INFORMATION-THEORETIC VOCABULARIZATION VIA OPTIMAL TRANSPORT FOR MACHINE TRANSLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

It is well accepted that the choice of token vocabulary largely affects the performance of machine translation. One dominant approach to construct a good vocabulary is the Byte Pair Encoding method (BPE). However, due to expensive trial costs, most previous studies only conduct simple trials with commonly used vocabulary sizes. This paper finds an exciting relation between an information-theoretic feature and BLEU scores with a given vocabulary. With this observation, we formulate the quest of vocabularization – finding the best token dictionary with a proper size – as an optimal transport problem. We then propose ***Info*-VOT**, a simple and efficient solution without the full and costly trial training. We evaluate our approach on multiple machine translation tasks, including WMT-14 English-German translation, TED bilingual translation, and TED multilingual translation. Empirical results show that *Info*-VOT can generate well-performing vocabularies on diverse scenarios. Also, one advantage of the proposed approach lies in its low consumption of computation resources. On TED bilingual translation, *Info*-VOT only spends a few CPU hours generating vocabularies, while the traditional BPE-Search solution takes hundreds of GPU hours.

1 INTRODUCTION

Due to the discreteness of text, it has been a standard practice for natural language processing (NLP) tasks (Mikolov et al., 2013; Vaswani et al., 2017; Gehrmann et al., 2018; Zhang et al., 2018; Devlin et al., 2019) to embed the sequence of input tokens using a vocabulary-based lookup table, with each row representing a token as a dense vector. As a necessary prerequisite, vocabulary construction bridges the gap between discrete symbols and continuous representations. Currently, the widely-used vocabularies (e.g., subword vocabularies) are mainly constructed by heuristic approaches (Sennrich et al., 2016; Costa-jussà & Fonollosa, 2016; Lee et al., 2017; Kudo & Richardson, 2018; Al-Rfou et al., 2019; Wang et al., 2020). In this work, we focus on machine translation to explore better vocabularization solutions.

Despite the promising performance, most of the current approaches inevitably require a large number of human efforts to adjust the granularity of segmentation units. While many previous studies (Sennrich & Zhang, 2019; Ding et al., 2019; Provilkov et al., 2020; Salesky et al., 2020) show that vocabulary size greatly impacts BLEU scores, especially on low-resource scenarios, very few existing studies carefully tune this hyper-parameter due to expensive computation resource costs. To address this problem, researchers recently pay increasing attention to automatic vocabulary search. However, to the best of our knowledge, there is still no literature that provides a unified theory to explore what quantitative features impact vocabularies' quality and how to formulate vocabularization as a learning problem such that the optimal vocabulary can be automatically learned by leveraging current machine learning techniques instead of heuristically defined.

We take the first step to a unified theory for automatic vocabulary learning (**AVL**). To be specific, we present a new perspective of information theory to quantitatively describe vocabularies and then formulate vocabularization into a discrete optimization problem, followed by a mathematically derived solution (*Info*-VOT). We start from the entropy of the token distribution. From the view of information theory, entropy represents the average level of information inherent in the tokens, or how many bits we need to define these tokens, also called Bits Per Character (BPC). We have an experimental finding that an information-theoretical feature BPC-AMD, short for AMD, correlates with BLEU scores. Formally, AMD, first introduced in this work, is defined as the amortized marginal difference over BPC (See Eq.2 for more explanations). Although it is hard to explain this connection based on our current knowledge, this feature can still be empirically used to search for the optimal vocabulary or guide vocabulary learning. Here we focus more on the learning setting to explore whether there is an efficient and promising AVL solution.

Motivated by our findings, we propose a novel two-step discrete optimization objective and an optimal transport solution that can efficiently find well-performing vocabularies. Our target is to find the optimal vocabulary with the highest AMD. To model AMD, a contrastive feature, here we define an incremental integer sequence $\mathbf{S}$ where each timestep t corresponds to a set of vocabularies $\mathbb{V}_{\mathbf{S}[t]}$ with each vocabulary containing up to $\mathbf{S}[t]$ items. Given $\mathbf{S}$, the optimization contains two steps. First, for each timestep t, we search for the optimal vocabulary $v(t) \in \mathbb{V}_{\mathbf{S}[t]}$ with the highest AMD based on the marginal difference between BPC values of $v(t) \in \mathbb{V}_{\mathbf{S}[t]}$ and $v(t-1) \in \mathbb{V}_{\mathbf{S}[t-1]}$. Second, we enumerate the optimal vocabularies from all timesteps and select the vocabulary with the highest AMD as the final vocabulary.

For simplification, we propose to optimize the lower bound of the objective in the first step. In the new objective function, AMD is based on the marginal difference between the maximum BPC of $\mathbb{V}_{\mathbf{S}[t]}$ and the maximum BPC of $\mathbb{V}_{\mathbf{S}[t-1]}$. Thus, the first objective is simplified into a problem finding a token set with the highest BPC scores in each timestep. Due to the exponential search space, we re-formulate this part into an optimal transport (OT) problem, which, therefore, can be solved in polynomial time by linear programming. To be specific, we can imagine vocabulary construction as a transport process that transports chars into token candidates. The number of chars is fixed, and different transport choices result in vocabularies with different costs. The target of OT is to find a transport matrix to minimize the transfer cost, i.e., negative BPC in our setting. We implement an entropy-based Sinkhorn algorithm to solve the OT problem.

We evaluate our approaches on multiple machine translation tasks, including WMT-14 English-German translation, TED bilingual translation, and TED multilingual translation. Empirical results show that *Info*-VOT can find well-performing vocabularies on diverse scenarios. Furthermore, *Info*-VOT is a lightweight solution and does not require expensive computation resources. On TED bilingual translation, *Info*-VOT only takes a few CPU hours to find vocabularies while the traditional BPE-Search solution takes hundreds of GPU hours.

2 RELATED WORK

With the development of deep learning, neural networks have achieved state-of-the-art results on natural language processing tasks. Initially, most neural models are built upon word-level vocabularies (Costa-jussà & Fonollosa, 2016; Vaswani et al., 2017; Zhao et al., 2019). While achieving state-of-the-art results, it is a common constraint that these word-level vocabularies fail on handling rare words under limited vocabulary size.

Researchers have proposed several advanced vocabularization approaches, like byte-level approaches (Wang et al., 2020), character-level approaches (Costa-jussà & Fonollosa, 2016; Lee et al., 2017; Al-Rfou et al., 2019), and subword-level approaches (Sennrich et al., 2016; Kudo & Richardson, 2018), to address this problem. Costa-jussà & Fonollosa (2016) propose a character-level vocabulary that adopts single characters as the minimum semantic unit. The surprisingly good performance brings new insights into token granularity. Byte-Pair Encoding (BPE) (Sennrich et al., 2016) is proposed to get subword-level vocabularies. The general idea is to merge pairs of frequent character sequences to create subword units. Subword-level vocabularies can be regarded as a trade-off between character-level vocabularies and word-level vocabularies. Compared to word-level vocabularies, it can decrease the sparsity of tokens and increase the shared features between similar words, which probably have similar semantic meanings, like “happy” and “happier”. Compared to character-level vocabularies, it has shorter sentence lengths without rare words. Following BPE, some variants recently have been proposed, like BPE-dropout (Provilkov et al., 2020), SentencePiece (Kudo & Richardson, 2018), and so on.

