Revisiting Multi-Domain Machine Translation

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

[‡]Université Paris-Saclay, CNRS, LIMSI, 91400, Orsay, France
firstname.lastname@limsi.fr
[†]SYSTRAN, 5 rue Feydeau, 75002 Paris, France
firstname.lastname@systrangroup.com

Abstract

When building machine translation systems, one often needs to make the best out of heterogeneous sets of parallel data in training, and to robustly handle inputs from unexpected domains in testing. This multi-domain scenario has attracted a lot of recent work, that fall under the general umbrella of transfer learning. In this study, we revisit multi-domain machine translation, with the aim to formulate the motivations for developing such systems and the associated expectations with respect to performance. Our experiments with a large sample of multi-domain systems show that most of these expectations are hardly met and suggest that further work is needed to better analyze the current behaviour of multi-domain systems and to make them fully hold their promises.

1 Introduction

Data-based Machine Translation (MT), whether statistical or neural, rests on well understood machine learning principles. Given a training sample of matched source-target sentence pairs $(\mathbf{f}, \mathbf{e})$ drawn from an underlying distribution $\mathcal{D}_s$, a model parameterized by θ (here, a translation function h_θ) is trained by minimizing the empirical expectation of a loss function $\ell(h_\theta(\mathbf{f}), \mathbf{e})$. This approach ensures that the translation loss remains low when translating more sentences drawn from the same distribution.

Owing to the great variability of language data, this ideal situation is rarely met in practice, warranting the study of an alternative scenario, where the test distribution $\mathcal{D}_t$ differs from $\mathcal{D}_s$. In this setting, *domain adaptation* (DA) methods are in order. DA has a long history in Machine Learning in general (e.g. (Shimodaira, 2000; Ben-David et al., 2010; Joaquin Quionero-Candela and Lawrence, 2008; Pan and Yang, 2010)) and in NLP in particular (eg. (Daumé III and Marcu, 2006; Blitzer, 2007; Jiang and Zhai, 2007)). Various techniques thus exist to handle both the situations where a (small) training sample drawn from $\mathcal{D}_t$ is available in training, or where only samples of source-side (or target-side) sentences are available (see (Foster and Kuhn, 2007; Bertoldi and Federico, 2009; Axelrod et al., 2011) for proposals from the statistical MT era, or (Chu and Wang, 2018) for a recent survey of DA for Neural MT).

A seemingly related problem is *multi-domain* (MD) machine translation (Sajjad et al., 2017; Farajian et al., 2017b; Kobus et al., 2017; Zeng et al., 2018; Pham et al., 2019) where one single system is trained and tested with data from multiple domains. MD machine translation (MDMT) corresponds to a very common situation, where all available data, no matter its origin, is used to train a robust system that performs well for any kind of new input. If the intuitions behind MDMT are quite simple, the exact specifications of MDMT systems are rarely spelled out: for instance, should MDMT perform well when the test data is distributed like the training data, when it is equally distributed across domains or when the test distribution is unknown? Should MDMT also be robust to new domains? How should it handle domain labeling errors?

A related question concerns the relationship between supervised domain adaptation and multi-domain translation. The latter task seems more challenging as it tries to optimize MT performance for a more diverse set of potential inputs, with an additional uncertainty regarding the distribution of test data. Are there still situations where MD systems can surpass single domain adaptation, as is sometimes expected?

In this paper, we formulate in a more precise fashion the requirements that an effective MDMT system should meet (Section 2). Our first contribution is thus of methodological nature and consists of lists of expected properties of MDMT systems and associated measurements to evaluate them (Section 3). In doing so, we also shed light on new problems that arise in this context, regarding for instance the accommodation of new domains in the course of training, or the computation of automatic domain tags. Our second main contribution is experimental and consists in a thorough reanalysis of eight recent multi-domain approaches from the literature, including a variant of a model initially introduced for DA. We show in Section 4 that existing approaches still fall short to match many of these requirements, notably with respect to the handling of a large amount of heterogeneous domains and to dynamically integrating new domains in training.

2 Requirements of multi-domain MT

In this section, we recap the main reasons for considering a multi-domain scenario and discuss their implications in terms of performance evaluation.

2.1 Formalizing multi-domain translation

We conventionally define a domain d as a distribution $\mathcal{D}_d(x)$ over some feature space $\mathcal{X}$ that is shared across domains (Pan and Yang, 2010): in machine translation, $\mathcal{X}$ is the representation space for source sentences; each domain corresponds to a specific source of data, and differs from the other data sources in terms of textual genre, thematic content (Chen et al., 2016; Zhang et al., 2016), register (Sennrich et al., 2016a), style (Niu et al., 2018), etc. Translation in domain d is formalized by a translation function $h_d(y|x)$ pairing sentences in a source language with sentences in a target language $y \in \mathcal{Y}$. h_d is usually assumed to be deterministic (hence $y = h_d(x)$), but can differ from one domain to the other.

A typical learning scenario in MT is to have access to samples from n_d domains, which means that the training distribution $\mathcal{D}^s$ is a mixture $\mathcal{D}^s(x) = \sum_d \lambda^s_d \mathcal{D}_d(x)$, with $\{\lambda^s_d, d = 1 \dots n_d\}$ the corresponding mixture weights ($\sum_d \lambda^s_d = 1$). Multi-domain learning, as defined in Dredze and Crammer (2008) further assumes that domain tags are also available in testing; the implication being that the test distribution is also as a mixture $\mathcal{D}^t(x) = \sum_d \lambda^t_d \mathcal{D}_d(x)$ of several domains, making the problem distinct from mere domain adaption. A multi-domain learner is then expected to use these tags effectively (Joshi et al., 2012) when computing the combined translation function $h(x, d)$, and to perform well in all domains (Finkel and Manning, 2009). This setting is closely related to the multi-source adaptation problem formalized in (Mansour et al., 2009a,b; Hoffman et al., 2018).

This definition seems to be the most accepted view of a multi-domain MT[1] and one that we also adopt here. Note that in the absence of further specification, the naive answer to the MD setting should be to estimate one translation function $\hat{h}_d(x)$ separately for each domain, then to translate using $\hat{h}(x, d) = \sum_{d'} h_{d'}(x)\mathbb{I}(d' = d)$, where $\mathbb{I}(x)$ is the indicator function. We now discuss the arguments that are put forward to proceed differently.

2.2 Reasons for building MDMT systems

A first motivation for moving away from the one-domain / one-system solution are practical (Sennrich et al., 2013; Farajian et al., 2017a): when faced with inputs that are potentially from multiple domains, it is easier and computationally cheaper to develop one single system instead of having to optimize and maintain multiple engines. The underlying assumption here is that the number of domains of interests can be large, a limiting scenario being fully personalized machine translation (Michel and Neubig, 2018).

A second line of reasoning rests on linguistic properties of the translation function and contends that domain specificities are mostly expressed lexically and will primarily affect content words or multi-word expressions; function words, on the other hand, are domain agnostic and tend to remain semantically stable across domains, motivating some cross-domain parameter sharing. An MDMT system should simultaneously learn lexical domain peculiarities, and leverage cross-domain similarities to improve the translation of generic contexts and words (Zeng et al., 2018; Pham et al., 2019). It is here expected that the MDMT scenario should be more profitable when the domain mix includes domains that are closely related and can share more information.

