gold alignments taken from (Farajian et al, 2014).

Technically, the gold alignment consists of *sure* links (S) and *possible* links (P) for each sentence pair. Counting the set of generated *alignment* links (A), we report the word alignment accuracy by the following measures (Och and Ney, 2003):

- *Precision* ($\frac{|A \cap P|}{|P|}$),
- *Recall* ($\frac{|A \cap S|}{|S|}$),
- *Alignment error rate* (AER) ($1 - \frac{|A \cap P| + |A \cap S|}{|A| + |S|}$).

Note that better results correspond to larger Precision, Recall and to lower AER.

For all experiments, we use the same training configuration for both the baseline/its latent domain alignment model: 5 iterations for IBM model 1/its latent domain model; 3 iterations for HMM alignment model/its 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).

5 Experiments with partial supervision

Our first set of experiments are designed to investigate how the induction of domains for word alignment, under the partial supervision of seed samples, can help improve word alignment accuracy. Our case study here is with *English-Spanish.*

More specifically, 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 Translation Automation Society (TAUS.com)) exemplifies a specific domain related to **Legal**, **Hardware** and **Pharmacy**.

5.1 Learning with partial supervision

We first discuss 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 $z \in [1, \ldots, (N+1)]$ 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. The idea of involving an out-domain in training latent variable model is proposed by (Cuong and Sima'an, 2014b), but for a different purpose.
- **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 z_i, we initialize $P(z_i | \mathbf{f}, \mathbf{e})$ close to 1 (i.e. $P(z_i | \mathbf{f}, \mathbf{e}) = 0.99$), and, $P(z_{i'} | \mathbf{f}, \mathbf{e})$ close to 0 for other domains i', $i' \neq i$ (i.e. $P(z_i | \mathbf{f}, \mathbf{e}) = 0.01$). Furthermore, we create a uniform distribution over the domain prior parameters for the rest of sentence pairs.
  Uniform initialization for the domain-focused 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.[4] That is, we train the model on each of the samples, assigning the derived probabilities as the initialization for their corresponding domain-focused alignment parameters. In our implementation, one EM iteration is dedicated for this.
- **Parameter Constraints** During training, it would be also necessary to keep the domain prior parameters fixed for all sentence pairs 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-focused LMs training** We now discuss how to train the domain-focused LMs with partial supervision. It would be reasonable to use the domain-specific seed samples to train their exemplifying domain-focused 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-focused 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 $P(z_{OUT} | \mathbf{f}, \mathbf{e})$ for all sentences, ranking them and select a small subset with highest score as the (on the fly) pseudo out-domain sample. Note that to train the LM probs, we construct interpolated 4-gram Kneser-Ney language models.

[4] 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.

Model	**Domain Prior**	**Prec.↑**	Δ	**Rec.↑**	Δ	**AER↓**	Δ
1 **Million**							
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**							
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**							
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.

5.2 Results

Learning with single domain: We first examine the binary case (i.e. $K = 2$) 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-focused statistics: in-domain vs. out-domain (z_1 and z_2 respectively). Once the model is trained, we combine the induced domain-focused statistics together (Eq. 20) 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 significantly 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 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.

How do domain-focused statistics look?: 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/-focused word translation statistics induced from the HMM alignment model/its latent domain model. Note that similar results are observed for transition tables. Formally, for a translation table, $\langle F,\ E\rangle$, its conditional entropy, $H(F|\ E)$ can be esti-

Model	Prior	Statistics	H(F| E)
Baseline	-	Domain-confused	**1348.53**
Latent	Hardware	z_1-conditioned	**1124.43**
		z_2-conditioned	**1354.58**
	Legal	z_1-conditioned	**1104.58**
		z_2-conditioned	**1385.35**
	Pharmacy	z_1-conditioned	**1115.52**
		z_2-conditioned	**1342.54**

Table 2 Conditional entropy of the statistics.

