Generic and Specialized Word Embeddings for Multi-Domain Machine Translation

Minh Quang Pham, Josep-Maria Crego, François Yvon, Jean Senellart

Generic and Specialized Word Embeddings for Multi-Domain Machine Translation

MinhQuang Pham[†‡], *Josep Crego*[†], *François Yvon*[‡], *Jean Senellart*[†]

[†]SYSTRAN / 5 rue Feydeau, 75002 Paris, France
firstname.lastname@systrangroup.com
[‡]LIMSI, CNRS, Université Paris-Saclay 91405 Orsay, France
firstname.lastname@limsi.fr

Abstract

Supervised machine translation works well when the train and test data are sampled from the same distribution. When this is not the case, *adaptation* techniques help ensure that the knowledge learned from out-of-domain texts generalises to in-domain sentences. We study here a related setting, *multi-domain adaptation*, where the number of domains is potentially large and adapting separately to each domain would waste training resources. Our proposal transposes to neural machine translation the feature expansion technique of (Daumé III, 2007): it isolates domain-agnostic from domain-specific lexical representations, while sharing the most of the network across domains. Our experiments use two architectures and two language pairs: they show that our approach, while simple and computationally inexpensive, outperforms several strong baselines and delivers a multi-domain system that successfully translates texts from diverse sources.

1. Introduction

Owing to the development of flexible and powerful architectures based on neural networks [1, 2, 3, 4], Machine Translation (MT) has made significant progresses over the past years and constitute to date the standard for most production engines. The development of MT systems, be they neural or statistical, require very large parallel corpora consisting of millions of sentence pairs, a resource that only exist in very few application domains and language pairs. A lot of the recent research effort has thus focused on developing MT systems in restricted data conditions, for instance building multilingual MTs which enable zero-shot translation [5, 6, 7]. Another important scenario for industrial MT is to adapt a neural system trained using parallel data in one domain to the peculiarity of other domains.

Domain Adaption (DA) in MT is an old issue [8, 9], which comes in various guises and for which a number of solutions have been studied. See the recent survey of [10] for neural MT. The typical setting is *supervised adaptation*, where a (small) amount of data in the target domain of interest is used to fine-tune the parameters of a system trained on a large amount of texts in a source domain. We study here a different scenario, *multi-domain adaptation* [11, 12], where we would like to use heterogenous data sources to train a unique system that would work well for all domains. This allows us to be both data efficient (all data is used to train all domains) and computationally efficient (we only train one system). Multi-domain adaptation is conceptually close to multilingual MT, or more generally to multi-task learning [13] and can be approached in a number of ways.

We adapt here ideas of [14] to neural MT. Our main hypothesis is that domains mostly differ at the lexical level, due to cross-domain polysemy, which motivates domain specific embeddings. By contrast, the deeper layers, which arguably model more abstract linguistic phenomena, are made shareable across domains. To this end, we design word embeddings containing a generic and several domain-specific regions. We experiment with four domains, two neural architectures and two language pairs and find that our technique yields effective multi-domain NMTs, outperforming several baselines. Our contributions are thus as follows: we adapt and implement the ideas of [14] for two NMT architectures; we provide experimental evidence that show the effectiveness of this technique; we evaluate the ability of our networks to dynamically accommodate new domains; and we introduce a new technique to analyze word polysemy using embeddings, which comforts the assumption that their variation across domains actually reflects a variation of senses.

2. Lexicalized domain representations

2.1. Multi-domain machine translation

Multi-domain machine translation is formalized as follows: we assume observations taking the form of domain-tagged sentence pairs $[(x, y), i]$, with x in the source language, y in the target language and i a domain tag in $[1 \dots d]$. We further assume a two-stage sampling process: first select a domain i according to $p(i)$, then select a sentence pair according to a domain specific distribution D_i. Our objective is to find a tuple of parameters $\{\theta_1 \dots \theta_d\} \in \mathbb{R}^D \times \dots \times \mathbb{R}^D$ minimizing:

$$\sum_{i \in [1..d]} p(i) E_{(x,y)\sim D_i}[-log(p_{\theta_i}(y|x, i))]. \quad (1)$$

The training data for domain i is denoted C_i.

A straightforward solution is to process each domain separately, computing the value θ_i^* that minimizes the empirical

loss on C_i. This strategy is only effective if we have sufficient training data for each domain; when this is not the case, some estimates θ_i^* may be far from their optimal value. The alternative we consider here constraints each parameter θ_i to be made of two parts: $\theta_i = [\theta_s; \theta_i']$. $\theta_s \in R^{D_g}$ is shared across all domains, while the second part $\theta_i' \in \mathbb{R}^{D_i}$ is only used in domain i. The parameter set is now much more constrained, yet we expect that tying parameters across domains will yield better estimates for θ_s due to a larger training corpus. In this setting, the optimization program defined by equation (1) can no longer be performed separately for each training corpus.

