EBBS: An Ensemble with Bi-Level Beam Search for Zero-Shot Machine Translation

Yuqiao Wen[1,*], **Behzad Shayegh**[1,*], **Chenyang Huang**[1], **Yanshuai Cao**[2], **Lili Mou**[1,3]
[1]Dept. Computing Science, Alberta Machine Intelligence Institute (Amii), University of Alberta
[2]RBC Borealis
[3]Canada CIFAR AI Chair, Amii
yq.when@gmail.com, the.shayegh@gmail.com, chenyangh@ualberta.ca
yanshuai.cao@borealisai.com, doublepower.mou@gmail.com

Abstract

The ability of zero-shot translation emerges when we train a multilingual model with certain translation directions; the model can then directly translate in unseen directions. Alternatively, zero-shot translation can be accomplished by pivoting through a third language (e.g., English). In our work, we observe that both direct and pivot translations are noisy and achieve less satisfactory performance. We propose EBBS, an ensemble method with a novel bi-level beam search algorithm, where each ensemble component explores its own prediction step by step at the lower level but all components are synchronized by a "soft voting" mechanism at the upper level. Results on two popular multilingual translation datasets show that EBBS consistently outperforms direct and pivot translations, as well as existing ensemble techniques. Further, we can distill the ensemble's knowledge back to the multilingual model to improve inference efficiency; profoundly, our EBBS-distilled model can even outperform EBBS as it learns from the ensemble knowledge.

Appendix — https://arxiv.org/abs/2403.00144
GitHub — https://github.com/MANGA-UOFA/EBBS

1 Introduction

Machine translation is a widely applicable NLP task that aims to translate a text from a source language to a target language (Brown et al. 1990; Bahdanau, Cho, and Bengio 2015). The Transformer architecture (Vaswani et al. 2017) and pretrained large language models (Radford et al. 2019; Lewis et al. 2020) have largely improved translation performance, especially in the supervised setting (Raffel et al. 2020), where a model can learn from large volumes of parallel corpora. However, machine translation remains challenging for low-resource languages, because there are not enough data for large neural networks to learn these languages (Radford et al. 2019; Muennighoff et al. 2023).

We specifically focus on multilingual translation in the *zero-shot* setting, where the system is required to translate between unseen language pairs. Since collecting parallel data and training individual models for every translation pair are prohibitively expensive, it is common to build a single multilingual system (Johnson et al. 2017; Fan et al. 2021) that can perform translation for all language pairs, most of which are zero-shot translation directions that do not involve a high-resource language (e.g., English). These models work by prepending a language-indicator token; the zero-shot translation ability emerges as the model generalizes from trained language pairs and is able to perform *direct translation* for unseen ones (Liu et al. 2021; Wicks and Duh 2022). The main drawback of such multilingual models is that they are noisy in the zero-shot setting due to the lack of supervision, and as a result, they tend to generate low-quality translations (Zhang et al. 2020; Liu et al. 2021).

Alternatively, zero-shot translation can be performed by *pivoting* (Wu and Wang 2007, 2009), where the model first translates the input into a high-resource language such as English, which is then translated to the target language. However, pivoting requires two translation steps, often leading to an accumulation of errors (Babych, Hartley, and Sharoff 2007; Gu et al. 2019).

In this paper, we propose an ensemble approach that aggregates direct and pivot translations in order to build a stronger multilingual translation model from weak ones. Building an ensemble for text generation is nuanced as it involves a sequence of word predictions. Word-level ensembles aggregate predictions at each generation step, which is usually achieved by averaging the predicted probabilities (Sennrich, Haddow, and Birch 2016a; Freitag, Al-Onaizan, and Sankaran 2017; Shanbhogue et al. 2023). This may not be ideal for zero-shot translation as the predictions are too noisy, making the averaged probabilities overly smooth. On the other hand, minimum Bayes risk decoding (MBR) (Bickel and Doksum 2015) can be considered a sequence-level voting ensemble, but existing MBR methods are only able to *select* from weak and noisy candidates given by the direct and pivot translations.

To this end, we propose an ensemble decoding algorithm with bi-level beam search (EBBS). Our EBBS performs two levels of beam search at each generation step: at the lower level, beam search is applied individually to each ensemble component; at the upper level, the ensemble maintains a shared beam by voting and synchronizing the candidates (sub-sequences) in lower-level beams. Unlike word-level ensembles (Freitag, Al-Onaizan, and Sankaran 2017;

[*]Work partially done during Mitacs internship at RBC Borealis

Shanbhogue et al. 2023), EBBS does not average the predicted distributions, encouraging individual predictors to explore their own preferences; unlike sequence-level MBR ensembles (Kobayashi 2018; Eikema and Aziz 2020), EBBS does not select from a candidate set, and thus is more flexible since votings are performed throughout the generation process.

We conducted experiments on IWSLT (Cettolo et al. 2017) and Europarl (Koehn 2005), two popular multilingual datasets for zero-shot machine translation. Results show that EBBS can generate high-quality translations and outperform existing ensemble techniques. In addition, we used EBBS-generated data for distillation to further improve the multilingual model. The experiment shows that such a distilling process encourages the model to learn from high-quality translations produced by EBBS, allowing it to outperform EBBS with no inference overhead compared with direct translation.

2 Related Work

Machine translation. In NLP, machine translation is a long-standing task that aims to rewrite text from one language to another without changing the meaning. Traditional research in translation has been mainly centered on the supervised setting, utilizing manually crafted rules (Forcada et al. 2011; Dugast, Senellart, and Koehn 2007) and statistical methods (Brown et al. 1990; Koehn 2009); more recently, neural machine translation systems have considerably improved the performance (Vaswani et al. 2017; Raffel et al. 2020). However, translation remains challenging for low-resource languages, where neural models do not have enough parallel data to train on.

Translation for low-resource languages largely relies on *zero-shot* techniques, where no parallel text is available for a particular translation direction. In general, zero-shot translation can be accomplished in a monolingual or multilingual setting. With monolingual data, the most common approach is to build language-specific autoencoders that share the same latent space of semantics; translation is then achieved by plugging in the decoder of the desired language (Lample et al. 2018a,b; Mohiuddin and Joty 2020).

