Analysis of Vocabulary and Subword Tokenization Settings for Optimal Fine-tuning of MT: A Case Study of In-domain Translation

Javad Pourmostafa Roshan Sharami, Dimitar Shterionov, Pieter Spronck
Department of Cognitive Science and Artificial Intelligence
Tilburg University, The Netherlands
{j.pourmostafa,d.shterionov,p.spronck}@tilburguniversity.edu

Abstract

The choice of vocabulary and subword (SW) tokenization has a significant impact on both training and fine-tuning of language and translation models. Fine-tuning is a common practice in optimizing a model with respect to new data. However, new data potentially introduces new words (or tokens), which, if not considered, may lead to suboptimal performance. In addition, the distribution of tokens in the new data can differ from the distribution of the original data. As such, the original SW tokenization model could be less suitable for the new data. With this work, we aim to gain better insights on the impact of SW tokenization and vocabulary generation on the performance of neural machine translation (NMT) models fine-tuned to a specific domain. To do so, we compare several strategies for SW tokenization and vocabulary generation and investigate the performance of the resulting models.

Our findings show that the best way to fine-tune for domain adaptation is to consistently use both BPE and vocabulary from the in-domain data, which helps the model pick up on important domain-specific terms. At the same time, it is crucial not to lose sight of the vocabulary of the base (pre-trained) model—maintaining coverage of this vocabulary ensures the model keeps its general language abilities. The most successful configurations are those that introduce plenty of frequent domain terms while still retaining a substantial portion of the base model vocabulary, leading to noticeably better translation quality and adaptation, as seen in higher BLEU scores. These benefits, however, often come with greater computational costs, such as longer training times, since the model must learn more new tokens. Conversely, approaches that skip important domain terms or combine mismatched tokenization and vocabulary do not perform as well, making it clear that both domain-specific adaptation and broad vocabulary coverage matter—and that these gains are realized when the vocabulary preserves a good portion of the base (pre-trained) model.

While using in-domain BPE and vocabulary yields the best domain adaptation, it substantially reduces out-of-domain translation quality. Hybrid configurations that combine base and domain vocabularies help balance this trade-off, maintaining broader translation capabilities alongside improved domain performance.

1 Introduction and background

Fine-tuning is a common practice in optimizing MT and pre-trained language models with respect to new data. It is often in the context of DA where an existing model is tuned to perform better on a specific domain (different from what the model was originally trained for) (Luong et al., 2015; Dakwale and Monz, 2017; Wang et al., 2019; Mahdieh et al., 2020; Chopra et al., 2023). The positive effect of fine-tuning has been demonstrated in various previous works. For example, Luong et al. (2015) trained an NMT model on English-German general-domain data and then fine-tuned it on a conversational data in the same languages, leading to an increase of 3.8 BLEU (Papineni et al., 2002) points compared to the original model. Sharami et al. (2022) show that fine-tuning is preferred (as it leads to better results) than training from scratch, even if the data allows the latter. To improve the translation performance on a new domain (without degrading the performance on the generic domain) is to ensemble the fine-tuned model with the already trained baseline, as done by Freitag and Al-Onaizan (2016). However, while they achieve a substantial increase of quality (+7.2 BLEU points), they note that because the in-domain data comprises of new vocabulary and linguistic features that are different from the generic data, the performance of the fine-tuned models drops for the generic domain task, especially when it comes to

Proceedings of Recent Advances in Natural Language Processing,pages 970–979
Varna, Sep 8–10, 2025
https://doi.org/10.26615/978-954-452-098-4-111

domain-specific contexts (e.g., medical and legal domains).

While newly introduced data brings in new information, i.e., new, unseen words, it could be that, statistically, segmentation into SWs is significantly different from the segmentation of the original model (Lim et al., 2018; Yeung, 2019; Sato et al., 2020; Hwang et al., 2024). If not properly addressed, this new information may have an adverse effect on the system.

To address this problem, Sato et al. (2020) proposed a method to adapt the embedding layers of the initial model to the target domain by projecting the general word embedding obtained from target-domain monolingual data onto source-domain embeddings. They reported a 3.86 and 3.28 BLEU points gain in English→Japanese and German→English translation, respectively.

