Stolen Subwords: Importance of Vocabularies for Machine Translation Model Stealing

Vilém Zouhar
2023 project report ETH Zurich
vzouhar@inf.ethz.ch

Abstract

In learning-based functionality stealing, the attacker is trying to build a local model based on the victim's outputs. The attacker has to make choices regarding the local model's architecture, optimization method and, specifically for NLP models, subword vocabulary, such as BPE. On the machine translation task, we explore (1) whether the choice of the vocabulary plays a role in model stealing scenarios and (2) if it is possible to extract the victim's vocabulary. We find that the vocabulary itself does not have a large effect on the local model's performance. Given gray-box model access, it is possible to collect the victim's vocabulary by collecting the outputs (detokenized subwords on the output). The results of the minimum effect of vocabulary choice are important more broadly for black-box knowledge distillation.

1 Introduction

NLP models are a key intellectual property, many of which are deployed online. This access creates an attack surface by which an adversarial agent can attempt to replicate the model at the fraction of the cost of the original model training or with the absence of proper training data. The goal of this task, called model functionality stealing, is to create a local copy (student) of a model (victim/teacher). In learning-based approaches, the victim model is queried to create a synthetic dataset, on which the student model is trained. Specifically for machine translation from L_1 to L_2, the attacker has access to a monolingual dataset $\mathcal{D}^{L_1}$, creates a synthetic dataset in L_2 by querining the victim and trains a model on $(\mathcal{D}^{L_1}, M_V(\mathcal{D}^{L_1}))$. This task is similar to model distillation (Hinton et al., 2015; Freitag et al., 2017; Wei et al., 2019; Tan et al., 2019; Gordon and Duh, 2020; Zhou et al., 2021), though the latter usually includes full model access and the goal of making the student model more efficient or privacy-preserving.

In the model stealing scenario, the attacker has to make decisions regarding the student model: data used for training D^{L_1}, model architecture and other preprocessing, including the subword algorithm and its vocabulary. While the effect of data and architecture choice in distillation and model stealing distillation has been explored (Orekondy et al., 2019; Krishna et al., 2019; Wallace et al., 2020; Zouhar, 2021), it is unclear what choice should be made regarding the model vocabulary. Specifically, we focus on BPE (Shibata et al., 1999; Sennrich et al., 2016), a popular subwording algorithm (Ding et al., 2019). We wish to quantify how much the increased student performance (gained by having access to the victim vocabulary) is worth the cost of obtaining such vocabulary. This leads to questions on vocabulary importance (**RQ1**) and vocabulary inference (**RQ2**):

> **RQ1**: Is it advantageous for learning-based model stealing to know the victim's BPE vocabulary or is domain-specificity more important?
> **RQ2**: Is it possible to efficiently recover the victim's vocabulary given black-box and gray-box access?

We describe the model stealing setup in Section 3 and the BPE algorithm in Section 4. In Section 5 we examine the task of learning-based model stealing and compare the effect of vocabularies on the student's performance (**RQ1**). Here we find that training on the victim's vocabulary is marginally worse than using a BPE vocabulary trained on the relevant domain, better than unrelated domains and when trained all data. In Section 6 we describe approaches for recovering the victim's BPE vocabulary based on the level of access and local data (**RQ2**). We discuss the results and conclude in Sections 7 and 8. Note the discussion on Limitations and Ethics in Appendices A and B. For clarity, we show all results in the main paper as averages of

[0] github.com/zouharvi/vocab-stealing

BLEU on English ↔ German language direction but plan to replicate the main findings for other languages and with other evaluation metrics.

2 Related Work

Model stealing has been explored mostly in the domain of computer vision where the output is a classification, possibly with probabilities (Tramèr et al., 2016; Orekondy et al., 2019; Atli et al., 2020; Kariyappa et al., 2021; Szyller et al., 2021; Liu et al., 2022). However, recent works have shown that even in NLP, where the output domain is more complicated, it is possible to infer training data (Carlini et al., 2021) and some weights (Zanella-Beguelin et al., 2021) of large language models. It is also possible to do learning-based model stealing of such models (Krishna et al., 2019; Keskar et al., 2020; Lyu et al., 2021; Xu et al., 2022).

Most work in this area for MT has been framed as knowledge distillation. If KL-divergence is to be used, it needs to be computed over matching distributions which assumes the same subword vocabulary. For MT distillation and immitation attackes, even without token-level KL-divergence optimization, Freitag et al. (2017); Wei et al. (2019); Tan et al. (2019); Gordon and Duh (2020) implicitly use the same BPE vocabulary as the teacher while (Wallace et al., 2020; Zouhar, 2021) choose to train their own BPE without any justification. The distillation work also dealt with the problem of mismatched student and teacher vocabularies (Khanuja et al., 2021; Kolesnikova et al., 2022).

Ding et al. (2019); Gowda and May (2020) examine the effect of BPE vocabulary size and Bogoychev and Chen (2021) experiment with using BPE trained on a different domain and is therefore suboptimal for the primary one.

3 Model Stealing

For most experiments, we consider black-box access (only translated outputs are available) to the victim model M_V. To answer **RQ2** we use additional grey-box access where the output is still segmented to subwords.[1] See Table 6 for an example of black-box and gray-box outputs.

While important in most adversarial scenarios, we assume a large but fixed query budget for the model of 10M sentence queries, in order to translate the respective datasets. For the victim model, we use the winning WMT19 English↔German model (Ng et al., 2019) as M_V. We translate all the available authentic data $\mathcal{D}_A$ using the victim model: $\mathcal{D}_V^{L_1} = M_V(\mathcal{D}_A^{L_2})$. We then train several student models on a combination of this data with various BPE models.

Student model description. The student model is Transformer-based on the Fairseq configuration `transformer_iwslt_de_en`. Following the victim model (Ng et al., 2019), we use a joint BPE vocabulary of 30k entries for the student model in order to reduce the number of configurations.

Data. We use 10M parallel sentences from the following datasets: ParaCrawl (Esplà-Gomis et al., 2019; Bañón et al., 2020), EuroPat (Heafield et al., 2022) and CommonCrawlAligned (El-Kishky et al., 2020) to which we refer to as PCrawl, EuroPat and CCrawl, respectively. They were chosen to represent different domains: general, legal publications and patents. For each of these datasets, we use 9.8M, 100k and 100k for the training, development and test splits. See Tables 5 and 8 for dataset overview and example sentences and translations.

4 Byte-Pair-Encoding

BPE is a method to reduce the output dimensionality by splitting target words into smaller units (subwords). This way, the MT system does not have to model the word *moonlight*, which occurs in our data 272 times, independently but rather model *moon*@@ and *light* which occur 7076 and 587321 times, respectively. See Table 6 for an example of subword units.

Vocabulary efficiency. BPE is a compression algorithm and as such, its goal is to encode the data in as little space as possible. Because BPE encodes the data as a sequence of subwords, we consider the number of subwords needed to encode a given dataset. With the same fixed vocabulary size (30k), we train multiple BPE models B_i on datasets $\mathcal{D}_i$ and define the efficiency of this model on dataset $\mathcal{D}_j$ as $\frac{|B_i(\mathcal{D}_j)|}{|B_j(\mathcal{D}_j)|}$. This is the number of subwords needed to encode $\mathcal{D}_j$ with BPE model B_i divided by the space requirements of the most efficient model with the same vocabulary size constraints (B_j).

We show the BPE model efficiencies on the three datasets in Table 1. The efficiencies confirm that BPE models trained on different datasets, even on the same language, are suboptimal on different do-

[1] Available e.g. by GPT-3 (Brown et al., 2020).

mains. It also shows that PCrawl and CCrawl contain similar distribution (scrapped from the web).

		Target dataset		
		PCrawl	CCrawl	EuroPat
Training	Victim	1.08	1.10	1.30
	All	1.04	1.04	1.05
	PCrawl	1.00 ⋆	1.03	1.26
	CCrawl	1.04	1.00 ⋆	1.28
	Patents	1.29	1.31	1.00 ⋆

Table 1: Efficiency of BPE encoding on various datasets in terms of the required number of subwords (lower is better). The ratio is computed against optimal BPE on that dataset (minimum in column, marked with star). *Victim* was trained on a larger version of PCrawl.

5 Model Stealing Results

In learning-based model stealing, we first translate the available data by the victim's model and then train a student model on this data. In Table 2 we show that training on authentic parallel data (Auth.+Auth.) is the most optimal but training on the victim's outputs (Auth.+Victim) is not far behind (-0.07).

Source+Target	PCrawl	CCrawl	EuroPat	Avg.
Auth.+Auth.	35.07	36.42	35.13	35.54
Auth.+Victim	34.95	36.25	35.20	35.47

Table 2: Within domain BLEU performance of student models trained on either authentic or victim's data (source and/or target). The used BPE is trained on the same data as the student model.

