Beyond Weight Tying: Learning Joint Input-Output Embeddings for Neural Machine Translation

Nikolaos Pappas[†] **Lesly Miculicich Werlen**[†◇] **James Henderson**[†]
[†]Idiap Research Institute, Martigny, Switzerland
[◇]École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
{npappas,lmiculicich,jhenderson}@idiap.ch

Abstract

Tying the weights of the target word embeddings with the target word classifiers of neural machine translation models leads to faster training and often to better translation quality. Given the success of this parameter sharing, we investigate other forms of sharing in between no sharing and hard equality of parameters. In particular, we propose a *structure-aware* output layer which captures the semantic structure of the output space of words within a joint input-output embedding. The model is a generalized form of *weight tying* which shares parameters but allows learning a more flexible relationship with input word embeddings and allows the effective capacity of the output layer to be controlled. In addition, the model shares weights across output classifiers and translation contexts which allows it to better leverage prior knowledge about them. Our evaluation on English-to-Finnish and English-to-German datasets shows the effectiveness of the method against strong encoder-decoder baselines trained with or without *weight tying*.

1 Introduction

Neural machine translation (NMT) predicts the target sentence one word at a time, and thus models the task as a sequence classification problem where the classes correspond to words. Typically, words are treated as categorical variables which lack description and semantics. This makes training speed and parametrization dependent on the size of the target vocabulary (Mikolov et al., 2013). Previous studies overcome this problem by truncating the vocabulary to limit its size and mapping out-of-vocabulary words to a single "unknown" token. Other approaches attempt to use a limited number of frequent words plus *sub-word units* (Sennrich et al., 2016), the combination of which can cover the full vocabulary, or to perform character-level modeling (Chung et al., 2016; Lee et al., 2017; Costa-jussà and Fonollosa, 2016; Ling et al., 2015); with the former being the most effective between the two. The idea behind these alternatives is to overcome the vocabulary size issue by modeling the morphology of rare words. One limitation, however, is that semantic information of words or sub-word units learned by the input embedding are not considered when learning to predict output words. Hence, they rely on a large amount of examples per class to learn proper word or sub-word unit output classifiers.

One way to consider information learned by input embeddings, albeit restrictively, is with *weight tying* i.e. sharing the parameters of the input embeddings with those of the output classifiers (Press and Wolf, 2017; Inan et al., 2016) which is effective for language modeling and machine translation (Sennrich et al., 2017; Klein et al., 2017). Despite its usefulness, we find that *weight tying* has three limitations: (a) It biases all the words with similar input embeddings to have a similar chance to be generated, which may not always be the case (see Table 1 for examples). Ideally, it would be better to learn distinct relationships useful for encoding and decoding without forcing any general bias. (b) The relationship between outputs is only implicitly captured by *weight tying* because there is no parameter sharing across output classifiers. (c) It requires that the size of the translation context vector and the input embeddings are the same, which in practice makes it difficult to control the output layer capacity.

In this study, we propose a *structure-aware* output layer which overcomes the limitations of previous output layers of NMT models. To achieve this, we treat words and subwords as units with textual descriptions and semantics. The model consists of a joint input-output embedding which learns what to share between input embeddings

Proceedings of the Third Conference on Machine Translation (WMT), Volume 1: Research Papers, pages 73–83

https://doi.org/10.18653/v1/W18-64008

	NMT		NMT-tied	NMT-joint	
Query	**Input**	**Output**	**Input/Output**	**Input**	**Output**
visited	attacked	visiting	visits	visiting	attended
(Verb past tense)	conquered	attended	attended	attended	witnessed
	contacted	visit	visiting	visits	discussed
	occupied	visits	frequented	visit	recognized
	consulted	discovered	visit	frequented	demonstrated
generous	modest	spacious	generosity	spacious	friendly
(Adjective)	extensive	generosity	spacious	generosity	flexible
	substantial	generously	generously	flexible	brilliant
	ambitious	massive	lavish	generously	fantastic
	sumptuous	huge	massive	massive	massive
friend	wife	friends	colleague	colleague	colleague
(Noun)	husband	colleague	friends	friends	fellow
	colleague	Fri@@	neighbour	neighbour	supporter
	friends	fellow	girlfriend	girlfriend	partner
	painter	friendship	companion	husband	manager

Table 1: Top-5 most similar input and output representations to two query words based on cosine similarity for an NMT trained without (NMT) or with *weight tying* (NMT-tied) and our *structure-aware* output layer (NMT-joint) on De-En ($|\mathcal{V}| \approx 32K$). Our model learns representations useful for encoding and generation which are more consistent to the dominant semantic and syntactic relations of the query such as verbs in past tense, adjectives and nouns (inconsistent words are marked in red).

and output classifiers, but also shares parameters across output classifiers and translation contexts to better capture the similarity structure of the output space and leverage prior knowledge about this similarity. This flexible sharing allows it to distinguish between features of words which are useful for encoding, generating, or both. Figure 1 shows examples of the proposed model's input and output representations, compared to those of a softmax linear unit with or without *weight tying*.