In this paper, we investigate the impact of using different SWs and vocabularies on the performance of fine-tuned NMT systems. We identify a best-case setup and preferable setups under constrained fine-tuning conditions, such as limited domain-specific data. That is, we aim to investigate which fine-tuning conditions (or settings) of a domain-specific model lead to the best performance. Specifically, our objectives are:

1. Identify optimal SW combination choices and vocabulary configurations for a given MT model and fine-tuning dataset.

2. Determine the best fine-tuning conditions under data limitations.

To achieve the aforementioned objectives, we use one large dataset ($\sim$12.7 million parallel sentences) for training and a smaller in-domain dataset ($\sim$248,000 parallel sentences) for fine-tuning multiple MT systems. This setup allows us to examine the extent to which a model trained on a substantial amount of general-domain data can be improved by fine-tuning with additional domain-specific data, which alone would be insufficient to train a robust model from scratch.

Each fine-tuned alternative, is trained on a different set of options of how the SWs and the vocabulary are created. We analyze these fine-tuning strategies to find the best setup based on available data. In our case study, for example, we have access to the data of both models (initial and fine-tuned). However, as already discussed in (Freitag and Al-Onaizan, 2016; Dakwale and Monz, 2017; Zimelewicz et al., 2024), initial models are mostly deployed in an application; thus data might not be available at the production time. As such, it is paramount to have a guideline based on the available data that determines how to best generate SWs and vocabularies. In this work, we use Byte-Pair Encoding (BPE) (Sennrich et al., 2016) for SW units.[1]

It is noteworthy that the point of this research is to investigate the best fine-tuning setup, rather than identifying the best model. Typically, fine-tuning involves tokenizing the new data using the vocabulary originally employed in training the model. This ensures consistency in SW segmentation and prevents discrepancies in word representations. However, it is not always evident whether this practice yields the best translation performance, especially when the new domain introduces a significantly different linguistic distribution or unseen vocabulary. Thus, we explore alternative approaches to SW segmentation and vocabulary creation to determine if different configurations could lead to better fine-tuning outcomes. Specifically, given a pre-training dataset A—whether in-domain, out-of-domain, synthetic, e.g., generated using methods like those in (Sharami et al., 2023), or authentic—and a fine-tuning dataset B, we investigate which tokenization and vocabulary configurations best enable the model to retain and adapt pre-training-derived parameters in a way that improves translation quality on B.

This paper is organized as follows. Section 2 presents the key decision points that frame our study. Section 3 describes the datasets used in our experiments. Section 4 details our experimental design and the overall training framework. Section 5 reports and analyzes the results, incorporating relevant recommendations. In Section 6, we discuss the limitations of our approach. Finally, Section 7 summarizes our findings and outlines directions for future work.

2 Decision points

Given a model M trained on a dataset D, which represents a specific domain d, and a fine-tuning dataset E representative for domain e, the following decision points need to be made:

[1] Throughout this paper, we use "subword" (SW) and "BPE" synonymously, as all experiments use BPE for subword segmentation.

2.1 SW segmentation

A key decision in fine-tuning is determining how to segment words into SW units. This choice affects how the model processes domain-specific terminology and generalizes across datasets. We consider three approaches:

- **Reusing the original SW model** trained on D (D_{SW}). This maintains consistency with the pre-trained model.
- **Training a new SW model on the fine-tuning dataset** (E_{SW}) to better capture domain-specific terminology.
- **Training a SW model on the combined dataset** ($(D+E)_{\text{SW}}$) to integrate both the original and fine-tuning data.

2.2 Vocabulary creation

SW segmentation techniques arose in response to two major challenges in NMT: (i) the lack of generalizability—models often fail to process words not seen during training, leading to out-of-vocabulary (OOV) problems and degraded performance; and (ii) the need to limit vocabulary size, as large vocabularies increase memory consumption and computational cost, which remains a practical constraint in current neural translation systems, particularly when working with large models or limited GPU resources.

Fine-tuning introduces a third, less often addressed challenge: whether the vocabulary used during adaptation adequately captures the token distribution of the fine-tuning dataset. If not, domain-specific content may be poorly represented, limiting the effectiveness of adaptation.