Decoding's Statistics	Prec.↑	Rec.↑	AER↓
z_1 (Pharmacy)	64.78	59.86	37.78
z_2 (Legal)	66.54	61.15	36.27
z_3 (Hardware)	66.98	61.36	35.95
z_4 (OUT)	68.46	63.01	34.38
$z_1 + z_2$	66.80	61.72	35.84
$z_1 + z_2 + z_3$	68.54	62.80	34.46
$z_1 + z_2 + z_3 + z_4$	**69.64**	**63.30**	**33.68**

Table 3 Domain-focused 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.

mated from its possible word pairs, $\langle e, \ f\rangle$:

$$H(F|\ E) = -\sum\nolimits_e P(e) \sum\nolimits_f P(f|\ e) \log P(f|\ e). \tag{21}$$

Table 2 reveals that the induced z_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 z_1-conditioned statistics are much more predictable compared to the domain-confused statistics. Meanwhile, the induced z_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.

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-focused statistics simultaneously. Under this setting, we obtain good results, as shown 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, it produces compatible with the best case yielded by learning the model with the binary domain case, i.e. slightly better precision, but slightly worse recall.

Domain-focused statistics combination: We investigate the relation between the number of domain-focused statistics involved in the Viterbi decoding (Eq. 20) and the word alignment accuracy. Table 3 presents the results in case of using only the induced z_1-/, z_2-/, z_3-/, z_4-conditioned statistics separately, and also using their different combinations. Interestingly, we observe that using more domain-focused statistics for decoding incrementally improves the word alignment accuracy over the heterogeneous data. While the domain-focused 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.

Translation experiment: 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).[5] 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-focused 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.

For each SMT sytem, we report translation accuracy 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. Note that better results correspond to larger BLEU, METEOR and to smaller TER. For every system reported, we run the optimizer three times, before running MultEval (Clark et al, 2011) for resampling and significance testing.

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.

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

[5] Note that adding word-level features from both translation sides does not help much, as observed by (Och et al, 2004; Huck et al, 2012). We thus add only an one from a translation side.

Data	System	BLEU↑	METEOR↑	TER↓
News test				
1M	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	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	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	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.

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.

and 4M sentence pairs, our system outperforms the baseline by 1.0 BLEU, 1.4 BLEU and 0.7 BLEU, respectively.

6 Experiments with unsupervised domain induction

We have shown that under heterogeneous corpora with partial supervision, the latent HMM-based alignment model produces significant improvements in the word alignment accuracy compared to its former HMM-based alignment model. Going beyond the findings, we surmise that virtually any large corpus harbors an arbitrary diversity of hidden domains, unknown in advance. Our goal is thus to explicitly abandon the assumption of known target domains, and with it the need for seed data exemplifying these domains.[6]

We now apply the model to arbitrary parallel corpora, i.e. to not only heterogeneous ones but also standard corpora often used to train SMT systems, e.g. *Hansards*, *Europarl* (Koehn, 2005), *JRC-Acquis* (Steinberger et al, 2012), *Common Crawl*, *United Nations* and *News Commentary*. More specifically,

- For *English-French*, we use the full French-English Hansards corpus of $808.39K$ sentence pairs.[7]
- For *English-Romanian*, *English-Spanish*, *English-Portuguese* and *English-Swedish*, we use the Europarl corpus of $370.11K/1M/1.85M$ and $1.73M$ sentence pairs respectively.[8]
- For *English-Italian*, we use the JRC-Acquis corpus of $780.08K$ sentence pairs.[9]
- Meanwhile, for *English-German*, our training data consists of $4.1M$ sentence pairs obtained from the WMT 2015 MT Shared Task (Bojar et al, 2015), including EuroParl, Common Crawl and News Commentary.

6.1 Unsupervised induction of domains

A fully unsupervised induction of domains for word alignment is obviously a harder problem compared to an induction of domains for word alignment with partial supervision. We now describe two variants of the training algorithm for our model as below:
Flat EM: Without any supervision, our EM starts the training with random initialization of model parameters. Note that to obtain better parameter estimates for word predictions $P(e_i|\ e_{i-n}^{i-1},\ z)$ and avoid overfitting for training language models, we find that it is necessary to apply an expected smoothing approach in the M-step. We chose *expected Kneser-Ney smoothing* technique (Zhang and Chiang, 2014) as it is simple and achieves state-of-the-art performance on the language modeling problem. We refer the algorithm as "Flat EM" to raise the fact that language model and alignment parameters are

[6] Naturally, the data, as any complex and large dataset, contains a wide variety of hidden sub-domains, yet they are not specified in advance. This motivates us to induce these domains automatically. In principle, we could induce domains without reference to the alignment problem and then use the latent domain variable within alignment models. However, we believe that this would not be an optimal choice as such domains are induced to capture phenomena potentially irrelevant to the word alignment problem (e.g. monolingual co-occurrence information).