2.2. Lexicalized domain embeddings

To actually implement this idea for NMT, we need to define the subset of parameters that will be shared across domains. Our hypothesis is that domain specificities can be confined to the lexical level and we define θ_s to contain all the network parameters except for a subpart of the word embeddings. For each word v, the embedding vector $e(v)$ is thus decomposed as $e(v) = [e_g(v); e_1(v); \ldots; e_d(v)]$, where $e_g(v)$ stores the generic lexical embedding, while $e_i(v)$ stores the subpart that is specific to domain i. In our NMT architectures, the actual embedding layer composes these vectors linearly to generate the word embedding for domain k according to:

$$\begin{aligned}\tilde{e}_k(v) =& M_g e_g(v) + \sum_{i \in [1,..,d]} M_i \times e_i(v) \times \delta(i = k) \\ =& M[e_g(v); e_1'(v,k) \ldots; e_d'(v,k)], \end{aligned} \quad (2)$$

where $\delta()$ is the indicator function, M is the matrix made of blocks $M_g, M_1 \ldots, M_d$, and $e_i'(v,k)$ is the masked embedding: $e_i'(v,k) = e_i(v) * \delta(i = k)$.

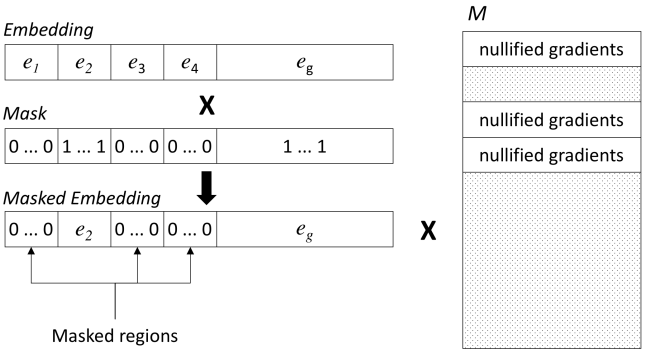

Figure 1: Lexicalized domain embeddings. When processing a sample from domain 2, we only activate the corresponding parameter region (θ_2) in the input embeddings; the remaining domain-specific parts are zeroed out and do not receive any update. The generic part is always active and is updated irrespective of the input domain.

Making sure that the actual embedding do not contain any zero is important for the Transformer model, since the lexical representations are then added to the positional encoding, which would undo the effect of domain masking, and propagate a gradient even to regions that should not be modified. With our design, we make sure that during backprogation, the matrix M receives gradient 0 at regions corresponding to deactivated regions in the word embedding. Those regions are also masked in forward step, thus do not interfere the training on the domains to which they are not assigned (see Figure 1). Our architecture is thus readily compatible with any NMT architecture, where we simply replace standard embedding layers by the embeddings defined in equation (2). In our experiments, we consider both the attentional RNN architecture of [2] and the Transformer architecture of [4].

3. Experiments

3.1. Domains and data

We experiment with two language pairs (English-French, English-German) and data originating from three domains, corresponding to texts from three European institutions: the European Parlement (EPPS) [15], the European Medicines Agency (EMEA), the European Central Bank (ECB) [16], In addition, for English-French we also use IT-domain corpora obtained from the OPUS web site[1] corresponding to KDE, Ubuntu, GNOME and PHP datasets (IT). We randomly split those corpora into training, validation and test sets (see statistics in Table 1). Validation sets are used to chose the best model according to the average BLEU score [18].

Corpus	Train	Valid	Test
English → French			
EMEA	1.09M	1,000	$1{,}000_{(300)}$
ECB	0.19M	1,000	1,000
EPPS	2.01M	1,000	1,000
IT	0.54M	1,000	1,000
English → German			
EMEA	1.11M	1,000	$1{,}000_{(300)}$
ECB	0.11M	1,000	1,000
EPPS	1.92M	1,000	1,000

Table 1: Corpora statistics.

Note that the EMEA dataset distributed on the OPUS site contains multiple sentence duplicates. We therefore report below two numbers as $S_{(T)}$: the first (S) is comparable to what has been published on earlier studies (eg. [17]), the second one (T) is obtained by making the test entirely disjoint from the training (700 duplicated sentences are discarded).

To reduce the number of lexical units and make our systems open-vocabulary, we apply Byte-Pair Encoding [22] separately for each language with 30,000 merge operations.