We consider three strategies for vocabulary construction:

- **Reusing the original vocabulary** — the vocabulary that the pre-trained model M was originally trained with (denoted $|D|$). This strategy ensures full compatibility with the pre-trained token embeddings and does not require any modifications to the embedding space.
- **Expanding the vocabulary** — augmenting the original vocabulary with additional tokens found in the fine-tuning dataset E, resulting in a combined vocabulary $|D+E|$. This approach aims to better cover domain-specific terms in E while retaining compatibility with M's original vocabulary.
- **Constructing a new vocabulary solely from the fine-tuning data** — generating the vocabulary exclusively from E (denoted $|E|$). This strategy maximizes domain-specific representational capacity but introduces a mismatch with the pre-trained vocabulary of M.

Handling vocabulary-embedding alignment. In the first strategy ($|D|$), the embedding space remains unchanged, as all tokens are already present in the pre-trained model.

In the second and third strategies ($|D+E|$ and $|E|$), we introduce new tokens absent from the original vocabulary. To accommodate these, we extend the embedding matrix by appending randomly initialized vectors for the new tokens while preserving the original embeddings.

The key distinction lies in the degree of divergence from the original model. Strategy 2 retains the original vocabulary and extends it with tokens from E, maintaining alignment with the pre-trained structure. In contrast, Strategy 3 derives both the vocabulary and BPE model entirely from E, resulting in a larger mismatch with the pre-trained model and necessitating greater adaptation during fine-tuning.

Since SW segmentation and vocabulary creation are interdependent, we explore all feasible combinations, resulting in **nine configurations**. These include applying each SW model ($D_{\text{SW}}, E_{\text{SW}}, (D+E)_{\text{SW}}$) with different vocabulary choices ($|D|, |D+E|, |E|$).

Following these decision points, given a fine-tuning dataset, we can consider three SW models. With these models, we (i) *tokenize the vocabulary sources*, and (ii) *tokenize the training sets for fine-tuning*. Typically, these two processes are tied to each other, i.e., once the SW model is learned and applied to the training data, the vocabulary is the set of SW units that appear in the (processed) data. However, this is not a hard constraint.

For instance, dataset E can be processed with E_{SW}, but the vocabulary used for training can still be based on D and derived from applying D_{SW}. Such mismatched configurations, though theoretically possible, can lead to tokenization inconsistencies and degrade model performance. Since they are suboptimal, we exclude them from this study.

3 Data

We used two datasets: (i) a large *out-of-domain* corpus consisting of approximately 12.7 million English-German sentence pairs drawn from the WMT18 dataset[2], and (ii) a smaller ($\sim$248K sentence pairs) *in-domain medical* corpus extracted from the multi-domain English-German data introduced by (Koehn and Knowles, 2017).

Out-of-domain dataset The out-of-domain corpus used to train our base model is a randomly selected subset of the WMT18 English-German dataset, which contains parallel data from various domains. We selected approximately 12.7 million sentence pairs to balance domain coverage with training efficiency.

In-domain dataset For fine-tuning, we used 248,099 English-German sentence pairs from the medical domain of the multi-domain dataset introduced by (Koehn and Knowles, 2017). We used the cleaned and re-split version provided by (Aharoni and Goldberg, 2020), which removes duplicates and prevents data leakage between train, dev, and test sets.

Combined dataset ($D + E$) For configurations requiring both D and E, we oversampled the in-domain medical data to match the size of the WMT18 subset and concatenated them. The combined data was shuffled and used to train BPE models or extract vocabularies. This ensures that both domains are equally represented, avoiding bias toward the larger out-of-domain corpus.

4 Experiments

To investigate the impact of SW and vocabulary generation choices on fine-tuning, we followed the decision points outlined in Section 2 and ran experiments using the English-German data described in Section 3. We compared the resulting fine-tuned models using BLEU (Papineni et al., 2002), TER (Snover et al., 2006), chrF2 (Popović, 2015). Additionally, we measured training time and estimated CO_2 emissions using `CodeCarbon` (Courty et al., 2024).