A third series of motivations are of statistical nature. The training data available for each domain is usually unevenly distributed, and domain-specific systems trained or adapted on small datasets are likely to have a high variance and generalize poorly. For some test domains, there

[1] An exception is (Farajian et al., 2017b), where test translations rely on similarity scores between test and train sentences, rather than on domain labels.

may even be no data at all (Farajian et al., 2017a). Training mix-domain systems is likely to reduce this variance, at the expense of a larger statistical bias (Clark et al., 2012). Under this view, MDMT would be especially beneficial for domains with little training data. This is observed for multilingual MT from English: an improvement for under-resourced languages due to positive transfer, at the cost of a decrease in performance for well-resourced languages (Arivazhagan et al., 2019).

Combining multiple domain-specific MTs can also be justified in the sake of distributional robustness (Mansour et al., 2009a,b), for instance when the test mixture differs from the train mixture, or when it includes new domains unseen in training. An even more challenging case is when the MT would need to perform well for any test distribution, as studied for statistical MT in (Huck et al., 2015). In all these cases, mixing domains in training and/or testing is likely to improve robustness against unexpected or adversarial test distribution (Oren et al., 2019).

A distinct line of reasoning is that mixing domains can have a positive regularization effect for all domains. By introducing variability in training, it prevents DA from overfitting the available adaptation data and could help improve generalization even for well-resourced domains. A related case is made in (Joshi et al., 2012), which shows that part of the benefits of MD training is due to an ensembling effect, where systems from multiple domains are simultaneously used in the prediction phase; this effect may subsist even in the absence of clear domain separations.

To recap, there are multiple arguments for adopting MDMT, some already used in DA settings, and some original. These arguments are not mutually exclusive; however each yields specific expectations with respect to the performance of this approach, and should also yield appropriate evaluation procedure. If the motivation is primarily computational, then a drop in MT quality with respect to multiple individual domains might be acceptable if compensated by the computational savings. If it is to improve statistical estimation, then the hope will be that MDMT will improve, at least for some under-resourced domains, over individually trained systems. If finally, it is to make the system more robust to unexpected or adversarial test distributions, then this is the setting that should be used to evaluate MDMT. The next section discusses ways in which these requirements of MDMT systems could be challenged.

3 Challenging multi-domain systems

In this section, we propose seven operational requirements that can be expected from an effective multi-domain system, and discuss ways to evaluate whether these requirements are actually met. All these evaluations will rest on comparison of translation performance, and do not depend on the choice of a particular metric. To make our results comparable with the literature, we will only use the BLEU score (Papineni et al., 2002) in Section 4, noting it may not be the best yardstick to assess subtle improvements of lexical choices that are often associated with domain adapted systems (Irvine et al., 2013). Other important figures of merit for MDMT systems are the computational training cost and the total number of parameters.

3.1 Multi-domain systems should be effective

A first expectation is that MDMT systems should perform well in the face of mixed-domain test data. We thus derive the following requirements.

[P1-LAB] A MDMT should perform better than the baseline which disregards domain labels, or reassigns them in a random fashion (Joshi et al., 2012). Evaluating this requirement is a matter of a mere comparison, assuming the test distribution of domains is known: if all domains are equally important, performance averages can be reported; if they are not, weighted averages should be used instead.

[P2-TUN] Additionally, one can expect that MDMT will improve over fine-tuning (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016), at least in domains where data is scarce, or in situations where several domains are close. To evaluate this, we perform two measurements, using a real as well as an artificial scenario. In the real scenario, we simply compare the performance of MDMT and fine-tuning for domains of varying sizes, expecting a larger gain for smaller domains. In the artificial scenario, we split a single domain in two parts which are considered as distinct in training. The expectation here is that a MDMT should yield a clear gain for both pseudo subdomains, which should benefit from the supplementary amount of relevant training. In this situation, MDMT should even outperform fine-tuning on either of the pseudo sub-domain.

3.2 Robustness to fuzzy domain separation

A second set of requirements is related to the definition of a domain. As repeatedly pointed out in the literature, parallel corpora in MT are often collected opportunistically and the view that each corpus constitutes a single domain is often a gross approximation.[2] MDMT should aim to make the best of the available data and be robust to domain assignments. To challenge these requirements we propose evaluating the following requirements.

[P3-HET] The notion of a domain being a fragile one, an effective MDMT system should be able to discover not only when cross-domain sharing is useful (cf. requirement [P2-TUN]), but also when intra-domain heterogeneity is hurting. This requirement is tested by artificially conjoining separate domains into one during training, hoping that the loss in performance with respect to the baseline (using correct domain tags) will remain small.

[P4-ERR] MDMTs should perform best when the true domain tag is known, but deteriorate gracefully in the face of tag errors; in this situation, catastrophic drops in performance are often observed. This requirement can be assessed by translating test texts with erroneous domain tags and reporting the subsequent loss in performance.

[P5-UNK] A related situation occurs when the domain of a test document is unknown. Several situations need be considered: for domains seen in training, using automatically predicted domain labels should not be much worse than using the correct one. For test documents from unknown domains (zero-shot transfer), a good MD system should ideally outperform the default baseline that merges all available data.

[P6-DYN] Another requirement, more of an operational nature, is that an MDMT system should smoothly evolve to handle a growing number of domains, without having to retrain the full system each time new data is available. This is a requirement [P6-DYN] that we challenge by dynamically changing the number of training and test domains.

3.3 Scaling to a large number of domains

[P7-NUM] As mentioned above, MDMT systems have often been motivated by computational arguments. This argument is all the more sensible as the number of domains increases, making the optimization of many individual systems both ineffective and undesirable. For lack of having access to corpora containing very large sets (eg. in the order of 100-1000) domains, we experiment with automatically learned domains.

4 Experimental settings

4.1 Data and metrics

We experiment with translation from English into French and use texts initially originating from 6 domains, corresponding to the following data sources: the UFAL Medical corpus V1.0 (MED)[3], the European Central Bank corpus (BANK) (Tiedemann, 2012); The JRC-Acquis Communautaire corpus (LAW) (Steinberger et al., 2006), documentations for KDE, Ubuntu, GNOME and PHP from Opus collection (Tiedemann, 2009), collectively merged in a IT-domain, Ted Talks (TALK) (Cettolo et al., 2012), and the Koran (REL). Complementary experiments also use v12 of the News Commentary corpus (NEWS). Most corpora are available from the Opus web site.[4] These corpora were deduplicated and tokenized with in-house tools; statistics are in Table 1. To reduce the number of types and build open-vocabulary systems, we use Byte-Pair Encoding (Sennrich et al., 2016b) with 30,000 merge operations on a corpus containing all sentences in both languages.

We randomly select in each corpus a development and a test set of 1,000 lines and keep the rest for training.[5] Validation sets are used to chose the best model according to the average BLEU score (Papineni et al., 2002).[6] Significance testing is performed using bootstrap resampling (Koehn, 2004), implemented in compare-mt[7] (Neubig et al., 2019). We report significant differences at the level of $p = 0.05$.

We measure the distance between domains using the $\mathcal{H}$-Divergence (Ben-David et al., 2010), which relates domain similarity to the test error of a domain discriminator: the larger the error,

[2] Two of our own "domains" actually comprise several subcorpora (IT and MED), see details in Section 4.1.

[3] https://ufal.mff.cuni.cz/ufal_medical_corpus. We only use the in-domain (medical) subcorpora: PATR, EMEA, CESTA, ECDC.

[4] http://opus.nlpl.eu

[5] The code for reproducing our train, dev and test datasets is available at https://github.com/qmpham/experiments.

[6] We use truecasing and the multibleu script.