In this paper, we focus on the multilingual setting, where one model can translate between multiple languages (Dabre, Chu, and Kunchukuttan 2020). Usually, parallel texts only exist for a high-resource language such as English, leaving translations between low-resource languages zero-shot (e.g., Italian to Dutch) (Johnson et al. 2017; Fan et al. 2021). In this setting, the most common approach is to train the multilingual model on English-centric data, and the zero-shot translation ability naturally emerges during the training process (Johnson et al. 2017; Scao et al. 2022).

A key challenge for multilingual models is task interference, where too many languages tend to degrade model performance (Zaremoodi, Buntine, and Haffari 2018; Wang, Lipton, and Tsvetkov 2020). As a result, research in this direction has been alleviating such interference by developing various parameter-separation schemes (Baziotis et al. 2022; Chronopoulou, Stojanovski, and Fraser 2023) and using gradient-based methods to update language-specific parameters (Wang and Zhang 2022; He et al. 2023). In our work, we use a standard Transformer model following Johnson et al. (2017) and Liu et al. (2021). Our proposed ensemble algorithm EBBS is compatible with the above approaches, as it is agnostic to model architectures.

Ensemble methods. In a model ensemble, multiple machine learning systems are integrated so as to form a stronger one (Dong et al. 2020; Yang, Lv, and Chen 2023). *Bagging*, a classic ensemble technique, works by training multiple models with different portions of data and combining their predictions through averaging or voting (Breiman 1996; Bühlmann and Yu 2002). Another popular ensemble approach is *boosting*, where different models are trained sequentially, with each subsequent model addressing the mistakes of the previous ones (Schapire 2003; Hastie et al. 2009; Natekin and Knoll 2013). Unfortunately, bagging and boosting are not compatible with our setting, because we build an ensemble with a single model. Alternatively, *stacking* combines the outputs by training a meta-model (Wolpert 1992; Ganaie et al. 2022), but this does not apply to our zero-shot setting either because we do not have groundtruth signals to train the meta-model. Even though these ensemble techniques may be directly applied to supervised generation (Freitag, Al-Onaizan, and Sankaran 2017; Kobayashi 2018; Hendy et al. 2021), they are not ideal as they do not take advantage of structured prediction. Our recent work has addressed the ensemble of tree structures (Shayegh et al. 2024; Shayegh, Wen, and Mou 2024; Shayegh et al. 2025), and in this paper we focus on text generation.

Unlike previous work, our EBBS performs bi-level beam search, exploring different components' own predictions and synchronizing them by a "soft voting" mechanism at every step. Our approach is specifically suited to the sequence generation process.

3 Approach

In this section, we first explain our ensemble components in §3.1. In §3.2, we propose EBBS, a novel ensemble decoding algorithm. Finally, we describe in §3.3 knowledge distillation with EBBS-decoded outputs for efficiency considerations.

3.1 Ensemble Components

In this work, we focus on zero-shot multilingual machine translation, which requires a system to perform translations for multiple languages, where some translation directions are unseen.

Specifically, our multilingual model is an encoder–decoder Transformer with a byte pair encoding tokenizer (Sennrich, Haddow, and Birch 2016b) shared among all languages. The encoder can capture the semantics of tokens in different languages, whereas the decoder translates the encoded text into the desired language based on a target-language indicator token (Johnson et al. 2017; Fan et al. 2021).

We follow the standard English-centric training (Johnson et al. 2017; Liu et al. 2021), where the multilingual model is trained using parallel data with En-

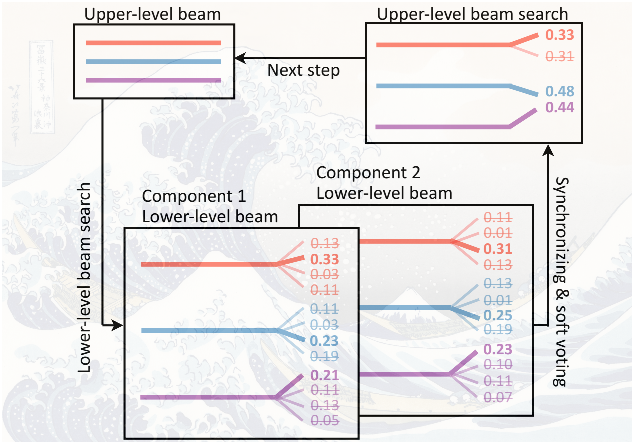

Figure 1: Illustration of our EBBS algorithm.

glish on one side (e.g., German-to-English and English-to-Romanian). As mentioned in §1, the zero-shot ability emerges during such training, and the model is able to perform direct translation between unseen language pairs (e.g., German-to-Romanian) (Dabre, Chu, and Kunchukuttan 2020; Ranathunga et al. 2023). An alternative approach is pivot translation, where the multilingual model performs two translations using a high-resource language as a pivot (e.g., first translating German to English, and then English to Romanian).

However, both direct and pivot translations have major weaknesses: the quality of direct translation tends to be low due to the lack of parallel data, whereas pivot translation suffers from error accumulation as it requires two translation steps (Babych, Hartley, and Sharoff 2007; Gu et al. 2019).

In this paper, we would like to build an ensemble of direct and pivot translations to boost translation quality, where each translation path results in an ensemble component. Commonly used ensemble methods such as averaging and voting may not work well for text generation. Voting, for example, chooses the most voted prediction, but in text generation, the components' votes often do not share anything in common, because there could be tens of thousands of tokens in the vocabulary. An averaging ensemble, on the other hand, averages the predicted distributions of all components, potentially leading to an overly smooth distribution. Despite early success by Razmara and Sarkar (2013), more recent studies report marginal or negative improvement for multi-pivot averaging ensemble (Fan et al. 2021; Gaikwad et al. 2024; Mohammadshahi, Vamvas, and Sennrich 2024).