BPE	PCrawl	CCrawl	EuroPat	Avg.
Victim	34.73	36.38	34.16	35.09
All	34.91	36.05	34.36	35.11
PCrawl	34.95	35.74	34.23	34.97
CCrawl	34.86	36.25	34.19	35.10
EuroPat	34.54	35.82	34.41	34.92

Table 3: Within domain student performance (BLEU) trained on Authentic + Victim data. The student models are trained with different BPE vocabularies (first column).

In Table 3 we show that training with the victim's vocabulary is marginally worse than training on the target domain (-0.11), beter on unrelated domains ($+0.23$) and similar when trained on all data ($+0.01$). The BPE optimality provides some explanation for student performance trained with the specific vocabulary. The higher the BPE inefficiency (further from optimal 1.00, Table 1), the lower the student BLEU. This is verified with negative correlations (Pearson: -58.27%, p = 0.023, Spearman: -57.20%, p = 0.026).[2] For model stealing and knowledge distillation applications it, therefore, plays only a trivial role to match the vocabulary of the local student model to that of the victim's (**RQ1**).

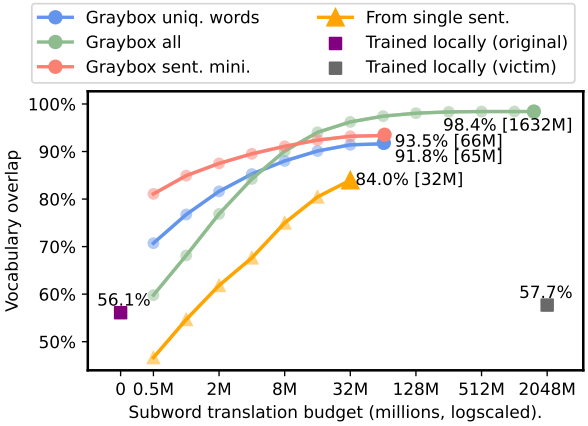

Figure 1: Overlap with victim's vocabulary. Numbers in square brackets indicate the subword translation budget. Points ■ are a BPEs trained locally, ● are vocabularies collected from model output and ▲ are vocabularies collected from model output starting from a single sentence (max of 5).

6 Recovering Victim's BPE Vocabulary

We operationalize the efficiency of recovering the victim's BPE vocabulary V against a hypothesis vocabulary V' as overlap $\frac{2\cdot|V\cap V'|}{|V|+|V'|}$ which ranges from 0 to 100%. We consider multiple baseline approaches and show their performance in Figure 1. The *Overlap* is not the only important variable but also *Subword budget* (how many subwords does the victim need to translate) and whether we assume black-box or gray-box access.

Training locally. First, we can train our own BPE model on the local authentic data. This does not require querying of the victim model at all but is sensitive to the original data selection and yields only 56.1% overlap with the victim's vocabulary. We can also train a BPE model on the victims' output. This requires the victim to translate the whole data, making it the most expensive. It is only marginally better (57.7%) than training on authentic data. This can be explained by the fact that

[2]Computed as correlation between mean-normalized BLEU scores for each domain and its BPE efficiency. Using scipy 1.9.3 (Virtanen et al., 2020).

the victim's BPE was trained on authentic and not synthetic data. The difference in data distributions between the authentic and the translated data is shown by the number of unique tokens (13.7M and 4.5M, respectively).

Gray-box access. Assuming that the model produces subword outputs (see Table 6), we can simply collect all the subwords that appear in the victim output. BPE is constructed by merging the most co-occurring pairs which leads to the selected subwords that are added to the vocabulary to be frequent in the training corpora. The overlap of 98.4% with the victim's vocabulary is therefore not surprising. However, this approach is severely inefficient. Consider the two English sentences and their translations into German in Table 4. We are spending the budget on repeatedly translating the same words which yield the same subwords on the output. Because we only care about the set of subwords in the output, we can simply translate all unique words in our dataset individually with a much smaller budget (65M). This results in a vocabulary that has 92.1% overlap. The reason for this degradation is that words are translated differently and hence into different subwords.[3] A compromise between translating whole sentences and unique words is to translate sentences but remove already-seen words. This shows promising results, especially within the low-budget area. A more elaborate approach with negative results is described in Appendix D. All the observation-based gray-box approaches start to plateau from some point on because those subwords become rarer. In Appendix C we analyze which subwords are missing.

The washing machine is broken. Die Wasch@@ maschine ist kap@@ utt@@ .
I broke the milling machine. Ich habe die F@@ rä@@ s@@ maschine kap@@ utt ge@@ macht@@ .

Table 4: Example gray-box translation with overlap.

From single sentence. Finally, we consider a scenario in which the attacker has not even monolingual data apart from a handful of sentences. We use these sentences as starting seeds to a vocabulary and then create "nonsense" sequences of words sampled from this vocabulary. We then translate this sequence, add resulting words into the opposing language's vocabulary, sample a "nonsense" sequence from that vocabulary and repeat. To encourage sampling of less-sampled words, we sample a word with the weight of $\frac{1}{\#\text{ sampled}}$. We repeat this process until no changes to either the source or target vocabularies are made (patience of 5 iterations). The algorithm is formally described in Listing 1. While the results in Figure 1 show that this method performs worse in recovering the victim's vocabulary, it should be appreciated in the context of starting with a single sentence. The 5 starting sentences are shown in Table 7 and Figure 2 shows that regardles of the starting seed, all runs converge to a similar vocabulary.

Take-away. The difficulty of inferring the victim's BPE vocabulary greatly depends on the level of access If only black-box access is available, then training a BPE on the available authentic data is the best option. If, however, the victim model produces subword outputs, simply collecting the output subwords makes it possible to construct a local copy of the vocabulary. Contrary to intuition, it is more effective to query the model on authentic (possibly minified) sentences and not on single deduplicated words. In the case where no authentic data is available, it is still possible to ineffectively infer a part of the vocabulary from a single seed sentence and "nonsense" sequence translations.

7 Discussion

The results clearly show that the choice of the BPE vocabulary is largely inconsequential as long as (1) it is that of the victim or (2) it matches the student's domain. Therefore the choice to either use the same vocabulary (Freitag et al., 2017; Wei et al., 2019; Tan et al., 2019; Gordon and Duh, 2020) or train a local one (Wallace et al., 2020; Zouhar, 2021) is justified.

8 Conclusion

In this paper, we explored the setting of learning-based model stealing and focused on the issue of BPE subword vocabularies. We find that it plays a very minor role whether the student shares the same vocabulary as the victim. We also document several approaches for inferring the victim's vocabulary based on its outputs. For both low and high translation budgets (0.5M and 128M) it is possible to infer the victim's vocabulary with the overlap of 93.5% and 98.1%, respectively.

[3] E.g. *I have* → *Ich habe* and *They have* → *Sie haben*.

A Limitations

We explored a specific NLP scenario (machine translation) and subwording method (BPE). While we believe that the exploreation of other subwording methods, such as Byte-level BPE (Wang et al., 2020), would yield similar results, more tokenization-sensitive tasks could show higher dependency on victims's vocabulary.

B Ethics statement

In our experiments we worked with publicly released models which we treated as belonging to the victim. Our research does not advocate for unlawful stealing of intellectual propety and does not facilitate it or provide any guidance. The exception is the vocabulary extraction, which we do not believe to be a key component in the intellectual propety.

References

Buse Gul Atli, Sebastian Szyller, Mika Juuti, Samuel Marchal, and N Asokan. 2020. Extraction of complex dnn models: Real threat or boogeyman? In *International Workshop on Engineering Dependable and Secure Machine Learning Systems*, pages 42–57. Springer.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, et al. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567.

Nikolay Bogoychev and Pinzhen Chen. 2021. The highs and lows of simple lexical domain adaptation approaches for neural machine translation. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 74–80.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Shuoyang Ding, Adithya Renduchintala, and Kevin Duh. 2019. A call for prudent choice of subword merge operations in neural machine translation. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 204–213.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 5960–5969, Online. Association for Computational Linguistics.

Miquel Esplà-Gomis, Mikel L Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII: Translator, Project and User Tracks*, pages 118–119.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *ArXiv*, abs/1702.01802.

Mitchell A. Gordon and Kevin Duh. 2020. Distill, adapt, distill: Training small, in-domain models for neural machine translation. In *NGT*.

Thamme Gowda and Jonathan May. 2020. Finding the optimal vocabulary size for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3955–3964.

Kenneth Heafield, Elaine Farrow, Jelmer Van Der Linde, Gema Ramírez-Sánchez, and Dion Wiggins. 2022. The EuroPat corpus: A parallel corpus of european patent data. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 732–740.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

Sanjay Kariyappa, Atul Prakash, and Moinuddin K Qureshi. 2021. Maze: Data-free model stealing attack using zeroth-order gradient estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13814–13823.

Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. The thieves on sesame street are polyglots-extracting multilingual models from monolingual APIs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6203–6207.

Simran Khanuja, Melvin Johnson, and Partha Talukdar. 2021. Mergedistill: Merging language models using pre-trained distillation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2874–2887.

Alina Kolesnikova, Yuri Kuratov, Vasily Konovalov, and Mikhail Burtsev. 2022. Knowledge distillation of russian language models with reduction of vocabulary. *arXiv preprint arXiv:2205.02340*.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. 2019. Thieves on sesame street! model extraction of bert-based apis. *arXiv preprint arXiv:1910.12366*.

Yupei Liu, Jinyuan Jia, Hongbin Liu, and Neil Zhenqiang Gong. 2022. StolenEncoder: Stealing pre-trained encoders in self-supervised learning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2115–2128.

Lingjuan Lyu, Xuanli He, Fangzhao Wu, and Lichao Sun. 2021. Killing two birds with one stone: Stealing model and inferring attribute from bert-based apis. *arXiv preprint arXiv:2105.10909*.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4954–4963.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. *ArXiv*, abs/1508.07909.

Yusuxke Shibata, Takuya Kida, Shuichi Fukamachi, Masayuki Takeda, Ayumi Shinohara, Takeshi Shinohara, and Setsuo Arikawa. 1999. Byte pair encoding: A text compression scheme that accelerates pattern matching.

Sebastian Szyller, Vasisht Duddu, Tommi Gröndahl, and N Asokan. 2021. Good artists copy, great artists steal: Model extraction attacks against image translation generative adversarial networks. *arXiv preprint arXiv:2104.12623*.

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. *ArXiv*, abs/1902.10461.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pages 601–618.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272.

Eric Wallace, Mitchell Stern, and Dawn Song. 2020. Imitation attacks and defenses for black-box machine translation systems. *arXiv preprint arXiv:2004.15015*.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020. Neural machine translation with byte-level subwords. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 05, pages 9154–9160.

Hao-Ran Wei, Shujian Huang, Ran Wang, Xinyu Dai, and Jiajun Chen. 2019. Online distilling from checkpoints for neural machine translation. In *NAACL*.

Qiongkai Xu, Xuanli He, Lingjuan Lyu, Lizhen Qu, and Gholamreza Haffari. 2022. Student surpasses teacher: Imitation attack for black-box NLP APIs. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2849–2860.

Santiago Zanella-Beguelin, Shruti Tople, Andrew Paverd, and Boris Köpf. 2021. Grey-box extraction of natural language models. In *International Conference on Machine Learning*, pages 12278–12286. PMLR.

Sheng Zhou, Yucheng Wang, Defang Chen, Jiawei Chen, Xin Wang, Can Wang, and Jiajun Bu. 2021. Distilling holistic knowledge with graph neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10387–10396.

Vilém Zouhar. 2021. Sampling and filtering of neural machine translation distillation data. *NAACL-HLT 2021*.

C Error Analysis of Missing Subwords

For the gray-box collection of subwords in translated sentences with 98.4% overlap, there were 139 subwords missing. The longest were obivous artefacts of the used dataset (such as *www.nachrichten.at*). However, for some subwords, such as *Bundesligist*, the model had similar subwords: *Bundeslig@@* (which would be combined with *ist*), *Bundesliga* and *Bundesligisten*. Despite the word *Shakespeare* appearing in the query data, on the output it was always presented as a single word (because of its frequency) and therefore the subword *Shakes* was not found. This also provides an explanation for the failure of the approach in Appendix D.

D Unique Word Minimization

Despite taking only unique words, we are still duplicating work by translating both e.g. by translating both *understand* → *verstehen* and *misunderstand* → *miss@@ verstehen*. We can therefore further minimize the budget by not querying words which

are subwords of ones that we query later. This is not an optimal approach because the composition of the larger word may be frequent enough to warrant its own subword, but still, this approach reaches 91.8% overlap with 53M subword budget. It is only marginally better than only taking unique words (91.8% with 65M, Figure 1).

```
def translate_random(model, vocab):
    sent_src ← sample(vocab, k=20)
    sent_tgt ← model(sent_src)
    return {w | w ∈ sent_tgt}

SENT_0 ← "Alice was beginning to..."

vocab_en ← {w | w ∈ SENT_0}
vocab_de ← {}

loop:
    vocab_de_ext ← translate_random(
        model_ende, vocab_en
    )
    if vocab_de_ext ⊆ vocab_de:
        exit
    vocab_de ← vocab_de ∪ vocab_de_ext

    vocab_en_ext ← translate_random(
        model_deen, vocab_de
    )
    if vocab_en_ext ⊆ vocab_en:
        exit
    vocab_en ← vocab_en ∪ vocab_en_ext
```

Listing 1: Cyclic backtranslation to steal victim's vocabulary from a single sentence.