This proposal is inspired by joint input-output models for zero-shot text classification (Yazdani and Henderson, 2015; Nam et al., 2016a), but innovates in three important directions, namely in learning complex non-linear relationships, controlling the effective capacity of the output layer and handling structured prediction problems.

Our contributions are summarized as follows:

- We identify key theoretical and practical limitations of existing output layer parametrizations such as softmax linear units with or without *weight tying* and relate the latter to joint input-output models.
- We propose a novel *structure-aware* output layer which has flexible parametrization for neural MT and demonstrate that its mathematical form is a generalization of existing output layer parametrizations.
- We provide empirical evidence of the superiority of the proposed structure-aware output layer on morphologically simple and complex languages as targets, including under challenging conditions, namely varying vocabulary sizes, architecture depth, and output frequency.

The evaluation is performed on 4 translation pairs, namely English-German and English-Finnish in both directions using BPE (Sennrich et al., 2016) of varying operations to investigate the effect of the vocabulary size to each model. The main baseline is a strong LSTM encoder-decoder model with 2 layers on each side (4 layers) trained with or without *weight tying* on the target side, but we also experiment with deeper models with up to 4 layers on each side (8 layers). To improve efficiency on large vocabulary sizes we make use of negative sampling as in (Mikolov et al., 2013) and show that the proposed model is the most robust to such approximate training among the alternatives.

2 Background: Neural MT

The translation objective is to maximize the conditional probability of emitting a sentence in a

target language $Y = \{y_1, ..., y_n\}$ given a sentence in a source language $X = \{x_1, ..., x_m\}$, noted $p_\Theta(Y|X)$, where Θ are the model parameters learned from a parallel corpus of length N:

$$\max_\Theta \frac{1}{N} \sum_{i=1}^{N} \log(p_\Theta(Y^{(i)}|X^{(i)})). \quad (1)$$

By applying the chain rule, the output sequence can be generated one word at a time by calculating the following conditional distribution:

$$p(y_t|y_1^{t-1}, X) \approx f_\Theta(y_1^{t-1}, X). \quad (2)$$

where f_Θ returns a column vector with an element for each y_t. Different models have been proposed to approximate the function f_Θ (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015; Cho et al., 2014; Gehring et al., 2017; Vaswani et al., 2017). Without loss of generality, we focus here on LSTM-based encoder-decoder model with attention Luong et al. (2015).

2.1 Output Layer parametrizations

2.1.1 Softmax Linear Unit

The most common output layer (Figure 3a), consists of a linear unit with a weight matrix $W \in \mathbb{R}^{\mathrm{d_h} \times |\mathcal{V}|}$ and a bias vector $b \in \mathbb{R}^{|\mathcal{V}|}$ followed by a softmax activation function, where V is the vocabulary, noted as NMT. For brevity, we focus our analysis specifically on the nominator of the normalized exponential which characterizes softmax. Given the decoder's hidden representation h_t with dimension size d_h, the output probability distribution at a given time, y_t, conditioned on the input sentence X and the previously predicted outputs y_1^{t-1} can be written as follows:

$$p(y_t|y_1^{t-1}, X) \propto \exp(W^T h_t + b)$$
$$\propto \exp(W^T I h_t + b), \quad (3)$$

where I is the identity function. From the second line of the above equation, we observe that there is no explicit output space structure learned by the model because there is no parameter sharing across outputs; the parameters for output class i, W_i^T, are independent from parameters for any other output class j, W_j^T.

2.1.2 Softmax Linear Unit with *Weight Tying*

The parameters of the output embedding W can be tied with the parameters of the input embedding $E \in \mathbb{R}^{|\mathcal{V}| \times \mathrm{d}}$ by setting $W = E^T$, noted as NMT-`tied`. This can happen only when the input dimension of W is restricted to be the same as that of the input embedding ($d = d_h$). This creates practical limitations because the optimal dimensions of the input embedding and translation context may actually be when $d_h \neq d$.

With tied embeddings, the parametrization of the conditional output probability distribution from Eq. 3 can be re-written as:

$$p(y_t|y_1^{t-1}, X) \propto \exp((E^T)^T h_t + b)$$
$$\propto \exp(E h_t + b). \quad (4)$$

As above, this model does not capture any explicit output space structure. However, previous studies have shown that the input embedding learns linear relationships between words similar to distributional methods (Mikolov et al., 2013). The hard equality of parameters imposed by $W = E^T$ forces the model to re-use this implicit structure in the output layer and increases the modeling burden of the decoder itself by requiring it to match this structure through h_t. Assuming that the latent linear structure which E learns is of the form $E \approx E_l \mathcal{W}$ where $E_l \in \mathbb{R}^{|\mathcal{V}| \times \mathrm{k}}$ and $\mathcal{W} \in \mathbb{R}^{\mathrm{k} \times \mathrm{d}}$ and $d = d_h$, then Eq. 4 becomes:

