Domain adapted machine translation: What does catastrophic forgetting forget and why?

Danielle Saunders* and Steve DeNeefe
RWS Language Weaver
danielle.saunders@cantab.net sdeneefe@rws.com
*Work completed while at RWS

Abstract

Neural Machine Translation (NMT) models can be specialized by domain adaptation, often involving fine-tuning on a dataset of interest. This process risks catastrophic forgetting: rapid loss of generic translation quality. Forgetting has been widely observed, with many mitigation methods proposed. However, the causes of forgetting and the relationship between forgetting and adaptation data are under-explored. This paper takes a novel approach to understanding catastrophic forgetting during NMT adaptation by investigating the impact of the data. We provide a first investigation of what is forgotten, and why. We examine the relationship between forgetting and the in-domain data, and show that the amount and type of forgetting is linked to that data's target vocabulary coverage. Our findings pave the way toward better informed NMT domain adaptation.

1 Introduction

The specialization of Neural Machine Translation (NMT) models for high performance in a specific domain, such as legal or healthcare, is of strong interest to academia [**?**] and industry [**?**]. Fine-tuning, sometimes known as transfer learning, is a well-established domain adaptation method that continues training a pre-trained NMT model on some new dataset from the domain of interest [**?**]. However, fine-tuning on domain-shifted data can result in catastrophic forgetting [**?**].

This apparently simple statement has been widely observed in NMT, but rarely examined. Catastrophic forgetting in NMT is described variously as 'degradation of general-domain performance' [**?**] or '[forgetting] previous domain knowledge' [**?**], typically referencing lower scores in some quality metric. However, prior work on forgetting in NMT focuses on mitigation, leaving two important gaps.

Firstly, prior work does not determine what is forgotten in concrete terms. Lower scores in a reference-based quality metric can indicate either poor translation or simply a vocabulary shift towards the new domain. This is especially true for string-based metrics like BLEU [**?**]. Prior work does not distinguish between quality drop and vocabulary shift, and further does not address whether vocabulary shift 'forgetting' is beneficial or detrimental to translation of the generic and new domains.

Secondly, prior work almost universally treats forgetting and its mitigation as independent of the adaptation dataset. In fact, the contents of the adaptation dataset will impact forgetting—consider the amount of forgetting expected if fine-tuning on a 1000-sentence sample of the pre-training dataset, versus 1000 copies of the same sentence pair. Understanding the relationship between adaptation data and forgetting is crucial for predicting how well domain adaptation will work, whether adapted performance is likely to generalise, or whether forgetting mitigation approaches are necessary.

The contributions of this paper lie in addressing these gaps. Specifically:

- We provide a first exploration of what domain-adapted NMT forgets. This includes quantifying the degree of detrimental vocabulary shift, and demonstrating that this shift is not well-characterised by common MT quality metrics.
- We show that forgetting can consist of using in-domain vocabulary inappropriately in out-of-vocabulary contexts—and, unexpectedly, that this can take place even when the source sentence has no in-domain triggers.
- We also provide a first investigation into the relationship between forgetting and adaptation dataset, examining the correlation between forgetting and several domain heuristics for eight domains across two language pairs.
- We find that some commonly used domain heuristics including sentence pair count and vocabulary distribution cannot explain how forgetting varies by domain, but that forgetting does have a strong relationship with generic vocabulary coverage.
- We support our findings by demonstrating significantly reduced forgetting with minimal, coverage-based mixed fine-tuning. In the process we show that much of the benefit of generic data mix-in comes from a relatively small vocabulary-covering set.

1.1 Related work

NMT adaptation with the goal of improved in-domain performance sometimes accounts for domain-specific data characteristics. Examples include selecting adaptation data by target domain similarity [**?**], gradually emphasising in-domain data during training [**?**], or determining hyperparameters via meta-learning [**?**].

Work focusing on catastrophic forgetting in NMT, by contrast, takes an adaptation-set-agnostic approach depending only on the generic dataset [**?**]. This can include regularizing parameters relative to the generic domain [**?**], training on complementary inputs via generic-trained teacher models [**?**] or mixing in generic data during adaptation [**?**].

The specific adaptation dataset is not typically considered beyond broad suggestions such as tuning for fewer steps on smaller datasets [**?**]. In this work, by contrast, we aim to understand forgetting based on the characteristics of the domain-specific adaptation dataset.

2 What does adapted NMT forget?

In this section, we explore what is forgotten during adaptation in concrete terms, using two quality metrics and a new measure for analysing vocabulary shift. We adapt pre-trained generic NMT models to eight diverse domains across two language pairs, intentionally triggering catastrophic forgetting, and analyse the degree of quality degradation versus vocabulary shift. In particular, we examine which tokens are forgotten and what replaces them after adaptation. We find that models experiencing forgetting produce in-domain vocabulary incorrectly and in entirely out-of-domain contexts.