Property	PCrawl	EuroPat	CCrawl
Line length (tokens)	17	30	11
Line length (chars)	97	183	61
Unique tokens	4.6M	4.1M	3.2M
English	2.2M	0.9M	1.7M
German	3.7M	3.5M	2.7M
Unique chars	6721	520	6897
English	6097	335	6062
German	4817	456	6335

Table 5: Overview of words and characters found in the three used datasets (10M sentences each). Per-language line length distribution is not shown as it is similar. Tokens are compared case-insensitive. The datasets differ not only in the diversity of used words but also symbols (characters).

Input:	Stolen Subwords: Importance of Vocabularies for Machine Translation Model Stealing
Black-box:	Gestohlene Subwörter: Bedeutung von Vokabeln für den Diebstahl maschineller Übersetzungsmodelle
Gray-box:	Gest@@ ohl@@ ene Sub@@ wör@@ ter@@ : Bedeutung von V@@ ok@@ ab@@ eln für den Diebstahl masch@@ in@@ eller Über@@ setz @@ ungs@@ modelle

Table 6: Examples of black-box vs gray-box outputs from English to German translation.

- Stolen subwords: importance of vocabularies for machine translation model stealing
- NLP models are a key intelectual property, many of which are deployed online.
- We present the Eyetracked Multi-Modal Translation (EMMT) corpus, a dataset containing monocular eye movement recordings, audio and 4-electrode electroencephalogram (EEG) data of 43 participants.
- Two roads diverged in a wood, and I- I took the one less traveled by, And that has made all the difference.
- One morning, when Gregor Samsa woke from troubled dreams, he found himself transformed in his bed into a horrible vermin.

Table 7: Starting sentences to seed the vocabulary in Listing 1.

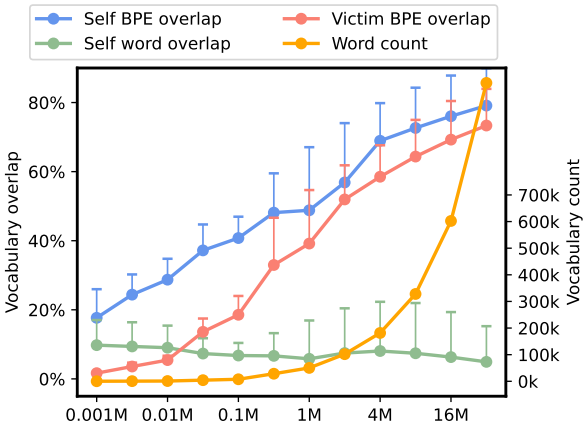

Figure 2: Overlaps between vocabularies of different runs of the cyclic translations from single sentence algorithm (Listing 1) and vocabulary sizes. Points show the average of 5 seeds and bars the maximum.

Dataset	Source	English	German
PCrawl ParaCrawl	Original	Airbnb®	Devala - Holiday Rentals & Places to Stay - Tamil Nadu, India	Airbnb®	Kuchanur – Ferienwohnungen & Unterkünfte - Tamil Nadu, Indien
	Translated	Airbnb	Kuchanur Apartments & Accommodations - Tamil Nadu, India	Airbnb ®	Devala - Ferienwohnungen und Unterkünfte - Tamil Nadu, Indien
EuroPat	Original	With this positioning, residual water can flow out of the chambers of the differential-pressure fluid gauge chamber due to the effect of gravity	Bei dieser Positionierung kann Restwasser aus den Kammern der Differenzdruckdose unter Schwerkraftwirkung abfließen.
	Translated	During this positioning, residual water can drain out of the chambers of the differential pressure box under gravity.	Bei dieser Positionierung kann aufgrund der Schwerkraft Restwasser aus den Kammern der Differenzdruck-Fluidmanometer-Kammer fließen.
CCrawl CCAligned	Original	Those who do good, good will be done to them!	Wer Gutes tut, dem wird Gutes widerfahren!
	Translated	He who does good will do good!	Diejenigen, die Gutes tun, werden Gutes tun!

Table 8: Example sentences from the three used English-German datasets with different domains. Rows marked with *Translated* contain the victim's output.