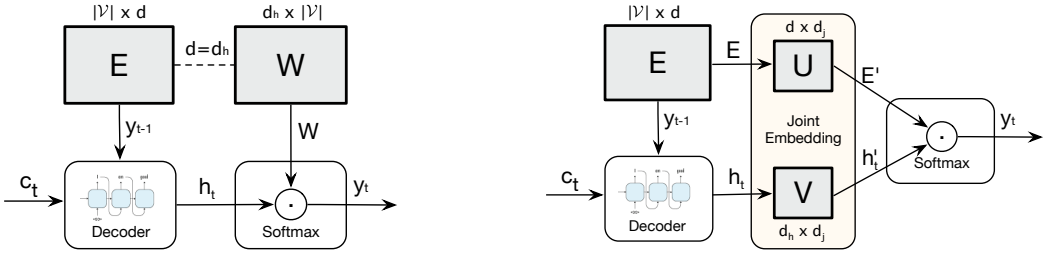
$$p(y_t|y_1^{t-1}, X) \propto exp(E_l \mathcal{W} h_t + b) \square. \quad (5)$$

The above form, excluding bias b, shows that *weight tying* learns a similar linear structure, albeit implicitly, to joint input-output embedding models with a bilinear form for zero-shot classification (Yazdani and Henderson, 2015; Nam et al., 2016a).[1] This may explain why *weight tying* is more sample efficient than the baseline softmax linear unit, but also motivates the learning of explicit structure through joint input-output models.

2.2 Challenges

We identify two key challenges of the existing parametrizations of the output layer: (a) their difficulty in learning complex structure of the output space due to their bilinear form and (b) their rigidness in controlling the output layer capacity due to their strict equality of the dimensionality of the translation context and the input embedding.

[1] The capturing of implicit structure could also apply for the output embedding W in Eq. 3, however that model would not match the bilinear input-output model form because it is based on the input embedding E.



(a) Typical output layer which is a softmax linear unit without or with *weight tying* ($W = E^T$).

(b) The *structure-aware* output layer is a joint embedding between translation contexts and word classifiers.

Figure 1: Schematic of existing output layers and the proposed output layer for the decoder of the NMT model with source context vector c_t, previous word $y_{t-1} \in \mathbb{R}^{\mathrm{d}}$, and decoder hidden states, $h_t \in \mathbb{R}^{\mathrm{d_h}}$.

2.2.1 Learning Complex Structure

The existing joint input-output embedding models (Yazdani and Henderson, 2015; Nam et al., 2016a) have the following bilinear form:

$$E \underbrace{\mathcal{W}}_{Structure} h_t \tag{6}$$

where $\mathcal{W} \in \mathbb{R}^{\mathrm{d} \times \mathrm{d_h}}$. We can observe that the above formula can only capture linear relationships between encoded text (h_t) and input embedding (E) through $\mathcal{W}$. We argue that for structured prediction, the relationships between different outputs are more complex due to complex interactions of the semantic and syntactic relations across outputs but also between outputs and different contexts. A more appropriate form for this purpose would include a non-linear transformation $\sigma(\cdot)$, for instance with either:

$$\text{(a)} \ \underbrace{\sigma(E\mathcal{W})}_{\text{Output structure}} h_t \quad \text{or} \quad \text{(b)} \ E \underbrace{\sigma(\mathcal{W}h_t)}_{\text{Context structure}} . \tag{7}$$

2.2.2 Controlling Effective Capacity

Given the above definitions we now turn our focus to a more practical challenge, which is the capacity of the output layer. Let Θ_{base}, Θ_{tied}, $\Theta_{bilinear}$ be the parameters associated with a softmax linear unit without and with *weight tying* and with a joint bilinear input-output embedding, respectively. The capacity of the output layer in terms of effective number of parameters can be expressed as:

$$\mathcal{C}_{base} \approx |\Theta_{base}| = |\mathcal{V}| \times d_h + |\mathcal{V}| \tag{8}$$

$$\mathcal{C}_{tied} \approx |\Theta_{tied}| \leq |\mathcal{V}| \times d_h + |\mathcal{V}| \tag{9}$$

$$\mathcal{C}_{bilinear} \approx |\Theta_{bilinear}| = d \times d_h + |\mathcal{V}|. \tag{10}$$

But since the parameters of Θ_{tied} are tied to the parameters of the input embedding, the effective number of parameters dedicated to the output layer is only $|\Theta_{tied}| = |\mathcal{V}|$.

The capacities above depend on *external* factors, that is $|\mathcal{V}|$, d and d_h, which affect not only the output layer parameters but also those of other parts of the network. In practice, for Θ_{base} the capacity d_h can be controlled with an additional linear projection on top of h_t (e.g. as in the OpenNMT implementation), but even in this case the parametrization would still be heavily dependent on $|\mathcal{V}|$. Thus, the following inequality for the effective capacity of these models holds true for fixed $|V|$, d, d_h:

$$\mathcal{C}_{tied} < \mathcal{C}_{bilinear} < \mathcal{C}_{base}. \tag{11}$$

This creates in practice difficulty in choosing the optimal capacity of the output layer which scales to large vocabularies and avoids underparametrization or overparametrization (left and right side of Eq. 11 respectively). Ideally, we would like to be able to choose the effective capacity of the output layer more flexibly moving freely in between $C_{bilinear}$ and C_{base} in Eq. 11.