2.1 Measuring vocabulary-shift forgetting

To determine which tokens are forgotten during adaptation we propose a new forgetting measure. Prior work measures forgetting via a drop in a corpus-level quality metric [**?**, **?**]. However, these do not mark which terms are forgotten. To measure vocabulary shift forgetting, a score should highlight terms that are used correctly before but not after adaptation.

We focus on unigram terms: these are easily interpretable with respect to the vocabulary, which often signifies domain [**?**]. Consider a test set where for each reference translation T_R we can compare a translation from an original model T_O and a translation from an adapted model T_A. We are interested in how the adapted model translation changes relative to the original model translation and the reference. For each reference T_R, we find the count of every reference token in original model and adapted model translations, $\#tok_O$ and $\#tok_A$, capped at the count in the reference $\#tok_R$:

$$
\begin{aligned}
O[tok]_{T_R} &= \min(\#tok_O, \#tok_R) \\
A[tok]_{T_R} &= \min(\#tok_A, \#tok_R) \\
ForgetGenUse[tok] &= \sum_{T_R} \max(O[tok]_{T_R} - A[tok]_{T_R}, 0)
\end{aligned}
$$

High $ForgetGenUse[tok]$ means we forget generic use of tok. For example, if the generic model correctly produced tok N times and the adapted model did not produce it at all, $ForgetGenUse[tok] = N$: all generic uses of tok are forgotten. If the generic and adapted models both fail to produce tok at all, $ForgetGenUse[tok] = 0$ – this is a quality problem but not specific to forgetting.

A normalized corpus-level score over a set of multiple tokens, V, is given by:

$$
ForgetGenUse_V = \frac{\sum_{tok \in V} ForgetGenUse[tok]}{\sum_{T_R} \sum_{tok \in V} \#tok_R} \tag{1}
$$

V could consist of all tokens (T_{All}) – in which case the denominator is the test set reference token count – or a subset, for example, out-of-domain (OOD) tokens, in which case the denominator is the count of just those tokens in all reference sentences. We report $ForgetGenUse$ over subword-level tokens for brevity and ease of interpretation, but could equally calculate over words or n-grams, if we wished to extend measurement to better reflect style or syntax.

$ForgetGenUse$ is related to change in unigram BLEU, but there are two crucial differences. First, it is defined for all occurrences of given tokens, whereas BLEU is defined on given segments which will include some instances of a token but not others. Secondly, BLEU masks detrimental vocabulary shift with beneficial shift where a token is translated correctly after adaptation but not before. If a score remains unchanged, some (e.g. out-of-domain) tokens may be translated worse, and others better. We are interested only in tokens which are translated worse. For this reason $ForgetGenUse$ minimises reward for beneficial vocabulary shift by only marking no-longer-correctly-output tokens per segment.

2.2 Intentionally triggering forgetting: Lower quality and detrimental vocabulary shift

Our first experiments intentionally trigger forgetting to explore what is forgotten. We pre-train one encoder-decoder Transformer model for each of German to English (de-en) and English to Japanese (en-ja) NMT

– all subsequent adaptation experiments are a fine-tuning run of one of these models. Appendix **??** gives details of model preparation.

Our generic test sets are concatenated WMT News/General test sets[1] for 2019–22 for de-en and 2020–22 for en-ja. While WMT news sets are often described as 'generic', each may feature quite specific vocabulary—for example, articles about recent news items. Combining test sets increases the reliability of forgetting evaluation via the increased segment count, as well as being more truly generic in topic coverage.

Our adaptation domains are drawn from widely-used datasets with standard test splits. For de-en, we adapt to the five domains from the OPUS multi-domain split produced by aharoni-goldberg-2020-unsupervised[2], including test sets. For en-ja we use three target domains: IWSLT [**?**], KFTT [**?**] and BSD [**?**]. We use test15 as test data for en-ja IWSLT and the standard test splits for the remainder. The datasets, listed in Table **??**, vary in domain and size.

We measure vocabulary shift forgetting via increased $ForgetGenUse$, and track quality degradation via decreases in a string-based metric, BLEU, and a neural metric, COMET[3] [**?**]. $ForgetGenUse$ expresses forgetting in the sense of vocabulary shift. Throughout this paper unless stated otherwise we report a drop in BLEU or COMET relative to the baseline as positive for brevity—high Δ meaning more forgetting. For reference, Table **??** gives generic and in-domain absolute BLEU and COMET scores for the pre-trained models, from which all other absolute values can be calculated.

We fine-tune our pre-trained models on domain-specific datasets until catastrophic forgetting is seen in the sense of quality drop on generic test sets. As we wish to understand the impact of dataset on forgetting independent of other variables, all experiments in this paper adapt for 20K steps. We found this caused similar forgetting to that previously described in the literature [**?**].