[7] The corpus consists of $1.1M$ sentence pairs, which is available at http://www.isi.edu/natural-language/download/hansard/index.html. We kept only $808.39K$ sentence pairs as the training data after removing duplicate sentences.

[8] The corpus is available at http://www.statmt.org/europarl.

[9] Similarly, the original corpus (which contains duplicate sentences) consists of $1.0M$ sentence pairs, which is available at http://optima.jrc.it/Acquis/JRC-Acquis.3.0/alignments/index.html.

trained simultaneously during learning.[10]

Hierarchical estimation: Unfortunately, the flat training procedure does not often work well in practice. Later experiments will show that inducing a large number of domains with the basic training procedure could significantly harm the alignment accuracy. But why does that happen? A combination of language and translation models are notorious hard to train jointly as the parameter space is probably too huge. With regards to learning algorithm, directly optimizing the likelihood does not lead to competitive performance as the EM algorithm gets stuck in bad local maxima as a result. With regards to the model by itself, having a more powerful model such as a bilingual neural network translation model (e.g. see (Devlin et al, 2014)) perhaps can improve the unsupervised induction of domains better.

For the sake of simplicity, our work addresses the problem in the spirit of improving the learning algorithm. Our solution is indeed very simple as follows. We start by estimating a simpler model which is basically a combination of only language models with hidden domains:

– Generate a domain z from a prior $P(z)$;
– Generate a sentence $\mathbf{e} = (e_1, \ldots, e_l)$ from the distribution $P(\mathbf{e}|\ z)$.

Note that word alignments are completely ignored at this stage. As before, we use EM to train the model. In the E-step, the posterior distributions for z are computed as

$$P(z|\ \mathbf{e}) \propto P(z)P(\mathbf{e}|\ z). \tag{22}$$

Then in the M-step, they are used to re-estimate the LM parameters $P(e_i|\ e_{i-n}^{i-1},\ z)$ and the priors $P(z)$.

Once the simpler model is estimated, we use the estimated $P(\mathbf{e}|\ z)$ and $P(z)$ within training our full model. Note that we considered only initializing $P(\mathbf{e}|\ z)$ using this procedure but found that keeping the LMs fixed resulted in a better word-alignment performance. This suggests that the simpler model plays a role of a regularizer.

For convenience, we refer the training algorithm as hierarchical EM. In experiments we draw a detailed comparison between the flat and hierarchical estimation in model performance. Despite its simplicity, the hierarchical procedure is observed to be particularly effective, providing consistently better alignment accuracy.

6.2 Results

We conduct our second set of experiments over a wide range of bilingual data, investigating whether our results are consistent across tasks. We start out with a comparison between $K = 8$ and the domain-confused baseline $K = 1$. Subsequently, we thoroughly investigate the model performance by exploring other

[10] We train the interpolated 3-grams latent domain LMs with expected Kneser-Ney smoothing in our experiments.