3 Structure-aware Output Layer for Neural Machine Translation

The proposed *structure-aware* output layer for neural machine translation, noted as NMT-`joint`, aims to learn the structure of the output space by learning a joint embedding between translation contexts and output classifiers, as well as, by learning what to share with input embeddings (Figure 1b). In this section, we describe the model in detail, showing how it can be trained efficiently for arbitrarily high number of effective parameters and how it is related to weight tying.

3.1 Joint Input-Output Embedding

Let $g_{inp}(h_t)$ and $g_{out}(e_j)$ be two non-linear projections of d_j dimensions of any translation context h_t and any embedded output e_j, where e_j is the j_{th} row vector from the input embedding matrix E, which have the following form:

$$e'_j = g_{out}(e_j) = \sigma(U e_j^T + b_u) \quad (12)$$

$$h'_t = g_{inp}(h_t) = \sigma(V h_t + b_v), \quad (13)$$

where the matrix $U \in \mathbb{R}^{\mathrm{d_j} \times \mathrm{d}}$ and bias $b_u \in \mathbb{R}^{\mathrm{d_j}}$ is the linear projection of the translation context and the matrix $V \in \mathbb{R}^{\mathrm{d_j} \times \mathrm{d_h}}$ and bias $b_v \in \mathbb{R}^{\mathrm{d_j}}$ is the linear projection of the outputs, and σ is a non-linear activation function (here we use `Tanh`). Note that the projections could be high-rank or low-rank for h'_t and e'_j depending on their initial dimensions and the target joint space dimension.

With $E' \in \mathbb{R}^{|\mathcal{V}| \times \mathrm{d_j}}$ being the matrix resulting from projecting all the outputs e_j to the joint space, i.e. $g_{out}(E)$, and a vector $b \in \mathbb{R}^{|\mathcal{V}|}$ which captures the bias for each output, the conditional output probability distribution of Eq 3 can be re-written as follows:

$$\begin{aligned} & p(y_t | y_1^{t-1}, X) \quad (14) \\ & \propto \exp\big(E' h'_t + b\big) \\ & \propto \exp\big(g_{out}(E) g_{inp}(h_t) + b\big) \\ & \propto \exp\big(\sigma(U E^T + b_u)\, \sigma(V h_t + b_v) + b\big). \end{aligned}$$

3.1.1 What Kind of Structure is Captured?

From the above formula we can derive the general form of the joint space which is similar to Eq. 7 with the difference that it incorporates both components for learning output and context structure:

$$\underbrace{\sigma(E \mathcal{W}_o)}_{\text{Output structure}} \quad \underbrace{\sigma(\mathcal{W}_c h_t)}_{\text{Context structure}} , \quad (15)$$

where $\mathcal{W}_o \in \mathbb{R}^{\mathrm{d} \times \mathrm{d_j}}$ and $\mathcal{W}_c \in \mathbb{R}^{\mathrm{d_j} \times \mathrm{d_h}}$ are the dedicated projections for learning output and context structure respectively (which correspond to U and V projections in Eq. 14). We argue that both nonlinear components are essential and validate this hypothesis empirically in our evaluation by performing an ablation analysis (Section 4.4).

3.1.2 How to Control the Effective Capacity?

The capacity of the model in terms of effective number of parameters (Θ_{joint}) is:

$$\mathcal{C}_{joint} \approx |\Theta_{joint}| = d \times d_j + d_j \times d_h + |\mathcal{V}|. \quad (16)$$

By increasing the joint space dimension d_j above, we can now move freely between $\mathcal{C}_{bilinear}$ and $\mathcal{C}_{base}$ in Eq .11 without depending anymore on the external factors (d, d_h, $|V|$) as follows:

$$\mathcal{C}_{tied} < \mathcal{C}_{bilinear} \leq \mathcal{C}_{joint} \leq \mathcal{C}_{base}. \quad (17)$$

However, for very large number of d_j the computational complexity increases prohibitively because the projection requires a large matrix multiplication between U and E which depends on $|\mathcal{V}|$. In such cases, we resort to sampling-based training, as explained in the next subsection.

3.2 Sampling-based Training

To scale up to large output sets we adopt the negative sampling approach from (Mikolov et al., 2013). The goal is to utilize only a sub-set $\mathcal{V}'$ of the vocabulary instead of the whole vocabulary $\mathcal{V}$ for computing the softmax. The sub-set $\mathcal{V}'$ includes all positive classes whereas the negative classes are randomly sampled. During back propagation only the weights corresponding to the sub-set $\mathcal{V}'$ are updated. This can be trivially extended to mini-batch stochastic optimization methods by including all positive classes from the examples in the batch and sampling negative examples randomly from the rest of the vocabulary.

Given that the joint space models generalize well on seen or unseen outputs (Yazdani and Henderson, 2015; Nam et al., 2016b), we hypothesize that the proposed joint space will be more sample efficient than the baseline NMT with or without *weight tying*, which we empirically validate with a sampling-based experiment in Section 4.5 (Table 2, last three rows with $|\mathcal{V}| \approx 128K$).