[1] `http://opus.nlpl.eu`

3.2. Baselines

To validate our findings, we compared lexicalized domain embedding models with standard models using both attentional Recurrent Neural Networks (RNNs) [2] and the Transformer architecture of [4]. Our baselines consist of:

- generic models trained with a simple concatenation of all corpora (`Mixed`);
- models tuned separately on each domain for respectively (10000, 15000, 5000) iterations using in-domain data ($\texttt{ft}_{EMEA}$, $\texttt{ft}_{EPPS}$, $\texttt{ft}_{ECB}$);
- models using domain tags as in [23] (`DC`);

For all models, we set the embeddings size equal to 512; the size of hidden layers is equal to 1024 for RNNs and 512 for Transformer. Other important configuration details are as follows: Transformer models use multi-head attention with 8 heads in each of the 6 layers; the inner feedforward layer contains 2048 cells; RNN models use 1 layer on both sides: a bidirectional LSTM encoder and a unidirectional LSTM decoder with attention. The domain control systems are exactly as their baseline counterparts (RNN and Transformer), with an additional 2 cells encoding the domain on the input layer. To train NMT systems, we use Adam, with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\alpha = 0.0005$ for RNNs; with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$, with Noam decay [4] for Transformer ($warmup\_steps = 4000$). In all cases, we use a batch size of 128 and a dropout rate of 0.1 for all layers. All our systems are implemented in OpenNMT-tf[2] [24].

3.3. Implementing Lexicalized Domain Representations

In order to implement lexicalised domain representations (henceforth `LDR`), we split the embedding vector into four regions: 3 are domain specific and 1 is generic, with sizes $[8, 8, 8, 488]$ respectively. If a sentence originates from domain i, the domain specific regions for all domains $j \neq i$ will be zeroed out while the other regions are activated (cf. Figure 1). We then use a dense layer of size 512 to fuse the region for the active domain and the "generic" region. Training is formalised in algorithm 1. Note that each iteration of algorithm 1 uses 2 batches: a "generic" batch updating only the generic region; and a "domain-specific" batch updating just the domain-specific parameters.

The batch selection procedure (step 2 of algorithm 1) ensures that the number of examples of each domain used in training follows the distribution of the training data. In our experiments, this means that sentences from the Europarl domain will be selected more frequently that the two other domains. We also consider a more balanced sampling procedure, where i is selected according to distribution $[\frac{\sqrt{|C_i|}}{\sum_{i\in[1,..,d]}\sqrt{|C_i|}}]$. The corresponding results are reported as $\texttt{LDR}^{0.5}$.

[2] `https://github.com/OpenNMT/OpenNMT-tf`

Procedure 1 Multi-domain Training

Input: Corpora $C_i, i \in [1, .., d]$ for d domains, Batch size B

1: **repeat**
2: Randomly pick $i \in [1, .., d]$ w.r.t the multinomial distribution $[\frac{|C_i|}{\sum_{i\in[1,..,d]}|C_i|}]$.
3: Randomly pick B sentences from C_i.
4: Activate only generic region to create generic batch, denoted W_g.
5: Compute gradient of θ_s, $\frac{\partial L}{\partial \theta_s}$ using W_g.
6: Activate domain-specific and generic regions to create domain-specific batch W_i
7: Compute gradient of domain-specific parameters θ_i, $\frac{\partial L}{\partial \theta_i}$ using W_d.
8: Update parameters θ_s using $\frac{\partial L}{\partial \theta_s}(W_g)$ and θ_i using $\frac{\partial L}{\partial \theta_i}(W_i)$
9: **until** convergence

3.4. Results

Results are summarized respectively in Table 2 for the Transformer systems and Table 3 for the RNN systems, where we report BLEU [18] scores computed after detokenization.[3]

First, we observe that Transformer is consistently better than RNNs and that fine-tuning on a domain-specific corpus, when applicable, is almost the best way to optimize the performance on that domain.[4] Note that fine-tuning however yields a marked (even sometimes catastrophic, eg. for the EMEA-tuned Transformer system) decrease in performance for the other domains.

Our approach ($\texttt{LDR}_{oracle}$) is consistently better than the `Mixed` strategy, with gains that range from very large (for EMEA and ECB) to unsignificant (for EPPS in most conditions). This means that our architecture is somehow able to compensate for the data unbalance and to raise the performance of the multi-domain system close to the best (fine-tuned) system in each domain. We even observe rare cases where the $\texttt{LDR}_{oracle}$ system outperforms fine-tuning (eg. Transformer en:de in the EMEA domain). $\texttt{LDR}_{oracle}$ is also better than Domain Control in three conditions out of four, `DC` being seemingly a better choice for the RNN than for the Transformer architecture. As expected, ignoring the true domain label yields a light drop in performance: this is reflected in the results of $\texttt{LDR}_{pred}$, which relies on automatically predicted domain labels.[5] Note that this decrease is however hardly significant, showing that our architecture is quite robust to noisy labels. Even in the worst case scenario where all domain tags are intentionally wrong ($\texttt{LDR}_{wrong}$), we see that the generic part still ensures a satisfying level of performance. A last contrast is with $\texttt{LDR}^{0.5}_{oracle}$ where we change

[3] As explained above, we report two numbers when testing with EMEA, except for the fine-tuning scenarios when tuning on ECB and EPPS.

[4] This is not so clear for EPPS, where fine-tuning does not seem to help.

[5] Our domain classifier uses a bi-LSTM RNN encoder, followed by a simple softmax layer. Its precision on a development set exceeds 95%.

the distribution of training sentences to decrease the weight of EPPS data and increase the number of ECB samples. As a result, we see a small decrease for EMEA and EPPS, and a large boost for ECB. This shows that our technique can be used in conjunction to other well known strategies for performing domain adaptation.