Table **??** shows generic forgetting after adaptation to each domain. The different domains exhibit a wide range of forgetting in terms of quality and vocabulary shift. Additionally, although ΔCOMET and ΔBLEU are strongly and significantly correlated across the sets of domains (Kendall's $\tau = 0.8, p < 0.05$) $ForgetGenUse_{All}$ does not have a significant correlation with either. This suggests that corpus-level quality metrics like BLEU and COMET do not sufficiently measure detrimental vocabulary shift. To confirm that the vocabulary shift measured by $ForgetGenUse$ is indeed detrimental despite not correlating with BLEU or COMET, we must analyse what replaces forgotten tokens.

2.3 Which tokens are forgotten, and what replaces them?

Vocabulary shift in a domain-adapted NMT system can be beneficial or detrimental. Beneficial vocabulary shift produces in-domain tokens in in-domain contexts, and out-of-domain tokens where more contextually appropriate. Detrimental vocabulary shift produces in-domain vocabulary tokens when it is not contextually appropriate.

To make this distinction and find the token-level replacements after adaptation to each domain, we compare the generic-model and adapted-model translations of the generic test sets. We align the two sets of translations using symmetrized fast align [**?**], which lets us identify which translation hypothesis tokens change after adaptation. We can also find the frequency of those tokens in the in-domain adaptation dataset.

Table **??** shows examples selected from the most 'forgotten' tokens for each de-en domain. Invariably, replacements have at least one token appearing in the in-domain adaptation dataset. Tokens which themselves

[1] See `https://machinetranslate.org`.

[2] We use the size-capped Subtitles set provided.

[3] wmt20-comet-da

Ref	Domain	#Train	#Test	BLEU / COMET
de-en	Gen (Generic)	43.9M	5769	35.3 / 0.54
de-en	IT (Software)	223K	2000	33.0 / 0.33
	Kor (Koran)	18K	2000	15.9 / -0.03
	Law	467K	2000	45.9 / 0.60
	Med (Medical)	248K	2000	44.8 / 0.55
	Sub (Subtitles)	500K	2000	26.4 / 0.21
en-ja	Gen (Generic)	22.4M	4037	22.5 / 0.43
en-ja	IWSLT (TED talks)	220K	1194	14.0 / 0.16
	KFTT (Kyoto/Wikipedia)	427K	1160	17.6 / 0.34
	BSD (Business/Dialog)	20K	2120	13.6 / 0.46

Table 1: Segment counts and absolute generic model BLEU and COMET on the generic domain test sets and on each in-domain test set.

	de-en					en-ja		
	IT	Kor	Law	Med	Sub	IWSLT	KFTT	BSD
ΔBLEU	5.0	22.3	4.7	8.3	5.7	4.8	2.0	7.0
ΔCOMET	0.11	0.78	0.08	0.16	0.09	0.07	0.06	0.29
$ForgetGenUse_{All}$	0.09	0.25	0.07	0.11	0.09	0.14	0.11	0.16

Table 2: Measuring forgetting on generic test sets for de-en and en-ja.

appear in the adaptation dataset are replaced less frequently, and only by alternatives with far more adaptation set occurrences. The replacements are often semantically similar, judged both by manual inspection and by average FastText embedding cosine similarity [**?**] between the original and replacing tokens.

Surprisingly, we find by inspection that the replacements tend to occur in very different contexts in the adaptation data and generic test set. Of the seven IT domain instances of *Donald*, two refer to computer scientist Knuth and five are subworded *Mc_* or *Mac_*+ *Donald* – none have the same referent as Trump. *Internet* is legitimately replaced by *web* after adaptation to Kor, but the Kor text only uses *web* in the sense of spider's web – including a different source term (*Internet* vs *Netz*). The Med domain only uses *match* as a verb in the context of experiments, not as a noun synonym for *game* as in Med-adapted test outputs – not only a different source term but a different source part of speech (*Spiel* vs e.g. *abstimmen*). The target vocabulary alone can influence forgetting, without requiring a contextually relevant source.

Focusing on the most-forgotten Kor domain, we perform a deeper analysis for two tokens that are forgotten vs two that are not forgotten. Using FastText embedding cosine similarity, we find the closest Kor-domain English tokens which can have the same part-of-speech. For each, the in-domain training count is in brackets:

- *satisfied* (3): *happy* (29) and *pleased* (90)
- *water* (236): *waters* (8) and *lake* (1)
- *England* (0): *Kingdom* (10)
- *genes* (0): *species* (3)

IT	Kor
satisfied 3 → *pleased* 90	*euros* 39 → *EUR* 4988
week 0 → *month* 35	*warranty* 20 → *guarantee* 2331
England 0 → *Kingdom* 10	*Donald* 0 → *President* 1685
accident 0 → *injury* 7	*touch* 5 → *contact* 948
Internet 0 → *web* 1	*wants* 9 → *intends* 416
Law	**Med**
danger 3 → *risk* 5466	*species* 2 → *types* 664
defeat 0 → *loss* 1377	*not* 21.1K → *n't* 50.6K
billion 0 → *million* 464	*citizens* 0 → *people* 292
guests 0 → *visitors* 11	*i.e.* 6273 → *so* 10.8K
game 0 → *match* 5	*infections* 0 → *cases* 207
Sub	
autumn 16 → *fall* 477	*victory* 1 → *win* 120
aircraft 38 → *plane* 435	*Trump* 0 → *Donald* 7
VAT 1 → *sales* 64	

Table 3: High $ForgetGenUse$ tokens for de-en domains—counts are for that token in the in-domain adaptation dataset. Left columns: Output from generic model. Right columns: Most frequent aligned replacements post-adaptation.

When the generic model translates the generic test set, it produces *satisfied* 11 times. The Kor-adapted model produces *pleased* for 10 of these, and *happy* for the remaining. Although all are in-domain, *satisfied* is far rarer. By contrast both models produce *water*, with no more frequent in-domain alternative, in the same locations.

By contrast, we consider out-of-domain tokens. The generic model produces *England* 14 times. The Kor-adapted model replaces 10 of these with *United Kingdom*, with the remaining four null-aligned – indicating undertranslation. Although the phrase *United Kingdom* does not occur in the Kor data, both words do occur separately. *United Kingdom* occurs in similar contexts to *England* during pre-training, making it a plausible, if incorrect, replacement. Interestingly, another out-of-domain term, *genes*, is not forgotten during adaptation. The closest in-domain alternative, *species*, is neither common nor a plausible replacement.

The token forget-replace effect can be triggered by individual subwords, not just whole-word tokens. For example, ignoring post-processing, the pre-trained model produces one token *October* where the Kor model produces two subwords *oc_* + *tober*. Neither word is in the Kor domain—but the subword *oc_* is, making *oc_*+ *tober* preferred.

2.4 Out-of-domain tokens are forgotten more

Given possible different requirements for in-domain (ID) and out-of-domain (OOD) tokens, it is interesting to calculate $ForgetGenUse_{set=\{ID|OOD\}}$ separately for these token subsets. For $ForgetGenUse_{ID}$ we sum and normalize in Equation **??** over all vocabulary tokens that appear in at least one adaptation set reference sentence for each domain. For $ForgetGenUse_{OOD}$ we do so for the complement, again for each domain.

The results in Table **??** show a striking difference in term shift between in-domain and out-of-domain tokens. $ForgetGenUse_{ID}$ has relatively small absolute values, and a small range of values. $ForgetGenUse_{OOD}$

	de-en					en-ja		
	IT	Kor	Law	Med	Sub	IWSLT	KFTT	BSD
$ForgetGenUse_{All}$	0.09	0.25	0.07	0.11	0.09	0.14	0.11	0.16
$ForgetGenUse_{OOD}$	0.37	0.60	0.27	0.47	0.11	0.34	0.65	0.49
$ForgetGenUse_{ID}$	0.07	0.17	0.07	0.09	0.09	0.14	0.11	0.12

Table 4: Calculating $ForgetGenUse$ over tokens that are out-of-domain (OOD) vs in-domain (ID) for each domain.

values by contrast are higher for every domain, meaning out-of-domain tokens are forgotten at a higher rate. $ForgetGenUse_{ID}$ and $ForgetGenUse_{All}$ are equal for Law and Sub (de-en) and IWSLT and KFTT (en-ja): for these domains, almost all generic test set tokens are in-domain.

It is not wholly surprising that out-of-domain tokens are forgotten more than in-domain tokens: a goal of adaptation is to use in-domain terminology instead of generic. However, the vocabulary shift reported by $ForgetGenUse$ does not just consist of generic terms being replaced by their in-domain equivalents. Instead, as shown in Table **??**, shifts can be technically correct but not domain-relevant (*game* $\rightarrow$ *match*, *Trump* $\rightarrow$ *Donald*) – these are unnecessary and can confuse users. While the definition of an NMT domain is an open question [**?**, **?**], it is not at all clear that a user or machine translation client would expect a subtitles domain to entail a shift to US-English terms like *fall* or *aircraft*, or expect an adapted model to no longer use standard but incidentally out-of-domain terms like *accident* or *species*. More serious still are meaning-changing errors (*billion* $\rightarrow$ *million*, *week* $\rightarrow$ *month*) – these unambiguously harm translation. Such vocabulary shift is clearly detrimental and lowers quality.

3 Why does forgetting vary by domain?

In this section we aim to understand the relationship between adaptation dataset and forgetting. This relationship is key for real world adaptation scenarios when deciding whether to adapt, how to adjust tuning hyperparameters, and which if any forgetting mitigation steps to take. To investigate, we compare datasets exhibiting varying degrees of forgetting in terms of multiple domain-differentiating heuristics. We find that many domain features do not correlate with forgetting, but that vocabulary coverage does.