3.3 Relation to *Weight Tying*

The proposed joint input-output space can be seen as a generalization of *weight tying* ($W = E^T$, Eq. 3), because its degenerate form is equivalent to *weight tying*. In particular, this can be simply derived if we set the non-linear projection functions in the second line of Eq. 14 to be the identity function, $g_{inp}(\cdot) = g_{out}(\cdot) = I$, as follows:

$$\begin{aligned} p(y_t | y_1^{t-1}, X) & \propto \exp\big((IE)\ (I h_t) + b\big) \\ & \propto \exp\big(E h_t + b\big)\ \square. \quad (18) \end{aligned}$$

Overall, this new parametrization of the output layer generalizes over previous ones and addresses their aforementioned challenges in Section 2.2.

		En → Fi		Fi → En		En → De		De → En	
	Model	$\|\Theta\|$	BLEU (Δ)	$\|\Theta\|$	BLEU (Δ)	$\|\Theta\|$	BLEU (Δ)	$\|\Theta\|$	BLEU (Δ)
32K	NMT	60.0M	12.68 (–)	59.8M	9.42 (–)	61.3M	18.46 (–)	65.0M	15.85 (–)
	NMT-tied	43.3M	12.58 (−0.10)	43.3M	9.59 (+0.17)	44.9M	18.48 (+0.0)	46.7M	16.51 (+0.66)†
	NMT-joint	47.5M	**13.03** (+0.35)‡	47.5M	**10.19** (+0.77)‡	47.0M	**19.79** (+1.3)‡	48.8M	**18.11** (+2.26)‡
64K	NMT	108.0M	13.32 (–)	106.7M	**12.29** (–)	113.9M	20.70 (–)	114.0M	20.01 (–)
	NMT-tied	75.0M	13.59 (+0.27)	75.0M	11.74 (−0.55)‡	79.4M	20.85 (+0.15)	79.4M	19.19 (−0.82)†
	NMT-joint	75.5M	**13.84** (+0.52)‡	75.5M	12.08 (−0.21)	79.9M	**21.62** (+0.92)‡	79.9M	**20.61** (+0.60)†
128K (∼)	NMT	201.1M	13.52 (–)	163.1M	11.64 (–)	211.3M	22.48 (–)	178.3M	19.12 (–)
	NMT-tied	135.6M	13.90 (+0.38)*	103.2M	11.97 (+0.33)*	144.2M	21.43 (−0.0)	111.6M	19.43 (+0.30)
	NMT-joint	137.7M	**13.93** (+0.41)†	103.7M	**12.07** (+0.43)†	146.3M	**22.73** (+0.25)†	115.8M	**20.60** (+1.48)‡

Table 2: Model performance and number of parameters ($|\Theta|$) with varying BPE operations (32K, 64K, 128K) on the English-Finish and English-German language pairs. The significance of the difference against the NMT baseline with p-values $<.05$, $<.01$ and $<.001$ are marked with *, † and ‡ respectively.

4 Evaluation

We compare the NMT-joint model to two strong NMT baselines trained with and without *weight tying* over four large parallel corpora which include morphologically rich languages as targets (Finnish and German), but also morphologically less rich languages as targets (English) from WMT 2017 (Bojar et al., 2017)[2]. We examine the behavior of the proposed model under challenging conditions, namely varying vocabulary sizes, architecture depth, and output frequency.

4.1 Datasets and Metrics

The English-Finnish corpus contains 2.5M sentence pairs for training, 1.3K for development (Newstest2015), and 3K for testing (Newstest2016), and the English-German corpus 5.8M for training, 3K for development (Newstest2014), and 3K for testing (Newstest2015). We preprocess the texts using the BPE algorithm (Sennrich et al., 2016) with 32K, 64K and 128K operations. Following the standard evaluation practices in the field (Bojar et al., 2017), the translation quality is measured using BLEU score (Papineni et al., 2002) (*multi-blue*) on *tokenized* text and the significance is measured with the paired bootstrap re-sampling method proposed by (Koehn et al., 2007).[3] The quality on infrequent words is measured with METEOR (Denkowski and Lavie, 2014) which has originally been proposed to measure performance on function words. To adapt it for our purposes on English-German pairs ($|\mathcal{V}| \approx 32K$), we set as *function words* different sets of words grouped according to three frequency bins, each of them containing $\frac{|\mathcal{V}|}{3}$ words of *high*, *medium* and *low* frequency respectively and set its parameters to $\{0.85, 0.2, 0.6, 0.\}$ and $\{0.95, 1.0, 0.55, 0.\}$ when evaluating on English and German respectively.

4.2 Model Configurations

The baseline is an encoder-decoder with 2 stacked LSTM layers on each side from OpenNMT (Klein et al., 2017), but we also experiment with varying depth in the range $\{1, 2, 4, 8\}$ for German-English. The hyperparameters are set according to validation accuracy as follows: maximum sentence length of 50, 512-dimensional word embeddings and LSTM hidden states, dropout with a probability of 0.3 after each layer, and Adam (Kingma and Ba, 2014) optimizer with initial learning rate of 0.001. The size of the joint space is also selected on validation data in the range $\{512, 2048, 4096\}$. For efficiency, all models on corpora with $\mathcal{V} \approx 128K$ (∼) and all *structure-aware* models with $d_j \geq 2048$ on corpora with $\mathcal{V} \leq 64K$ are trained with 25% negative sampling.[4]

4.3 Translation Performance

Table 2 displays the results on four translation sets from English-German and English-Finish language pairs when varying the number of BPE operations. The NMT-tied model outperforms the

[2] http://www.statmt.org/wmt17/

[3] multi-bleu.perl and bootstrap-hypothesis-difference-significance.pl scripts.