Model	EMEA	EPPS	ECB	Avg.
English→French				
Mixed	$67.69_{47.60}$	37.50	53.49	52.89
FT_{EMEA}	$76.77_{49.43}$	17.16	11.99	35.30
FT_{EPPS}	20.86	37.04	24.53	27.47
FT_{ECB}	26.93	27.09	**56.52**	36.84
DC	$67.87_{45.42}$	37.31	54.14	53.10
LDR_{oracle}	$74.26_{\mathbf{49.90}}$	37.67	54.07	55.33
$\text{LDR}^{0.5}_{oracle}$	$\mathbf{74.95}_{49.38}$	37.35	55.91	**56.07**
LDR_{pred}	$74.29_{49.84}$	**37.73**	54.01	55.34
LDR_{wrong}	$72.95_{49.78}$	37.62	53.35	54.64
English→German				
Mixed	$64.57_{42.99}$	26.47	68.67	53.23
FT_{EMEA}	$68.35_{42.97}$	17.02	32.87	39.41
FT_{EPPS}	36.19	26.29	40.71	34.39
FT_{ECB}	24.72	18.36	**74.05**	39.04
DC	$63.48_{42.98}$	26.27	66.95	52.23
LDR_{oracle}	$70.90_{\mathbf{46.12}}$	26.30	68.90	55.36
$\text{LDR}^{0.5}_{oracle}$	$\mathbf{71.31}_{45.23}$	25.98	73.74	**57.01**
LDR_{pred}	$70.89_{\mathbf{46.12}}$	**26.53**	68.63	55.35
LDR_{wrong}	$69.51_{43.50}$	26.31	66.86	54.22

Table 2: BLEU scores for Transformer systems

We also compare our architecture with the multi-domain model of [17] (WDCMT) for the pair English→French. We use the author's implementation[6] that is composed of one bidirectional Gated recurrent units (GRU) layer on the encoder side; and one unidirectional conditional GRU layer on the decoder side; the dimension of "domain" layers is 300. The direct comparison with our RNN is difficult, as both networks differ in many ways: framework, cell types, *etc*. Results in Table 4 therefore use a variant of our model that makes it more similar to the WDCMT network. In particular, this variant also uses a single GRU layer in the encoder and a single conditional GRU layer in the decoder ($\text{LDR}^{condgru}_{pred}$). As can be seen in this table, our model is on average comparable to WDCMT, while using a much simpler design.

4. Complementary experiments

4.1. Balancing generic and domain representations

An important practical question concerns the balance between the generic and the domain-specific part of the embeddings. In the limit where the domain specific part is very small, we should recover the performance of the Mixed system; conversely, we expect to see a less effective sharing of data across domains by increasing the domain-specific regions. Table 5 reports the result of a series of experiments for the Transformer architecture (English-French) with varying domain-specific sizes allocating between 4 and 64 cells for domain-specific information, and the complement to 512 for the generic part. The differences are overall quite small in our experimental setting, where the training data is relatively limited and does not require to use a large embedding size. We therefore decided to allocate 8 cells for the domain specific part. This suggests that we could easily accommodate more domains with the same architecture and even reserve some regions to handle supplementary data (see below).

[6] http://github.com/DeepLearnXMU/WDCNMT

Model	EMEA	EPPS	ECB	Avg.
English→French				
Mixed	$65.42_{45.11}$	34.70	51.38	50.50
FT_{EMEA}	$72.06_{\mathbf{47.33}}$	18.62	16.78	35.82
FT_{EPPS}	35.47	34.61	39.56	36.55
FT_{ECB}	21.93	22.60	51.53	32.02
DC	$68.26_{43.76}$	35.13	50.09	51.16
LDR_{oracle}	$71.73_{46.30}$	**35.21**	50.91	52.62
$\text{LDR}^{0.5}_{oracle}$	$71.70_{46.41}$	34.24	**52.37**	**52.77**
LDR_{pred}	$\mathbf{72.76}_{46.35}$	35.10	50.38	52.75
LDR_{wrong}	$62.10_{43.29}$	34.17	48.79	48.35
English→German				
Mixed	$57.37_{37.94}$	23.10	63.54	48.00
FT_{EMEA}	$65.64_{\mathbf{44.71}}$	12.36	15.93	31.31
FT_{EPPS}	24.90	22.98	26.26	24.71
FT_{ECB}	41.80	15.97	71.07	42.95
DC	$62.53_{39.25}$	**23.74**	65.71	50.66
LDR_{oracle}	$\mathbf{63.43}_{40.04}$	22.66	64.40	50.16
$\text{LDR}^{0.5}_{oracle}$	$63.27_{38.16}$	21.83	**69.55**	**51.55**
LDR_{pred}	$63.17_{39.92}$	22.51	64.00	49.89
LDR_{wrong}	$56.84_{37.05}$	22.06	61.66	46.85

Table 3: BLEU scores for RNN systems

Model	EMEA	EPPS	ECB	Avg.
English→French				
LDR_{pred}	$\mathbf{72.76}_{46.35}$	35.10	50.38	**52.75**
$\text{LDR}^{condgru}_{pred}$	$71.70_{46.21}$	35.09	51.22	52.67
WDCMT	$68.76_{45.29}$	**35.71**	**52.75**	52.40

Table 4: BLEU scores for RNN systems. Comparison between WDCMT and LDR_{pred} built using conditional GRUs.

4.2. Additional Domain Scenario

We now evaluate the ability of our model to integrate new domains, a very common scenario for industrial MT. In this setting, we consider that we have a model (LDR_{oracle}) trained as before for EMEA, EPPS and ECB during 200,000 itera-

$\texttt{LDR}_{oracle}$	EMEA	EPPS	ECB	Avg.
English→French				
size=4	$74.65_{49.61}$	37.42	54.49	55.52
size=8	$74.26_{49.90}$	37.67	54.07	55.33
size=16	$74.15_{49.10}$	**37.78**	**54.56**	55.50
size=32	$\mathbf{75.10}_{48.61}$	37.64	54.29	**55.68**
size=64	$74.50_{\mathbf{50.17}}$	37.27	54.50	55.42

Table 5: BLEU scores for the Transformer architecture for varying domain-specific embedding sizes

tions, which needs to process new training data from the IT domain. Assuming that we have reserved extra empty embedding cells[7] for this domain, we resume training with 4 domains during 100,000 additional iterations, yielding an updated model $\texttt{LDR}^{*}_{oracle}$. Results for the English→French language pair are in Table 6, where for comparison purposes we also report numbers obtained with continued training with the `Mixed` model, training for the same number of iterations and using the same four datasets ($\texttt{Mixed}^{*}$).

Model	EMEA	EPPS	ECB	IT	Avg.
English→French					
`Mixed`	$67.69_{47.60}$	37.50	53.49	13.91	43.15
$\texttt{Mixed}^{*}$	$66.49_{45.79}$	37.59	55.07	51.78	52.73
$\texttt{LDR}_{oracle}$	$74.26_{\mathbf{49.90}}$	**37.67**	54.07	13.40	44.85
$\texttt{LDR}^{*}_{oracle}$	$\mathbf{76.17}_{49.71}$	37.48	**55.12**	**55.24**	**56.00**

Table 6: BLEU scores for the Transformer architecture when including IT as additional domain

As expected, a huge improvement in performance is observed for the IT test set when learning includes in-domain data for both models, with $\texttt{LDR}^{*}_{oracle}$ outperforming $\texttt{Mixed}^{*}$ by a wide margin. It is interesting to see that this additional data has also a positive impact on other test sets: both models similarly increase their performance for the ECB domain, and $\texttt{LDR}^{*}_{oracle}$ additionally improves the results for the EMEA test, which is not the case for $\texttt{Mixed}^{*}$; finally, using IT data does not impact the quality of translations for the EPPS domain of any of the models. Overall better results are obtained by our $\texttt{LDR}^{*}_{oracle}$ model trained with data from an additional source, demonstrating the ability of our architecture to seamlessly integrate a new domain.

4.3. Analysis of Word Embeddings

One of our main assumptions is that the difference between domains can be confined at the lexical level, warranting our decision to specialise lexical representations for each domain, while the remaining part of the network is shared across domains. Linguistically, this assumption relates to the classical "one sense per collocation" [25] and corresponds to the fact that in many cases, polysemy corresponds to variation of use across domain. In its weaker form, it allows us to assume that all occurrences of a given form in a given domain correspond to the same sense and share the same representation; the same form occurring in different domains is allowed to have one distinct embedding per domain, which may help capture polysemy and lexical ambiguity in translation.

To check this hypothesis, we performed the following analysis of embeddings learned with the multi-domain Transformer system for English:French. For each unit[8] in our English dictionary, we compute the k nearest neighbours for each domain $i \in [1 \dots d]$, where the distance between unit u and v for domain i is the cosine distance in the corresponding embedding space, ie. assuming that the actual embedding of v for domain i is $e(v, i) = M_g e_g(u) + M_i e_i(v)$ (cf. equation (2)). This process yields d lists of k nearest neighbours. A small intersection should then be a sign of a variation of use across domains; conversely, an near-identical set of neighbours across domains should reflect the stability of word use. Table 7 list the 10 units with the smaller (respectively larger) intersection (we use $k = 10$ and $d = 3$).

Polysemic "words"	Monosemic "words"
ases (0)	obtain (10)
impairment (1)	virtually (10)
convenience (1)	represent (10)
oring (1)	safety (10)
ums (1)	defence (10)
turnover (1)	coordinated (10)
occurrence (1)	handling (10)
tent (2)	July (10)
ture (2)	previous (10)
mation (2)	better (10)

Table 7: Analyzing the variation of embeddings across domains. For each word or subword we also report the size of the intersection (between 0 and 10).

Let us first consider the full words in the left column of Table 7. The case of *impairment* is pretty clear, occuring in EMEA mostly in terms such as "hepatic impairment" or "renal impairment", and translating into French as *insuffisance*. In ECB, its collocates are quite different and impairment often occurs in terms such as "cost subject to impairments" (French: *coût soumis à des réductions de valeur*). Likewise, "convenience" seems to have its general meaning ("for convenience") in EMEA, but appears in ECB in the specific context of "convenience credit card" (French *carte de crédit à remboursement différé*). We finally see the same phenomena with "turnover", which is consistently translated with its economic meaning (French *chiffre d'affaire*) in ECB

[7] For this experiment, word embeddings contain 480 cells for the generic region and 32 cells for domain specific regions (8 cells x 4 regions).

[8] In this study, we work with BPE units meaning that in many cases we observe the variation of use of *word parts*. As we work with a large inventory, many of these units still correspond to actual words and we focus on these in our comments. We also restrict our analysis to words that occur at least 30 times in each domain, to ensure that each domain-specific region is updated during training.

and EPPS, but whose collocates in EMEA ("bone turnover", "hepatic turnover") are associated with the idea of the cell renewall process, yielding translations such as *remodelage osseux* in French. Subword units can be analysed in the same ways: "ums", for instance, appears in words such as "gums", "serums", "vacuums" in EMEA; in ECB, "ums" is mostly the suffix of "maximums", "minimums", or "premiums"; EPPS finally contains a more diverse set of "-ums" ending words ("stadium", "forum", equilibrium", etc).

Let us now consider the list of putative monosemic words (on the right part of Table 7), ie. words for which the nearest neighbors are the same in all domains. This list contains mostly words for which we do not expect much variation in translation: adjectives ("previous", "better"), adverbs ("virtually"), generic verbs ("handling", "coordinated"). Further down this list, we will also find prepositions ("at", "in"), auxiliary ("been") etc.

5. Related Work

Domain adaptation (DA) is a vexing problem in NLP, which appears in a wide range of practical situations and data scenarios (eg. supervised vs. unsupervised adaptation), and has been thoroughly studied from a number of perspectives, ranging from theoretical analysis to more applied work, and for which many solutions have been proposed. The literature of DA for Machine Translation reflects this diversity and typically distinguishes data-based approaches from model-based approaches [26, 10].

The most common adaptation scenario uses (mostly) out-of-domain data in training, while testing on a low-resource in-domain set of texts. In this setting, *data-based approaches* aim to bias the distribution of the train (out-of-domain) data towards matching that of the target domain, using data selection techniques [27, 9, 28], or generating adapted pseudo-parallel data through back-translation [29, 30, 31, 32]. *Model-centric approaches* build domain-adapted models by combining (eg. with mixture weights) multiple data sources or multiple systems [8, 33], or by biasing the training objective towards the desired domain using in-domain adaptation data [34, 35, 36]. Another approach worth mentioning integrates domain information (for instance a domain language model) in the decoding algorithm [37]. Our scenario is a bit different as we aim to train a unique system that will work well for several domains. This corresponds to a practical scenario in the industry, where one would like to maintain one single multi-domain engine, trained on all the available (heteregeneous) sources of data. Our primary source of inspiration is the proposal of [14] who proposes to use several copies of the same features (one "generic" shared across domains, and one for each domain), letting the training adjust their respective weights. This work has been reanalyzed in a Bayesian framework in [38], and revisited notably in [39]. Following up on [40], the recent proposals of [41] apply the same idea with neural architectures, using domain specific masking to zero out the parameters modeling domains irrelevant to the current input sentence. Compared to our work, these techniques are used in deep network layers and applied to sequence labelling tasks.

The multi-domain scenario in NMT has been studied in a number of recent works. [12] learn a generic system and propose to dynamically adapt the network weights in an unsupervised manner using a small sample of training data that resembles the test data, an idea already explored in [11]. Their main contribution is to propose a method to relate the amount of adaptation of the network parameters to the similarity between the adaptation sample and the test sentence: the higher the similarity, the more agressive the adaptation. By analogy with the proposal of [7] for multilingual NMT, [23] and [42] separately propose to extend the representation of the source text with a domain tag. Our model also modifies input representations, but allows each source word to have a domain-specific representation, thereby improving training of the shared parts of the network.

Similarly to our approach, [17] attempts to separate on the encoder side domain-specific representation from generic representations in two different sub-networks, where generic versus domain-specific representations automatically emerge from two adversarial networks. The decoder side can thus attend separately to these two representations to generate its output. In this approach, the shared and domain-specific parts are kept separated in the deeper layers of the network, whereas we try to localise the differences between domains at the lexical level, based on a much cheaper computational architecture. Another trait of this proposal is the ability to automatically infer domain information for test sentences; as we have shown, our architecture can also effectively accommodate sentences lacking domain information.