3.2 Our Proposed EBBS Algorithm

We propose an ensemble with bi-level beam search (EBBS), a novel decoding algorithm that enables different ensemble components to collaborate and vote on each other's partial generations with two levels of beam search.

At the lower level, each ensemble component performs beam search individually, exploring its own preferred regions of the sentence space. At the upper level, EBBS synchronizes the lower-level beam candidates through a voting mechanism, only keeping the most promising partial generations in a shared, upper-level beam. This allows the ensemble components to vote out spurious partial candidates and improve zero-shot translation performance.

Concretely, we assume there are K ensemble components $p_1, \cdots, p_K$, each predicting the probability of the next word given some prefix.

For the 0th decoding step, EBBS initializes the upper-level beam by $\overline{B}_0 = \langle \textsc{bos}, 1 \rangle$, suggesting that a sequence is forced to start with a special beginning-of-sequence token BOS with probability 1.

For step t, each ensemble component performs lower-level beam search individually, based on the prefixes in the last step's shared beam $\overline{B}_{t-1}$:

$$\underline{B}_{t,k} = \text{top-}Z\{\ \langle \mathbf{y}_{1:t-1} \oplus \mathrm{y},\ \ p \cdot p_k(\mathrm{y}|\mathbf{y}_{1:t-1}, \mathbf{x}) \rangle : \langle \mathbf{y}_{1:t-1}, p \rangle \in \overline{B}_{t-1},\ \ \mathrm{y} \in V\ \} \tag{1}$$

for $k = 1, \cdots, K$. Here, top-Z selects Z-many sequences with the highest probabilities, $\oplus$ represents string concatenation, V is the vocabulary, and $p_k(\mathrm{y}|\mathbf{y}_{1:t-1}, \mathbf{x})$ is the kth ensemble component's predicted probability at step t given the prefix $\mathbf{y}_{1:t-1}$ and input $\mathbf{x}$.

At the upper level, EBBS synchronizes the lower-level individual beams $\underline{B}_{t,k}$, for $k = 1, \cdots, K$, into a shared, upper-level beam through a soft-voting mechanism, where the candidate set C_t is the union of the sequences in lower-level beams:

$$C_t = \bigcup\nolimits_k \{\mathbf{y} : \langle \mathbf{y}, p \rangle \in \underline{B}_{t,k}\} \tag{2}$$

We evaluate each candidate in C_t and compute its overall vote as the sum of the probabilities.

$$\overline{B}_t = \text{top-}Z \left\{ \left\langle \mathbf{y}, \sum_{\substack{k:\ k=1,\cdots,K \\ \langle \mathbf{y}', p \rangle \in \underline{B}_{t,k}:\ \mathbf{y}'=\mathbf{y}}} p \right\rangle : \ \mathbf{y} \in C_t \right\} \tag{3}$$

In this way, the upper level synchronizes all ensemble components with the shared beam $\overline{B}_t$ for the next step of generation.

Intuitively, our voting scheme gives an ensemble component Z-many votes, each weighted by the predicted probability. The votes (probabilities) are then tallied (summed) for each candidate to form the upper-level beam. Our bi-level beam search terminates when we have Z-many terminated sequences in the shared beam, and returns the sequence with the highest score[1] as the ensemble output. We provide the detailed pseudocode for EBBS in Algorithm 1 and an illustration in Figure 1.

Discussion. Traditional beam search keeps a fixed-size beam of high-likelihood partial sequences. To build an ensemble with multiple predictors, it is tempting to directly average their probabilities $p(\mathbf{y}|\mathbf{x}) = \frac{1}{K}\sum_{k=1}^{K} p_k(\mathbf{y}|\mathbf{x})$ as

[1]For selecting the final output, we follow standard implementations and normalize the joint probability by length, i.e., taking the geometric mean of step-wise probabilities (Wolf et al. 2019; Ott et al. 2019). Otherwise, beam search algorithms are often biased towards short sequences (Meister, Cotterell, and Vieira 2020).

Algorithm 1: Our EBBS Algorithm

Input: $\mathbf{x}$: input sentence; Z: beam size
K: number of scorers; $p_1, \cdots, p_K$: scorers

1 $H \leftarrow \emptyset$ ▷ candidate outputs
2 $\overline{B}_0 \leftarrow \{\langle \text{BOS}, 1\rangle\}$ ▷ upper-level beam
3 **for** $t = 1, 2, \cdots$ **do**
4 ▷ lower: individual beam search
5 **for** $\langle \mathbf{y}_{1:t-1}, p\rangle \in \overline{B}_{t-1}$ **do**
6 **for** $k = 1, \cdots, K$ **do**
7 $\underline{B}_{t,k} \leftarrow \emptyset$ ▷ lower-level beam
8 **for** $\mathrm{y} \in V$ **do**
9 $p' \leftarrow p_k(\mathrm{y}|\mathbf{y}_{1:t-1}, \mathbf{x})$
10 $\underline{B}_{t,k}.\mathrm{add}(\langle \mathbf{y}_{1:t-1} \oplus \mathrm{y}, p \cdot p'\rangle)$
11 $\underline{B}_{t,k} \leftarrow \underline{B}_{t,k}.\mathrm{top}(Z)$
12 ▷ upper: beam synchronization
13 $D \leftarrow$ empty dictionary
14 **for** $k = 1, \cdots, K$ **do**
15 **for** $\langle \mathbf{y}, p\rangle \in \underline{B}_{t,k}$ **do**
16 **if** $\mathbf{y} \in D$ **then**
17 $D[\mathbf{y}] \leftarrow p + D[\mathbf{y}]$
18 **else**
19 $D[\mathbf{y}] \leftarrow p$
20 $\overline{B}_t \leftarrow D.\mathrm{top}(Z)$
21 ▷ check for termination
22 **for** $\langle \mathbf{y}, p\rangle \in \underline{B}_t$ **do**
23 **if** $\mathrm{y}_t = \text{EOS}$ **then**
24 $H.\mathrm{add}(\langle \mathbf{y}, p\rangle)$
25 **if** $|H| = Z$ **then**
26 **return** $H.\mathrm{top}(1)$

the score for beam search, which has been experimented in previous work (Sennrich, Haddow, and Birch 2016a; Shanbhogue et al. 2023).

However, our intuition suggests that such an approach may suffer from the *over-smoothing problem* (Wei et al. 2019; Wen et al. 2023b): when multiple translations (known as *modes*) are plausible given an input, the ensemble process will overly smooth out the modes by probability averaging.

By contrast, EBBS allows each ensemble component to explore its own mode (Lines 4–11, Algorithm 1). In Figure 1, for example, the top sequence yields two plausible next tokens, suggested by each component in the lower level; their probabilities are not smoothed out in our approach, unlike averaging ensembles. The upper level performs soft voting (Lines 12–19, Algorithm 1) so as to maintain tractable inference.

3.3 EBBS-Based Distillation

To improve inference efficiency, we perform knowledge distillation based on the outputs of our EBBS algorithm. In particular, we follow (Kim and Rush 2016) and apply a sequence-level knowledge distillation loss, treating the output $\hat{\mathbf{y}}$ of our ensemble (serving as a *teacher*) as the pseudo-groundtruth for finetuning the multilingual translation model (serving as a *student*):

$$\mathcal{L}_{\text{KD}} = -\sum_{t=1}^{|\hat{\mathbf{y}}|} \log p(\hat{\mathrm{y}}_t | \hat{\mathbf{y}}_{1:t-1}, \mathbf{x}) \tag{4}$$

Our distilling method is an ensemble-then-distill process. This differs from a straightforward practice of multi-teacher distillation, where the student learns from the union of teachers' outputs (Wu, Wu, and Huang 2021). The commonly applied cross-entropy loss is known to yield overly smooth distributions (Wen et al. 2023a,b), and the problem becomes more severe with multiple teachers, leading to less satisfactory performance of union distillation (Shayegh et al. 2024). On the contrary, our approach provides the student with a consolidated pseudo-groundtruth translation, causing less confusion during the distillation process especially when teachers disagree.

4 Experiments

4.1 Settings

We evaluated EBBS on two popular benchmark datasets for zero-shot machine translation: IWSLT (Cettolo et al. 2017), which contains 4 languages (with English) and 6 zero-shot directions; and Europarl v7 (Koehn 2005), which contains 9 languages and 56 zero-shot directions.

We used BLEU scores (Papineni et al. 2002) (in particular, SacreBLEU (Post 2018)) as our main evaluation metric,[2] which is one of the most widely used metrics for translation (Fan et al. 2021; Scao et al. 2022). For in-depth analyses, we further adopted other popular translation metrics, including the character-level n-gram F score (chrF2++) (Popović 2017), the translation edit rate (TER) (Snover et al. 2006), and a more recent, neural network-based metric called COMET (Rei et al. 2020).

We replicated (Liu et al. 2021) and trained a multilingual translation system as our base model. Specifically, the neural architecture in (Liu et al. 2021) is a 5-layer encoder–decoder Transformer for IWSLT, but has 8 layers for Europarl to accommodate more training data and languages.

For EBBS, we used a beam size of five for both upper- and lower-level beams. In our experiment, we implemented standard beam search for comparison, where we also used a beam size of five, following the common practice (Meister, Cotterell, and Vieira 2020). A comprehensive beam analysis can be found in our appendix.

4.2 Competing Methods

We comprehensively compare our EBBS with direct/pivot translation and other ensemble methods.

Direct/pivot translation. For direct translation, we applied beam search on the multilingual model to translate

[2]We use BLEUn to denote the n-gram overlap and BLEU to denote the brevity-penalized geometric mean of BLEUn for $n = 1, \cdots, 4$. The exact evaluation scripts are available in our codebase (Footnote 1).

in unseen directions. For pivot translation (Wu and Wang 2007, 2009; Vamvas and Sennrich 2022), we used English as the pivot because we have parallel data for translations both from and to English.

Word-level averaging ensemble. Averaging is one of the most widely used ensemble techniques in text generation (Sennrich, Haddow, and Birch 2016a; Freitag, Al-Onaizan, and Sankaran 2017; Shanbhogue et al. 2023). Essentially, the ensemble components' probabilities are first averaged before being fed to the standard beam search.

Word-level voting ensemble. The voting ensemble, common in classification tasks, picks the output class based on the number of votes from ensemble components (given by argmax). However, voting is not common in text generation, because argmax may select completely different words by the ensemble components due to the large vocabulary size, making voting ineffective. As a remedy, we pick the word by the highest probability when there is a tie for votes.

Sequence-level voting ensemble. Minimum Bayes risk (MBR) decoding is originally designed as a single-model decoding algorithm, where it selects a sequence from a set of beam search results based on similarity (Eikema and Aziz 2020; Müller and Sennrich 2021). Here, we use it as a sequence-level ensemble technique, where the candidates are the output sequences from different ensemble components. Let $C = \{\mathbf{y}_1, \cdots, \mathbf{y}_K\}$ be the set of candidate outputs given by K ensemble components. The best output is selected as

$$\mathbf{y}^* = \operatorname*{argmax}_{\mathbf{y} \in C} \sum_{\mathbf{y}' \in C \setminus \{\mathbf{y}\}} \mathrm{BLEU}(\mathbf{y}, \mathbf{y}') \tag{5}$$

where $\mathrm{BLEU}(\mathbf{h}, \mathbf{r})$ computes the BLEU score between a hypothesis $\mathbf{h}$ and a reference $\mathbf{r}$. In essence, MBR selects an output that resembles others most, using BLEU as the similarity metric.

4.3 Results and Analysis

Main results. Our experiment starts by a replication of the base multilingual model (Liu et al. 2021). As shown in Rows 1–2, Table 1, the results are generally close, indicating that our replication is successful and ready for ensemble research. Further, we tried English pivoting (Row 3), a common zero-shot translation method. In our experiments, we find that it does not outperform direct translation, as pivoting methods may suffer from the error accumulation problem due to two-step translation.

We then compare different ensemble techniques, including our proposed EBBS. We notice that IWSLT contains four languages (with English); thus we have two available pivoting directions (excluding source and target), which, along with direct translation, are our three ensemble components. For Europarl, it contains nine languages; for performance and efficiency concerns (to be shown in Figure 2), we also consider three translation paths as our ensemble components: direction translation, English pivoting, and a second pivot.[3]

We study the common ensemble technique of word-level averaging (Row 4), which has been used in previous translation research (Freitag, Al-Onaizan, and Sankaran 2017). As we can see, the averaging ensemble performs similarly to direct translation on both datasets. Our zero-shot results are different from (Freitag, Al-Onaizan, and Sankaran 2017), which shows a word-level averaging ensemble of random seeds can improve performance in the supervised setting. This is because models trained with different random seeds exhibit similar behavior, and averaging their probabilities achieves a denoising effect. However, our ensemble components differ drastically in terms of their strengths and expertise due to the different translation paths (direct and pivot translations). Thus, word averaging fails to improve translation quality in our setting.

Alternatively, voting ensembles can also be applied, at either the word level or the sequence level. As seen, word-level voting is not effective, as it is worse than direct translation on both datasets (Row 5). This is expected because the voted words (top predictions) by the ensemble components may not overlap due to the large vocabulary size. In such cases, the algorithm defaults to choosing the word with the highest probability, causing the ensemble to follow the most peaked distributions.

Sequence-level voting should also be done in a soft manner, and minimum Bayes risk (MBR) decoding can be thought of as using a Bayes risk to softly "vote" the candidate outputs. As seen from Row 6, such a method works relatively well on Europarl, achieving the second-highest performance across all ensemble methods; however, it works poorly on the IWSLT dataset. The main drawback of sequence-level voting is that it can only *select* one of the ensemble components' output. This may not work well when the individual ensemble components are weak, especially with the small IWSLT dataset. Such a selective sequence-level ensemble cannot integrate different expertise of its components during generation.

Unlike existing ensemble methods, our EBBS algorithm achieves higher performance in most directions on both datasets. Noticing that Europarl contains 56 zero-shot directions, we could only present in Table 1 the first seven directions based on the order provided by the dataset, due to the space limit. Table 2 further shows a pairwise comparison against direct translation (a strong baseline in our experiment) in all zero-shot directions. As seen, EBBS achieves higher performance in 56 out of 62 cases across two datasets, showing strong statistical evidence for its effectiveness, with a p-value of 3e-11 in a two-sided binomial test.

We also evaluate EBBS-based distillation (Row 8, Table 1). Again, since Europarl has 56 zero-shot directions, we follow the standard practice (Fan et al. 2021) and select a subset of directions, namely, Danish to other languages, to save computational cost. As seen in Row 8, EBBS-based distillation consistently achieves the highest performance in

[3]We use the first available language in the order of Spanish, German, and French. For example, Spanish-to-German translation will have to use French as the pivot. These languages are chosen because they have the most content on the Internet according to the Web Technology Surveys (https://w3techs.com/technologies/overview/content_language).

Dataset	#	Method	Average	it-nl	it-ro	nl-it	nl-ro	ro-it	ro-nl	
IWSLT	1	Direct translation (Liu et al. 2021)†	17.7	18.5	17.8	17.9	15.5	19.6	16.8	
	2	Direct translation (our replication)	17.29	17.46	17.48	18.23	14.63	19.65	16.26	
	3	Pivoting (en)	16.19	17.49	15.09	16.79	13.05	18.34	16.37	
	4	Word-level averaging ensemble	17.28	17.29	17.44	18.33	14.65	19.69	16.30	
	5	Word-level voting ensemble	16.99	17.58	16.38	17.78	14.13	19.21	16.84	
	6	Sequence-level voting ensemble (MBR)	16.72	16.64	16.53	17.83	13.74	19.48	16.08	
	7	EBBS (ours)	18.24	19.52	17.09	19.06	14.58	20.75	18.45	
	8	Direct w/ EBBS distillation (ours)	**18.92**	**19.86**	**18.80**	**19.73**	**15.39**	**21.23**	**18.48**	
	#	Method	Average	da-de	da-es	da-fi	da-fr	da-it	da-nl	da-pt
Europarl	1	Direct translation (Liu et al. 2021)†	26.9	24.2	33.1	18.1	30.6	26.1	26.3	29.9
	2	Direct translation (our replication)	27.74	26.24	33.64	18.95	31.01	26.58	27.36	30.38
	3	Pivoting (en)	27.69	25.17	33.87	18.70	31.44	27.12	26.75	30.79
	4	Word-level averaging ensemble	27.76	26.13	33.72	18.91	31.01	26.67	27.39	30.50
	5	Word-level voting ensemble	27.45	25.76	33.24	18.39	30.96	26.83	26.63	30.37
	6	Sequence-level voting ensemble (MBR)	27.90	25.90	33.95	19.15	31.50	27.15	27.09	30.55
	7	EBBS (ours)	28.36	26.32	34.28	19.43	31.97	27.67	**27.78**	31.08
	8	Direct w/ EBBS distillation (ours)	**28.54**	**26.75**	**34.68**	**19.89**	**32.00**	**27.69**	27.61	**31.19**

Table 1: Main results of BLEU scores on IWSLT and Europarl. The best results are in **bold**; the second best results are underlined. † indicates cited results; others were obtained by our experimentation.

Dataset	Method	Avg. BLEU	Wins	Losses
IWSLT	Direct translation	17.29	2	4
	EBBS (ours)	**18.24**	**4**	**2**
Europarl	Direct translation	27.85	4	52
	EBBS (ours)	**28.44**	**52**	**4**
Overall	Direct translation	26.83	6	56
	EBBS (ours)	**27.45**	**56**	**6**
p-value	3e-11			

Table 2: Pairwise comparison on all 62 zero-shot directions in both datasets. The p-value is given by a two-sided binomial test.

all directions (except for Danish-to-Dutch translation). This shows that an EBBS-distilled model can outperform EBBS, which is not surprising because learning can smooth out the noise of various heuristics (Deshmukh et al. 2021; Jolly et al. 2022), such as the ensemble algorithm in our scenario. Importantly, EBBS-based distillation achieves significantly higher translation quality with *no inference overhead* compared with direct translation.

Distillation analysis. We compare EBBS-based distillation with other distilling methods. Here, we only focus on Italian-to-Dutch[4] translation to save computational cost.

In particular, we consider two alternative distilling methods: direct and union distillation. Direct distillation finetunes the multilingual model with its own predictions based on direct translation. Union distillation, on the other hand, takes the union of the teachers' outputs (direct and pivot translations) for training, which is under a controlled experimental setup, because it uses exactly the same translation paths as our EBBS-based distillation.

As seen in Table 3, both direct and union distillation marginally improve the performance compared with no distillation. Intriguingly, learning from the union of multiple teachers is not necessarily better than learning from the best teacher (namely, direct translation). This is because multiple teachers may provide conflicting training signals and confuse the student model.

On the contrary, our EBBS-based distillation consistently outperforms direct and union distillation on both datasets. This shows that our ensemble-then-distill approach is able to consolidate the knowledge of multiple teachers to better train the student model.

Further, the analysis suggests that our EBBS-distilled model achieves a speedup of multiple times compared with EBBS, because after distillation the model is used by direct translation. This is a significant result, because our EBBS-based distillation not only speeds up the EBBS ensemble approach, but also improves the translation quality of EBBS as shown in Row 8, Table 1.

Analysis of ensemble components. In Table 4, we analyze the ensemble components to better understand our ensemble technique for zero-shot machine translation. As seen, direct translation is an effective approach, which is consistent with previous literature (Fan et al. 2021; Liu et al. 2021). English pivoting achieves higher performance for some metrics but lower for others; it is not conclusively better than direct translation, probably because of the error accumulation problem. Pivoting through non-English languages degrades the performance to a large extent because lacking supervision along the pivoting path leads to two steps of zero-shot translation. EBBS, on the other hand, combines the strengths of individual components and consistently outperforms them in all metrics.

We further study how EBBS performs with different num-

[4]We could only afford one translation direction for this analysis, because we need to train different models for all competing distilling methods. This differs from Table 1, where we follow previous work and perform EBBS-based distillation for Danish to other languages. We chose Italian-to-Dutch translation here, because it is the first one in IWSLT, conveniently also available in Europarl.

Dataset	Method		BLEU$^\uparrow$	BLEU1$^\uparrow$	BLEU2$^\uparrow$	BLEU3$^\uparrow$	BLEU4$^\uparrow$	chrF2++$^\uparrow$	TER$^\downarrow$	COMET$^\uparrow$
IWSLT	EBBS		19.52	51.87	25.12	13.88	8.02	45.63	71.36	0.7341
	Direct Translation	No distillation	17.46	50.49	23.01	12.01	6.66	43.73	72.02	0.7088
		Direct distillation	18.10	50.37	23.53	12.63	7.17	44.48	72.86	0.7144
		Union distillation	17.80	49.21	23.01	12.51	7.10	44.93	75.92	0.7221
		EBBS distillation	**20.13**	**53.20**	**26.06**	**14.33**	**8.26**	**46.46**	**69.28**	**0.7428**
Europarl	EBBS		26.10	57.07	31.00	19.76	13.28	52.75	65.63	0.8340
	Direct Translation	No distillation	25.33	56.32	30.08	19.01	12.78	52.32	66.56	0.8276
		Direct distillation	25.44	56.54	30.28	19.13	12.79	52.61	66.34	0.8286
		Union distillation	25.53	56.58	30.34	19.18	12.91	52.63	66.27	0.8282
		EBBS distillation	**25.92**	**56.76**	**30.68**	**19.57**	**13.24**	**52.73**	**66.04**	**0.8307**

Table 3: Comparison of various distilling methods for Italian-to-Dutch translation. $^{\uparrow/\downarrow}$The higher/lower, the better.

Method	BLEU$^\uparrow$	chrF2++$^\uparrow$	TER$^\downarrow$	COMET$^\uparrow$
Direct translation	25.33	52.32	66.56	0.8276
Pivoting (en)	25.08	51.92	66.24	0.8322
Pivoting (es)	24.40	51.71	67.91	0.8192
Pivoting (pt)	24.34	51.61	67.68	0.8191
Pivoting (fr)	24.20	51.61	67.84	0.8208
Pivoting (de)	23.65	50.70	67.89	0.8157
Pivoting (da)	23.12	50.36	69.00	0.8156
Pivoting (fi)	20.74	48.11	70.59	0.8051
Our EBBS	**26.10**	**52.75**	**65.63**	**0.8340**

Table 4: The performance of direct/pivot translation and our EBBS for Italian-to-Dutch translation on Europarl.

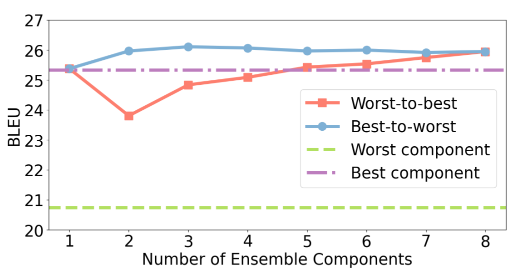

Figure 2: Analysis of the number of ensemble components for Italian-to-Dutch translation on Europarl.

bers of ensemble components. Specifically, we analyze two incremental ensemble settings: best-to-worst and worst-to-best. In both cases, we start with direct translation; then we incrementally add the next "best" or "worst" pivot translation according to Table 4.

Figure 2 shows the trends of incremental ensembles. If we add the best pivot directions, the performance peaks at three ensemble components; interestingly, the inclusion of weaker components does not affect EBBS much. On the other hand, adding the worst pivot translation at the beginning leads to an immediate drop of 1.6 BLEU points, which then largely recovers with the second pivot. This is reasonable because the worst pivot (Finnish) is 4.6 BLEU points lower than direct translation, and EBBS cannot decide on which of the two ensemble components to trust; despite this, the performance of EBBS is still much better than the average performance of the components. With a second pivot, there is a third "opinion" when the first two components "disagree." The performance continues to rise if more and stronger components are added. In fact, our ensemble even surpasses the baseline with 4 weakest pivot translations, each of which is at least 1 BLEU point lower than the baseline. This demonstrates that EBBS is flexible and works well with both strong and weak ensemble components.

Appendix. The full version of the paper is available at https://arxiv.org/abs/2403.00144, where we present additional details and results in the appendix:

A. Beam search,
B. Experimental details,
C. Analysis of inference efficiency,
D. Average performance across tasks,
E. Analysis of beam size,
F. Entropy of distilled models,
G. Analysis of voting methods in EBBS, and
H. Case study.

5 Conclusion

In this work, we address ensemble-based zero-shot machine translation by directly translating and pivoting through different languages. We further design a novel bi-level beam search algorithm (called EBBS) for decoding. We evaluated EBBS on two popular zero-shot translation datasets, IWSLT and Europarl. Results show that EBBS outperforms existing ensemble techniques, and that the high-quality translations produced by EBBS can be used for distillation to improve translation efficiency (and sometimes also output quality).

Acknowledgments

The research is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), a Mitacs Accelerate project, the Amii Fellow Program, the Canada CIFAR AI Chair Program, an Alberta Innovates Program, and the Digital Research Alliance of Canada (alliancecan.ca). We used the open-domain artwork "The Great Wave off Kanagawa" by Katsushika Hokusai as the background for Figure 1.

References

Babych, B.; Hartley, A.; and Sharoff, S. 2007. Translating from under-resourced languages: Comparing direct transfer against pivot translation. In *MTSummit*.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Baziotis, C.; Artetxe, M.; Cross, J.; and Bhosale, S. 2022. Multilingual machine translation with hyper-adapters. In *EMNLP*, 1170–1185.

Bickel, P. J.; and Doksum, K. A. 2015. *Mathematical Statistics: Basic Ideas and Selected Topics*. CRC Press.

Breiman, L. 1996. Bagging predictors. *Machine Learning*, 24: 123–140.

Brown, P. F.; et al. 1990. A statistical approach to machine translation. *CL*, 16(2): 79–85.

Bühlmann, P.; and Yu, B. 2002. Analyzing bagging. *The Annals of Statistics*, 30(4): 927–961.

Cettolo, M.; et al. 2017. Overview of the IWSLT 2017 evaluation campaign. In *IWSLT*, 2–14.

Chronopoulou, A.; Stojanovski, D.; and Fraser, A. 2023. Language-family adapters for low-resource multilingual neural machine translation. In *Proceedings of the Workshop on Technologies for Machine Translation of Low-Resource Languages*, 59–72.

Dabre, R.; Chu, C.; and Kunchukuttan, A. 2020. A survey of multilingual neural machine translation. *ACM Computing Surveys*, 53(5).

Deshmukh, A. A.; Zhang, Q.; Li, M.; Lin, J.; and Mou, L. 2021. Unsupervised chunking as syntactic structure induction with a knowledge-transfer approach. In *EMNLP Findings*, 3626–3634.

Dong, X.; Yu, Z.; Cao, W.; Shi, Y.; and Ma, Q. 2020. A survey on ensemble learning. *Frontiers of Computer Science*, 14: 241–258.

Dugast, L.; Senellart, J.; and Koehn, P. 2007. Statistical post-editing on SYSTRAN's rule-based translation system. In *WMT*, 220–223.

Eikema, B.; and Aziz, W. 2020. Is MAP decoding all you need? The inadequacy of the mode in neural machine translation. In *COLING*, 4506–4520.

Fan, A.; et al. 2021. Beyond English-centric multilingual machine translation. *JMLR*, 22(107): 1–48.

Forcada, M. L.; et al. 2011. Apertium: A free/open-source platform for rule-based machine translation. *Machine Translation*, 25: 127–144.

Freitag, M.; Al-Onaizan, Y.; and Sankaran, B. 2017. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*.

Gaikwad, P.; Doshi, M.; Dabre, R.; and Bhattacharyya, P. 2024. How effective is multi-source pivoting for translation of low resource Indian languages? *arXiv preprint arXiv:2406.13332*.

Ganaie, M.; Hu, M.; Malik, A.; Tanveer, M.; and Suganthan, P. 2022. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115: 105151.

Gu, J.; Wang, Y.; Cho, K.; and Li, V. O. 2019. Improved zero-shot neural machine translation via ignoring spurious Correlations. In *ACL*, 1258–1268.

Hastie, T.; Rosset, S.; Zhu, J.; and Zou, H. 2009. Multi-class AdaBoost. *Statistics and Its Interface*, 2(3): 349–360.

He, D.; et al. 2023. Gradient-based gradual pruning for language-specific multilingual neural machine translation. In *EMNLP*, 654–670.

Hendy, A.; et al. 2021. Ensembling of distilled models from multi-task teachers for constrained resource language pairs. In *WMT*, 130–135.

Johnson, M.; et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*, 5: 339–351.

Jolly, S.; Zhang, Z. X.; Dengel, A.; and Mou, L. 2022. Search and learn: Improving semantic coverage for data-to-text generation. In *AAAI*, 10858–10866.

Kim, Y.; and Rush, A. M. 2016. Sequence-level knowledge distillation. In *EMNLP*, 1317–1327.

Kobayashi, H. 2018. Frustratingly easy model ensemble for abstractive summarization. In *EMNLP*, 4165–4176.

Koehn, P. 2005. Europarl: A parallel corpus for statistical machine translation. In *MTSummit*, 79–86.

Koehn, P. 2009. *Statistical Machine Translation*. Cambridge University Press.

Lample, G.; Conneau, A.; Denoyer, L.; and Ranzato, M. 2018a. Unsupervised machine translation using monolingual corpora only. In *ICLR*.

Lample, G.; Ott, M.; Conneau, A.; Denoyer, L.; and Ranzato, M. 2018b. Phrase-based & neural unsupervised machine translation. In *EMNLP*, 5039–5049.