4.1 Experimental design

Given our task—fine-tuning a model trained on the WMT18 out-of-domain dataset (D) using in-domain medical data (E)—a total of 9 theoretical configurations exist, arising from three possible vocabulary sources (D, E, or $D + E$) and three BPE models trained on the same sources. Each configuration couples one vocabulary source with one BPE model, which is used for both vocabulary construction and fine-tuning data segmentation.

However, as discussed in Section 2, we imposed constraints to ensure segmentation consistency. Specifically, we excluded configurations where the vocabulary is derived from one source (D or E), but the fine-tuning segmentation is performed using a BPE model trained on the combined dataset ($D + E$). These mismatches introduce inconsistencies, as the vocabulary may not align with how the data is tokenized. Since both D and E are available, we ensure that the same BPE model is used for both vocabulary construction and fine-tuning segmentation.

In total, there are 9 possible configurations (from all combinations of BPE models and vocabulary sources). However, we exclude 2 inconsistent configurations, leaving 7 valid configurations for our experiments (see Table 1).

Config.	BPE for vocab. + FT data	Vocabulary source
C1	D_{BPE}	D
C2	D_{BPE}	$D + E$
C3	D_{BPE}	E
C4	E_{BPE}	D
C5	E_{BPE}	$D + E$
C6	E_{BPE}	E
C7	$(D + E)_{\text{BPE}}$	$D + E$

Table 1: **Valid fine-tuning configurations.** Each row represents a consistent setup where the same BPE model is used for both segmenting the fine-tuning data and constructing the vocabulary. D refers to the WMT18 out-of-domain dataset; E refers to the in-domain medical dataset.

4.2 Model architecture and training Setup

Framework and model architecture We used the OpenNMT-py[3] framework (Klein et al., 2017) to train and fine-tune Transformer-based NMT models (Vaswani et al., 2017). Each model had 6 encoder and 6 decoder layers, 512-dimensional

[2] `http://statmt.org/wmt18/translation-task.html`. The original WMT18 dataset is considerably larger; we selected a representative subset to reduce computational costs. The selected subset and all preprocessing scripts are released for reproducibility.

[3] `https://opennmt.net/OpenNMT-py/`

embeddings, 8 attention heads, and a feed-forward size of 2048. We used the Noam optimizer schedule with a learning rate of 2.0, 8,000 warmup steps, and label smoothing of 0.1. Batching was done over 10,240 tokens with gradient accumulation over 4 steps.

Training setup All models were trained for up to 200,000 steps, with validation and checkpointing every 1,000 steps. We applied early stopping after 10 validations without improvement. All experiments—including the base and fine-tuned models—were run on a single NVIDIA A40 GPU.

Base model The base model was trained on the WMT18 out-of-domain dataset. We applied BPE with 50K merge operations to both source and target sides. The resulting vocabularies and tokenized data were used to train the initial Transformer model, which served as the starting point for all fine-tuning experiments.

Fine-tuning Fine-tuning was done on the in-domain medical dataset using the same model architecture and training settings. Each configuration (C1–C7) used a specific combination of vocabulary source and BPE model (see Table 1). The base model checkpoint was reused across all configurations, and only the vocabulary and tokenized data differed. BLEU was used to track validation performance.

BPE settings We trained separate BPE models for the source and target sides. The number of merge operations depended on dataset size: 8K merges for corpora with fewer than 100K lines, 30K for those between 100K and 1M, and 50K for larger ones. This choice is supported by prior work, which shows that smaller vocabularies benefit Transformer models (Kudo, 2018), and that 2K–8K merges perform best for low-resource datasets (Adlaon and Marcos, 2024). Our BPE models were used consistently for both vocabulary construction and fine-tuning data segmentation.

5 Results and analysis

In this section, we present the evaluation results and statistical comparisons of our fine-tuning setups. We also explore vocabulary overlaps to understand how token and vocabulary choices impact performance and adaptation.

5.1 Analysis of fine-tuning results

Table 2 summarizes the performance of all fine-tuning configurations. To better interpret these results, we performed pairwise bootstrap tests on BLEU scores (Section 5.1.1), interpreting p-values as a continuous measure of confidence without enforcing a strict threshold. TER and chrF2 metrics supplemented the analysis to refine the ranking.