Despite promising results, these subword-level approaches still require expensive computation costs to tune vocabulary size. More recently, some best-practice studies notice this problem and propose some practical solutions (Kreutzer & Sokolov, 2018; Cherry et al., 2018; Chen et al., 2019; Salesky et al., 2020).

Unlike these approaches, this work takes the first step to a unified theory for automatic vocabulary learning. We propose a discrete optimization objective function and a principled solution based on optimal transport for AVL.

3 INFORMATION-THEORETIC PERSPECTIVE OF VOCABULARY

More and more researchers have recently accepted that information theory and machine learning are the two sides of the same coin, first mentioned by MacKay (2003). Information theory studies the shortest code-length and uncertainty, while machine learning studies how to compress data with a low-dimension vector. Also, learning can be regarded as a process of reducing uncertainty. Following this view, many studies are proposed to understand current machine learning systems from the perspective of information theory (Saxe et al., 2018; Gabrié et al., 2018; Goldfeld et al., 2019).

In this section, we describe vocabularization from the perspective of information theory. Considering that BPE is the dominant solution, this section mainly focuses on BPE to explore whether there are essential features strongly correlated to BLEU scores.

From Frequency to BPC Almost all existing vocabularies are built upon an information-theoretic concept: frequency. In information theory, frequency is a token-level feature, which describes the information of a single token. To get a full understanding of vocabulary, here we explore several vocabulary-level features, such as entropy, BPC. Entropy is a common feature evaluating the average level of “information”, or “uncertainty” inherent in the distribution. It represents the shortest code length to represent all tokens, short for Bits Per Token (BPT). One of the variants of BPT is Bits-Per-Char (BPC), which normalizes BPT with the averaged length of tokens. We argue that BPC is a more fair evaluation feature than BPT, which avoids the effects of token lengths. Given a vocabulary $v(T)$ with size T, BPC is computed as:

$$\boldsymbol{B}_{v(T)} = -\frac{1}{l_{v(T)}} \sum_{t \in v(T)} P(t) \log P(t), \tag{1}$$

where $P(t)$ is the probability of token t and $l_{v(T)}$ is the average length of tokens in vocabulary v_T. Here we study the relation between BPC and its variants with downstream results.

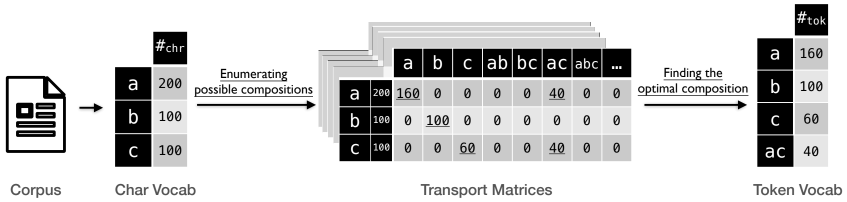
From BPC to AMD Empirical results demonstrate that BPC-AMD, short for AMD, which is first introduced in this work, is strongly related to BLEU scores. Formally, AMD is defined as the amortized marginal difference over BPC, which is normalized by the size of vocabulary:

$$\boldsymbol{D}(v(k+m)) = -\frac{\boldsymbol{B}(v(k+m)) - \boldsymbol{B}(v(k))}{m}, \tag{2}$$

where $\boldsymbol{D}(v(k+m))$ represents the AMD score for vocabulary $v(k+m)$. m represents the increased vocabulary size and v_k, v_{k+m} are vocabularies generated by k, $k+m$ merge operations, respectively. Imagine a vocabulary search policy that incrementally increases the number of merge operations. AMD is a dynamic feature that describes how information changes.

New Finding: AMD Correlates with BLEU scores. To evaluate the relationship between AMD and BLEU scores, we conduct experiments on 45 language pairs from TED and calculate their Spearman correlation scores. The full results are shown in Appendix A. The median correlation score is 0.49. In general, [0.8, 1] means very strong correlations, [0.6, 0.8] means strong correlations, [0.4, 0.6] means moderate correlations, [0.2, 0.4] means weak correlations. Almost two-thirds of pairs show obvious positive correlations (greater than 0.4)[1]. Considering that other factors (e.g. model size, corpus size) also affect BLEU scores, we believe that it is good evidence to support the finding. Experiment settings can be found in Section 5. Please refer to Appendix A for more implementation details.

[1] https://www.statstutor.ac.uk/resources/uploaded/spearmans.pdf


Corpus
Char Vocab
Enumerating possible compositions
Transport Matrices
Finding the optimal composition
Token Vocab

	#chr
a	200
b	100
c	100

		a	b	c	ab	bc	ac	abc	...
a	200	160	0	0	0	0	40	0	0
b	100	0	100	0	0	0	0	0	0
c	100	0	0	60	0	0	40	0	0

	#tok
a	160
b	100
c	60
ac	40

Figure 1: An illustration of vocabulary construction from a transport view. Given a corpus, we can calculate a char distribution and a token candidate distribution. A vocabulary can be built upon a transport matrix deciding how much chars are transported to different tokens.

Based on this finding, we have two natural choices to get the final vocabulary: search and learning. In the search-based direction, the optimal vocabularies can be obtained by enumerating all candidate vocabularies. While being simple, the main limitation lies in the vast search space. Assuming we have N token candidates, the target of search-based approaches is to find the optimal vocabulary from 2^N subsets of tokens. In a real-world scenario with limited resources, we need a more efficient examination solution. In this work, we take the first step to the learning-based direction to explore “how far an information-theoretic learning approach can reach”.

4 Our Proposed Approach: *Info*-VOT

This section describes the details of the proposed approach. We first show the general idea of *Info*-VOT in Section 4.1, then describe the optimal transport solution in Section 4.2, followed by the implementation details in Section 4.3.

4.1 Overview

We formulate vocabulary construction as an optimization problem whose target is to find the vocabulary with the highest AMD based on Eq. 2. Since AMD is a dynamic feature depending on a marginal difference, we also formulate a dynamic process here. Given an incremental integer sequence $\boldsymbol{S} = \{k, k+i, k+2\cdot i, ..., k+(t-1)\cdot i, \cdots\}$ where $k+(t-1)\cdot i$ is the upper bound of vocabulary size at t-th timestep and k represents the number of characters. With sequence $\boldsymbol{S}$, the target to find the optimal vocabulary $v(t)$ with the highest AMD can be formulated as:

$$\underset{v(t-1)\in\mathbb{V}_{\boldsymbol{S}[t-1]}, v(t)\in\mathbb{V}_{\boldsymbol{S}[t]}}{\arg\max} \boldsymbol{D}(v(t)) = \underset{v(t-1)\in\mathbb{V}_{\boldsymbol{S}[t-1]}, v(t)\in\mathbb{V}_{\boldsymbol{S}[t]}}{\arg\max} -\frac{1}{i}\big[\boldsymbol{B}(v(t)) - \boldsymbol{B}(v(t-1))\big] \tag{3}$$

where $\mathbb{V}_{\boldsymbol{S}[t-1]}$ and $\mathbb{V}_{\boldsymbol{S}[t]}$ are two sets containing all vocabularies with upper bound of size $\boldsymbol{S}[t-1]$ and $\boldsymbol{S}[t]$. For simplification, we propose to optimize the surrogated loss which is the lower bound of Eq. 3:

$$\underset{t}{\arg\max} -\frac{1}{i}\big[\max_{v(t)\in\mathbb{V}_{\boldsymbol{S}[t]}} \boldsymbol{B}(v(t)) \; - \max_{v(t-1)\in\mathbb{V}_{\boldsymbol{S}[t-1]}} \boldsymbol{B}(v(t-1))\big] \tag{4}$$

Based on this equation, the whole solution is split into two steps: 1) search for the optimal vocabulary with the highest BPC at each timestep t; 2) enumerate all timesteps and output the vocabulary corresponding to the time step satisfying Eq. 4. Section 4.2 shows the details of the optimal transport solution in the first step. and Section 4.3 shows the implementation details of *Info*-VOT.

4.2 Maximization of BPC via Optimal Transport

The first step of our approach is to search for the vocabulary with the highest BPC from $\mathbb{V}_{\boldsymbol{S}[t]}$. Formally, the goal is to find a vocabulary $v(t)$ such that BPC is maximized,

$$\underset{v(t)\in\mathbb{V}_{\boldsymbol{S}[t]}}{\arg\max} -\frac{1}{l_{v(t)}} \sum_{x\in v(t)} P(x)\log P(x), \tag{5}$$

where l_v is the average length for tokens in $v(t)$, $P(x)$ is the probability of token x. However, notice that this problem is in general intractable due to the extensive vocabulary size. Therefore, we instead propose a relaxation in the formulation of discrete Optimal Transport, which can then be solved efficiently via the well-known Sinkhorn algorithm.

To be specific, vocabulary construction can be viewed as a transport process that transfers char distributions into token distributions. Given two sets of chars and tokens, we can define a transport matrix with each item (i, j) deciding how many chars are transported from char i to token j. Since the number of chars is limited, and not all token candidates can get enough chars, different transport metrics result in vocabularies with different costs. Figure 1 illustrates an example to understand this process. The objective function is to find the transport matrix with the lowest costs.

4.2.1 Background: Optimal Transport

More precisely, given a cost matrix $\boldsymbol{C} \in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{T}|}$ and two discrete distributions: char distribution $\mathcal{C}$ and token distribution $\mathcal{T}$, $\{a_i\}_{i=1}^{|\mathcal{C}|}, \{b_j\}_{j=1}^{|\mathcal{T}|}$ are the corresponding probability mass. The discrete OT considers the following optimization problem

$$\min_{\boldsymbol{A}\in\mathbb{R}^{m\times n}} \langle \boldsymbol{A}, \boldsymbol{C}\rangle\,, \quad \text{s.t.} \quad \boldsymbol{A}\cdot\mathbf{1}_n = \vec{a}, \quad \boldsymbol{A}^{\mathsf{T}}\cdot\mathbf{1}_m = \vec{b}, \tag{6}$$

where $\boldsymbol{A}$ is the transport matrix. Intuitively, optimal transport is about finding the best plan of transporting mass from the source distribution $\mathcal{C}$ to the target distribution $\mathcal{T}$ with the minimum work defined by $\langle \boldsymbol{A}, \boldsymbol{C}\rangle$. Since the original OT problem is a linear programming which requires $O(N^3 \log N)$ time complexity to solve, Cuturi (2013) proposed to add an entropy regularization term to accelerate the convergence (P163). The objective function for entropy regularized OT is

$$\min_{\boldsymbol{A}\in\mathbb{R}^{m\times n}} \langle \boldsymbol{A}, \boldsymbol{C}\rangle - \gamma H(\boldsymbol{A}), \quad \text{s.t.} \quad \boldsymbol{A}\cdot\mathbf{1}_n = \vec{a}, \quad \boldsymbol{A}^{\mathsf{T}}\cdot\mathbf{1}_m = \vec{b}, \tag{7}$$

where $H(\boldsymbol{A}) = -\sum A_{ij}\log A_{ij}$ is the entropy term. The entropy regularization makes the problem convex. Moreover, there is an efficient algorithm, the Sinkhorn algorithm, that allows us to solve the problem in nearly linear time.

4.2.2 Solution

A tractable lower bound of BPC Given a set of vocabularies $\mathbb{V}_{\boldsymbol{S}[t]}$, we want to find a vocabulary with the highest BPC. Consequently, the objective function in Eq. 5 becomes

$$\min_{v\in\mathbb{V}_{\boldsymbol{S}[t]}} \frac{1}{l_v}\sum_{i\in v} P(i)\log P(i),$$
$$\text{s.t.} \quad P(i) = \frac{\text{Token}(i)}{\sum_{i\in v}\text{Token}(i)},\ l_v = \frac{\sum_{i\in v} len(i)}{|v|}$$

where $\text{Token}(i)$ is the frequency of token i in the vocabulary v. Notice that both the distribution $P(i)$ and the average length l_v depend on the choice of $v \in \mathbb{V}_{\boldsymbol{S}}$.

To obtain a tractable lower bound of BPC, it suffices to give a tractable upper bound of the above objective function. To this end, let $\mathbb{T} \in \mathbb{V}_{\boldsymbol{S}[t]}$ be the vocabulary containing top $S[t]$ most frequent tokens, $\mathbb{C}$ be the set of chars and $|\mathbb{T}|, |\mathbb{C}|$ be their sizes respectively. Clearly, we have

$$\min_{v\in\mathbb{V}_{\boldsymbol{S}}} \frac{1}{l_v}\sum_{i\in v} P(i)\log P(i) \le \frac{1}{l_{\mathbb{T}}}\sum_{i\in\mathbb{T}} P(i)\log P(i). \tag{8}$$

Here we start from the upper bound of the above objective function, that is $\frac{1}{l_{\mathbb{T}}}\sum_{i\in\mathbb{T}} P(i)\log P(i)$ and then search for a refined token set from $\mathbb{T}$ with larger BPC. In this way, we reduce the search space into the subsets of $\mathbb{T}$. Let $P(i, j)$ be the joint probability distribution of the tokens and chars that we want to learn. Then we have