3.1 Controlling for dataset size

Dataset size is recognized as having an impact on MT adaptation performance [**?**], and has been associated in forgetting [**?**]. However, its relationship with forgetting across domains is unclear. We can assess correlation of forgetting with number of lines per dataset for our results in Table **??**. Surprisingly, Kendall's τ is not significant between data size and either of ΔCOMET or ΔBLEU. $ForgetGenUse$ does show significant negative correlation with dataset size ($\tau = 0.7$ $p < 0.05$), suggesting smaller datasets have a greater likelihood of vocabulary shift, but not necessarily general quality degradation.

We further investigate by controlling for dataset size in terms of tokens. We randomly subsample each de-en dataset except for Kor to the same approximate number of tokens as Kor, and likewise for the en-ja domains and BSD. As previously, we adapt the same pre-trained model for 20K steps.

While the forgetting metrics in Table **??** are certainly more clustered than those in Table **??**, there is still significant variation for de-en. None of the subsampled corpora result in the same forgetting as Kor by any

	de-en					en-ja		
	IT-s	Kor	Law-s	Med-s	Sub-s	IWSLT-s	KFTT-s	BSD
ΔBLEU	10.4	22.3	12.3	14.4	11.5	7.2	6.9	7.0
ΔCOMET	0.24	0.78	0.28	0.36	0.24	0.20	0.28	0.29
$ForgetGenUse_{All}$	0.14	0.25	0.15	0.17	0.15	0.17	0.17	0.16

Table 5: Forgetting when adapting on subsampled (-s) domains. All de-en sets except Kor, and all en-ja except BSD, subsampled randomly to approximately the same token count as Kor/BSD respectively.

metric. The order of the domains changes in terms of forgetting: Law-s is in the middle while Law had the least forgetting, and Sub-s now shows slightly more forgetting than IT-s. For en-ja, forgetting is closer across domains, but there is still noticeable variation in ΔCOMET. Dataset size clearly affects absolute amount of forgetting, with all metrics increasing from Tables **??** to **??**. However, forgetting still has no clear relationship with the domain heuristic of token count after subsampling for equivalent size.

3.2 Controlling segment length and quality

Segment length and alignment quality both have potential causative links with catastrophic forgetting. Segment length distribution has been used as a domain feature [**?**]. Short segments in particular can be ambiguous to translate, making them candidates for problematic adaptation [**?**]. Poorly aligned segment pairs likewise can cause hallucinations when used for adaptation [**?**].

We investigate the effect of length on forgetting by adapting to subsets of the domains with the shortest segment pairs. We subsample again to the approximate token count of Kor/BSD, allowing direct comparison with the Table **??** subsampling results. We focus on de-en Law vs Sub, which originally have the longest and shortest average segment lengths respectively.

If change in segment length corresponds to domain shift, we might expect a large forgetting change for Law-ss relative to Law-s, and a small change for Sub-ss relative to Sub-s. Surprisingly, Table **??** shows precisely the reverse. Forgetting for short-segment Law-ss is similar to random-segment Law-s, even though the change in average example length is 49 tokens. By contrast, tuning on Sub-ss results in extreme forgetting relative to Sub-s, which is only 10.5 tokens longer on average. For en-ja, the larger length shift KFTT s-to-ss does result in a larger forgetting shift than for the IWSLT domain. However, in both cases the results are between the extremes seen for de-en. We propose that relative segment length between domains is not necessarily informative, but that a high proportion of very short segment lengths accelerates forgetting.

On inspection the Sub-ss dataset contains many badly aligned source-target pairs. We hypothesize that these may encourage forgetting. To test this we produce a short-subsampled version filtered for quality, Sub-ssf. The shortest examples are sampled after alignment-filtering using LASER[4]. The scores when adapting to Sub-ssf are less dramatically different to Sub-s, although still quite different to the Law-s-to-Law-ss forgetting.

The only other domain with a significant proportion of low LASER score segments is Kor. Adapting to a similarly LASER-filtered Kor set gives scores 0.3 BLEU worse and 0.01 COMET better than adapting to the full Kor set, with no change in $ForgetGenUse$: dataset quality cannot fully explain forgetting. Indeed, the low-quality Sub-ss pairs are also present in the full Sub dataset, which showed little forgetting (Table **??**). Data noise in small enough proportions is not too harmful in these experiments, in line with the findings of

[4] `https://github.com/facebookresearch/LASER`, cutoff score 0.8 selected by inspection.

	Law-s	Law-ss	Sub-s	Sub-ss	Sub-ssf	IWSLT-s	IWSLT-ss	KFTT-s	KFTT-ss
Av. #toks	66.1	17.0	23.0	12.5	13.6	44.4	15.7	60.8	11.2
ΔBLEU	12.3	12.4	11.5	30.4	15.1	7.2	10.4	6.9	11.3
ΔCOMET	0.28	0.27	0.24	1.27	0.34	0.20	0.44	0.28	0.59
$ForgetGenUse_{All}$	0.15	0.15	0.15	0.42	0.19	0.17	0.23	0.17	0.23

Table 6: Forgetting on generic sets, adapting on subsampled datasets. We sample randomly (-s) or sample the shortest (-ss) lines by source plus target token count. Sub-ssf pre-filters the shortest lines using LASER.

khayrallah-koehn-2018-impact.

3.3 Corpus-level score domain heuristics

We investigate the use of corpus-level scores as domain heuristics: negative log-likelihood (NLL) under the pre-trained model, Jensen-Shannon Divergence (JSD) [**?**] between the pre-training dataset and in-domain vocabulary distributions, and the generic vocabulary coverage of each in-domain dataset. Unlike the previous heuristics, we cannot easily control for these by subsampling to obtain datasets with equivalent values. Instead we find their correlation with forgetting metrics. Table **??** gives both heuristics and forgetting metrics.

For generic model likelihood, we score a 10K segment sample of the domain under the generic, pre-trained model. We use length-normalized NLL to indicate similarity to the generic domain without conflating with average segment length. The results do not show a clear relationship with forgetting: Kor and Sub for example have similar NLL but very different forgetting characteristics. Overall we find NLL has a weakly significant correlation with ΔBLEU ($\tau = 0.5$, $p < 0.1$) and no significant correlation with ΔCOMET or $ForgetGenUse$.

We calculate vocabulary distribution divergence between the generic and in-domain vocabularies using JSD[5]. The de-en domain with the greatest divergence from the generic vocabulary—Kor—is indeed the domain with the most forgetting. For en-ja the highest-forgetting domain has a similar JSD to other domains. As well, neither source nor target JSD varies strongly between domains, reducing its utility as a forgetting heuristic. Neither source nor target JSD have a significant Kendall's τ with any of the three forgetting metrics.

Finally, we calculate vocabulary coverage for each domain. We define coverage as the proportion of the generic subword vocabulary that appears at all in the preprocessed segments for a given domain, calculated separately over source and target segments. Both source and target vocabulary coverage vary strongly across domains and have a significant inverse correlation with ΔCOMET ($\tau = 0.6/0.9$ source/target, $p < 0.05$). ΔBLEU has a significant correlation with target coverage ($\tau = 0.7$ $p < 0.05$) and weakly significant with source coverage ($\tau = 0.6$ $p < 0.1$). Interestingly, $ForgetGenUse$ only has a significant correlation with source coverage ($\tau = 0.7$ $p < 0.05$). Although we observed in Section **??** that detrimental vocabulary shift occurs regardless of source content in the test sentence, it does correlate with lower vocabulary similarity between source adaptation sentences.

Vocabulary coverage could also explain the increase in forgetting when aggressively subsampling a dataset, as the number of sentence pairs correlates strongly and significantly with the number of unique vocabulary tokens. Indeed, when we include metrics and coverage for the subsampled domains (final lines of Table **??**), we see significant and strong correlation between coverage and all forgetting metrics.

[5]As proposed by lu-etal-2020-diverging we use unweighted JSD to disentangle vocabulary distribution from relative data size.

	de-en					en-ja		
	IT	Kor	Law	Med	Sub	IWSLT	KFTT	BSD
ΔBLEU	5.0	22.3	4.7	8.3	5.7	4.8	2.0	7.0
ΔCOMET	0.11	0.78	0.08	0.16	0.09	0.07	0.06	0.29
$ForgetGenUse_{All}$	0.09	0.25	0.07	0.11	0.09	0.14	0.11	0.16
Generic NLL	-2.1	-2.4	-1.4	-1.8	-2.6	-1.7	-1.6	-2.1
Src-vcb JSD	0.42	0.50	0.44	0.42	0.42	0.30	0.38	0.39
Trg-vcb JSD	0.39	0.46	0.39	0.40	0.40	0.32	0.42	0.39
Src-vcb cover	0.69	0.23	0.70	0.63	0.75	0.61	0.79	0.31
Trg-vcb cover	0.52	0.23	0.59	0.48	0.62	0.62	0.72	0.29
Src-vcb cover (-s)	0.50	-	0.43	0.44	0.52	0.45	0.52	-
Trg-vcb cover (-s)	0.39	-	0.34	0.34	0.45	0.41	0.46	-

Table 7: Corpus-level score domain heuristics, with forgetting measures for reference. Generic NLL and vocab JSD: closer to 0 is more similar to generic. Final lines: vocab coverage for downsampled domains of Table **??**.

4 Understanding generic data mix-in

In the previous section, we found that adaptation dataset vocabulary coverage has a strong negative correlation with forgetting. A natural question follows: what would be the effect of ensuring all of these adaptation datasets had 100% vocabulary coverage? To answer, we propose and perform Minimal Mix-in, a targeted variant of mixed fine-tuning [**?**]. We focus on target coverage and the quality degradation metrics BLEU and COMET, both for brevity and as these had the strongest relationship with vocabulary coverage.

Mixed fine-tuning aims to mitigate forgetting by mixing examples from the generic training set into the adaptation set. For Minimal Mix-in, we add generic examples to the adaptation set if they include a target token that is not in the adaptation dataset so far. Aside from the novelty of targeted mix-in data, our goal is to examine the effect of improving the vocabulary coverage with minimal other change to the adaptation data and no change at all to the model architecture, adaptation or inference procedure. Excepting Kor and BSD, Minimal Mix-in produces an adaptation dataset where fewer than 10% of examples are generic.

To benchmark the forgetting mitigation possible with a similar non-minimal data augmentation, we follow a popular mixed fine-tuning recipe found in the literature [**?**, **?**] which uses a 1:1 ratio of randomly sampled generic segments to in-domain segments. We refer to this as Random 1:1. Table **??** summarizes the size of the different mix-in interventions.

4.1 Less than 10% of the mix-in data can mitigate 80% of the forgetting

In Table **??** we verify that Random 1:1 generic data mix-in does mitigate more forgetting than Minimal Mix-in, or fine-tuning with no mix-in at all. However, Minimal Mix-in mitigates a large proportion of the forgetting, within 1 BLEU of Random 1:1 for 6 domains and within 0.03 COMET for 7.

Assuming Random 1:1 benchmarks the forgetting mitigation possible while adjusting only data mix-in, the proportion of that mitigation achieved by Minimal Mix-in is $\frac{NoMix-in-MinimalMix-in}{NoMix-in-Random1:1}$. This value is at least 80% of the Random 1:1 forgetting mitigation for 6 domains when measuring ΔBLEU and for 4 domains when measuring ΔCOMET – and at least 70% for 6 domains over both metrics. Minimal Mix-in achieves this while mixing in less than 10% as much generic data for all domains except Kor and BSD. It

Domain	Random 1:1	Minimal Mix-in
de-en IT	222927	18861
de-en Kor	17982	22769
de-en Law	467309	17485
de-en Med	248099	19605
de-en Sub	500000	16632
en-ja IWSLT	219716	11205
en-ja KFTT	427353	8547
en-ja BSD	20000	16860

Table 8: Number of generic mix-in lines for each strategy. Random 1:1 is by definition the same size as the in-domain dataset, and Minimal Mix-in is often far smaller.

is worth noting that Sub and IWSLT, for which Minimal Mix-in performs less well, have high vocabulary coverage of the generic test set, as indicated in the discussion of Table **??**.

Minimal Mix-in also reduces forgetting variation across the domains. This is in line with our prior finding that vocabulary coverage for an in-domain adaptation set correlates strongly with forgetting. Our experiment effectively sets vocabulary coverage to be the same – 100% – for every domain, which results in correspondingly very similar forgetting across all domains even if ensuring coverage does not mitigate forgetting entirely. This finding also supports work by gu-feng-2020-investigating showing that, when adapting with frozen parameters, decoder embeddings are most correlated with preserved generic performance. Our results, from a data perspective, suggest that future work might focus on specifically decoder embeddings for tokens not in the in-domain data.

Finally we examine vocabulary shift. We confirm by inspection that the less desirable replacements from Table **??** are no more. For example, for the Med domain, $ForgetGenUse[billion]$ drops from 18 to 0, meaning everywhere the baseline model produces *billion* correctly, so does the adapted model. Analysing $ForgetGenUse$, we find a pattern generally the same as for COMET and BLEU—Minimal Mix-in is on par with a 1:1 generic ratio. The main exception is en-ja IWSLT. It is possible that domains where generic test vocabulary is almost entirely covered by the in-domain data already may benefit less from Minimal Mix-in.

As noted in Section **??**, $ForgetGenUse$ has less correlation with target coverage. A richer mix-in set may be required to address detrimental vocabulary shift.

4.2 Minimal mix-in, better in-domain scores

A primary goal of adapting NMT is improved in-domain translation. Table **??** gives quality metric deltas on the in-domain test sets: higher values are now better. A 1:1 generic ratio has a negative impact, with noticeable BLEU and COMET drops relative to unmixed fine-tuning. By contrast, Minimal Mix-in scores similarly to unmixed fine-tuning for all except de-en Med. Improvement in terms of ΔCOMET shows less variation than under ΔBLEU, possibly because COMET assigns higher scores to paraphrases which may not use domain-specific terminology. Mixing in large amounts of generic data reduces scores relative to Minimal Mix-in. It is interesting to note that for the smallest domain, Kor, mixing no generic data also leads to reduced in-domain performance.