[7] https://github.com/neulab/compare-mt

	MED	LAW	BANK	IT	TALK	REL	NEWS
# lines	2609 (0.68)	501 (0.13)	190 (0.05)	270 (0.07)	160 (0.04)	130 (0.03)	260 (0)
# tokens	133 / 154	17.1 / 19.6	6.3 / 7.3	3.6 / 4.6	3.6 / 4.0	3.2 / 3.4	7.8 / 9.2
# types	771 / 720	52.7 / 63.1	92.3 / 94.7	75.8 / 91.4	61.5 / 73.3	22.4 / 10.5	-
# uniq	700 / 640	20.2 / 23.7	42.9 / 40.1	44.7 / 55.7	20.7 / 25.6	7.1 / 2.1	-

Table 1: Corpora statistics: number of parallel lines ($\times 10^3$) and proportion in the basic domain mixture (which does not include the NEWS domain), number of tokens in English and French ($\times 10^6$), number of types in English and French ($\times 10^3$), number of types that only appear in a given domain ($\times 10^3$). MED is the largest domain, containing almost 70% of the sentences, while REL is the smallest, with only 3% of the data.

	LAW	BANK	TALK	IT	REL
MED	1.93	1.97	1.9	1.93	1.97
LAW		1.94	1.97	1.93	1.99
BANK			1.98	1.94	1.99
TALK				1.92	1.93
IT					1.99

Table 2: The $\mathcal{H}$-divergence between domains

the closer the domains. Our discriminator is a SVM independently trained for each pair of domains, with sentence representations derived via mean pooling from the source side representation of the generic Transformer model. We used the scikit-learn[8] implementation with default values. Results in Table 2 show that all domains are well separated from all others, with REL being the furthest apart, while TALK is slightly more central.

4.2 Baselines

Our baselines are standard for multi-domain systems.[9] Using Transformers (Vaswani et al., 2017) implemented in OpenNMT-tf[10] (Klein et al., 2017), we build the following systems:

- a generic model trained on a concatenation of all corpora (Mixed). We develop two versions[11] of this system, one where the domain unbalance reflects the distribution of our training data given in Table 1 (Mixed-Nat) and one where all domains are equally represented in training (Mixed-Bal). The former is the best option when the train mixture $\mathcal{D}^s$ is also expected in testing; the latter should be used when the test distribution is uniform across domains. Accordingly, we report two aggregate scores: a weighted average reflecting the training distribution, and an unweighted average, meaning that test domains are equally important.
- fine-tuned models (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016), based on the Mixed-Nat system, further trained on each domain for at most 20 000 iterations, with early stopping when the dev BLEU stops increasing. The full fine-tuning (FT-Full) procedure may update all the parameters of the initial generic model, resulting in six systems adapted for one domain, with no parameter sharing across domains.

All models use embeddings and the hidden layers sizes of dimension 512. Transformers contain with 8 attention heads in each of the 6+6 layers; the inner feedforward layer contains 2048 cells. The adapter-based systems (see below) additionally use an adaptation block in each layer, composed of a 2-layer perceptron, with an inner ReLU activation function operating on normalized entries of dimension 1024. Training uses batches of 12,288 tokens, Adam with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$, Noam decay ($warmup\_steps = 4000$), and a dropout rate of 0.1 in all layers.

4.3 Multi-domain systems

Our comparison of multi-domain systems includes our own reimplementations of recent proposals from the literature:[12]

- a system using domain control as in (Kobus et al., 2017): domain information is introduced either as an additional token for each source sentence (DC-Tag), or as a supplementary feature for each word (DC-Feat).

[8] https://scikit-learn.org

[9] We however omit domain-specific systems trained only with the corresponding subset of the data, which are always inferior to the mix-domain strategy (Britz et al., 2017).

[10] https://github.com/OpenNMT/OpenNMT-tf

[11] In fact three: to enable a fair comparison with WDCMT, a RNN-based variant is also trained and evaluated. This system appears as Mixed-Nat-RNN in Table 3.

[12] Further implementation details are in Appendix A.

- a system using lexicalized domain representations (Pham et al., 2019): word embeddings are composed of a generic and a domain specific part (LDR);

- the three proposals of Britz et al. (2017). TTM is a feature-based approach where the domain tag is introduced as an extra word *on the target side*. Training uses reference tags and inference is usually performed with predicted tags, just like for regular target words. DM is a multi-task learner where a domain classifier is trained on top the MT encoder, so as to make it aware of domain differences; ADM is the adversarial version of DM, pushing the encoder towards learning domain-independent source representations. These methods thus only use domain tags in training.

- the multi-domain model of Zeng et al. (2018) (WDCMT), where a domain-agnostic and a domain-specialized representation of the input are simultaneously processed; supervised classification and adversarial training are used to compute these representations. Again, inference does not use domain tags.[13]

- two multi-domain versions of the approach of Bapna and Firat (2019), denoted FT-Res and MDL-Res, where a domain-specific adaptation module is added to all the Transformer layers; within each layer, residual connections enable to short-cut this adapter. The former variant corresponds to the original proposal of Bapna and Firat (2019) (see also (Sharaf et al., 2020)). It fine-tunes the adapter modules of a Mixed-Nat system independently for each domain, keeping all the other parameters frozen. The latter uses the same architecture, but a different training procedure and learns all parameters jointly from scratch with a mix-domain corpus.

This list includes systems that slightly depart from our definition of MDMT: standard implementations of TTM and WDCMT rely on infered, rather than on gold, domain tags - which must somewhat affect their predictions; DM and ADM make no use of domain tags at all. We however did not consider the proposal of (Farajian et al., 2017b) which performs on-the-fly tuning for each test sentence and diverges more strongly from our notion of MDMT.

5 Results and discussion

5.1 Performance of MDMT systems

In this section, we discuss the basic performance of MDMT systems trained and tested on 6 domains. Results are in Table 3. As expected, balancing data in the generic setting makes a great difference (the unweighted average is 2 BLEU points better, notably owing to the much better results for REL). As explained above, this setting should be the baseline when the test distribution is assumed to be balanced across domains. As all other systems are trained with an unbalanced data distribution, we use the weighted average to perform global comparisons.

Fine-tuning each domain separately yields a better baseline, outperforming Mixed-Nat for all domains, with significant gains for domains that are distant from MED: REL, IT, BANK, LAW.

All MDMTs (except DM and ADM) slightly improve over Mixed-Nat(for most domains), but these gains are rarely significant. Among systems using an extra domain feature, DC-Tag has a small edge over DC-Feat and also requires less parameters; it also outperforms TTM, which however uses predicted rather than gold domain tags. TTM is also the best choice among the systems that do not use domain tags in inference. The best contenders overall are FT-Res and MDL-Res, which significantly improve over Mixed-Nat for a majority of domains, and are the only ones to clearly fulfill [P1-LAB]; WDCMT also improves on three domains, but regresses on one. The use of a dedicated adaptation module thus seems better than feature-based strategies, but yields a large increase of the number of parameters. The effect of the adaptation layer is especially significant for small domains (BANK, IT and REL).

All systems fail to outperform fine-tuning, sometimes by a wide margin, especially for an "isolated" domain like REL. This might be due to the fact that domains are well separated (cf. Section 4.1) and are hardly helping each other. In this situation, MDMT systems should dedicate a sufficient number of parameters to each domain, so as to close the gap with fine-tuning.

[13] For this system, we use the available RNN-based system from the authors (https://github.com/DeepLearnXMU/WDCNMT) which does not directly compare to the other, Transformer-based, systems; the improved version of (Su et al., 2019) seems to produce comparable, albeit slightly improved, results.