$$\begin{aligned}\sum_{i\in\mathbb{T}} P(i)\log P(i) &= \sum_{i\in\mathbb{T}}\sum_{j\in\mathbb{C}} P(i,j)\log P(i)\\ &= \underbrace{\sum_{i\in\mathbb{T}}\sum_{j\in\mathbb{C}} P(i,j)\log P(i,j)}_{\mathcal{L}_1} + \underbrace{\sum_{i\in\mathbb{T}}\sum_{j\in\mathbb{C}} P(i,j)(-\log P(j|i))}_{\mathcal{L}_2}.\end{aligned} \tag{9}$$

```
Algorithm 1: Info-VOT
Input: Token candidate sequence $\mathbb{L}$ ranked by frequencies, incremental integer sequence
  $\boldsymbol{S} = \{k, k+i, k+2\cdot i, \cdots, k+(t-1)\cdot i\}$ where the last item of $\boldsymbol{S}$ is less than $|\mathbb{L}|$, character
  sequence $\mathbb{C}$, training corpus $D_c$
Parameters: u $\in \mathbb{R}_+^{|\mathbb{C}|}$, v $\in \mathbb{R}_+^{|\mathbb{T}|}$
vocabularies = []
for item in $\boldsymbol{S}$ do
    // Begin of Sinkhorn algorithm
    Initialize u = ones() and v = ones()
    $\mathbb{T} = \mathbb{L}[: item]$
    Calculate all token frequencies $P(\mathbb{T})$ based on $D_c$
    Calculate all char frequencies $P(\mathbb{C})$ based on $D_c$
    Calculate $\boldsymbol{K}$ based on Eq. 10
    while not converge do
        u = $P(\mathbb{T})/\boldsymbol{K}$v
        v = $P(\mathbb{C})/\boldsymbol{K}^T$u
    optimal_matrix = u.reshape(-1, 1) * $\boldsymbol{K}$ * v.reshape(1, -1)
    // End of Sinkhorn algorithm
    BPC, vocab = get_vocab(optimal_matrix)
    // Generate a vocabulary based on the transport matrix
    vocabularies.append(BPC,vocab)
Select the optimal vocabulary $\boldsymbol{v}^*$ satisfying Eq. 4 from vocabularies
Output:$\boldsymbol{v}^*$
```

The details of proof can be found at Appendix C. Since $\mathcal{L}_1$ is nothing but the negative entropy of the joint probability distribution $P(i,j)$, we shall denote it as $-H(P)$. Let $\boldsymbol{K}$ be the $|\mathbb{C}| \times |\mathbb{T}|$ matrix whose (i,j)-th entry is given by $-\log P(j|i)$, then we can write

$$\mathcal{L}_2 = \langle \boldsymbol{P}, \boldsymbol{K} \rangle \tag{10}$$

where $K_{ij} = -\log P(j|i) = +\infty$ if $j \notin i$ and $-\log \frac{\#c \in t}{len(t)}$ otherwise.

Note that we have the hard constraints $\sum_j P(i,j) = P(i)$ and $\sum_i P(i,j) = P(j)$ where $P(i), P(j)$ are the char distribution and candidate token distribution of $\mathbb{T}$, respectively. However, in order to obtain a refined token set from $\mathbb{T}$ with larger BPC, we need to relax the hard constraint on the token distribution matching to a soft constraint. This formulation then allows us to drop out tokens with low joint probability distribution. See the discussion at the end of this section for more implementation details. In summary, our final objective function is

$$\begin{aligned} &\underset{\boldsymbol{P} \in \mathbb{R}^{|\mathbb{C}| \times |\mathbb{T}|}}{\arg\min} \; -H(\boldsymbol{P}) + \langle \boldsymbol{P}, \boldsymbol{K} \rangle \,, \\ \text{s.t.} \quad &\sum_i \boldsymbol{P}(i,j) = P(j), \quad |\sum_j \boldsymbol{P}(i,j) - P(i)| \leq \epsilon, \quad \forall i,j. \end{aligned}$$

with small $\epsilon > 0$. This objective function has the same form as the entropy regularized OT Eq. 7 with $\gamma = 1$ except for the soft constraint on the token distribution matching. Strictly speaking, this is an unbalanced entropy regularized Optimal Transport problem. Nonetheless, we can still use the generalized Sinkhorn algorithm to efficiently find the target vocabulary as detailed in Section 4.6 of Peyré & Cuturi (2020). The algorithm details are shown in Algorithm 1. At each timestep t, we can generate a new vocabulary associated with BPC scores based on the transport matrix $\boldsymbol{P}$. Here we filter several tokens which do not get enough characters in the transport matrix. Please refer to Section 4.3 for more details. Then, we collect these vocabularies associated with BPC scores, and output the vocabulary satisfying Eq. 4.

4.3 Implementation

Algorithm 1 lists the whole process of *Info*-VOT. First, we rank all token candidates according to their frequencies. Due to the large space of token candidates, we adopt BPE generated tokens (e.g. BPE-100K) as the target token candidates. It is important to note that any segmentation algorithms can be used to initialize token candidates. Each merge action represents a new token. We range all tokens based on the generation rank. In this way, we can get a sequence of tokens associated with their probabilities $\boldsymbol{P}_{bpe}$ that is then used to initialize $\mathbb{L}$ in Algorithm 1. The size of the incremental integer sequence $\boldsymbol{S}$ is a hyper-parameter.

At each timestep, we can get the vocabulary with the maximum BPC score based on the transport matrix. It is inevitable to handle illegal transport case in the transport matrix. We remove tokens with distributed chars less than one-tenth of token frequencies. Then, we enumerate all timesteps and select the vocabulary satisfying Eq. 4 as the final vocabulary.

5 EXPERIMENTS

To evaluate the performance of *Info*-VOT, we conduct experiments on three datasets, including WMT-14 English-German translation, TED bilingual translation, and TED multilingual translation.

5.1 SETTINGS

We run experiments on the following machine translation datasets. See Appendix B for more model and training details.

1. WMT-14 English-German (En-De) dataset: This dataset has 4.5M sentence pairs. The dataset is processed following Ott et al. (2018). We choose newstest14 as the test set.
2. TED bilingual dataset: We include two settings: many-to-English multilingual translation and English-to-many multilingual translation. We choose 12 language-pairs for evaluation. We use the language code according to ISO-639-1 standard[2]. TED data is provided by Ye et al. (2018).
3. TED multilingual dataset: We conduct experiments with 45 language pairs on a many-to-English setting. The network is trained on all language pairs. We adopt the same pre-processing pipeline in the WMT-14 En-De dataset.

6 RESULTS AND ANALYSIS

Vocabularies Searched by *Info*-VOT are Better than Widely-used BPE Vocabularies. Ding et al. (2019) gather 42 papers that have been accepted by the research track of Conference of Machine Translation (WMT) through 2017 and 2018. Among these papers, the authors find that 30K–40K is the most popular range for the number of BPE merge actions. Following this work, we first compare our methods with dominant BPE-30K. The results are listed in Table 1. As we can see, the vocabularies searched by *Info*-VOT achieve competitive BLEU scores with smaller sizes. The promising results demonstrate that *Info*-VOT is a practical approach that can find a well-performing vocabulary with higher BLEU and smaller size.

***Info*-VOT Works Well on Multilingual Settings.** We conduct a multilingual experiment covering 45 language pairs. These languages come from multiple language families and have diverse characters. Table 7 in Appendix B lists the full results. We compare *Info*-VOT with BPE-60K, the most popular setting in multilingual translation tasks. As we can see, *Info*-VOT achieves better BLEU scores on 30 out of 45 datasets.

Vocabularies Searched by *Info*-VOT are on Par with BPE-1K Recommended by Ding et al. (2019) on Low-resource Datasets. Ding et al. (2019) study how the size of BPE affects the model performance. They conduct experiments on 4 language pairs and find that smaller vocabularies are more suitable for low-resource datasets. For Transformer architectures, the optimal vocabulary size lays between 0–4K, around 0-2K merge actions. We compare *Info*-VOT and BPE-1K on a many-to-English bilingual setting. The results are shown in Table 2. We can see that *Info*-VOT can find a good vocabulary that is on par with heuristically searched vocabularies in terms of BLEU scores. Note that the advantage of *Info*-VOT lies in its high-efficiency on searching for well-performing vocabularies on diverse translation settings, including high-resource bilingual translation, low-resource bilingual

[2] http://www.lingoes.net/en/translator/langcode.htm

Table 1: Comparison between vocabularies search by *Info*-VOT and widely-used BPE vocabularies. BPE-30K is the most popular setting in 42 papers accepted by the research track of Conference of Machine Translation (WMT) through 2017 and 2018. * means WMT translation results, and the rest columns are TED results. *Info*-VOT achieves higher BLEU scores with massive size reduction.