We ranked configurations using the following criteria:

1. BLEU scores, weighted by the strength of statistical evidence from p-values.
2. TER to resolve ties or unclear BLEU differences.
3. chrF2 as a final tiebreaker if both BLEU and TER were inconclusive.

Accordingly, the ranking from best to worst is:

$$C6 \succ C1 \succ C5 \succ C2 \succ C7 \succ C3 \succ C4,$$

where $\succ$ denotes a configuration that performs better or more reliably than the next.

Top configuration ($C6$). Configuration $C6$ uses both BPE and vocabulary exclusively from the in-domain data E, resulting in the highest BLEU and best TER and chrF2 scores. Statistical tests show that $C6$ significantly outperforms all other configurations, confirming the advantage of aligning segmentation and vocabulary strictly with the fine-tuning domain.

Strong middle tier ($C1$, $C5$, $C2$). Configurations $C1$, $C5$, and $C2$ achieve similar BLEU scores, with statistical evidence showing no clear superiority among them. $C1$ (BPE and vocabulary from out-of-domain D) slightly leads numerically, while $C5$ (in-domain BPE, combined vocabulary) offers better TER than $C2$ (out-of-domain BPE, combined vocabulary), which justifies the order. These results suggest incorporating some in-domain vocabulary or combining datasets can yield competitive results if full in-domain access (for both BPE and vocabulary) is not possible.

Lower performing configurations ($C7$, $C3$, and $C4$). Configurations $C7$ and $C3$ perform moderately but are consistently behind the mid-tier cluster. Configuration $C4$ ranks last, likely because of a mismatch between its in-domain BPE and out-of-domain vocabulary, which impairs tokenization and reduces fine-tuning effectiveness.

Config	BPE Model (vocab. + FT)	Vocabulary Source	BLEU↑	chrF2↑	TER↓	CO$_2$ (g)↓	Time (h)↓
C1	D_{BPE}	D	53.6	69.4	49.3	1658.69	07:45
C2	D_{BPE}	$D + E$	53.4	69.5	49.9	1198.66	05:15
C3	D_{BPE}	E	51.7	68.4	50.9	907.24	04:00
C4	E_{BPE}	D	46.6	64.5	53.0	723.94	03:11
C5	E_{BPE}	$D + E$	53.1	68.9	49.7	729.04	03:15
C6	E_{BPE}	E	**54.8**	**69.8**	**48.9**	1587.41	09:30
C7	$(D + E)_{\text{BPE}}$	$D + E$	53.2	69.1	50.1	543.84	03:08

Table 2: **Evaluation scores of fine-tuned models.** Each configuration pairs a specific BPE model and vocabulary source consistently. All models were fine-tuned on the in-domain dataset E and evaluated on the same test set. CO$_2$ emissions and training times are recorded during fine-tuning.

Practical recommendations. For optimal fine-tuning, use both BPE and vocabulary consistently derived from the in-domain data, as exemplified by configuration $C6$. When full access to in-domain data or vocabulary is limited—due to privacy, proprietary constraints, or resource availability—fine-tuning remains possible but may yield reduced adaptation effectiveness. In such cases, configurations like $C1$ and $C2$ offer robust alternatives by leveraging available data while balancing performance and practicality. It is important to avoid mixing BPE and vocabulary from mismatched domains, as this often leads to suboptimal tokenization and degraded translation quality. Overall, aligning tokenization and vocabulary with domain data maximizes fine-tuning benefits, but adapting with limited data can still provide meaningful improvements compared to no adaptation.

5.1.1 BLEU score statistical comparison

We conducted pairwise bootstrap tests on BLEU scores using 1,000 iterations. Table 3 shows the p-values for all configuration pairs. Diagonal entries represent self-comparisons.

	C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**
C1	–	0.545	0.000	0.000	0.200	0.852	0.102
C2	0.447	–	0.000	0.000	0.172	0.792	0.063
C3	1.000	1.000	–	0.000	0.995	1.000	0.989
C4	1.000	1.000	1.000	–	1.000	1.000	1.000
C5	0.774	0.813	0.006	0.000	–	0.966	0.286
C6	0.149	0.174	0.000	0.000	0.026	–	0.012
C7	0.896	0.929	0.010	0.000	0.701	0.987	–

Table 3: **Pairwise bootstrap p-values for BLEU scores** (1,000 iterations). Diagonal entries represent self-comparisons. *All values are provided for reference only; no statistical significance threshold is applied.*

5.2 Training time and CO$_2$ emissions

The training times and estimated CO$_2$ emissions in Table 2 show the resource demands of each fine-tuning setup. The best-performing configuration, $C6$, took the longest—about 9.5 hours—and had a higher carbon footprint, likely because it had to learn many new domain-specific tokens.