Model / Domain		MED	LAW	BANK	TALK	IT	REL	WAVG	AVG
Mixed-Nat	[65m]	37.3	54.6	50.1	33.5	43.2	77.5	41.1	49.4
Mixed-Bal	[65m]	35.3	54.1	52.5	31.9	44.9	89.5	40.3	51.4
FT-Full	[6×65m]	37.7	**59.2**	**54.5**	34.0	**46.8**	**90.8**	**42.7**	**53.8**
DC-Tag	[+4k]	38.1	55.3	49.9	33.2	43.5	**80.5**	41.6	50.1
DC-Feat	[+140k]	37.7	54.9	49.5	32.9	43.6	**79.9**	41.4	49.9
LDR	[+1.4m]	37.0	54.7	49.9	33.9	43.6	**79.9**	40.9	49.8
TTM	[+4k]	37.3	54.9	49.5	32.9	43.6	**79.9**	41.0	49.7
DM	[+0]	35.6	49.5	45.6	29.9	37.1	62.4	38.1	43.4
ADM	[+0]	36.4	53.5	48.3	32.0	41.5	73.4	38.9	47.5
FT-Res	[+12.4m]	37.3	**57.9**	**53.9**	33.8	**46.7**	**90.2**	**42.3**	**53.3**
MDL-Res	[+12.4m]	37.9	**56.0**	**51.2**	33.5	44.4	**88.3**	42.0	**51.9**
Mixed-Nat-RNN	[51m]	36.8	53.8	47.2	30.0	35.7	60.2	39.2	44.0
WDCMT	[73m]	36.0	53.3	**48.8**	31.1	**38.8**	58.5	39.0	44.4

Table 3: Translation performance of MDMT systems based on the same Transformer (top) or RNN (bottom) architecture. The former contains 65m parameters, the latter has 51m. For each system, we report the number of additional domain specific parameters, BLEU scores for each domain, domain-weighted (WAVG) and unweighted (AVG) averages. For weighted-averages, we take the domain proportions from Table 1. Boldface denotes significant gains with respect to Mix-Nat (or Mix-Nat-RNN, for WDCMT), underline denotes significant losses.

5.2 Redefining domains

Table 4 summarizes the results of four experiments where we artificially redefine the boundaries of domains, with the aim to challenge requirements [P2-TUN], [P3-HET], and [P4-ERR]. In first three, we randomly *split* one corpus in two parts and proceed as if this corresponded to two actual domains. A MD system should detect that these two pseudo-domains are mutually beneficial and should hardly be affected by this change with respect to the baseline scenario (no split). In this situation, we expect MDMT to even surpass fine-tuning separately on each of these dummy domains, as MDMT exploits all data, while fine-tuning focuses only on a subpart. In testing, we decode the test set twice, once with each pseudo-domain tag. This makes no difference for TTM, DM, ADM and WDCMT, which do not use domain tags in testing. In the *merge* experiment, we merge two corpora in training, in order to assess the robustness with respect to heterogenous domains [P3-HET]. We then translate the two corresponding tests with the same (merged) system.

Our findings can be summarized as follows. For the split experiments, we see small variations that can be positive or negative compared to the baseline situation, but these are hardly significant. All systems show some robustness with respect to fuzzy domain boundaries; this is mostly notable for ADM, suggesting that when domain are close, ignoring domain differences is effective. In contrary, FT-Full incurs clear losses across the board, especially for the small data condition (Miceli Barone et al., 2017). Even in this very favourable case however, very few MDMT systems are able to significantly outperform FT-Full and this is only observed for the smaller part of the MED domain. The merge condition is hardly different, with again large losses for FT-full and FT-Res, and small variations for all systems. We even observe some rare improvements with respect to the situation where we use actual domains.

5.2.1 Handling wrong or unknown domains

In the last two columns of Table 4, we report the drop in performance when the domain information is not correct. In the first (RND), we use test data from the domains seen in training, presented with a random domain tag. In this situation, the loss with respect to using the correct tag is generally large (more than 10 BLEU points), showing an overall failure to meet requirement [P4-ERR], except for systems that ignore domain tags in testing.

In the second (NEW), we assess [P5-UNK] by translating sentences from a domain unseen in training (NEWS). For each sentence, we automatically predict the domain tag and use it for decoding.[14] In this configuration, again, systems us-

[14] Domain tags are assigned as follows: we train a language model for each domain and assign tag on a per-sentence basis

Set-up	Split		Split		Split		Merge		Wrong	
Model	MED (0.5 / 0.5)		MED (0.25 / 0.75)		LAW (0.5 / 0.5)		BANK+LAW		RND	NEW
	MED$_1$	MED$_2$	MED$_1$	MED$_2$	LAW$_1$	LAW$_2$	BANK	LAW	ALL	NEWS
FT-Full	-0.1	-0.6	-1.5	-0.2	-2.3	-5.1	-1.6	-1.4	-19.6	-3.3
DC-Tag	-0.2	-0.3	**+0.1**	+0.2	-0.4	-0.4	-0.5	-0.4	-13.4	-1.7
DC-Feat	-0.5	0.0	**+0.3**	+0.3	+0.3	+0.3	+0.3	+0.1	-14.2	-1.8
LDR	+0.1	+0.1	+0.4	+0.4	0.0	0.0	0.0	+0.1	-12.0	-1.4
TTM (*)	-0.2	-0.2	-0.2	-0.2	-0.3	-0.3	0.0	-0.3	0.0	-0.1
DM (*)	-0.3	-0.3	+0.4	+0.4	+0.3	+0.3	+0.9	+0.1	0.0	-0.9
ADM (*)	+0.6	+0.6	+0.4	+0.4	+0.4	+0.4	+0.1	-0.4	0.0	-0.2
FT-Res	-0.1	-0.4	-0.3	-0.3	-2.2	-2.9	-2.4	-3.2	-13.3	-3.0
MDL-Res	-0.2	-0.1	**+0.2**	+0.0	-0.9	-0.9	+0.7	-0.3	-18.6	-1.3
WDCMT (*)	-0.0	-0.0	+0.2	+0.2	+0.8	+0.8	-0.4	-0.8	0.0	+0.2

Table 4: Translation performance with variable domain definitions. In the Split/Merge experiments, we report BLEU differences for the related test set(s). Underline denotes significant loss when domains are changed wrt. the baseline situation; bold for a significant improvement over FT-Full; (*) tags systems ignoring test domains.

ing domain tags during inference perform poorly, significantly worse than the Mixed-Nat baseline (Bleu=23.5).

5.2.2 Handling growing numbers of domains

Another set of experiments evaluate the ability to dynamically handle supplementary domains (requirement [P6-DYN]) as follows. Starting with the existing MD systems of Section 5.1, we introduce an extra domain (NEWS) and resume training with this new mixture of data[15] for 50,000 additional iterations. We contrast this approach with training all systems from scratch and report differences in performance in Figure 1 (see also Table 7 in appendix B).[16] We expect that MDMT systems should not be too significantly impacted by the addition of a new domain and reach about the same performance as when training with this domain from scratch. From a practical viewpoint, dynamically integrating new domains is straightforward for DC-Tag, DC-Feat or TTM, for which new domains merely add new labels. It is less easy for DM, ADM and WDCMT, which include a built-in domain classifier whose outputs have to be prespecified, or for LDR, FT-Res and MDL-Res for which the number of possible domains is built in the architecture and has to be anticipated from the start. This makes a difference between domain-bounded systems, for which the number of domains is limited and truly open-domain systems.

We can first compare the results of coldstart training with six or seven domains in Table 7: a first observation is that the extra training data is hardly helping for most domains, except for NEWS, where we see a large gain, and for TALK. The picture is the same when one looks at MDMTs, where only the weakest systems (DM, ADM) seem to benefit from more (out-of-domain) data. Comparing now the coldstart with the warmstart scenario, we see that the former is always significantly better for NEWS, as expected, and that resuming training also negatively impacts the performance for other domains. This happens notably for DC-Tag, TTM and ADM. In this setting MDL-Res and DM show the smaller average loss, with the former achieving the best balance of training cost and average BLEU score.