En-X	**De***	**Es**	**PTbr**	**Fr**	**Ru**	**He**	**Ar**	**Ko**	**It**	**Nl**	**Ro**	**Tr**	**De**
BPE-30K	29.51	39.97	40.32	43.98	20.11	28.76	18.25	10.40	36.88	33.42	28.50	19.12	30.28
***Info*-VOT**	**30.00**	**40.75**	**41.65**	**45.02**	**20.38**	**29.82**	**18.65**	**10.41**	**37.24**	**33.83**	**29.56**	**20.06**	**31.52**
X-En	**De***	**Es**	**PTbr**	**Fr**	**Ru**	**He**	**Ar**	**Ko**	**It**	**Nl**	**Ro**	**Tr**	**De**
BPE-30K	-	44.37	47.08	42.70	28.21	39.93	33.83	22.20	41.44	39.43	37.65	28.89	38.91
***Info*-VOT**	-	**45.47**	**47.72**	**43.49**	**28.78**	**41.31**	**35.01**	**22.78**	**41.67**	**39.80**	**39.15**	**30.20**	**39.95**
Vocab Size (K)	**De***	**Es**	**PTbr**	**Fr**	**Ru**	**He**	**Ar**	**Ko**	**It**	**Nl**	**Ro**	**Tr**	**De**
BPE-30K	33.6	29.9	29.8	29.8	30.1	30.0	30.3	33.5	29.8	29.8	29.9	30.0	29.9
***Info*-VOT**	**8.5**	**1.9**	**1.7**	**1.5**	**1.7**	**1.5**	**1.7**	**5.2**	**1.9**	**2.0**	**1.9**	**1.8**	**1.7**

Table 2: Comparison between vocabularies search by *Info*-VOT and BPE-1K, recommended by Ding et al. (2019) for low-resource datasets. Here we take TED En-X bilingual translation as an example. This table demonstrate that vocabularies searched by *Info*-VOT are on par with heuristically searched vocabularies in terms of BLEU scores.

X-En	**Es**	**PTbr**	**Fr**	**Ru**	**He**	**Ar**	**Ko**	**It**	**Nl**	**Ro**	**Tr**	**De**	**avg**
BPE-1K	44.99	**47.85**	43.13	28.37	**41.31**	**35.04**	22.67	41.47	**40.19**	38.83	30.10	39.24	37.76
***Info*-VOT**	**45.47**	47.72	**43.49**	**28.78**	**41.31**	35.01	**22.78**	**41.67**	39.80	**39.15**	**30.20**	**39.95**	**37.94**
Vocab Size (K)	**Es**	**PTbr**	**Fr**	**Ru**	**He**	**Ar**	**Ko**	**It**	**Nl**	**Ro**	**Tr**	**De**	**avg**
BPE-1K	**1.4**	**1.3**	**1.3**	**1.4**	**1.3**	**1.5**	**4.7**	**1.2**	**1.2**	**1.2**	**1.2**	**1.2**	**1.6**
***Info*-VOT**	1.9	1.7	1.5	1.7	1.5	1.7	5.2	1.9	2.0	1.9	1.8	1.7	2.0

translation, and multilingual translation. In contrast, BPE-1K is selected based on plenty of training trials towards low-resource settings.

***Info*-VOT is a Green Vocabularization Solution.** One advantage of *Info*-VOT lies in its low consumption of computation resources. For traditional BPE-Search solution, full training is necessary to select the best-performing vocabularies. We compare *Info*-VOT with BPE-Search in Table 3. In BPE-Search, we first define a vocabulary set including BPE-1K, BPE-2K, BPE-3K, BPE-4K, BPE-5K, BPE-6K, BPE-7K, BPE-8K, BPE-9K, BPE-10K, BPE-20K, BPE-30K. Then, we run full experiments to select the best vocabulary. Table 3 demonstrates that *Info*-VOT is a green solution that can find a competitive vocabulary within a few hours on a single CPU, compared to BPE-Search that takes hundreds of GPU hours. The cost of BPE-Search is the sum of the training time on all vocabularies.

A Simple Baseline with *Info*-VOT-generated Vocabularies Beats Existing Strong Approaches. We implement *Info*-VOT on a widely-used baseline, Transformer-big. We are curious about how much a baseline can reach only with the change of vocabularyies. We compare *Info*-VOT and several strong approaches on WMT-14 En-De dataset. Table 4 shows surprisingly good results of our method. Compared to existing approaches in the top block, *Info*-VOT achieves almost the best performance with a much smaller vocabulary. These results demonstrate that a simple baseline can achieve good results with a well-defined vocabulary.

***Info*-VOT Beats SentencePiece and WordPiece.** SentencePiece and WordPiece are two variants of subword vocabularies. We also compare our approach with them on WMT-14 En-De dataset to evaluate the effectiveness of *Info*-VOT. The middle block of Table 4 lists the results of SentenPiece and WordPiece. We implement these two approaches with the default setting. We can observe that *Info*-VOT outperforms SentencePiece and WordPiece by a large margin, with over 1 BLEU score improvements.

Table 3: Comparison between *Info*-VOT and BPE-Search on bilingual settings. In BPE-Search solution, the best-performing vocabulary BPE-5K is selected based on its average performance from BPE-1K, BPE-2K, BPE-3K, BPE-4K, BPE-5K, BPE-6K, BPE-7K, BPE-8K, BPE-9K, BPE-10K, BPE-20K, and BPE-30K. BPE-Search requires full training and takes 288 GPU hours to search for the optimal vocabulary while *Info*-VOT only takes 0.5 CPU hours.

X-En	**Es**	**PTbr**	**Fr**	**Ru**	**He**	**Ar**	**Ko**	**It**	**Nl**	**Ro**	**Tr**	**De**	**avg**
BPE-Search	45.26	**47.98**	43.25	**29.18**	**41.47**	**35.59**	**23.14**	41.64	**40.16**	38.77	**30.78**	39.26	**38.03**
***Info*-VOT**	**45.47**	47.72	**43.49**	28.78	41.31	35.01	22.78	**41.67**	39.80	**39.15**	30.20	**39.95**	37.94
Vocab Size (K)	**Es**	**PTbr**	**Fr**	**Ru**	**He**	**Ar**	**Ko**	**It**	**Nl**	**Ro**	**Tr**	**De**	**Cost**
BPE-Search	5.3	5.3	5.3	5.4	5.3	5.5	8.8	9.9	5.2	5.2	5.2	5.2	288 GPU hours
***Info*-VOT**	**1.9**	**1.7**	**1.5**	**1.7**	**1.5**	**1.7**	**5.2**	**1.9**	**2.0**	**1.9**	**1.8**	**1.7**	**0.5 CPU hours**

Table 4: Comparison between *Info*-VOT and strong baselines. *Info*-VOT achieves almost the best performance with a much smaller vocabulary.