Other configurations such as $C7$, $C4$, and $C5$ completed training notably faster, around three hours, and had lower CO$_2$ emissions compared to $C6$. This can be attributed to several factors. Configurations $C7$ and $C5$ utilize vocabularies with higher overlap to the baseline tokens, meaning fewer new domain-specific tokens need to be learned, which reduces training complexity and time. On the other hand, $C4$ exhibits both a vocabulary and BPE mismatch, which limits effective fine-tuning and results in quicker but less effective training. In summary, configurations with less vocabulary adaptation or mismatched tokenization require less training time and energy but tend to yield lower translation quality.

Overall, while domain-aligned fine-tuning boosts performance, it can require more time and energy—something to consider in real-world applications.

5.3 Vocabulary overlap analysis

To better understand how vocabulary choice influences fine-tuning, we measured overlap between each configuration's vocabulary and the baseline WMT vocabulary in terms of token frequency coverage. This approach more accurately reflects the practical impact of commonly used tokens on model performance, as it weights tokens by how often they occur in the data. Table 4 summarizes the key statistics:

Config	BPE Model (Vocab+FT)	Vocab SRC	BLEU↑	SRC Overlap %	TGT Overlap %	New Tokens SRC	New Tokens TGT
C6	E_{BPE}	E	54.8	82.84	77.81	13,022	13,559
C1	D_{BPE}	D	53.6	100	100	0	0
C5	E_{BPE}	$D + E$	53.1	83.04	77.95	14,289	14,157
C2	D_{BPE}	$D + E$	53.4	83.04	77.95	11,736	11,804
C7	$(D + E)_{\text{BPE}}$	$D + E$	53.2	97.61	95.72	14,300	15,077
C3	D_{BPE}	E	51.7	83.04	77.95	11,736	11,804
C4	E_{BPE}	D	46.6	90.70	90.46	0	0

Table 4: **Vocabulary overlap and BLEU scores per configuration.** Configurations are listed in order of their overall performance ranking. SRC and TGT overlap percentages indicate the proportion of baseline tokens retained. New Tokens columns count tokens unique to the configuration's vocabulary.

- **SRC/TGT Overlap (%)**: The percentage of total token frequency (i.e., the sum of token counts) in the baseline WMT vocabulary that is also present in the configuration's vocabulary, calculated for source (English) and target (German) separately. This reflects not just the number of shared tokens, but their practical frequency in baseline data.
- **New Tokens**: The number of tokens in the configuration's vocabulary that do not appear in the baseline vocabulary (after filtering), representing domain-specific or new tokens introduced by the configuration.

The results demonstrate that configurations incorporating in-domain vocabulary (e.g., $C6$, $C5$, and $C7$) introduce a substantial number of new domain-specific tokens, which is associated with their superior BLEU scores and more effective domain adaptation. In contrast, $C1$, relying solely on the baseline vocabulary, achieves complete overlap but lacks critical domain-specific terms, limiting its adaptability. The notably poor performance of $C4$ corresponds with its lower vocabulary overlap and the evident mismatch between its BPE model and vocabulary source, underscoring the detrimental impact of inconsistent tokenization strategies.

These findings robustly support our practical recommendation: for optimal fine-tuning, vocabulary and BPE should be consistently derived from the same in-domain data. Such alignment ensures richer domain-specific token representation, ultimately leading to enhanced translation accuracy and better overall model performance.

5.4 Out-of-domain performance analysis

To quantify the impact of different fine-tuning strategies on generalization, we evaluated all configurations on the original out-of-domain (WMT18, D) test set. Table 5 reports BLEU scores for each configuration, alongside the absolute and relative drop with respect to the pre-trained base model (before fine-tuning).

Config	BPE/Vocab Source	BLEU	Drop	Drop (%)
Base	D_{BPE}, D (pre-trained)	33.9	–	–
C2	$D_{\text{BPE}}, D + E$	15.1	-18.8	-55.5
C7	$(D + E)_{\text{BPE}}, D + E$	15.0	-18.9	-55.8
C3	D_{BPE}, E	13.3	-20.6	-60.8
C1	D_{BPE}, D	13.1	-20.8	-61.4
C4	E_{BPE}, D	10.2	-23.7	-69.9
C6	E_{BPE}, E	7.7	-26.2	-77.3
C5	$E_{\text{BPE}}, D + E$	7.0	-26.9	-79.4

Table 5: **Out-of-domain BLEU scores.** Performance on the WMT18 (D) test set for all configurations, ranked by smallest drop relative to the pre-trained base model.

The results illustrate the trade-off introduced by domain adaptation: as the model is adapted to the in-domain data, out-of-domain performance drops substantially across all configurations. This degradation is most pronounced when both BPE and vocabulary are derived solely from in-domain data (C5, C6), indicating strong domain specialization at the expense of generalization.

For practitioners seeking to balance domain adaptation and general translation quality, we recommend hybrid configurations such as C_2 and C_7, which use either the original BPE with a combined vocabulary or a combined BPE and vocabulary. These setups moderate the drop in out-of-domain BLEU, preserving more general-domain competence while still offering improved domain adaptation.

6 Limitations

Despite the systematic and thorough analysis, we acknowledge several drawbacks and limitations of our work. Addressing these in the future would complement this research and expand the understanding of the impact of data processing on model performance.

- **Focus on MT:** Our evaluation focused on NMT systems. However, neural language models are also impacted by how the training and fine-tuning data is processed and used, as well as the limitations placed on the vocabulary. This is even more pertinent with the progress in large language models (LLMs). We did not analyze the performance of LLMs, which is a more complex task, especially in the case of multi-lingual LLMs capable of translation.

- **Fine-tuning data:** Our study focused exclusively on the medical domain for fine-tuning. Future research could consider additional specialized domains to evaluate the generalizability of the findings.

- **Use of BPE only:** We employed BPE only and did not consider other methods such as SentencePiece (Kudo and Richardson, 2018) or LMVR (Ataman et al., 2017). This was a deliberate choice, as it was up to us which model and method to use during training and fine-tuning.

- **Hyperparameters:** We used the model's default hyperparameters and did not perform hyperparameter optimization or tuning. This was not necessary as we aimed to compare the impact of the SW algorithms under the same conditions. However, we acknowledge that fine-tuning hyperparameters would impact the performance of original and fine-tuned models, and we hypothesize a correlation with how the vocabulary is constructed.

7 Conclusion and Future Work

In this work, we presented a systematic analysis of vocabulary and SW tokenization settings for fine-tuning NMT models, using a large out-of-domain corpus (WMT18) and a specialized in-domain medical dataset as a case study. By comparing seven realistic fine-tuning setups that varied in BPE segmentation and vocabulary generation, we identified clear practical guidelines for domain adaptation.

Our results show that the most effective fine-tuning is achieved when both BPE and vocabulary are derived from the in-domain data, allowing the model to better capture frequent and relevant domain-specific terms. At the same time, we find that maintaining a substantial overlap with the vocabulary of the base model (originally trained on out-of-domain data) is essential for preserving general language coverage and ensuring stable adaptation. The best-performing configurations in our experiments balanced these two needs: they introduced many new, high-frequency in-domain tokens while still retaining a good portion of the base model vocabulary. However, this approach tends to require more computational resources, such as increased training time and higher energy consumption, due to the need for the model to learn and integrate more new tokens.

It is important to note that while maximizing domain adaptation can significantly boost in-domain performance, it may lead to a substantial drop in out-of-domain translation quality. Hybrid configurations that combine base and domain vocabularies help balance this trade-off, preserving broader translation capabilities while still delivering improved domain performance.

If, in addition to the in-domain data, the original out-of-domain data or its BPE/vocabulary are also accessible, combining these resources can help preserve general language coverage and stabilize adaptation. In all cases, our findings highlight the importance of aligning both BPE and vocabulary with the domain of the adaptation data, while retaining overlap with the base model's vocabulary to ensure generalization.

For future work, we plan to extend our evaluation to other domains and language pairs, and to investigate how these findings generalize to LLMs and multilingual systems. We are also interested in exploring adaptive methods for selecting which tokens to retain or introduce during fine-tuning, with the aim of optimizing both performance and computational efficiency.

All datasets, models, and scripts from our work are publicly available at: `https://github.com/JoyeBright/subword-ft-guide`.

References

Kristine Mae M. Adlaon and Nelson Marcos. 2024. Finding the optimal byte-pair encoding merge operations for neural machine translation in a low-resource setting. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14673–14682, Miami, Florida, USA. Association for Computational Linguistics.

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from turkish to english. *The Prague Bulletin of Mathematical Linguistics*, 108.

Shivang Chopra, Suraj Kothawade, Houda Aynaou, and Aman Chadha. 2023. Transcending domains through text-to-image diffusion: A source-free approach to domain adaptation. *arXiv.org*, abs/2310.01701.

Benoit Courty, Victor Schmidt, Sasha Luccioni, Goyal-Kamal, MarionCoutarel, Boris Feld, Jérémy Lecourt, LiamConnell, Amine Saboni, Inimaz, supatomic, Mathilde Léval, Luis Blanche, Alexis Cruveiller, ouminasara, Franklin Zhao, Aditya Joshi, Alexis Bogroff, Hugues de Lavoreille, Niko Laskaris, Edoardo Abati, Douglas Blank, Ziyao Wang, Armin Catovic, Marc Alencon, Michał Stęchły, Christian Bauer, Lucas Otávio N. de Araújo, JPW, and MinervaBooks. 2024. mlco2/codecarbon: v2.4.1.

Praveen Dakwale and Christof Monz. 2017. Fine-tuning for neural machine translation with limited degradation across in- and out-of-domain data. In *Proceedings of the 16th Machine Translation Summit (MT-Summit 2017)*, pages 156–169.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation.

Dongjun Hwang, Seong Joon Oh, and Junsuk Choe. 2024. Overcoming domain limitations in open-vocabulary segmentation. *x*.

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.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Robert Lim, Kenneth Heafield, Hieu Hoang, Mark Briers, and Allen Malony. 2018. Exploring hyper-parameter optimization for neural machine translation on gpu architectures.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Mahdis Mahdieh, Mia Xu Chen, Yuan Cao, and Orhan Firat. 2020. Rapid domain adaptation for machine translation with monolingual data.

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 2002)*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Shoetsu Sato, Jin Sakuma, Naoki Yoshinaga, Masashi Toyoda, and Masaru Kitsuregawa. 2020. Vocabulary adaptation for domain adaptation in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4269–4279, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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 Technologies*, pages 35–40, San Diego, California. Association for Computational Linguistics.

Javad Pourmostafa Roshan Sharami, Dimitar Shterionov, and Pieter Spronck. 2022. Selecting parallel in-domain sentences for neural machine translation using monolingual texts.

Javad Pourmostafa Roshan Sharami, Dimitar Shterionov, and Pieter Spronck. 2023. A Python tool for selecting domain-specific data in machine translation. In *Proceedings of the 1st Workshop on Open Community-Driven Machine Translation*, pages 29–30, Tampere, Finland. European Association for Machine Translation.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Xu Wang, Chunyang Chen, and Zhenchang Xing. 2019. Domain-specific machine translation with recurrent neural network for software localization. *Empirical Software Engineering*, 24(6):3514–3545.

Chin Man Yeung. 2019. Effects of inserting domain vocabulary and fine-tuning bert for german legal language.

Eduardo Zimelewicz, Marcos Kalinowski, Daniel Méndez, Görkem Giray, Antônio Pedro Santos Alves, Niklas Lavesson, Kelly Azevedo, Hugo Villamizar, Tatiana Escovedo, Hélio Lopes, Stefan Biffl, Juergen Musil, Michael Felderer, Stefan Wagner, María Teresa Baldassarre, and Tony Gorschek. 2024. Ml-enabled systems model deployment and monitoring: Status quo and problems. In *x*, pages 112 – 131. Springer Science+Business Media.