5.3 Automatic domains

In this section, we experiment with automatic domains, obtained by clustering sentences into $k = 30$ classes using the k-means algorithm based on generic sentence representations obtained via mean pooling (cf. Section 4.1). This allows us to evaluate requirement [P7-scale], training and testing our systems as if these domains were fully separated. Many of these clusters are mere splits of the large MED, while a fewer number of classes are mixtures of two (or more) existing domains

based on the language model log-probability (assuming uniform domain priors). The domain classifier has an average prediction error of 16.4% for in-domain data.

[15] The design of a proper balance between domains in training is critical for achieving optimal performance: as our goal is to evaluate all systems in the same conditions, we consider a basic mixing policy based on the new training distribution. This is detrimental to the small domains, for which the "negative transfer" effect is stronger than for larger domains.

[16] WDCMT results are excluded from this table, as resuming training proved difficult to implement.

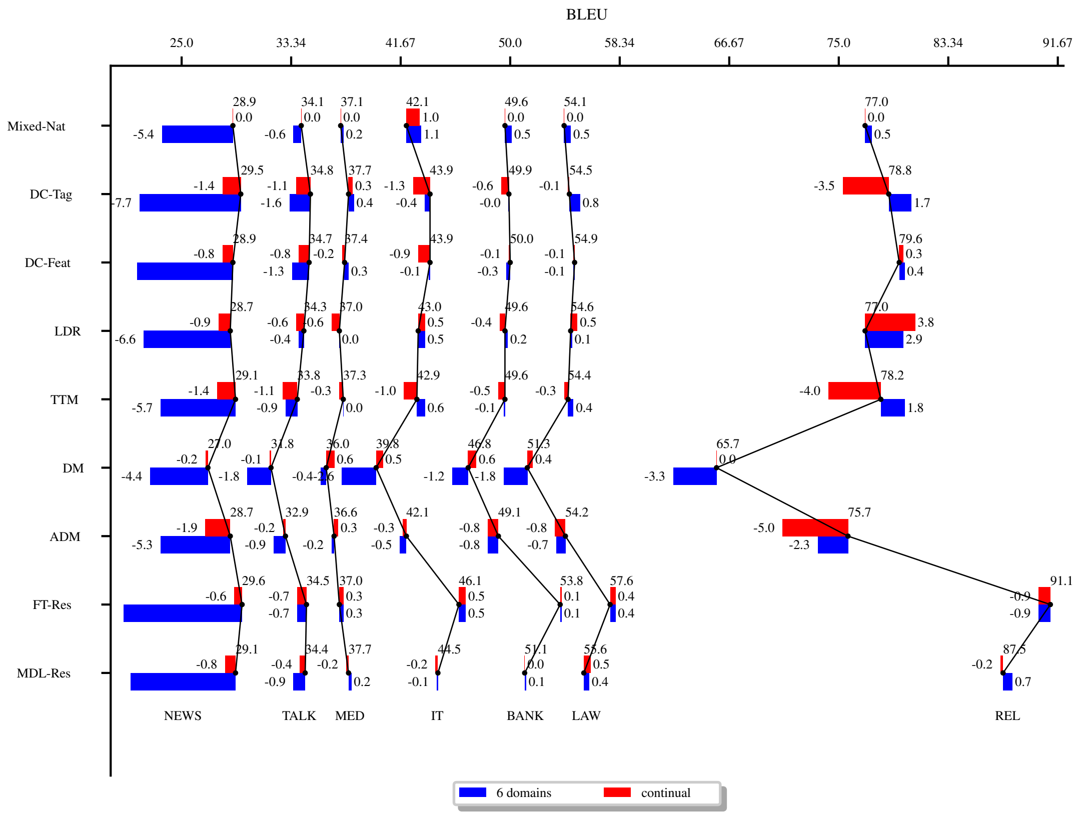

Figure 1: Ability to handle a new domain. We report BLEU scores for a complete training session with 7 domains, as well as differences (in blue) with training with 6 domains (from Table 3); and (in red) differences with continual training.

(full details are in Appendix C). We are thus in a position to reiterate, at a larger scale, the measurements of Section 5.2 and test whether multi-domain systems can effectively take advantage from the cross-domain similarities and to eventually perform better than fine-tuning. The results in Table 5 also suggest that MDMT can surpass fine-tuning for the smaller clusters; for the large clusters, this is no longer true. The complete table (in Appendix C) shows that this effect is more visible for small subsets of the medical domain.

Finally, Table 6 reports the effect of using automatic domain for each of the 6 test sets: each sentence was first assigned to an automatic class, translated with the corresponding multi-domain system with 30 classes; aggregate numbers were then computed, and contrasted with the 6-domain scenario. Results are clear and confirm previous observations: even though some clusters are very close, the net effect is a loss in performance for almost all systems and conditions. In this setting, the best MDMT in our pool (`MDL-Res`) is no longer able to surpass the `Mix-Nat` baseline.

6 Related work

The multi-domain training regime is more the norm than the exception for natural language processing (Dredze and Crammer, 2008; Finkel and Manning, 2009), and the design of multi-domain systems has been proposed for many language processing tasks. We focus here exclusively on MD machine translation, keeping in mind that similar problems and solutions (parameter sharing, instance selection / weighting, adversarial training, etc) have been studied in other contexts.

Multi-domain translation was already proposed for statistical MT, either considering as we do multiple sources of training data (eg. (Banerjee et al., 2010; Clark et al., 2012; Sennrich et al., 2013; Huck et al., 2015)), or domains made of multiple

Model/	Train	Mixed	FT	FT	MDL	DC	DC	TTM	ADM	DM	LDR
Clusters	size	Nat	Full	Res	Res	Feat	Tag				
10 small	29.3k	68.3	70.0	70.7	**71.2**	70.6	53.1	67.3	69.8	67.0	70.2
10 mid	104.7k	44.8	**48.0**	46.0	45.7	44.8	44.3	44.5	43.7	41.6	44.5
10 large	251.1k	50.4	**52.9**	52.0	51.3	49.6	43.2	49.1	48.5	44.3	49.5
Avg	128.4k	54.5	**57.0**	56.2	56.1	55.0	46.9	53.6	54.0	51.0	54.7

Table 5: BLEU scores computed by merging the 10 smaller, medium, and larger cluster test sets. Best score for each group is in boldface. For the small clusters, full-fine tuning is outperformed by several MDMT systems - see details in Appendix C.

Domain / Model	MED	LAW	BANK	TALK	IT	REL	WAVG	AVG
DC-Tag	38.5	54.0	49.0	33.6	42.2	76.7	41.6	49.0
DC-Feat	37.3	54.2	49.3	33.6	41.9	75.8	40.8	48.7
LDR	37.4	54.1	48.7	32.5	41.4	75.9	39.1	48.3
TTM	37.4	53.7	48.9	32.8	41.3	75.8	40.7	48.3
DM	35.4	49.3	45.2	29.7	37.1	60.0	37.8	42.8
ADM	36.1	53.5	48.0	32.0	41.1	72.1	39.5	47.1
FT-Res	37.5	55.7	51.1	33.1	44.1	86.7	41.6	51.4
MDL-Res	37.3	55.5	50.2	32.2	42.1	86.7	41.2	50.7
WDCMT	35.6	53.1	48.4	30.5	37.7	56.0	38.5	43.6

Table 6: Translation performance with automatic domains, computed with the original test sets. Significance tests are for comparisons with the 6-domain scenario (Table 3).

topics (Eidelman et al., 2012; Hasler et al., 2014). Two main strategies were considered: instance-based, involving a measure of similarities between train and test domains; feature-based, where domain/topic labels give rise to additional features.

The latter strategy has been widely used in NMT: Kobus et al. (2017) inject an additional domain feature in their seq2seq model, either in the form of an extra (initial) domain-token or in the form of an additional domain-feature associated to each word. These results are reproduced by Tars and Fishel (2018), who also consider automatically induced domain tags. This technique also helps control the style of MT outputs in (Sennrich et al., 2016a; Niu et al., 2018), and to encode the source or target languages in multilingual MT (Firat et al., 2016; Johnson et al., 2017). Domain control can also be performed on the target side, as in (Chen et al., 2016), where a topic vector describing the whole document serves as an extra context in the softmax layer of the decoder. Such ideas are further developed in (Chu and Dabre, 2018; Pham et al., 2019), where domain differences and commonalties are encoded in the network architecture: some parameters are shared across domains, while others are domain-specific.

Techniques proposed by (Britz et al., 2017) aim to ensure that domain information is actually used in a mix-domain system. Three methods are considered, using either domain classification (or domain normalization, via adversarial training) on the source or target side. There is no clear winner in either of the three language pairs considered. One contribution of this work is the idea of normalizing representations through adversarial training, so as to make the mixture of heterogeneous data more effective; representation normalization has since proven a key ingredient in multilingual transfer learning. The same basic techniques (parameter sharing, automatic domain identification / normalization) are simultaneously at play in (Zeng et al., 2018; Su et al., 2019): in this approach, the lower layers of the MT use auxiliary classification tasks to disentangle domain specific representations on the one hand from domain-agnostic representations on the other hand. These representations are then processed as two separate inputs, then recombined to compute the translation.

Another parameter-sharing scheme is in (Jiang et al., 2019), which augments a Transformer model with domain-specific heads, whose contributions are regulated at the word/position level: some words have "generic" use and rely on mixed-domain heads, while for some other words it is preferable to use domain specific heads, thereby reintroducing the idea of ensembling at the core of (Huck et al., 2015; Saunders et al., 2019). The results for three language pairs outperform several standard baselines for a two-domain systems (in fr:en and de:en) and a 4-domain system (zh:en).

Finally, Farajian et al. (2017b), Li et al. (2018) and Xu et al. (2019) adopt a different strategy. Each test sentence triggers the selection of a small set of related instances; using these, a generic NMT is tuned for some iterations, before delivering its output. This approach entirely dispenses with the notion of domain and relies on data selection techniques to handle data heterogeneity.

7 Conclusion and outlook

In this study, we have carefully reconsidered the idea of multi-domain machine translation, which seems to be taken for granted in many recent studies. We have spelled out the various motivations for building such systems and the associated expectations in terms of system performance. We have then designed a series of requirements that MDMT systems should meet, and proposed a series of associated test procedures. In our experiments with a representative sample of MDMTs, we have found that most requirements were hardly met for our experimental conditions. If MDMT systems are able to outperform the mixed-domain baseline, at least for some domains, they all fall short to match the performance of fine-tuning on each individual domain, which remains the best choice in multi-source single domain adaptation. As expected however, MDMTs are less brittle than fine-tuning when domain frontiers are uncertain, and can, to a certain extend, dynamically accommodate additional domains, this being especially easy for feature-based approaches. Our experiments finally suggest that all methods show decreasing performance when the number of domains or the diversity of the domain mixture increases.

Two other main conclusions can be drawn from this study: first, it seems that more work is needed to make MDMT systems make the best out of the variety of the available data, both to effectively share what needs to be shared while at the same time separating what needs to be kept separated. We notably see two areas worthy of further explorations: the development of parameter sharing strategies when the number of domains is large; the design of training strategies that can effectively handle a change of the training mixture, including an increase in the number of domains. Both problems are of practical relevance in industrial settings. Second, and maybe more importantly, there is a general need to adopt better evaluation methodologies for evaluating MDMT systems, which require systems developers to clearly spell out the testing conditions and the associated expected distribution of testing instances, and to report more than comparisons with simple baselines on a fixed and known handful of domains.

Acknowledgements

The work presented in this paper was partially supported by the European Commission under contract H2020-787061 ANITA.

This work was granted access to the HPC resources of [TGCC/CINES/IDRIS] under the allocation 2020- [AD011011270] made by GENCI (Grand Equipement National de Calcul Intensif).

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv e-prints*, abs/1907.05019.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 355–362, Edinburgh, United Kingdom.

Pratyush Banerjee, Jinhua Du, Baoli Li, Sudip Kumar Naskar, Andy Way, and Josef van Genabith. 2010. Combining multi-domain statistical machine translation models using automatic classifiers. In *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas*, AMTA 2010, Denver, CO, USA.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference* *on Natural Language Processing*, EMNLP-IJCNLP, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jenn Wortman. 2010. A theory of learning from different domains. *Machine Learning*, 79(1):151–175.

Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 182–189, Athens, Greece. Association for Computational Linguistics.

John Blitzer. 2007. *Domain adaptation of natural language processing systems*. Ph.D. thesis, School of Computer Science, University of Pennsylvania.

Denny Britz, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126, Copenhagen, Denmark. Association for Computational Linguistics.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. In *Proceedings of the Twelth Biennial Conference of the Association for Machine Translation in the Americas*, AMTA 2012, Austin, Texas.

Chenhui Chu and Raj Dabre. 2018. 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, pages 909–912, Okayama, Japan.

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

Jonathan H. Clark, Alon Lavie, and Chris Dyer. 2012. One system, many domains: Open-domain statistical machine translation via feature augmentation. In *Proceedings of the Tenth Biennial Conference of the Association for Machine Translation in the Americas*, (AMTA 2012), San Diego, CA.

Hal Daumé III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research (JAIR)*, 26:101–126.

Mark Dredze and Koby Crammer. 2008. Online methods for multi-domain learning and adaptation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 689–697, Honolulu, Hawaii.

Vladimir Eidelman, Jordan Boyd-Graber, and Philip Resnik. 2012. Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 115–119, Jeju Island, Korea. Association for Computational Linguistics.

M. Amin Farajian, Marco Turchi, Matteo Negri, Nicola Bertoldi, and Marcello Federico. 2017a. Neural vs. phrase-based machine translation in a multi-domain scenario. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 280–284, Valencia, Spain. Association for Computational Linguistics.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017b. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark.

Jenny Rose Finkel and Christopher D. Manning. 2009. 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*, pages 602–610, Boulder, Colorado.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. 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*, pages 866–875. Association for Computational Linguistics.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Prague, Czech Republic.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897.

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

Judy Hoffman, Mehryar Mohri, and Ningshan Zhang. 2018. Algorithms and theory for multiple-source adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8246–8256. Curran Associates, Inc.

Matthias Huck, Alexandra Birch, and Barry Haddow. 2015. Mixed domain vs. multi-domain statistical machine translation. In *Procedings of the Machine Translation Summit*, MT Summit XV, pages 240–255, Miami Florida.

Ann Irvine, John Morgan, Marine Carpuat, Hal Daum, and Dragos Munteanu. 2013. Measuring machine translation errors in new domains. *Transactions of the Association for Computational Linguistics*, 1:429–440.

Haoming Jiang, Chen Liang, Chong Wang, and Tuo Zhao. 2019. Multi-domain neural machine translation with word-level adaptive layer-wise domain mixing. *CoRR*, abs/1911.02692.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, Prague, Czech Republic. Association for Computational Linguistics.

Anton Schwaighofer Joaquin Quionero-Candela, Masashi Sugiyama and Neil D. Lawrence, editors. 2008. *Dataset Shift in Machine Learning*. Neural Information Processing series. MIT Press. ISBN : 9780262170055.

Melvin Johnson, Mike Schuster, Quoc Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernand a Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Mahesh Joshi, Mark Dredze, William W Cohen, and Carolyn P Rose. 2012. Multi-domain learning: When do domains matter? In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1302–1312.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

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

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2018. One sentence one model for neural machine translation. In *Proceedings of the Eleventh International Conference on*

Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA).

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *Proceedings of the International Workshop on Spoken Language Translation*, IWSLT, Da Nang, Vietnam.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009a. Domain adaptation with multiple sources. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1041–1048. Curran Associates, Inc.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009b. Multiple source adaptation and the Rényi divergence. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, UAI 2009, pages 367–374.

Antonio Valerio Miceli Barone, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1489–1494, Copenhagen, Denmark. Association for Computational Linguistics.

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

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, Xinyi Wang, and John Wieting. 2019. compare-mt: A tool for holistic comparison of language generation systems. *CoRR*, abs/1903.07926.

Xing Niu, Sudha Rao, and Marine Carpuat. 2018. Multi-task neural models for translating between styles within and across languages. In *Proceedings of the 27th International Conference on Computational Linguistics*, COLING, pages 1008–1021, Santa Fe, New Mexico, USA.

Yonatan Oren, Shiori Sagawa, Tatsunori Hashimoto, and Percy Liang. 2019. Distributionally robust language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4227–4237, Hong Kong, China. Association for Computational Linguistics.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

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

Minh Quang Pham, Josep-Maria Crego, Jean Senellart, and François Yvon. 2019. Generic and Specialized Word Embeddings for Multi-Domain Machine Translation. In *Proceedings of the 16th International Workshop on Spoken Language Translation*, IWSLT, page 9p, Hong-Kong, CN.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Yonatan Belinkov, and Stephan Vogel. 2017. Neural machine translation training in a multi-domain scenario. In *Proceedings of the 14th International Workshop on Spoken Language Translation*, IWSLT 2017, Tokyo, Japan.

Danielle Saunders, Felix Stahlberg, and Bill Byrne. 2019. UCAM biomedical translation at WMT19: Transfer learning multi-domain ensembles. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 169–174, Florence, Italy. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Tech-*

nologies, pages 35–40, San Diego, California. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. 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)*, pages 1715–1725, Berlin, Germany.

Rico Sennrich, Holger Schwenk, and Walid Aransa. 2013. A multi-domain translation model framework for statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 832–840, Sofia, Bulgaria. Association for Computational Linguistics.

Amr Sharaf, Hany Hassan, and Hal Daumé III. 2020. Meta-learning for few-shot NMT adaptation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 43–53, Online. Association for Computational Linguistics.

Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227 – 244.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Toma Erjavec, Dan Tufis, and Dniel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, LREC'06, Genoa, Italy. European Language Resources Association (ELRA).

Jinsong Su, Jiali Zeng, Jun Xie, Huating Wen, Yongjing Yin, and Yang Liu. 2019. Exploring discriminative word-level domain contexts for multi-domain neural machine translation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, pages 1–1.

Sander Tars and Mark Fishel. 2018. Multi-domain neural machine translation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, EAMT, pages 259–269, Alicante, Spain. EAMT.

Jörg Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, LREC'12, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Jitao Xu, Josep Crego, and Jean Senellart. 2019. Lexical micro-adaptation for neural machine translation. In *Proceedings of the 16th International Workshop on Spoken Language Translation*, IWSLT 2019, Hong Kong, China.

Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. 2018. Multi-domain neural machine translation with word-level domain context discrimination. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, Brussels, Belgium. Association for Computational Linguistics.

Jian Zhang, Liangyou Li, Andy Way, and Qun Liu. 2016. Topic-informed neural machine translation. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, COLING 2016, pages 1807–1817, Osaka, Japan. The COLING 2016 Organizing Committee.

Appendices

A - Description of multi-domain systems

We use the following setups for MDMT systems.

- Mixed-Nat, FT-full, TTM, DC-Tag use a medium Transformer model of (Vaswani et al., 2017) with the following settings: embeddings size and hidden layers size are set to 512. Multi-head attention comprises 8 heads in each of the 6 layers; the inner feedforward layer contains 2048 cells. Training use a batch size of $12,288$ tokens; optimization uses Adam with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$ and Noam decay ($warmup\_steps = 4000$), and a dropout rate of 0.1 for all layers.

- FT-Res and MDL-res use the same medium Transformer and add residual layers with a bottleneck dimension of size 1024.

- ADM, DM use medium Tranformer model and a domain classifier composing of 3 dense layers of size 512×2048, 2048×2048 and $2048 \times domain\_num$. The two first layers of the classifier use the $\mathrm{ReLU}()$ as activation function, the last layer uses $\tanh()$ as activation function.

- DC-Feat uses medium Transformer model and domain embeddings of size 4. Given a sentence of domain i in a training batch, the embedding of domain i is concatenated to the embedding of each token in the sentence.

- LDR uses medium Transformer model and for each token we introduce a LDR feature of size $4 \times domain\_num$. Given a sentence of domain $i \in [1, .., K]$ in the training batch, for each token of the sentence, the LDR units of the indexes outside of the range $[4(i-1), .., 4i-1]$ are masked to 0, and the masked LDR feature will be concatenated to the embedding of the token. Details are in (Pham et al., 2019).

- Mixed-Nat-RNN uses one bidirectional LSTM layer in the encoder and one LSTM layer in the decoder. The size of hidden layers is 1024, the size of word embeddings is 512.

- WDCNMT uses one bidirectional GRU layer in the encoder and one GRU-conditional layer in the decoder. The size of hidden layers is 1024, the size of word embeddings is 512.

Training For each domain, we create train/dev/test sets by randomly splitting each corpus. We maintain the size of validation sets and of test sets equal to 1,000 lines for every domain. The learning rate is set as in (Vaswani et al., 2017). For the fine-tuning procedures used for FT-full and FT-Res, we continue training using the same learning rate schedule, continuing the incrementation of the number of steps. All other MDMT systems reported in Tables 3 and 4 use a combined validation set comprising 6,000 lines, obtained by merging the six development sets. For the results in Table 7 we also append the validation set of NEWS to the multi-domain validation set. In any case, training stops if either training reaches the maximum number of iterations (50,000) or the score on the validation set does not increase for three consecutive evaluations. We average five checkpoints to get the final model.

B - Experiments with continual learning

C - Experiments with automatic domains

This experiment aims to simulate with automatic domains a scenario where the number of "domains" is large and where some "domains" are close and can effectively share information. Full results in Table 8. Cluster size vary from approximately 8k sentences (cluster 24) up to more than 350k sentences. More than 2/3 of these clusters mostly comprise texts from one single domain, as for cluster 12 which is predominantly MED, the remaining clusters typically mix 2 domains. Fine-tuning with small domains is often outperformed by other MDMT techniques, an issue that a better regularization strategy might mitigate. Domain-control (DC-Feat) is very effective for small domains, but again less so in larger data conditions. Among the MD models, approaches using residual adapters have the best average performance.