WMT-14 En-De	**BLEU**	**Merge Actions**	**Vocabulary Size**	**Parameters**
Vaswani et al. (2017)	28.4	32K	33.6K	210M
Shaw et al. (2018)	29.2	32K	33.6K	213M
Ott et al. (2018)	29.3	32K	33.6K	210M
So et al. (2019)	29.8	32K	33.6K	218M
Liu et al. (2020)	30.1	32K	33.6K	256M
SentencePiece	28.7	32K	33.6K	210M
WordPiece	29.0	32K	33.6K	210M
***Info*-VOT**	**30.0**	**8.5K**	**8.7K**	**188M**

***Info*-VOT Works Well on Normal-size Architectures.** This work mainly focus on Transformer-big model in experiments. We are still curious about whether *Info*-VOT also works on other architectures. We take WMT-14 En-De translation as an example and implement a Transformer network and a Convolutional Seq2Seq model. All networks use the default settings from Fairseq[3]. We set the maximum epochs to 100 and average the last five models as the final network for evaluation. Please refer to Appendix B for more results. Table 6 in Appendix B demonstrates that vocabularies search by *Info*-VOT work well on different architectures. It is important to note that model size also affects BLEU scores. In this work, we verify the effectiveness of *Info*-VOT on normal-size architectures. For those small architectures, we recommend larger vocabularies associated with more embedding parameters.

7 Conclusion

In this work, we propose a unified information-theoretic vocabulary learning framework. The whole framework starts from an exciting finding that AMD, an information-theoretic feature, correlates with BLEU scores. Based on this finding, we design a two-step discrete optimization objective and a principled optimal transport solution: *Info*-VOT. Experiments show that *Info*-VOT is an effective approach. It can quickly find a well-performing vocabulary on diverse settings, inlcuding high-resource bilingual translation, low-resource bilingual translation, and multilingual translation.

Despite promising results, *Info*-VOT still has several limitations that need to be improved in future work. First, *Info*-VOT relies on the initialized token distribution. For simplification, we directly adopt BPE-100K generated tokens associated with their probabilities as initialization. Second, the transport matrix in *Info*-VOT still needs an additional post-processing pipeline to get the final vocabulary. Although a recommended post-processing setting is given, we believe that an advanced algorithm in the future can reduce the dependency on post-processing.

[3] https://github.com/pytorch/fairseq/tree/master/examples/translation

References

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. Character-level language modeling with deeper self-attention. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 3159–3166. AAAI Press, 2019.

Wenhu Chen, Yu Su, Yilin Shen, Zhiyu Chen, Xifeng Yan, and William Yang Wang. How large a vocabulary does text classification need? A variational approach to vocabulary selection. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 3487–3497. Association for Computational Linguistics, 2019.

Colin Cherry, George F. Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. Revisiting character-based neural machine translation with capacity and compression. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 4295–4305. Association for Computational Linguistics, 2018.

Marta R. Costa-jussà and José A. R. Fonollosa. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics, 2016.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pp. 2292–2300, 2013.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

Shuoyang Ding, Adithya Renduchintala, and Kevin Duh. A call for prudent choice of subword merge operations in neural machine translation. In Mikel L. Forcada, Andy Way, Barry Haddow, and Rico Sennrich (eds.), *Proceedings of Machine Translation Summit XVII Volume 1: Research Track, MTSummit 2019, Dublin, Ireland, August 19-23, 2019*, pp. 204–213. European Association for Machine Translation, 2019.

Marylou Gabrié, Andre Manoel, Clément Luneau, Jean Barbier, Nicolas Macris, Florent Krzakala, and Lenka Zdeborová. Entropy and mutual information in models of deep neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pp. 1826–1836, 2018.

Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. Bottom-up abstractive summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 4098–4109. Association for Computational Linguistics, 2018.

Ziv Goldfeld, Ewout van den Berg, Kristjan H. Greenewald, Igor Melnyk, Nam Nguyen, Brian Kingsbury, and Yury Polyanskiy. Estimating information flow in deep neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2299–2308. PMLR, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Julia Kreutzer and Artem Sokolov. Learning to segment inputs for NMT favors character-level processing. *CoRR*, abs/1810.01480, 2018.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pp. 66–71. Association for Computational Linguistics, 2018.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5: 365–378, 2017.

Xiaodong Liu, Kevin Duh, Liyuan Liu, and Jianfeng Gao. Very deep transformers for neural machine translation. *CoRR*, abs/2008.07772, 2020.

David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pp. 3111–3119, 2013.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. In Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana L. Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor (eds.), *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pp. 1–9. Association for Computational Linguistics, 2018.

Gabriel Peyré and Marco Cuturi. Computational optimal transport, 2020.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. Bpe-dropout: Simple and effective subword regularization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 1882–1892. Association for Computational Linguistics, 2020.

Elizabeth Salesky, Andrew Runge, Alex Coda, Jan Niehues, and Graham Neubig. Optimizing segmentation granularity for neural machine translation. *Machine Translation*, pp. 1–19, 2020.

Andrew M. Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D. Tracey, and David D. Cox. On the information bottleneck theory of deep learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Rico Sennrich and Biao Zhang. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, July 2019.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pp. 464–468. Association for Computational Linguistics, 2018.

David R. So, Quoc V. Le, and Chen Liang. The evolved transformer. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5877–5886. PMLR, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. Neural machine translation with byte-level subwords. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 9154–9160. AAAI Press, 2020.

Qi Ye, Sachan Devendra, Felix Matthieu, Padmanabhan Sarguna, and Neubig Graham. When and why are pre-trained word embeddings useful for neural machine translation. In *HLT-NAACL*, 2018.

Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253, 2018.

Yi Zhao, Yanyan Shen, and Junjie Yao. Recurrent neural network for text classification with hierarchical multiscale dense connections. In Sarit Kraus (ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 5450–5456. ijcai.org, 2019.

APPENDIX A: AMD IS CORRELATED TO BLEU SCORES

To evaluate the relationship between AMD and BLEU scores, we conduct experiments on 45 language pairs from TED and calculate their Spearman correlation scores. To avoid the effects of unsteady BLEU scores, we use a multilingual network to initialize bilingual networks. Specifically, the BLEU scores come from bilingual models pre-trained on multilingual datasets segmented by BPE-30K, BPE-60K, BPE-80K, BPE-100K, BPE-120K, BPE-140K. Here BPE size refers to the number of BPE merge actions from all languages. Formally, we define a vocabulary sequence $\mathbf{I}$ = $\{$v(BPE-20K), v(BPE-30K), v(BPE-60K), v(BPE-80K), v(BPE-100K), v(BPE-120K), v(BPE-140K)$\}$. We set the maximum training epoch to 50 and average the last five models as the final network for evaluation. For each translation pair p with the t-th vocabulary in $\mathbf{I}$, the AMD score is calculated as follows:

$$\mathbf{D}(p, \mathbf{I}_t) = \frac{\mathbf{B}(\mathbf{I}_t) - \mathbf{B}(\mathbf{I}_{t-1})}{\mathbf{S}(\mathbf{I}_t) - \mathbf{S}(\mathbf{I}_{t-1})}$$

where $\mathbf{B}(\mathbf{I}(t))$ represents the BPC value of training data segmented by vocabulary $\mathbf{I}_t$. $\mathbf{S}(\mathbf{I}_t)$ represents the number of unique tokens of training data segmented by vocabulary $\mathbf{I}_t$.