[4] Training the models with a full 128K vocabulary without sampling runs out of memory on our machines.

	Model	Layer form	BLEU	$\|\Theta\|$
	NMT	$W^T h_t$	15.85	65.0M
	NMT-tied	Eh_t	16.51	46.7M
NMT-joint	Eq. 6	$E\mathcal{W}h_t$	16.23	47.0M
NMT-joint	Eq. 7 a	$\sigma(E\mathcal{W})h_t$	16.01	47.0M
NMT-joint	Eq. 7 b	$E\sigma(\mathcal{W}h_t)$	17.52	47.0M
NMT-joint	Eq. 15 (512)	$\sigma(E\mathcal{W}_o)\sigma(\mathcal{W}_c h_t)$	17.54	47.2M
NMT-joint	Eq. 15 (2048)	$\sigma(E\mathcal{W}_o)\sigma(\mathcal{W}_c h_t)$	**18.11**	48.8M

Table 3: BLEU scores on De → En ($|\mathcal{V}| \approx 32K$) for the ablation analysis of NMT-joint.

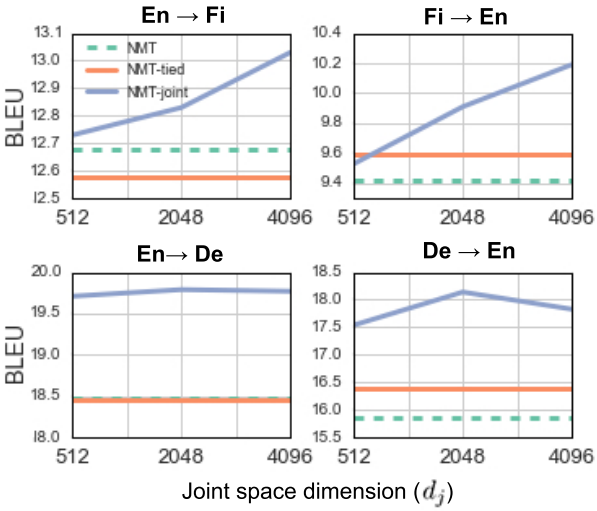


Figure 2: BLEU scores for the NMT-joint model when varying its dimension (d_j) with $|\mathcal{V}| \approx 32K$.

NMT baseline in many cases, but the differences are not consistent and it even scores significantly lower than NMT baseline in two cases, namely on Fi → En and De → En with $\mathcal{V} \approx 64K$. This validates our claim that the parametrization of the output space of the original NMT is not fully redundant, otherwise the NMT-tied would be able to match its BLEU in all cases. In contrast, the NMT-joint model outperforms consistently both baselines with a difference up to +2.2 and +1.6 BLEU points respectively,[5] showing that the NMT-tied model has a more effective parametrization and retains the advantages of both baselines, namely sharing weights with the input embeddings, and dedicating enough parameters for generation.

Overall, the highest scores correlate with a high number of BPE operations, namely 128K, 64K, 128K and 64k respectively. This suggests that the larger the vocabulary the better the performance, especially for the morphologically rich target languages, namely En → Fi and En → De. Lastly, the NMT baseline seems to be the least robust to sampling since its BLEU decreases in two cases. The other two models are more robust to sampling, however the difference of NMT-tied with the NMT is less significant than that of NMT-joint.

4.4 Ablation Analysis

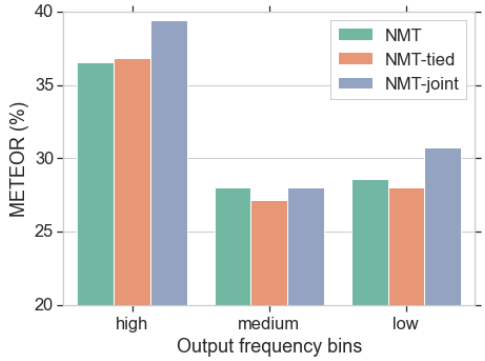
To demonstrate whether all the components of the proposed joint input-output model are useful and to which extend they contribute to the performance, we performed an ablation analysis; the results are displayed in Table 3. Overall, all the variants of the NMT-joint outperform the baseline with varying degrees of significance. The NMT-joint with a bilinear form (Eq. 6) as in (Yazdani and Henderson, 2015; Nam et al., 2016b) is slightly behind the NMT-tied and outperforms the NMT baseline; this supports our theoretical analysis in Section 2.1.2 which demonstrated that *weight tying* is learning an implicit linear structure similar to bilinear joint input-output models.