K	Pre.	Rec.	AER	Pre.	Rec.	AER	K	Pre.↑	Rec.	AER	Pre.	Rec.	AER
	English → German							**German → English**					
	Data: 125K			*Data: 1M*				*Data: 125K*			*Data: 1M*		
1	57.53	57.88	42.30	67.77	67.00	32.61	1	62.87	66.06	35.59	70.11	73.44	28.28
8	59.51	59.49	40.50	69.58	68.38	31.01	8	63.21	65.72	35.57	71.23	74.33	27.27
	Data: 250K			*Data: 2M*				*Data: 250K*			*Data: 2M*		
1	63.81	63.66	36.26	68.64	67.85	31.75	1	66.78	69.83	31.75	70.88	73.80	27.71
8	65.68	65.09	34.62	70.23	69.14	30.31	8	67.89	70.90	30.65	72.27	74.94	26.44
	Data: 500K			*Data: 4M*				*Data: 500K*			*Data: 4M*		
1	66.65	66.29	33.53	68.09	67.85	32.03	1	69.70	72.94	28.74	70.54	73.87	27.86
8	68.50	67.86	31.82	69.27	68.82	30.95	8	70.76	73.70	27.82	72.13	74.83	26.56
	English → Portuguese							**Portuguese → English**					
	Data: 250K			*Data: 1M*				*Data: 250K*			*Data: 1M*		
1	72.70	81.84	23.55	73.40	82.10	23.00	1	76.75	83.40	20.45	77.30	83.14	20.23
8	73.92	82.23	22.63	75.83	83.79	20.85	8	78.44	84.18	19.12	80.47	84.57	17.78
	Data: 500K			*Data: 1.9M*				*Data: 500K*			*Data: 1.9M*		
1	71.96	82.10	23.90	73.75	83.01	22.44	1	76.98	83.27	20.37	79.50	84.44	18.40
8	73.98	83.27	22.20	75.30	83.79	21.17	8	78.74	84.05	19.01	81.32	84.05	17.51
	English → Swedish							**Swedish → English**					
	Data: 200K			*Data: 850K*				*Data: 200K*			*Data: 850K*		
1	74.54	78.56	23.58	76.70	80.21	21.66	1	71.23	82.01	24.02	73.87	83.32	21.95
8	75.24	78.95	23.02	78.22	81.53	20.23	8	71.55	82.10	23.79	74.43	83.86	21.39
	Data: 425K			*Data: 1.7M*				*Data: 425K*			*Data: 1.7M*		
1	76.13	79.55	22.27	77.31	80.81	21.06	1	72.98	82.25	22.91	75.53	84.88	20.34
8	77.91	80.81	20.73	78.91	82.01	19.64	8	73.33	82.75	22.47	76.31	85.51	19.61
	English → Spanish							**Spanish → English**					
	Data: 125K			*Data: 500K*				*Data: 125K*			*Data: 500K*		
1	73.59	79.40	23.90	76.67	81.56	21.21	1	78.60	82.75	19.57	79.91	83.83	18.37
8	76.71	81.92	21.04	78.36	82.87	19.68	8	79.68	83.35	18.69	80.46	83.47	18.21
	Data: 250K			*Data: 1M*				*Data: 250K*			*Data: 1M*		
1	74.89	80.24	22.79	76.70	80.72	21.55	1	79.23	83.23	19.01	81.32	84.19	17.41
8	77.83	82.63	20.08	78.90	82.28	19.62	8	80.61	84.19	17.81	82.18	84.07	16.98
	English → French							**French → English**					
	Data: 400K			*Data: 800K*				*Data: 400K*			*Data: 800K*		
1	79.86	80.11	16.13	81.15	92.55	14.92	1	82.58	91.90	14.00	83.36	92.03	13.45
8	81.19	92.17	15.00	81.99	93.07	14.17	8	83.00	92.17	13.62	84.14	92.37	12.82
	English → Italian							**Italian → English**					
	Data: 400K			*Data: 800K*				*Data: 400K*			*Data: 800K*		
1	77.60	80.11	21.17	77.41	80.16	21.26	1	85.77	81.76	16.26	85.63	81.78	16.33
8	79.20	81.26	19.79	78.83	81.36	19.94	8	87.13	82.98	14.98	87.09	83.08	14.94
	English → Romanian							**Romanian → English**					
	Data: 185K			*Data: 370K*				*Data: 185K*			*Data: 370K*		
1	46.43	40.34	56.83	48.24	42.61	54.75	1	63.87	58.02	39.20	64.32	58.44	38.76
8	50.09	43.18	53.62	52.77	46.40	50.62	8	62.15	56.37	40.88	63.50	57.60	39.59

Table 6 A systematic comparison of alignment quality for *seven* language pairs.

values for K. We present results with our hierarchical estimation, but later on, we provide a detailed comparison between the flat and hierarchical estimation. To have a better understanding of the result, we report the alignment accuracy for the two translation directions (separately from source to target, and from target to source). For all seven language pairs, we vary the amount of training data, considering not only the entire dataset but also its half (and for the larger 5 datasets also 1/4, 1/8 etc).