The full results are shown in Table 5. The median correlation score is 0.49. In general, [0.8, 1] means very strong correlations, [0.6, 0.8] means strong correlations, [0.4, 0.6] means moderate correlations, [0.2, 0.4] means weak correlations. Almost two-thirds of pairs show obvious positive correlations (greater than 0.4). Considering that other factors (e.g. model size, corpus size) also affect BLEU scores, we believe that it is good evidence to support the claim.

Table 5: The correlation coefficient between AMD and BLEU scores. Among 45 tasks, almost two-thirds of tasks show obvious correlations (greater than 0.4) between AMD and BLEU scores. Here we list tasks with top-30 correlation scores. In general, [0.8, 1] means very strong correlations, [0.6, 0.8] means strong correlations, [0.4, 0.6] means moderate correlations, [0.2, 0.4] means weak correlations, [0.0, 0.2] means extremely weak correlations.

X-En	Correlation	BPE-30K	BPE-60K	BPE-80K	BPE-100K	BPE-120K	BPE-140K
Sr	0.90	36.52 8.93e-06	36.91 1.12e-05	36.67 9.79e-06	36.24 9.15e-06	36.24 8.45e-06	36.07 7.45e-06
Zh-cn	0.89	22.31 1.10e-06	22.73 1.14e-05	22.96 7.68e-06	22.43 6.29e-06	22.59 7.21e-06	22.41 6.71e-06
Hu	0.89	27.79 1.18e-05	28.41 1.67e-05	28.25 1.50e-05	28.2 1.51e-05	27.96 1.47e-05	28.17 1.46e-05
Ro	0.77	36.23 9.28e-06	36.8 1.44e-05	36.91 1.41e-05	36.83 1.38e-05	36.77 1.18e-05	36.09 1.17e-05
Sl	0.75	26.11 1.58e-05	26.53 2.12e-05	26.9 2.36e-05	26.63 2.12e-05	26.37 2.25e-05	25.6 1.95e-05
Uk	0.75	28.74 6.85e-06	29.58 1.80e-05	29.48 1.92e-05	29.12 1.70e-05	29.18 1.70e-05	29.31 1.56e-05
Hy	0.74	23.18 2.49e-05	23.63 4.47e-05	23.56 5.78e-05	23.71 4.47e-05	23.56 4.14e-05	23.23 3.70e-05
Vi	0.66	28.39 9.36e-06	28.63 2.16e-05	28.65 1.93e-05	28.8 1.97e-05	28.62 1.73e-05	28.35 1.75e-05
Nl	0.66	37.27 1.53e-05	37.17 1.84e-05	37.67 1.69e-05	37.45 1.63e-05	37.05 1.48e-05	36.89 1.44e-05
Et	0.64	22.4 2.23e-05	22.35 3.48e-05	21.65 3.14e-05	22.75 3.60e-05	22.71 3.60e-05	21.89 3.41e-05
Sv	0.61	39.66 1.76e-05	40.16 2.24e-05	40.12 2.07e-05	40.2 1.92e-05	40.02 1.83e-05	40.02 1.93e-05
Sq	0.60	37.68 1.62e-05	38.47 1.83e-05	38.07 1.78e-05	38.25 1.88e-05	37.68 1.59e-05	38.29 1.62e-05
My	0.60	18.38 1.35e-05	19.5 3.30e-05	18.98 2.98e-05	18.87 3.05e-05	18.74 3.13e-05	18.83 2.47e-05
Tr	0.60	27.82 1.31e-05	28.44 1.68e-05	28.68 1.63e-05	28.19 1.51e-05	28.46 1.40e-05	28.31 1.37e-05
Ja	0.60	16.94 -6.71e-06	17.02 1.65e-05	17.23 1.34e-05	17.07 1.17e-05	17.09 1.24e-05	16.85 1.08e-05
Sk	0.59	31.49 1.04e-05	32.74 1.61e-05	32.12 1.57e-05	32.52 1.57e-05	32.35 1.44e-05	32.74 1.51e-05
Ka	0.58	22.24 2.10e-06	22.32 5.52e-05	22.91 5.52e-05	21.75 4.90e-05	21.86 4.27e-05	21.36 4.18e-05
Mk	0.54	33.68 1.24e-05	34.44 2.92e-05	34.52 2.73e-05	34.58 2.62e-05	33.38 2.47e-05	33.52 2.22e-05
Da	0.54	43.43 2.22e-05	44.57 2.50e-05	44.97 2.41e-05	44.16 2.40e-05	44.55 2.07e-05	44.27 2.25e-05
Lt	0.54	26.09 1.27e-05	26.4 2.18e-05	26.3 2.04e-05	26.58 2.13e-05	26.61 2.02e-05	25.78 2.01e-05
Nb	0.54	41.84 2.54e-05	43.44 3.09e-05	43.87 3.10e-05	44.13 2.98e-05	44.14 3.15e-05	44.06 2.85e-05
Pl	0.49	25.69 8.24e-06	25.94 1.48e-05	26.17 1.43e-05	26.16 1.57e-05	26.07 1.37e-05	26.13 1.30e-05
Zh-tw	0.49	21.17 -3.59e-06	21.86 8.30e-06	21.41 8.15e-06	21.57 4.74e-06	21.47 5.17e-06	21.55 5.51e-06
Ru	0.46	26.25 7.36e-06	26.56 1.68e-05	26.74 1.68e-05	26.79 1.60e-05	26.69 1.59e-05	26.63 1.44e-05
Bg	0.46	40.47 9.00e-06	40.53 1.56e-05	40.23 1.45e-05	40.52 1.47e-05	40.25 1.34e-05	40.25 1.35e-05
Cs	0.44	30.13 1.15e-05	30.85 1.85e-05	30.24 1.76e-05	30.53 1.76e-05	30.85 1.60e-05	30.45 1.66e-05
Ku	0.43	19.2 9.83e-06	19.79 3.34e-05	19.85 3.28e-05	18.9 2.76e-05	19.27 2.60e-05	18.2 2.89e-05
It	0.31	39.33 1.63e-05	38.5 1.98e-05	38.94 1.77e-05	38.65 1.55e-05	38.73 1.45e-05	38.46 1.36e-05
Es	0.26	42.56 1.79e-05	43.02 1.91e-05	43.06 1.76e-05	43.05 1.55e-05	42.44 1.36e-05	42.67 1.35e-05
Fr	0.20	40.87 2.12e-05	40.79 2.14e-05	40.94 1.81e-05	40.72 1.74e-05	40.95 1.46e-05	40.69 1.37e-05

Table 6: *Info*-VOT can find better vocabularies than widely-used vocabularies on normal-size architectures. Here "better" means competitive results but smaller sizes.

WMT-14 En-De	BLEU	Vocabulary Size
Transformer-big	29.30	33.6K
	30.00	**8.5K**
Transformer	**27.71**	33.6K
	27.60	**8.5K**
Convolutional Seq2Seq	**26.35**	33.6K
	26.35	**8.5K**

APPENDIX B: EXPERIMENTS

Models. We use Fairseq to train a Transformer-big model with the same setting in the original paper (Ott et al., 2018). The input embedding and output embeddings are shared. We use the Adam optimizer (Kingma & Ba, 2015) with a learning rate 5e-4 and an inverse_sqrt decay schedule. The warm-up step is $4,000$, the dropout rate is 0.3, the update frequency is 8, the number of tokens is $9,600$, or $4,800$ in a single batch.

Training and Evaluation. We run WMT-14 En-De experiments with 4 GPUs, TED bilingual translation with 2 GPUs, TED multilingual translation with 8 GPUs. We set a beamwidth to 4 for En-De and 5 for the other. We average the last five models on all datasets and use the averaged model to generate translation results. We calculate case-sensitive tokenized BLEU for evaluation.

***Info*-VOT Works on Normal-size Architectures.** We take WMT'14 En-De as an example and implement a Transformer network and a Convolutional Seq2Seq network. All networks use the default settings from Fairseq[4]. We set the maximum epochs to 100 and average the last five models as the final network for evaluation. Table 6 shows that *Info*-VOT can find better vocabularies than widely-used vocabularies on diverse architectures.

***Info*-VOT can Find Better Vocabularies on Multilingual Translation.** Table 7 lists the comparison on multilingual translation. These languages come from multiple language families and have diverse characters. BPE-60K is the most popular setting in multilingual translation tasks. As we can see, *Info*-VOT achieves better BLEU scores on 30 out of 45 language pairs.

APPENDIX C: PROOFS FOR EQ. 9

$$\begin{aligned}
\sum_{i\in\mathbb{T}} P(i)\log P(i) &= \sum_{i\in\mathbb{T}}\sum_{j\in\mathbb{C}} P(i,j)\log P(i) \\
&= \sum_{i\in\mathbb{T}}\sum_{j\in\mathbb{C}} P(i,j)\log P(i,j)\cdot\frac{P(i)}{P(i,j)} \\
&= \sum_{i\in\mathbb{T}}\sum_{j\in\mathbb{C}} P(i,j)\log P(i,j) + \sum_{i\in\mathbb{T}}\sum_{j\in\mathbb{C}} P(i,j)\log\frac{P(i)}{P(i,j)} \\
&= \underbrace{\sum_{i\in\mathbb{T}}\sum_{j\in\mathbb{C}} P(i,j)\log P(i,j)}_{\mathcal{L}_1} + \underbrace{\sum_{i\in\mathbb{T}}\sum_{j\in\mathbb{C}} P(i,j)(-\log P(j|i))}_{\mathcal{L}_2}.
\end{aligned}$$

□

[4] https://github.com/pytorch/fairseq/tree/master/examples/translation

Table 7: Comparison between *Info*-VOT and widely-used BPE vocabularies on multilingual translation. Here we show the results on 45 language pairs. BPE-60K is the most popular setting in multilingual translation tasks. *Info*-VOT achieves better BLEU scores on 30 out of 45 datasets. The size of vocabulary generated by *Info*-VOT is around 90K.

X-En	**Es**	**PT-br**	**Fr**	**Ru**	**He**	**Ar**	**Ko**	**Zh-cn**	**It**	**Ja**	**Zh-tw**	**Nl**	**Ro**
BPE-60K	35.00	37.73	33.53	**24.22**	30.58	26.32	**19.41**	20.98	32.63	16.21	**20.00**	30.77	30.58
***Info*-VOT**	**35.49**	**38.36**	**34.00**	24.20	**30.96**	**26.52**	19.36	**21.15**	**32.67**	**16.30**	19.99	**31.22**	**30.92**

X-En	**Tr**	**De**	**Vi**	**Pl**	**Pt**	**Bg**	**El**	**Fa**	**Sr**	**Hu**	**Hr**	**Uk**	**Cs**
BPE-60K	23.64	29.96	25.20	**23.82**	35.88	32.87	32.52	24.53	**30.26**	**24.07**	32.04	26.40	**27.18**
***Info*-VOT**	**23.74**	**30.56**	**25.47**	23.78	**36.16**	**33.36**	**33.39**	**25.03**	30.24	**24.07**	**32.13**	**26.44**	27.07

X-En	**Id**	**Th**	**Sv**	**Sk**	**Sq**	**Lt**	**Da**	**My**	**Sl**	**Mk**	**Fr-ca**	**Fi**	**Hy**
BPE-60K	27.51	**22.34**	33.21	29.31	**31.78**	**23.73**	**37.72**	**17.82**	**24.99**	29.92	30.42	21.13	**21.70**
***Info*-VOT**	**28.05**	22.24	**33.94**	**29.36**	**31.78**	22.91	**37.72**	17.43	24.83	**31.49**	**31.04**	**21.30**	21.40

X-En	**Hi**	**Nb**	**Ka**	**Et**	**Ku**	**Gl**
BPE-60K	**23.87**	38.36	**21.51**	**22.05**	**18.45**	**31.39**
***Info*-VOT**	22.69	**39.72**	20.91	21.27	17.85	**31.39**

APPENDIX D: SUPPLEMENTAL EXPERIMENTS

We conduct bilingual experiments on 14 language pairs with the most training data from TED. The below table shows that BPE-2K is not a universal solution. Different datasets have varying optimal sizes.

Table 8: BPE-2K is not a universal solution. Different datasets have varying optimal sizes.

X-En	**BPE-1K**	**BPE-2K**	**BPE-3K**	**BPE-5K**	**BPE-6K**	**BPE-7K**	**BPE-8k**	**BPE-9K**	**BPE-10K**	**BPE-20K**
Es	44.99	45.29	45.28	45.26	**45.31**	44.86	44.99	45.04	45.11	44.44
PT-br	47.85	47.88	**48.00**	47.98	47.5	47.61	47.66	47.39	47.21	47.05
Fr	43.13	43.58	43.45	43.25	43.58	43.61	43.54	**43.71**	43.12	43.05
Ru	28.37	28.68	29.19	29.18	29.10	29.07	29.24	**29.28**	29.10	29.32
He	41.31	41.28	**41.54**	41.47	40.89	40.99	41.1	41.13	41.31	40.38
Ar	35.04	**35.63**	35.61	35.59	35.59	35.43	34.54	35.17	35.26	34.12
Ko	22.67	22.53	22.97	23.14	22.92	22.01	**23.18**	22.84	22.66	22.35
Zh-cn	**24.49**	24.42	24.11	23.99	23.96	24.33	24.09	24.32	24.00	23.47
It	41.47	41.72	41.82	41.64	41.59	41.65	**41.83**	41.60	41.96	41.22
Ja	16.55	17.24	17.47	**17.68**	17.08	17.58	17.65	17.59	17.59	16.8
Zh-tw	22.65	22.95	23.04	**23.38**	23.12	22.39	22.79	22.80	22.71	22.74
Nl	40.19	**40.80**	40.27	40.16	40.24	40.18	39.76	39.8	40.33	39.57
Ro	38.83	**39.18**	39.09	38.77	38.88	38.94	38.9	38.33	38.44	38.29
Tr	30.10	30.35	30.22	**30.78**	29.22	30.37	29.99	29.61	29.97	29.73