6. Conclusions

In this paper, we have presented a new technique for multi-domain machine translation, adapting the "frustratingly easy" idea of [14] to two standard NMT architectures. Our experiments have shown that for both architectures and for two language pairs, our multi-domain models improve over several baselines of the literature and that it is robust to noise in domain labels. It is noticeable that these results are obtained without impacting the architecture or training complexity , making our approach an effective baseline for further studies in multi-domain translation. We have also shown that our approach can dynamically handle new domains; and that the domain-specific embeddings often reflect differences of senses. In our future work, we intend to develop these ideas so as to make the architecture more self-configurable and able to adapt the size of the domain-specific regions depending upon the actual variation of use across domains; we also would like to find additional ways to make the architecture able to integrate an arbitrary number of new domains in a dynamic fashion, as this is an important requirement in industrial systems.

7. References

[1] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, Doha, Qatar, October 2014, pp. 103–111. [Online]. Available: http://www.aclweb.org/anthology/W14-4012

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations*, ser. ICLR, San Diego, CA, 2015. [Online]. Available: https://arxiv.org/pdf/1409.0473.pdf

[3] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70, International Convention Centre, Sydney, Australia, 2017, pp. 1243–1252. [Online]. Available: http://proceedings.mlr.press/v70/gehring17a.html

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[5] O. Firat, K. Cho, and Y. Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2016, pp. 866–875. [Online]. Available: http://www.aclweb.org/anthology/N16-1101

[6] T.-H. Ha, J. Niehues, and A. Waibel, "Toward multilingual neural machine translationwith universal encoder and decoder," in *Proceedings of the International Workshop on Spoken Language Translation*. Vancouver, Canada: IWSLT, 2016.

[7] M. Johnson, M. Schuster, Q. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. a. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, 2017. [Online]. Available: https://transacl.org/ojs/index.php/tacl/article/view/1081

[8] G. Foster and R. Kuhn, "Mixture-model adaptation for SMT," in *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic, 2007, pp. 128–135. [Online]. Available: http://www.aclweb.org/anthology/W/W07/W07-0717

[9] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11, Edinburgh, United Kingdom, 2011, pp. 355–362. [Online]. Available: http://dl.acm.org/citation.cfm?id=2145432.2145474

[10] C. Chu and R. Wang, "A survey of domain adaptation for neural machine translation," in *Proceedings of the 27th International Conference on Computational Linguistics*, ser. COLING 2018, Santa Fe, New Mexico, USA, 2018, pp. 1304–1319. [Online]. Available: http://aclweb.org/anthology/C18-1111

[11] R. Sennrich, H. Schwenk, and W. Aransa, "A multi-domain translation model framework for statistical machine translation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 832–840. [Online]. Available: https://www.aclweb.org/anthology/P13-1082

[12] M. A. Farajian, M. Turchi, M. Negri, and M. Federico, "Multi-domain neural machine translation through unsupervised adaptation," in *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark, Sept. 2017, pp. 127–137. [Online]. Available: https://www.aclweb.org/anthology/W17-4713

[13] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, July 1997. [Online]. Available: https://doi.org/10.1023/A:1007379606734

[14] H. Daumé III, "Frustratingly easy domain adaptation," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, 2007, pp. 256–263. [Online]. Available: http://aclweb.org/anthology/P07-1033

[15] P. Koehn, "Europarl: A parallel corpus for Statistical Machine Translation," in *2nd Workshop on EBMT of MT-Summit X*, Phuket, Thailand, 2005, pp. 79–86.

[16] J. Tiedemann, "News from OPUS - A collection of multilingual parallel corpora with tools and interfaces," in *Recent Advances in Natural Language Processing*, N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, Eds. Borovets, Bulgaria: John Benjamins, Amsterdam/Philadelphia, 2009, vol. V, pp. 237–248.

[17] J. Zeng, J. Su, H. Wen, Y. Liu, J. Xie, Y. Yin, and J. Zhao, "Multi-domain neural machine translation

with word-level domain context discrimination," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 447–457. [Online]. Available: http://aclweb.org/anthology/D18-1041

[18] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02, Stroudsburg, PA, USA, 2002, pp. 311–318.

[19] O. Dušek, J. Hajič, J. Hlaváčová, J. Libovický, P. Pecina, A. Tamchyna, and Z. Urešová, "Khresmoi summary translation test data 2.0," 2017, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. [Online]. Available: http://hdl.handle.net/11234/1-2122

[20] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, and A. Tamchyna, "Findings of the 2014 workshop on statistical machine translation," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Baltimore, Maryland, USA: Association for Computational Linguistics, June 2014, pp. 12–58. [Online]. Available: http://www.aclweb.org/anthology/W/W14/W14-3302

[21] M. Paul, M. Federico, and S. Stüker, "Overview of the IWSLT 2010 evaluation campaign," in *International Workshop on Spoken Language Translation (IWSLT) 2010*, ser. IWSLT, Paris, France, 2010, pp. 3–27. [Online]. Available: https://www.isca-speech.org/archive/iwslt_10/papers/slta_003.pdf

[22] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, Aug. 2016, pp. 1715–1725. [Online]. Available: https://www.aclweb.org/anthology/P16-1162

[23] C. Kobus, J. Crego, and J. Senellart, "Domain control for neural machine translation," in *Proceedings of the International Conference Recent Advances in Natural Language Processing*, ser. RANLP 2017. Varna, Bulgaria: INCOMA Ltd., 2017, pp. 372–378. [Online]. Available: https://doi.org/10.26615/978-954-452-049-6_049

[24] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, "OpenNMT: Open-source toolkit for neural machine translation," in *Proceedings of ACL 2017, System Demonstrations*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 67–72. [Online]. Available: http://aclweb.org/anthology/P17-4012

[25] D. Yarowsky, "One sense per collocation," in *Proceedings of the Workshop on Human Language Technology*, ser. HLT '93. Stroudsburg, PA, USA: Association for Computational Linguistics, 1993, pp. 266–271. [Online]. Available: https://doi.org/10.3115/1075671.1075731

[26] C. Chu, R. Dabre, and S. Kurohashi, "An empirical comparison of domain adaptation methods for neural machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 385–391. [Online]. Available: http://aclweb.org/anthology/P17-2061

[27] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proceedings of the ACL 2010 Conference Short Papers*. Uppsala, Sweden: Association for Computational Linguistics, 2010, pp. 220–224. [Online]. Available: http://aclweb.org/anthology/P10-2041

[28] K. Duh, G. Neubig, K. Sudoh, and H. Tsukada, "Adaptation data selection using neural language models: Experiments in machine translation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, 2013, pp. 678–683. [Online]. Available: http://aclweb.org/anthology/P13-2119

[29] M. Utiyama and H. Isahara, "Reliable measures for aligning Japanese-English news articles and sentences," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, 2003. [Online]. Available: http://aclweb.org/anthology/P03-1010

[30] R. Wang, H. Zhao, B.-L. Lu, M. Utiyama, and E. Sumita, "Neural network based bilingual language model growing for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 189–195. [Online]. Available: http://aclweb.org/anthology/D14-1023

[31] ——, "Connecting phrase based statistical machine translation adaptation," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, 2016, pp. 3135–3145. [Online]. Available: http://aclweb.org/anthology/C16-1295

[32] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 86–96. [Online]. Available: http://aclweb.org/anthology/P16-1009

[33] R. Wang, M. Utiyama, L. Liu, K. Chen, and E. Sumita, "Instance weighting for neural machine translation domain adaptation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017, pp. 1483–1489. [Online]. Available: http://aclweb.org/anthology/D17-1155

[34] M.-T. Luong and C. D. Manning, "Stanford neural machine translation systems for spoken language domain," in *International Workshop on Spoken Language Translation*, Da Nang, Vietnam, 2015.

[35] M. Freitag and Y. Al-Onaizan, "Fast domain adaptation for neural machine translation," *CoRR*, vol. abs/1612.06897, 2016.

[36] B. Chen, C. Cherry, G. Foster, and S. Larkin, "Cost weighting for neural machine translation domain adaptation," in *Proceedings of the First Workshop on Neural Machine Translation*. Vancouver: Association for Computational Linguistics, 2017, pp. 40–46. [Online]. Available: http://aclweb.org/anthology/W17-3205

[37] C. Gulcehre, O. Firat, K. Xu, K. Cho, and Y. Bengio, "On integrating a language model into neural machine translation," *Comput. Speech Lang.*, vol. 45, no. C, pp. 137–148, Sept. 2017. [Online]. Available: https://doi.org/10.1016/j.csl.2017.01.014

[38] J. R. Finkel and C. D. Manning, "Hierarchical Bayesian domain adaptation," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado, June 2009, pp. 602–610. [Online]. Available: https://www.aclweb.org/anthology/N09-1068

[39] M.-W. Chang, M. Connor, and D. Roth, "The necessity of combining adaptation methods," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA, Oct. 2010, pp. 767–777. [Online]. Available: https://www.aclweb.org/anthology/D10-1075

[40] Y. Yang and T. M. Hospedales, "A unified perspective on multi-domain and multi-task learning," in *Proceedings of the International Conference on Learning Representations*, ser. ICLR, San Diego, CA, 2015. [Online]. Available: https://arxiv.org/abs/1412.7489

[41] N. Peng and M. Dredze, "Multi-task domain adaptation for sequence tagging," in *Proceedings of the 2nd Workshop on Representation Learning for NLP*, ser. REP4NLP@ACL, Vancouver, Canada, 2017, pp. 91–100. [Online]. Available: https://aclanthology.info/papers/W17-2612/w17-2612

[42] C. Chu and R. Dabre, "Multilingual and multi-domain adaptation for neural machine translation," in *Proceedings of the 24st Annual Meeting of the Association for Natural Language Processing (NLP 2018)*, Okayama, Japan, 2018, pp. 909–912.