Domain Model	MED	LAW	BANK	TALK	IT	REL	NEWS	WAVG	AVG
Mixed-Nat	37.1	54.1	49.6	34.1	42.1	77.0	28.9	40.8	49.0
	+0.2 | –	+0.5 | —	+0.5 | –	-0.6 | –	+1.1 | –	+0.5 | –	-5.4 | –	+0.3 | –	+0.4 | –
DC-Tag	37.7	54.5	49.9	34.8	43.9	78.8	29.5	41.4	49.9
	+0.3 | +0.3	**+0.8** | -0.1	-0.04 | -0.6	-1.6 | -1.1	-0.4 | -1.3	**+1.7** | -3.5	-7.7 | -1.4	+0.2 | -0.1	+0.1 | -1.1
DC-Feat	37.4	54.9	50.0	34.7	43.9	79.6	28.9	41.2	50.1
	+0.3 | -0.2	-0.1 | -0.1	-0.3 | -0.1	-1.3 | -0.6	-0.1 | -0.9	+0.4 | +0.3	-7.3 | -0.8	+0.1 | -0.2	-0.2 | -0.3
LDR	37.0	54.6	49.6	34.3	43.0	77.0	28.7	40.8	49.2
	0.0 | -0.6	+0.1 | +0.5	+0.2 | -0.4	-0.4 | -0.6	+0.5 | +0.5	**+2.9** | **+3.8**	-6.6 | -0.9	+0.6 | +0.5	+0.1 | -0.4
TTM	37.3	54.4	49.6	33.8	42.9	78.2	29.1	41.0	49.4
	0.0 | -0.3	+0.4 | -0.3	-0.1 | -0.5	-0.9 | -1.1	+0.6 | -1.0	**+1.8** | -4.0	-5.7 | -1.4	0.0 | -0.5	+0.3 | -1.2
DM	36.0	51.3	46.8	31.8	39.8	65.7	27.0	38.9	45.2
	-0.4 | +0.6	-1.8 | +0.4	-1.2 | +0.6	-1.8 | -0.1	-2.6 | +0.5	-3.3 | 0.0	-4.4 | -1.2	-0.8 | +0.5	-1.8 | +0.3
ADM	36.6	54.2	49.1	32.9	42.1	75.7	28.7	40.2	48.4
	-0.2 | +0.3	-0.7 | -0.8	-0.8 | -0.8	-0.9 | -0.2	-0.5 | -0.4	-2.3 | -5.0	-5.4 | -1.9	-0.5 | -0.2	-0.9 | -1.1
FT-Res	37.0	57.6	53.8	34.5	46.1	91.1	29.6	42.2	53.3
	+0.3 | +0.3	+0.4 | +0.4	+0.1 | +0.1	-0.7 | -0.7	+0.5 | +0.5	-0.9 | -0.9	-9.0 | -0.6	-0.1 | -0.1	+0.2 | +0.2
MDL-Res	37.7	55.6	51.1	34.4	44.5	87.5	29.1	41.9	51.8
	+0.2 | -0.2	+0.4 | +0.5	+0.1 | 0.0	-0.9 | -0.4	-0.1 | -0.2	+0.9 | -0.2	-8.0 | -0.8	+0.1 | -0.2	+0.1 | -0.1

Table 7: Ability to handle a new domain. We report BLEU scores for a complete training session with 7 domains, as well as differences with (left) training with 6 domains (from Table 3); (right) continuous training mode. Averages only take into account six domains (NEWS excluded). Underline denotes a significant loss, bold a significant gain.

Model Cluster	size train / test	Mixed Nat	FT Full	FT Res	MDL Res	DC Feat	DC Tag	TTM	ADM	DM	LDR
24 [med]	8.1k / 3	90.4	90.4	90.4	90.4	100.0	65.6	100.0	90.4	100.0	100.0
13 [-]	17.3k / 52	67.6	75.4	74.3	74.3	75.0	54.7	74.7	75.9	65.9	76.9
28 [-]	25.6k / 54	71.6	68.7	68.1	70.2	71.0	42.5	72.0	71.3	65.6	72.6
19 [IT]	27.2k / 88	58.5	63.0	60.9	63.9	63.7	57.2	59.4	61.1	60.5	60.3
0 [-]	27.4k / 72	43.9	33.3	45.4	45.4	49.9	15.4	46.8	49.2	46.6	47.8
22 [-]	27.5k / 103	91.5	93.7	93.4	93.9	92.5	72.8	92.3	93.2	91.4	93.4
25 [-]	28.2k / 56	57.0	44.8	48.2	49.1	54.6	47.2	49.8	54.2	45.1	52.4
16 [med]	30.4k / 18	57.2	70.4	77.4	73.5	61.8	54.2	58.4	58.1	52.5	58.3
23 [med]	47.0k / 23	24.5	27.2	26.5	28.5	30.5	27.3	32.0	24.4	29.0	29.8
17 [med]	54.4k / 26	39.9	40.3	41.6	38.0	37.1	36.6	35.2	35.4	31.3	33.7
8 [IT]	61.4k / 214	46.9	53.1	55.8	53.6	48.9	45.1	48.8	50.9	43.0	46.7
1 [-]	68.1k / 122	47.2	47.5	48.7	45.1	46.8	39.1	45.4	44.2	40.7	44.9
7 [med]	91.5k / 30	41.3	35.5	41.4	39.9	41.4	36.5	37.3	37.1	40.7	41.8
11 [med]	93.0k / 38	31.6	42.6	31.8	35.4	36.0	29.6	36.7	32.7	26.5	36.6
29 [law]	109.2k / 242	65.9	69.2	67.6	67.7	66.0	63.8	65.1	64.7	62.4	65.9
27 [med]	109.3k / 49	11.0	9.6	8.7	9.2	10.0	19.4	9.4	7.9	10.7	10.6
5 [-]	109.9k / 267	46.3	47.4	46.9	45.4	44.0	42.9	43.7	44.3	40.9	45.7
6 [med]	133.4k / 73	37.2	38.9	38.7	36.8	37.5	27.5	38.0	37.2	31.3	35.9
26 [-]	134.8k / 428	31.8	30.8	31.8	31.2	31.9	32.6	32.2	30.5	29.6	31.2
15 [bank]	136.9k / 674	46.5	51.5	47.9	48.0	46.6	46.0	45.8	45.7	42.9	46.0
4 [rel]	137.4k / 1016	77.1	85.3	83.5	83.3	75.8	46.1	74.2	73.3	63.2	75.9
2 [med]	182.6k / 85	70.6	75.8	71.7	69.4	68.2	67.3	67.3	68.6	65.6	68.2
20 [med]	183.0k / 71	47.4	47.2	46.8	47.2	48.4	47.5	48.8	47.3	47.1	46.8
21 [-]	222.8k / 868	38.7	38.8	39.0	37.2	37.5	35.9	36.9	37.1	33.4	37.0
10 [med]	225.4k / 115	40.0	42.6	40.0	38.2	39.9	35.8	39.5	39.1	36.3	40.7
18 [med]	245.0k / 106	57.7	60.3	58.7	58.6	58.4	56.3	57.3	56.1	54.9	55.9
9 [med]	301.6k / 145	37.2	37.3	36.5	36.1	36.4	37.7	36.4	35.2	34.2	37.0
3 [law]	323.5k / 680	50.1	52.0	50.8	50.1	49.1	48.3	49.0	48.2	44.4	49.1
14 [med]	334.0 / 146	31.6	31.4	31.9	33.0	32.5	34.1	31.4	32.1	30.5	31.8
12 [med]	356.4k / 148	36.3	36.6	35.9	35.9	35.8	37.0	36.4	35.4	34.2	36.3

Table 8: Complete results for the experiments with automatic domains. For each cluster, we report: the majority domain when one domain accounts for more than 75% of the class; training and test sizes; and BLEU scores obtained with the various systems used in this study. Most test sets are too small to report significance tests.