Lewis, M.; et al. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 7871–7880.

Liu, D.; Niehues, J.; Cross, J.; Guzmán, F.; and Li, X. 2021. Improving zero-shot translation by disentangling positional information. In *ACL-IJCNLP*, 1259–1273.

Meister, C.; Cotterell, R.; and Vieira, T. 2020. If beam search is the answer, what was the question? In *EMNLP*, 2173–2185.

Mohammadshahi, A.; Vamvas, J.; and Sennrich, R. 2024. Investigating multi-pivot ensembling with massively multilingual machine translation models. In *Proceedings of the Workshop on Insights from Negative Results in NLP*, 169–180.

Mohiuddin, T.; and Joty, S. 2020. Unsupervised word translation with adversarial autoencoder. *CL*, 46(2): 257–288.

Muennighoff, N.; et al. 2023. Scaling data-constrained language models. In *NeurIPS*, 50358–50376.

Müller, M.; and Sennrich, R. 2021. Understanding the properties of minimum Bayes risk decoding in neural machine translation. In *ACL-IJCNLP*, 259–272.

Natekin, A.; and Knoll, A. 2013. Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, 7: 1–21.

Ott, M.; et al. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *NAACL-HLT: Demonstrations*, 48–53.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: A method for automatic evaluation of machine translation. In *ACL*, 311–318.

Popović, M. 2017. chrF++: Words helping character n-grams. In *WMT*, 612–618.

Post, M. 2018. A call for clarity in reporting BLEU scores. In *WMT*, 186–191.

Radford, A.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Raffel, C.; et al. 2020. Exploring the limits of transfer learning with a unified text-to-text Transformer. *JMLR*, 21(140): 1–67.

Ranathunga, S.; Lee, E.-S. A.; Prifti Skenduli, M.; Shekhar, R.; Alam, M.; and Kaur, R. 2023. Neural machine translation for low-resource languages: A survey. *ACM Computing Survey*, 55(11).

Razmara, M.; and Sarkar, A. 2013. Ensemble triangulation for statistical machine translation. In *IJCNLP*, 252–260.

Rei, R.; Stewart, C.; Farinha, A. C.; and Lavie, A. 2020. COMET: A neural framework for MT evaluation. In *EMNLP*, 2685–2702.

Scao, T. L.; et al. 2022. BLOOM: A 176B-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Schapire, R. E. 2003. The boosting approach to machine learning: An overview. *Nonlinear Estimation and Classification*, 149–171.

Sennrich, R.; Haddow, B.; and Birch, A. 2016a. Edinburgh neural machine translation systems for WMT 16. In *WMT*, 371–376.

Sennrich, R.; Haddow, B.; and Birch, A. 2016b. Neural machine translation of rare words with subword units. In *ACL*, 1715–1725.

Shanbhogue, A. V. K.; Xue, R.; Saha, S.; Zhang, D.; and Ganesan, A. 2023. Improving low resource speech translation with data augmentation and ensemble strategies. In *IWSLT*, 241–250.

Shayegh, B.; Cao, Y.; Zhu, X.; Cheung, J. C.; and Mou, L. 2024. Ensemble distillation for unsupervised constituency parsing. In *ICLR*.

Shayegh, B.; Wen, Y.; and Mou, L. 2024. Tree-averaging algorithms for ensemble-based unsupervised discontinuous constituency parsing. In *ACL*, 15135–15156.

Shayegh, B.; et al. 2025. Error diversity matters: An error-resistant ensemble method for unsupervised dependency parsing. In *AAAI*.

Snover, M.; Dorr, B.; Schwartz, R.; Micciulla, L.; and Makhoul, J. 2006. A study of translation edit rate with targeted human annotation. In *Association for Machine Translation in the Americas*, 223–231.

Vamvas, J.; and Sennrich, R. 2022. NMTScore: A multilingual analysis of translation-based text similarity measures. In *EMNLP Findings*, 198–213.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

Wang, Q.; and Zhang, J. 2022. Parameter differentiation based multilingual neural machine translation. In *AAAI*, 11440–11448.

Wang, Z.; Lipton, Z. C.; and Tsvetkov, Y. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In *EMNLP*, 4438–4450.

Wei, B.; Lu, S.; Mou, L.; Zhou, H.; Poupart, P.; Li, G.; and Jin, Z. 2019. Why do neural dialog systems generate short and meaningless replies? A comparison between dialog and translation. In *ICASSP*, 7290–7294.

Wen, Y.; Hao, Y.; Cao, Y.; and Mou, L. 2023a. An equal-size hard EM algorithm for diverse dialogue generation. In *ICLR*.

Wen, Y.; Li, Z.; Du, W.; and Mou, L. 2023b. f-divergence minimization for sequence-level knowledge distillation. In *ACL*, 10817–10834.

Wicks, R.; and Duh, K. 2022. The effects of language Token prefixing for multilingual machine translation. In *AACL-IJCNLP*, 148–153.

Wolf, T.; et al. 2019. Huggingface's Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Wolpert, D. H. 1992. Stacked generalization. *Neural Networks*, 5(2): 241–259.

Wu, C.; Wu, F.; and Huang, Y. 2021. One teacher is enough? Pre-trained language model distillation from multiple teachers. In *ACL-IJCNLP Findings*, 4408–4413.

Wu, H.; and Wang, H. 2007. Pivot language approach for phrase-based statistical machine translation. In *ACL*, 856–863.

Wu, H.; and Wang, H. 2009. Revisiting pivot language approach for machine translation. In *ACL-IJCNLP*, 154–162.

Yang, Y.; Lv, H.; and Chen, N. 2023. A survey on ensemble learning under the era of deep learning. *Artificial Intelligence Review*, 56(6): 5545–5589.

Zaremoodi, P.; Buntine, W.; and Haffari, G. 2018. Adaptive knowledge sharing in multi-task learning: Improving low-resource neural machine translation. In *ACL*, 656–661.

Zhang, B.; Williams, P.; Titov, I.; and Sennrich, R. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *ACL*, 1628–1639.