The NMT-joint model without learning explicit translation context structure (Eq. 7 a) performs similar to the bilinear model and the NMT-tied model, while the NMT-joint model without learning explicit output structure (Eq. 7 b) outperforms all the previous ones. When keeping same capacity (with d_j=512), our full model, which learns both output and translation context structure, performs similarly to the latter model and outperforms all the other baselines, including joint input-output models with a bilinear form (Yazdani and Henderson, 2015; Nam et al., 2016b). But when the capacity is allowed to increase (with d_j=2048), it outperforms all the other models. Since both nonlinearities are necessary to allow us to control the effective capacity of the joint space, these results show that both types of structure induction are important for reaching the top performance with NMT-joint.

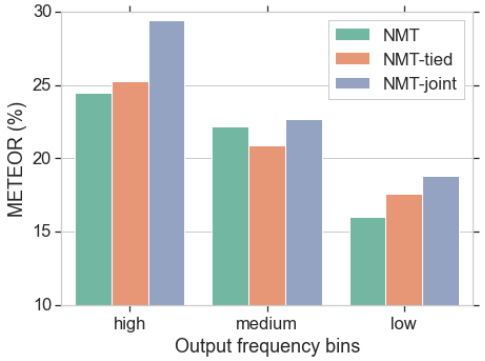
4.5 Effect of Embedding Size

Performance Figure 2 displays the BLEU scores of the proposed model when varying the size of the joint embedding, namely $d_j \in \{512, 2048, 4096\}$, against the two baselines. For English-Finish pairs, the increase in embedding size leads to a consistent increase in BLEU in favor of the NMT-joint model. For the English-German pairs, the difference with the baselines is much more evident

[5] Except in the case of Fi → En with $|\mathcal{V}| \approx 64K$, where the NMT baseline performed the best.



(a) Results on En → De ($|\mathcal{V}| \approx 32K$).



(b) Results on De → En ($|\mathcal{V}| \approx 32K$).

Figure 3: METEOR scores (%) on both directions of German-English language pair for all the models when focusing the evaluation on different frequency outputs grouped into three bins (high, medium, low).

		Sampling		
Model	d_j	50%	25%	5%
NMT	-	4.3K	5.7K	7.1K
NMT-`tied`	-	5.2K	6.0K	7.8K
NMT-`joint`	512	4.9K	5.9K	7.2K
NMT-`joint`	2048	2.8K	4.2K	7.0K
NMT-`joint`	4096	1.7K	2.9K	6.0K

Table 4: Target tokens processed per second during training with negative sampling on En → De pair with a large BPE vocabulary $|\mathcal{V}| \approx 128K$.

and the optimal size is observed around 2048 for De → En and around 512 on En → De. The results validate our hypothesis that there is parameter redundancy in the typical output layer. However the ideal parametrization is data dependent and is achievable systematically only with the `joint` output layer which is capacity-wise in between the typical output layer and the `tied` output layer.

Training speed Table 4 displays the target tokens processed per second by the models on En → DE with $|\mathcal{V}| \approx 128K$ using different levels of negative sampling, namely 50%, 25%, and 5%. In terms of training speed, the 512-dimensional NMT-`joint` model is as fast as the baselines, as we can observe in all cases. For higher dimensions of the joint space, namely 2048 and 4096 there is a notable decrease in speed which is remidiated by reducing the percentage of the negative samples.

4.6 Effect of Output Frequency and Architecture Depth

Figure 3 displays the performance in terms of METEOR on both directions of German-English language pair when evaluating on outputs of different frequency levels (high, medium, low) for all the competing models. The results on De → EN show that the improvements brought by the NMT-`joint` model against baselines are present consistently for all frequency levels including the low-frequency ones. Nevertheless, the improvement is most prominent for high-frequency outputs, which is reasonable given that no sentence filtering was performed and hence frequent words have higher impact in the absolute value of METEOR. Similarly, for En → De we can observe that NMT-`joint` outperforms the others on high-frequency and low-frequency labels while it reaches parity with them on the medium-frequency ones.

We also evaluated our model in another challenging condition in which we examine the effect of the NMT architecture depth in the performance of the proposed model. The results are displayed in Table 5. The results show that the NMT-`joint` outperforms the other two models consistently when varying the architecture depth of the encoder-decoder architecture. The NMT-`joint` overall is much more robust than NMT-`tied` and it outperforms it consistently in all settings. Compared to the NMT which is overparametrized the improvement even though consistent it is smaller for layer depth 3 and 4. This happens because NMT has a much higher number of parameters than NMT-`joint` with d_j=512.

Increasing the number of dimensions d_j of the joint space should lead to further improvements, as shown in Fig. 2. In fact, our NMT-`joint` with $d_j = 2048$ reaches 18.11 score with a 2-layer deep model, hence it outperforms all other NMT and NMT-`tied` models even with a deeper architecture (3-layer and 4-layer) regardless of the fact that it utilizes fewer parameters than them (48.8M vs 69.2-73.4M and 50.9-55.1M respectively).

Model	d_j	1-layer	$\|\Theta\|$	2-layer	$\|\Theta\|$	3-layer	$\|\Theta\|$	4-layer	$\|\Theta\|$
NMT	-	16.49	60.8M	15.85	65.0M	17.71	69.2M	17.74	73.4M
NMT-`tied`	-	15.93	42.5M	16.51	46.7M	17.72	50.9M	17.60	55.1M
NMT-`joint`	512	**16.93**	43.0M	**17.54**	47.2M	**17.83**	51.4M	**18.13**	55.6M

Table 5: BLEU scores on De $\rightarrow$ En ($|\mathcal{V}| \approx 32K$) for the NMT-`joint` with $d_j = 512$ against baselines when varying the depth of both the encoder and the decoder of the NMT model.