Esti.	K	Pre.↑	Rec.↑	AER↓	Pre.↑	Rec.↑	AER↓
		English → Spanish			Spanish → English		
Flat	8	75.62	80.84	22.12	79.56	82.99	18.92
Hier	8	76.71	81.92	21.04	79.68	83.35	18.69
Flat	16	75.21	80.36	22.56	79.03	83.35	19.06
Hier	16	75.78	81.08	21.93	79.68	83.23	18.75
Flat	32	75.02	80.36	22.67	78.32	82.51	19.82
Hier	32	76.78	82.04	20.94	79.58	83.23	18.80
Flat	48	74.20	80.00	23.29	77.38	82.40	20.40
Hier	48	75.96	81.44	21.67	79.69	83.23	18.74
Flat	64	73.61	80.12	23.58	77.26	81.92	20.68
Hier	64	75.96	81.44	21.67	79.98	83.59	18.42
Flat	128	73.02	80.36	23.81	76.36	81.68	21.28
Hier	128	75.89	81.44	21.71	79.62	83.47	18.67

Table 7 Comparison between *flat* and *hierarchical* estimation.

Unsupervised domain induction for word alignment: Table 6 presents the results in detail. Overall, domain induction leads to a significant improvement in alignment accuracy. Specifically, we observe the following:

- For all *seven* language pairs, the domain induction framework yields a significant reduction in alignment error rate. For instance, with a corpus of $125K$ English-Spanish sentence pairs, we have a reduction in AER from 23.9 to 21.0 (the English-to-Spanish direction). With the training data of $250K$, AER goes down from 22.8 to 20.1. Given that each language and the corresponding parallel corpus has very different properties, the consistent improvements in alignment accuracy confirm that the benefits from inducing domains are genuine.
- Interestingly, for *five* language pairs (English-French, English-Portuguese, English-Swedish, English-German and English-Italian), the improvements are *roughly as large as* the ones resulting from doubling the training data (for both directions).
- The improvement is consistent across different dataset sizes: from hundreds of thousands to millions of sentence pairs. This is encouraging as it shows that our method is stable across different regimes.

Flat vs. hierarchical estimation: We also compare the model performance for *flat* and *hierarchical* estimation. Table 7 presents the results for a corpus of $125K$ sentence pairs on English-Spanish. As expected, the flat estimation is less stable, while the hierarchical estimation provides consistently better alignment quality. Moreover, the difference tends to be significantly larger with larger K's, confirming the intuition that the hierarchical estimation scheme should be particularly effective for more complex models.

Unsupervised domain induction with sparse Dirichlet priors: (Riley and Gildea, 2012) show that using sparse Dirichlet priors results in improvements for standard alignment models (including the HMM model), as the priors effectively regularize them. We hypothesized that the improvements may be even more significant with our richer models ($K > 1$). In our experiments, we chose the same values of hyperparameters for both translations

	K	**Pre.↑**	**Rec.↑**	**AER↓**	**Pre.↑**	**Rec.↑**	**AER↓**
		English → Spanish			**Spanish → English**		
✗	1	73.59	79.40	23.90	78.60	82.75	19.57
✓	1	74.91	80.48	22.68	78.63	82.51	19.65
✗	8	76.71	81.92	21.04	79.68	83.35	18.69
✓	8	77.21	80.96	21.15	80.49	82.75	18.49
✗	16	75.78	81.08	21.93	79.68	83.23	18.75
✓	16	77.53	81.20	20.86	80.47	82.75	18.50
✗	32	76.78	82.04	20.94	79.58	83.23	18.80
✓	32	78.71	82.28	19.73	80.96	82.99	18.12
✗	48	75.96	81.44	21.67	79.69	83.23	18.74
✓	48	78.17	81.68	20.28	81.12	82.99	18.03
✗	64	75.96	81.44	21.67	79.98	83.59	18.42
✓	64	78.68	82.28	19.74	81.18	82.87	18.05
✗	128	75.89	81.44	21.71	79.62	83.47	18.67
✓	128	78.27	82.28	19.96	80.22	82.28	18.85

Table 8 The influence of anti-smoothing (marked with notation ✓).

and alignment probabilities (10^{-4}).[11] As in (Riley and Gildea, 2012), we used the variational EM algorithm (Beal, 2003), resulting in a small change in the M-step (passing both the expected counts computed at the E-step and the hyperparameters through the exponentiated digamma function).

Table 8 presents our results. We conduct experiments on a small English-Spanish training data of $125K$ sentence pairs. Interestingly, Table 8 shows that this anti-smoothing helps but it helps both the domain-confused baselines and our domain-informed models similarly. Consequently, we treat this modification as an orthogonal issue and do not use it in any other experiments. Still, it suggests that in practical systems it makes sense to use the anti-smoothing in a combination with our approach.

Translation experiment: Finally, we investigate whether the improvements in alignment quality result in better translation models. We evaluate translation for *three* language pairs on the WMT shared task - English-Spanish, English-French and English-German. Again, we measure translation on three corpora with different size, including a small corpus of $125K$ for English-Spanish, a medium corpus of $800K$ sentence pairs for English-French, and a large corpus of $4.1M$ sentence pairs of English-German.

For evaluation, we conducted experiments over standard News-test sets from the WMT shared task. In detail, for English-Spanish, we use the News-test 2008 of $2,051$ sentence pairs for development and the News-test 2013 of $3,000$ sentence pairs for testing. Similarly, for English-French, we use the News-test 2009 of $2,525$ sentence pairs as a development set and the News-test 2013 of $3,000$ sentence pairs as a test set. Finally, for English-German, we use the News-test 2008 of $2,524$ sentence pairs as a development set and the News-test 2015 of $3,003$ sentence pairs as a test set.

[11] Other choices of the hyperparameter have also been tried, yet we did not observe significant differences in the model performance.

K	BLEU↑	METEOR↑	TER↓
Spanish → English			
1	24.5	30.7	58.2
8	24.7**	31.0**	57.8**
French → English			
1	21.5	28.8	62.1
8	21.6**	28.9**	61.9**
German → English			
1	22.2	28.3	60.2
8	22.3**	28.4**	60.3**

Table 9 Translation experiments, where scores marked with ** are significant (the p-level of 5% under paired bootstrap resampling). Note that better results correspond to larger BLEU, METEOR and to smaller TER.

The results are reported in Table 9, demonstrating a modest but significant translation improvement on all three languages.

7 Related work

Word alignment by itself is an interesting topic (e.g. see recent work (Simion et al, 2013; Tamura et al, 2014; Chang et al, 2014; Shen et al, 2015; Wang et al, 2015; Liu et al, 2015)). Like any statistical models, word alignment models suffer significantly from lacking of in-domain data for training. For instance, (Duh et al, 2010) suggests that training phrase-based SMT system might benefit from deploying a simple trick: They train statistical alignment models on a concatenation of both in-domain and a much larger out-of-domain dataset. In the end, they exclude out-of-domain data during phrase extraction. Similar findings are reported in (Gao et al, 2011). A domain-specific alignment model would produce better word alignment accuracy over the data, if it is interpolated with another general-domain alignment model that is trained on a much larger out-of-domain dataset. (Shah et al, 2010) shows that it would benefit from training word alignment with weighting sentence pairs according to their relevance to the domain. All the preliminary experiments are with translation accuracy instead of word alignment accuracy.

In terms of domain-focused 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. Another distant research line (Hua et al, 2005) trains different alignment models independently on different datasets. In the end, we can interpolate those models, with translation probability interpolation, distortion probability interpolation and fertility probability interpolation to improve word alignment accuracy. 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). In terms of working with the heterogeneous data, another distantly related research line focuses on data selection (Axelrod et al, 2011; Kirchhoff and Bilmes, 2014; Cuong and Sima'an, 2014b).

8 Conclusion

Besides the novelty of exploring the quality of word alignment in heterogeneous corpora, the 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 get by, but has not been exploited before.

With this work we also hope to draw attention of the research community's attention to the diversity of hidden domains in corpora usually used to train SMT systems - *Hansards*, *Europarl*, *JRC-Acquis*, *Common Crawl*, *United Nations* and *News Commentary*. We show that a fully unsupervised induction of domains for word alignment is hard to perform, but it is possible. Moreover, extensive experiments over seven different language pairs demonstrate significant improvements in alignment accuracy with an unsupervised induction of domains for word alignment.

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). The idea of disentangling domains for translation is not restricted to alignment models. In future work we hope to apply our research to other translation models, such as phrase-based models and reordering models.

References

Axelrod A, He X, Gao J (2011) Domain adaptation via pseudo in-domain data selection. In: EMNLP

Beal MJ (2003) Variational algorithms for approximate bayesian inference. PhD thesis, Gatsby Computational Neuroscience Unit, University College London

Bojar O, Buck C, Callison-Burch C, Federmann C, Haddow B, Koehn P, Monz C, Post M, Soricut R, Specia L (2013) Findings of the 2013 Workshop on Statistical Machine Translation. In: WMT

Bojar O, Chatterjee R, Federmann C, Haddow B, Huck M, Hokamp C, Koehn P, Logacheva V, Monz C, Negri M, Post M, Scarton C, Specia L, Turchi M (2015) Findings of the 2015 workshop on statistical machine translation. In: WMT

Brown PF, Pietra VJD, Pietra SAD, Mercer RL (1993) The mathematics of statistical machine translation: parameter estimation. Comput Linguist

Carpuat M, Goutte C, Foster G (2014) Linear mixture models for robust machine translation. In: WMT

Chang YW, Rush AM, DeNero J, Collins M (2014) A constrained viterbi relaxation for bidirectional word alignment. In: ACL

Cherry C, Foster G (2012) Batch tuning strategies for statistical machine translation. In: NAACL HLT

Clark JH, Dyer C, Lavie A, Smith NA (2011) Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In: ACL HLT (Short Papers)

Cuong H, Sima'an K (2014a) Latent domain phrase-based models for adaptation. In: EMNLP

Cuong H, Sima'an K (2014b) Latent domain translation models in mix-of-domains haystack. In: COLING

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the em algorithm. JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B 39(1):1–38

Denkowski M, Lavie A (2011) Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In: WMT

Devlin J, Zbib R, Huang Z, Lamar T, Schwartz R, Makhoul J (2014) Fast and robust neural network joint models for statistical machine translation. In: ACL

Duh K, Sudoh K, Tsukada H (2010) Analysis of translation model adaptation in statistical machine translation. In: IWSLT

Farajian MA, Bertoldi N, Federico M (2014) Online word alignment for online adaptive machine translation. In: Proceedings of the EACL 2014 Workshop on Humans and Computer-assisted Translation

Fraser A, Marcu D (2006) Semi-supervised training for statistical word alignment. In: COLING-ACL

Galley M, Manning CD (2008) A simple and effective hierarchical phrase reordering model. In: EMNLP

Gao Q, Vogel S (2010) Consensus versus expertise: A case study of word alignment with mechanical turk. In: NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk

Gao Q, Bach N, Vogel S (2010) A semi-supervised word alignment algorithm with partial manual alignments. In: WMT

Gao Q, Lewis W, Quirk C, Hwang MY (2011) Incremental training and intentional over-fitting of word alignment. In: MT Summit

Graca J, Pardal JP, Coheur L, Caseiro D (2008) Building a golden collection of parallel multi-language word alignment. In: LREC

Graça JaV, Ganchev K, Taskar B (2010) Learning tractable word alignment models with complex constraints. Comput Linguist 36(3):481–504, DOI 10.1162/coli˙a˙00007, URL http://dx.doi.org/10.1162/coli˙a˙00007

Holmqvist M, Ahrenberg L (2011) A gold standard for english-swedish word alignment. In: Proceedings of the 18th Nordic Conference of Computational Linguistics NODALIDA 2011, 11

Hua W, Haifeng W, Zhanyi L (2005) Alignment model adaptation for domain-specific word alignment. In: ACL

Huck M, Peitz S, Freitag M, Nuhn M, Ney H (2012) The rwth aachen machine translation system for wmt 2012. In: WMT

Kirchhoff K, Bilmes J (2014) Submodularity for data selection in machine translation. In: EMNLP

Koehn P (2005) Europarl: A Parallel Corpus for Statistical Machine Translation. In: Proceedings of MTSummit

Koehn P, Och FJ, Marcu D (2003) Statistical phrase-based translation. In: NAACL HLT

Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen W, Moran C, Zens R, Dyer C, Bojar O, Constantin A, Herbst E (2007) Moses: Open source toolkit for statistical machine translation. In: ACL on Interactive Poster and Demonstration Sessions

Liang P, Taskar B, Klein D (2006) Alignment by agreement. In: HLT-NAACL

Liu C, Liu Y, Sun M, Luan H, Yu H (2015) Generalized agreement for bidirectional word alignment. In: Proceedings of the EMNLP

Mansour Y, Mohri M, Rostamizadeh A (2009a) Domain adaptation with multiple sources. In: Proceedings of NIPS

Mansour Y, Mohri M, Rostamizadeh A (2009b) Multiple source adaptation and the rÉnyi divergence. In: Proceedings of UAI

Mihalcea R, Pedersen T (2003) An evaluation exercise for word alignment. In: Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond - Volume 3

Och FJ, Ney H (2003) A systematic comparison of various statistical alignment models. Comput Linguist 29(1):19–51, DOI 10.1162/089120103321337421, URL http://dx.doi.org/10.1162/089120103321337421

Och FJ, Gildea D, Khudanpur S, Sarkar A, Yamada K, Fraser A, Kumar S, Shen L, Smith D, Eng K, Jain V, Jin Z, Radev D (2004) A smorgasbord of features for statistical machine translation. In: HLT-NAACL

Papineni K, Roukos S, Ward T, Zhu WJ (2002) Bleu: A method for automatic evaluation of machine translation. In: ACL

Riley D, Gildea D (2012) Improving the ibm alignment models using variational bayes. In: Proceedings of ACL (Short Paper)

Shah K, Barrault L, Schwenk H (2010) Translation model adaptation by resampling. In: WMT

Shen S, Liu Y, Sun M, Luan H (2015) Consistency-aware search for word alignment. In: Proceedings of the EMNLP

Simion A, Collins M, Stein C (2013) A convex alternative to ibm model 2. EMNLP

Snover M, Dorr B, Schwartz R, Micciulla L, Makhoul J (2006) A study of translation edit rate with targeted human annotation. In: AMTA

Steinberger R, Pouliquen B, Widiger A, Ignat C, Erjavec T, Tufis D, Varga D (2006) The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. In: LREC

Steinberger R, Eisele A, Klocek S, Pilos S, Schlüter P (2012) Dgt-tm: A freely available translation memory in 22 languages. In: LREC

Tam YC, Lane I, Schultz T (2007) Bilingual lsa-based adaptation for statistical machine translation. Machine Translation 21(4):187–207, DOI 10.1007/s10590-008-9045-2, URL http://dx.doi.org/10.1007/s10590-008-9045-2

Tamura A, Watanabe T, Sumita E (2014) Recurrent neural networks for word alignment model. In: ACL

Vogel S, Ney H, Tillmann C (1996) Hmm-based word alignment in statistical translation. In: COLING, pp 836–841, URL http://dblp.uni-trier.de/db/conf/coling/coling1996.html#VogelNT96

Wang X, Utiyama M, Finch A, Watanabe T, Sumita E (2015) Leave-one-out word alignment without garbage collector effects. In: Proceedings of the EMNLP

Zhang H, Chiang D (2014) Kneser-ney smoothing on expected counts. In: Proceedings of ACL

Zhao B, Xing EP (2008) Hm-bitam: Bilingual topic exploration, word alignment, and translation. In: NIPS