	de-en						en-ja			
Mix-in method	IT	Kor	Law	Med	Sub	Mean	IWSLT	KFTT	BSD	Mean
ΔBLEU										
No mix-in	5.0	22.3	4.7	8.3	5.7	9.2	4.8	2.0	7.0	4.6
Random 1:1	0.5	5.0	0.6	0.7	0.8	1.5	-0.2	0.1	1.3	0.4
Minimal Mix-in	1.1	4.1	1.3	1.7	3.0	2.2	3.2	0.1	1.7	1.7
ΔCOMET										
Fine-tune, no mix-in	0.11	0.78	0.08	0.16	0.09	0.24	0.07	0.06	0.29	0.14
Random 1:1	0.01	0.07	0.01	0.01	0.01	0.02	-0.02	0.0	0.03	0.00
Minimal Mix-in	0.04	0.09	0.03	0.04	0.04	0.05	0.03	0.01	0.04	0.03
$ForgetGenUse$										
No mix-in	0.09	0.25	0.07	0.11	0.09	0.12	0.14	0.11	0.16	0.14
Random 1:1	0.03	0.08	0.02	0.03	0.03	0.04	0.05	0.03	0.08	0.05
Minimal Mix-in	0.05	0.07	0.04	0.05	0.06	0.05	0.12	0.08	0.09	0.10

Table 9: Forgetting metrics on generic test sets, varying the mix-in dataset when fine-tuning for 20K iterations in each case. Lower is better for all metrics. Negative scores indicate improvement.

	de-en						en-ja			
Mix-in method	IT	Kor	Law	Med	Sub	Mean	IWSLT	KFTT	BSD	Mean
ΔBLEU										
No mix-in	8.2	5.3	5.8	6.2	3.6	5.8	3.5	10.5	4.8	6.3
Random 1:1	6.5	6.1	4.4	3.4	2.7	4.6	3.2	9.1	4.5	5.6
Minimal Mix-in	7.9	6.4	5.6	4.7	3.4	5.6	3.6	10.3	4.5	6.1
ΔCOMET										
No mix-in	0.26	0.09	0.05	0.05	0.07	0.10	0.04	0.12	0.04	0.07
Random 1:1	0.23	0.11	0.04	0.04	0.04	0.09	0.05	0.10	0.07	0.07
Minimal Mix-in	0.26	0.12	0.05	0.05	0.06	0.11	0.05	0.12	0.05	0.07

Table 10: ΔBLEU and ΔCOMET on in-domain test sets for the same experiments as in Table **??**. Higher is better.

5 Conclusions

This paper investigates what is forgotten during NMT domain adaptation, and why. We show that vocabulary shift during adaptation is not necessarily beneficial, and that detrimental shift can be orthogonal to quality metrics. We find forgetting correlates with in-domain vocabulary coverage, allowing better prediction of how adaptation will behave on a particular dataset. Our findings emphasise that NMT adaptation research should not be dataset agnostic: in-domain data characteristics are critical to how adaptation can succeed or fail.

6 Limitations

Since our investigation is dataset-dependent, it is necessarily limited by the data and languages we have used. We report on a selection of widely used, diverse domain-specific datasets, as available for two language pairs

with contrasting resources and distance. Additional language pairs or domains would allow us to generalise better.

Another limitation is model variety. In the interests of brevity, time and cost we only conduct our experiments with moderately sized Transformers trained for NMT. There has been much recent interest in machine translation by prompting Large Language Models (LLMs) pre-trained on huge uncurated datasets [**?**]. Work concurrent with ours by pang-etal-2024-salute observe that LLMs also struggle with domain-specific translation. Indeed, when fine-tuning on the same de-en OPUS domain-specific datasets as us, they report that LLMs exhibit similar behaviour in terms of 'forgetting' domain-specific terminology in preference to tokens appearing in the adaptation set, although they do not attempt to explain or mitigate this. We leave confirming experiments to future work.

Acknowledgments

We thank the anonymous reviewers for their helpful feedback. This work was completed while the first author was at RWS.

A Experimental setup

We pre-train two Transformer models using the Tensorflow T2T toolkit [**?**], one for each of German-English (de-en) and English-Japanese (en-ja). Both use BPE vocabulary [**?**], with details given in Table **??**. Following findings from the most recent WMT shared task [**?**] on Transformer NMT models, we use deep encoders with relatively shallow decoders for a balance of speed and quality. We found a slightly deeper encoder and smaller, not shared BPE vocabulary gave better results for en-ja in initial testing.

The de-en model is pre-trained on 43.9M lines of parallel data made available via the WMT shared task: Paracrawl v9, Europarl v10, NewsCommentary v14, Tilde and WikiMatrix [**?**]. The en-ja model is pre-trained on 22.4M lines of JParacrawl v3.0 [**?**].

When calculating BLEU for en-ja, we use Sacrebleu v2.0 [**?**] with the Mecab tokenizer.

To minimize our computational and energy use, we pre-train each model only once on 4 GPUs for approximately two days. Each fine-tuning run of 20K steps takes approximately 1 additional hour of training.

	de-en	en-ja
Encoder layers	15	18
Decoder layers	3	3
Hidden size	2560	2560
Filter size	640	640
# BPE merges	32K	16K
Shared BPE	Y	N

Table 11: Pre-trained model specifications