5 Related Work

Several studies focus on learning joint input-output representations grounded to word semantics for zero-shot image classification (Weston et al., 2011; Socher et al., 2013; Zhang et al., 2016), but there are fewer such studies for NLP tasks. (Yazdani and Henderson, 2015) proposed a zero-shot spoken language understanding model based on a bilinear joint space trained with hinge loss, and (Nam et al., 2016b), proposed a similar joint space trained with a WARP loss for zero-shot biomedical semantic indexing. In addition, there exist studies which aim to learn output representations directly from data such as (Srikumar and Manning, 2014; Yeh et al., 2018; Augenstein et al., 2018); their lack of semantic grounding to the input embeddings and the vocabulary-dependent parametrization, however, makes them data hungry and less scalable on large label sets. All these models, exhibit similar theoretical limitations as the softmax linear unit with *weight tying* which were described in Sections 2.2.

To our knowledge, there is no existing study which has considered the use of such joint input-output labels for neural machine translation. Compared to previous joint input-label models our model is more flexible and not restricted to linear mappings, which have limited expressivity, but uses non-linear mappings modeled similar to energy-based learning networks (Belanger and McCallum, 2016). Perhaps, the most similar embedding model to ours is the one by (Pappas and Henderson, 2018), except for the linear scaling unit which is specific to sigmoidal linear units designed for multi-label classification problems and not for structured prediction, as here.

6 Conclusion and Perspectives

We proposed a re-parametrization of the output layer for the decoder of NMT models which is more general and robust than a softmax linear unit with or without *weight tying* with the input word embeddings. Our evaluation shows that the *structure-aware* output layer outperforms *weight tying* in all cases and maintains a significant difference with the typical output layer without compromising much the training speed. Furthermore, it can successfully benefit from training corpora with large BPE vocabularies using negative sampling. The ablation analysis demonstrated that both types of structure captured by our model are essential and complementary, as well as, that their combination outperforms all previous output layers including those of bilinear input-output embedding models. Our further investigation revealed the robustness of the model to sampling-based training, translating infrequent outputs and to varying architecture depth.

As future work, the *structure-aware* output layer could be further improved along the following directions. The computational complexity of the model becomes prohibitive for a large joint projection because it requires a large matrix multiplication which depends on $|\mathcal{V}|$; hence, we have to resort to sampling based training relatively quickly when gradually increasing d_j (e.g. for $d_j >= 2048$). A more scalable way of increasing the output layer capacity could address this issue, for instance, by considering multiple consecutive additive transformations with small d_j. Another useful direction would be to use more advanced output encoders and additional external knowledge (contextualized or generically defined) for both words and sub-words. Finally, to encourage progress in joint input-output embedding learning for NMT, our code is available on Github: `http://github.com/idiap/joint-embedding-nmt`.

Acknowledgments

We are grateful for the support from the European Union through its Horizon 2020 program in the SUMMA project n. 688139, see `http://www.summa-project.eu` and for the valuable feedback from the anonymous reviewers.

References

Isabelle Augenstein, Sebastian Ruder, and Anders Sgaard. 2018. Multi-task learning of pairwise sequence classification tasks over disparate label spaces. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1896–1906, New Orleans, Louisiana. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, USA.

David Belanger and Andrew McCallum. 2016. Structured prediction energy networks. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 983–992, New York, New York, USA. PMLR.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703, Berlin, Germany. Association for Computational Linguistics.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361, Berlin, Germany. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2016. Tying word vectors and word classifiers: A loss framework for language modeling. *arXiv preprint arXiv:1611.01462*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Philipp Koehn et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL (Demo and Poster Sessions)*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

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

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2015. Character-based neural machine translation. *arXiv preprint arXiv:1511.04586*.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016a. All-in text: learning document, label, and word representations jointly. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016b. All-in text: Learning document,

label, and word representations jointly. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 1948–1954, Phoenix, AR, USA.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Nikolaos Pappas and James Henderson. 2018. Joint input-label embedding for neural text classification. *arXiv pre-print arXiv:1806.06219*.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the ACL (Vol. 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. 2013. Zero-shot learning through cross-modal transfer. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 935–943, Lake Tahoe, Nevada.

Vivek Srikumar and Christopher D. Manning. 2014. Learning distributed representations for structured output prediction. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3266–3274, Cambridge, MA, USA. MIT Press.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. WSABIE: Scaling up to large vocabulary image annotation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2764–2770. AAAI Press.

Majid Yazdani and James Henderson. 2015. A model of zero-shot learning of spoken language understanding. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 244–249.

Chih-Kuan Yeh, Wei-Chieh Wu, Wei-Jen Ko, and Yu-Chiang Frank Wang. 2018. Learning deep latent spaces for multi-label classification. In *In Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, New Orleans, USA.

Yang Zhang, Boqing Gong, and Mubarak Shah. 2016. Fast zero-shot image tagging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA.