Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation

Melvin Johnson*, Mike Schuster*, Quoc V. Le, Maxim Krikun,
Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas,
Martin Wattenberg, Greg Corrado, Macduff Hughes, Jeffrey Dean
Google
{melvinp,schuster}@google.com

Abstract

We propose a simple solution to use a single Neural Machine Translation (NMT) model to translate between multiple languages. Our solution requires no changes to the model architecture from a standard NMT system but instead introduces an artificial token at the beginning of the input sentence to specify the required target language. Using a shared wordpiece vocabulary, our approach enables Multilingual NMT systems using a single model. On the WMT'14 benchmarks, a single multilingual model achieves comparable performance for English→French and surpasses state-of-the-art results for English→German. Similarly, a single multilingual model surpasses state-of-the-art results for French→English and German→English on WMT'14 and WMT'15 benchmarks, respectively. On production corpora, multilingual models of up to twelve language pairs allow for better translation of many individual pairs. Our models can also learn to perform implicit bridging between language pairs never seen explicitly during training, showing that transfer learning and zero-shot translation is possible for neural translation. Finally, we show analyses that hints at a universal interlingua representation in our models and also show some interesting examples when mixing languages.

1 Introduction

End-to-end Neural Machine Translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015; Cho et al., 2014) is an approach to machine translation that has rapidly gained adoption in many large-scale settings (Zhou et al., 2016; Wu et al., 2016; Crego and et al., 2016). Almost all such systems are built for a single language pair — so far there has not been a sufficiently simple and efficient way to handle multiple language pairs using a single model without making significant changes to the basic NMT architecture.

In this paper we introduce a simple method to translate between multiple languages using a single model, taking advantage of multilingual data to improve NMT for all languages involved. Our method requires no change to the traditional NMT model architecture. Instead, we add an artificial token to the input sequence to indicate the required target language, a simple amendment to the data only. All other parts of the system — encoder, decoder, attention, and shared wordpiece vocabulary as described in Wu et al., (2016) — stay exactly the same. This method has several attractive benefits:

- **Simplicity**: Since no changes are made to the architecture of the model, scaling to more languages is trivial — any new data is simply added, possibly with over- or under-sampling such that all languages are appropriately represented, and used with a new token if the target language changes. Since no changes are made to the training procedure, the mini-batches for training are just sampled from the overall mixed-language training data just like for the single-language case. Since no a-priori decisions about how to allocate parameters for different languages are made, the system adapts automatically to use the total number of param-

*Corresponding authors.

Transactions of the Association for Computational Linguistics, vol. 5, pp. 339–351, 2017. Action Editor: Colin Cherry.
Submission batch: 11/2016; Revision batch: 3/2017; Published 10/2017.

eters efficiently to minimize the global loss. A multilingual model architecture of this type also simplifies production deployment significantly since it can cut down the total number of models necessary when dealing with multiple languages. Note that at Google, we support a total of over 100 languages as source and target, so theoretically 100^2 models would be necessary for the best possible translations between all pairs, if each model could only support a single language pair. Clearly this would be problematic in a production environment. Even when limiting to translating to/from English only, we still need over 200 models. Finally, batching together many requests from potentially different source and target languages can significantly improve efficiency of the serving system. In comparison, an alternative system that requires language-dependent encoders, decoders or attention modules does not have any of the above advantages.

- **Low-resource language improvements**: In a multilingual NMT model, all parameters are implicitly shared by all the language pairs being modeled. This forces the model to generalize across language boundaries during training. It is observed that when language pairs with little available data and language pairs with abundant data are mixed into a single model, translation quality on the low resource language pair is significantly improved.

- **Zero-shot translation**: A surprising benefit of modeling several language pairs in a single model is that the model can learn to translate between language pairs it has never seen in this combination during training (zero-shot translation) — a working example of transfer learning within neural translation models. For example, a multilingual NMT model trained with Portuguese→English and English→Spanish examples can generate reasonable translations for Portuguese→Spanish although it has not seen any data for that language pair. We show that the quality of zero-shot language pairs can easily be improved with little additional data of the language pair in question (a fact that has been previously confirmed for a related approach which is discussed in more detail in the next section).

In the remaining sections of this paper we first discuss related work and explain our multilingual system architecture in more detail. Then, we go through the different ways of merging languages on the source and target side in increasing difficulty (many-to-one, one-to-many, many-to-many), and discuss the results of a number of experiments on WMT benchmarks, as well as on some of Google's large-scale production datasets. We present results from transfer learning experiments and show how implicitly-learned bridging (zero-shot translation) performs in comparison to explicit bridging (i.e., first translating to a common language like English and then translating from that common language into the desired target language) as typically used in machine translation systems. We describe visualizations of the new system in action, which provide early evidence of shared semantic representations (interlingua) between languages. Finally we also show some interesting applications of mixing languages with examples: code-switching on the source side and weighted target language mixing, and suggest possible avenues for further exploration.

2 Related Work

Interlingual translation is a classic method in machine translation (Richens, 1958; Hutchins and Somers, 1992). Despite its distinguished history, most practical applications of machine translation have focused on individual language pairs because it was simply too difficult to build a single system that translates reliably from and to several languages.

Neural Machine Translation (NMT) (Kalchbrenner and Blunsom, 2013) was shown to be a promising end-to-end learning approach in Sutskever et al. (2014); Bahdanau et al. (2015); Cho et al. (2014) and was quickly extended to multilingual machine translation in various ways.

An early attempt is the work in Dong et al., (2015), where the authors modify an attention-based encoder-decoder approach to perform multilingual NMT by adding a separate decoder and attention mechanism for each target language. In Luong et al., (2015a) multilingual training in a multitask learning setting is described. This model is also an encoder-decoder

network, in this case without an attention mechanism. To make proper use of multilingual data, they extend their model with multiple encoders and decoders, one for each supported source and target language. In Caglayan et al., (2016) the authors incorporate multiple modalities other than text into the encoder-decoder framework.

Several other approaches have been proposed for multilingual training, especially for low-resource language pairs. For instance, in Zoph and Knight (2016) a form of multi-source translation was proposed where the model has multiple different encoders and different attention mechanisms for each source language. However, this work requires the presence of a multi-way parallel corpus between all the languages involved, which is difficult to obtain in practice. Most closely related to our approach is Firat et al., (2016a) in which the authors propose multi-way multilingual NMT using a single shared attention mechanism but multiple encoders/decoders for each source/target language. Recently in Lee et al., (2016) a CNN-based character-level encoder was proposed which is shared across multiple source languages. However, this approach can only perform translations into a single target language.

Our approach is related to the multitask learning framework (Caruana, 1998). Despite its promise, this framework has seen limited practical success in real world applications. In speech recognition, there have been many successful reports of modeling multiple languages using a single model ((Schultz and Kirchhoff, 2006) for an extensive reference and references therein). Multilingual language processing has also shown to be successful in domains other than translation (Gillick et al., 2016; Tsvetkov et al., 2016).

There have been other approaches similar to ours in spirit, but used for very different purposes. In Sennrich et al.,(2016a), the NMT framework has been extended to control the politeness level of the target translation by adding a special token to the source sentence. The same idea was used in Yamagishi et al., (2016) to add the distinction between 'active' and 'passive' tense to the generated target sentence.

Our method has an additional benefit not seen in other systems: it gives the system the ability to perform zero-shot translation, meaning the system can translate from a source language to a target language without having seen explicit examples from this specific language pair during training. Zero-shot translation was the direct goal of Firat et al., (2016c). Although they were not able to achieve this direct goal, they were able to do what they call "zero-resource" translation by using their pre-trained multi-way multilingual model and later fine-tuning it with pseudo-parallel data generated by the model. It should be noted that the difference between "zero-shot" and "zero-resource" translation is the additional fine-tuning step which is required in the latter approach.

To the best of our knowledge, our work is the first to validate the use of true multilingual translation using a single encoder-decoder model, and is incidentally also already used in a production setting. It is also the first work to demonstrate the possibility of zero-shot translation, a successful example of transfer learning in machine translation, without any additional steps.

3 System Architecture

The multilingual model architecture is identical to Google's Neural Machine Translation (GNMT) system (Wu et al., 2016) (with the optional addition of direct connections between encoder and decoder layers which we have used for some of our experiments).

To be able to make use of multilingual data within a single system, we propose one simple modification to the input data, which is to introduce an artificial token at the beginning of the input sentence to indicate the target language the model should translate to. For instance, consider the following En→Es pair of sentences:

```
How are you? -> ¿Cómo estás?
```

It will be modified to:

```
<2es> How are you? -> ¿Cómo estás?
```

to indicate that Spanish is the target language. Note that we don't specify the source language – the model will learn this automatically.

After adding the token to the input data, we train the model with all multilingual data consisting of multiple language pairs at once, possibly after over- or undersampling some of the data to adjust for the relative ratio of the language data available. To address the issue of translation of unknown words and to limit the vocabulary for computational efficiency,

we use a shared wordpiece model (Schuster and Nakajima, 2012) across all the source and target data used for training, usually with 32,000 word pieces. The segmentation algorithm used here is very similar (with small differences) to Byte-Pair-Encoding (BPE) which was described in Gage (1994) and was also used in Sennrich et al., (2016b) for machine translation. All training is carried out similar to (Wu et al., 2016) and implemented in TensorFlow (Abadi and et al., 2016).

In summary, this approach is the simplest among the alternatives that we are aware of. During training and inference, we only need to add one additional token to each sentence of the source data to specify the desired target language.

4 Experiments and Results

In this section, we apply our proposed method to train multilingual models in several different configurations. Since we can have models with either single or multiple source/target languages we test three interesting cases for mapping languages: 1) *many to one*, 2) *one to many*, and 3) *many to many*. As already discussed in Section 2, other models have been used to explore some of these cases already, but for completeness we apply our technique to these interesting use cases again to give a full picture of the effectiveness of our approach.

We will also show results and discuss benefits of bringing together many (un)related languages in a single large-scale model trained on production data. Finally, we will present our findings on zero-shot translation where the model learns to translate between pairs of languages for which no explicit parallel examples existed in the training data, and show results of experiments where adding additional data improves zero-shot translation quality further.

4.1 Datasets, Training Protocols and Evaluation Metrics

For WMT, we train our models on the WMT'14 En→Fr and the WMT'14 En→De datasets. In both cases, we use newstest2014 as the test sets to be able to compare against previous work (Luong et al., 2015c; Sébastien et al., 2015; Zhou et al., 2016; Wu et al., 2016). For WMT Fr→En and De→En we use newstest2014 and newstest2015 as test sets. Despite training on WMT'14 data, which is somewhat smaller than WMT'15, we test our De→En model on newstest2015, similar to Luong et al., (2015b). The combination of newstest2012 and newstest2013 is used as the development set.

In addition to WMT, we also evaluate the multilingual approach on some Google-internal large-scale production datasets representing a wide spectrum of languages with very distinct linguistic properties: En↔Japanese(Ja), En↔Korean(Ko), En↔Es, and En↔Pt. These datasets are two to three orders of magnitude larger than the WMT datasets.

Our training protocols are mostly identical to those described in Wu et al., (2016). We find that some multilingual models take a little more time to train than single language pair models, likely because each language pair is seen only for a fraction of the training process. We use larger batch sizes with a slightly higher initial learning rate to speed up the convergence of these models.

We evaluate our models using the standard BLEU score metric and to make our results comparable to previous work (Sutskever et al., 2014; Luong et al., 2015c; Zhou et al., 2016; Wu et al., 2016), we report tokenized BLEU score as computed by the `multi-bleu.pl` script, which can be downloaded from the public implementation of Moses.[1]

To test the influence of varying amounts of training data per language pair we explore two strategies when building multilingual models: a) where we oversample the data from all language pairs to be of the same size as the largest language pair, and b) where we mix the data as is without any change. The wordpiece model training is done after the optional oversampling taking into account all the changed data ratios. For the WMT models we report results using both of these strategies. For the production models, we always balance the data such that the ratios are equal.

One benefit of the way we share all the components of the model is that the mini-batches can contain data from different language pairs during training and inference, which are typically just random samples from the final training and test data distributions. This is a simple way of preventing "catastrophic forgetting" - tendency for knowledge of previously

[1] `http://www.statmt.org/moses/`

learned task(s) (e.g. language pair A) to be abruptly forgotten as information relevant to the current task (e.g. language pair B) is incorporated (French, 1999). Other approaches to multilingual translation require complex update scheduling mechanisms to prevent this effect (Firat et al., 2016b).

4.2 Many to One

In this section we explore having multiple source languages and a single target language — the simplest way of combining language pairs. Since there is only a single target language no additional source token is required. We perform three sets of experiments:

- The first set of experiments is on the WMT datasets, where De→En and Fr→En are combined to train a multilingual model. Our baselines are two single language pair models: De→En and Fr→En trained independently. We perform these experiments once with oversampling and once without.
- The second set of experiments is on production data where we combine Ja→En and Ko→En, with oversampling. The baselines are two single language pair models trained independently.
- Finally, the third set of experiments is on production data where we combine Es→En and Pt→En, with oversampling. The baselines are again two single language pair models trained independently.

All of the multilingual and single language pair models have the same total number of parameters as the baseline NMT models trained on a single language pair (using 1024 nodes, 8 LSTM layers and a shared wordpiece model vocabulary of 32k, a total of 255M parameters per model). A side effect of this equal choice of parameters is that it is presumably unfair to the multilingual models as the number of parameters available per language pair is reduced by a factor of N compared to the single language pair models, if N is the number of language pairs combined in the multilingual model. The multilingual model also has to handle the combined vocabulary of all the single models. We chose to keep the number of parameters constant for all models to simplify experimentation. We relax this constraint for some of the large-scale experiments shown further below.

Table 1: Many to One: BLEU scores on for single language pair and multilingual models. *: no oversampling

Model	Single	Multi	Diff
WMT De→En	30.43	30.59	+0.16
WMT Fr→En	35.50	35.73	+0.23
WMT De→En*	30.43	30.54	+0.11
WMT Fr→En*	35.50	36.77	+1.27
Prod Ja→En	23.41	23.87	+0.46
Prod Ko→En	25.42	25.47	+0.05
Prod Es→En	38.00	38.73	+0.73
Prod Pt→En	44.40	45.19	+0.79

The results are presented in Table 1. For all experiments the multilingual models outperform the baseline single systems despite the above mentioned disadvantage with respect to the number of parameters available per language pair. One possible hypothesis explaining the gains is that the model has been shown more English data on the target side, and that the source languages belong to the same language families, so the model has learned useful generalizations.

For the WMT experiments, we obtain a maximum gain of +1.27 BLEU for Fr→En. Note that the results on both the WMT test sets are better than other published state-of-the-art results for a single model, to the best of our knowledge.

4.3 One to Many

In this section, we explore the application of our method when there is a single source language and multiple target languages. Here we need to prepend the input with an additional token to specify the target language. We perform three sets of experiments very similar to the previous section.

Table 2 summarizes the results when performing translations into multiple target languages. We see that the multilingual models are comparable to, and in some cases outperform, the baselines, but not always. We obtain a large gain of +0.9 BLEU for En→Es. Unlike the previous set of results, there are less significant gains in this setting. This is perhaps due to the fact that the decoder has a more difficult time translating into multiple target languages which may even have different scripts, which are combined into a single shared wordpiece vocabulary. Note that even for languages with entirely different scripts (e.g., Korean and Japanese) there is significant overlap in

wordpieces when real data is used, as often numbers, dates, names, websites, punctuation etc. are actually using a shared script (ASCII).

Table 2: One to Many: BLEU scores for single language pair and multilingual models. *: no oversampling

Model	Single	Multi	Diff
WMT En→De	24.67	24.97	+0.30
WMT En→Fr	38.95	36.84	-2.11
WMT En→De*	24.67	22.61	-2.06
WMT En→Fr*	38.95	38.16	-0.79
Prod En→Ja	23.66	23.73	+0.07
Prod En→Ko	19.75	19.58	-0.17
Prod En→Es	34.50	35.40	+0.90
Prod En→Pt	38.40	38.63	+0.23

We observe that oversampling helps the smaller language pair (En→De) at the cost of lower quality for the larger language pair (En→Fr). The model without oversampling achieves better results on the larger language compared to the smaller one as expected. We also find that this effect is more prominent on smaller datasets (WMT) and much less so on our much larger production datasets.

4.4 Many to Many

In this section, we report on experiments when there are multiple source languages and multiple target languages within a single model — the most difficult setup. Since multiple target languages are given, the input needs to be prepended with the target language token as above.

The results are presented in Table 3. We see that the multilingual production models with the same model size and vocabulary size as the single language models are quite close to the baselines – the average relative loss in BLEU score across all experiments is only approximately 2.5%.

Although there are some significant losses in quality from training many languages jointly using a model with the same total number of parameters as the single language pair models, these models reduce the total complexity involved in training and productionization.

4.5 Large-scale Experiments

This section shows the result of combining 12 production language pairs having a total of 3B parameters (255M per single model) into a single multilingual model. A range of multilingual models were trained, starting from the same size as a single language pair model with 255M parameters (1024 nodes) up to 650M parameters (1792 nodes). As above, the input needs to be prepended with the target language token. We oversample the examples from the smaller language pairs to balance the data as explained above.

Table 3: Many to Many: BLEU scores for single language pair and multilingual models. *: no oversampling

Model	Single	Multi	Diff
WMT En→De	24.67	24.49	-0.18
WMT En→Fr	38.95	36.23	-2.72
WMT De→En	30.43	29.84	-0.59
WMT Fr→En	35.50	34.89	-0.61
WMT En→De*	24.67	21.92	-2.75
WMT En→Fr*	38.95	37.45	-1.50
WMT De→En*	30.43	29.22	-1.21
WMT Fr→En*	35.50	35.93	+0.43
Prod En→Ja	23.66	23.12	-0.54
Prod En→Ko	19.75	19.73	-0.02
Prod Ja→En	23.41	22.86	-0.55
Prod Ko→En	25.42	24.76	-0.66
Prod En→Es	34.50	34.69	+0.19
Prod En→Pt	38.40	37.25	-1.15
Prod Es→En	38.00	37.65	-0.35
Prod Pt→En	44.40	44.02	-0.38

The results for single language pair models versus multilingual models with increasing numbers of parameters are summarized in Table 4. We find that the multilingual models are on average worse than the single models (about 5.6% to 2.5% relative depending on size, however, some actually get better) and as expected the average difference gets smaller when going to larger multilingual models. It should be noted that the largest multilingual model we have trained has still about five times less parameters than the combined single models.

The multilingual model also requires only roughly 1/12-th of the training time (or computing resources) to converge compared to the combined single models (total training time for all our models is still in the order of weeks). Another important point is that since we only train for a little longer than a standard single model, the individual language pairs can see as little as 1/12-th of the data in comparison to their single language pair models but still produce satisfactory results.

In summary, multilingual NMT enables us to

Table 4: Large-scale experiments: BLEU scores for single language pair and multilingual models.

Model	Single	Multi	Multi	Multi	Multi
#nodes	1024	1024	1280	1536	1792
#params	3B	255M	367M	499M	650M
En→Ja	23.66	21.10	21.17	21.72	21.70
En→Ko	19.75	18.41	18.36	18.30	18.28
Ja→En	23.41	21.62	22.03	22.51	23.18
Ko→En	25.42	22.87	23.46	24.00	24.67
En→Es	34.50	34.25	34.40	34.77	34.70
En→Pt	38.40	37.35	37.42	37.80	37.92
Es→En	38.00	36.04	36.50	37.26	37.45
Pt→En	44.40	42.53	42.82	43.64	43.87
En→De	26.43	23.15	23.77	23.63	24.01
En→Fr	35.37	34.00	34.19	34.91	34.81
De→En	31.77	31.17	31.65	32.24	32.32
Fr→En	36.47	34.40	34.56	35.35	35.52
ave diff	-	-1.72	-1.43	-0.95	-0.76
vs single	-	-5.6%	-4.7%	-3.1%	-2.5%

group languages with little loss in quality while having the benefits of better training efficiency, smaller number of models, and easier productionization.

4.6 Zero-Shot Translation

The most straight-forward approach of translating between languages where no or little parallel data is available is to use explicit bridging, meaning to translate to an intermediate language first and then to translate to the desired target language. The intermediate language is often English as xx→En and En→yy data is more readily available. The two potential disadvantages of this approach are: a) total translation time doubles, b) the potential loss of quality by translating to/from the intermediate language.

An interesting benefit of our approach is that it allows to perform directly *implicit bridging* (zero-shot translation) between a language pair for which no explicit parallel training data has been seen without any modification to the model. Obviously, the model will only be able to do zero-shot translation between languages it has seen individually as source and target languages during training at some point, not for entirely new ones.

To demonstrate this we will use two multilingual models — a model trained with examples from two different language pairs, Pt→En and En→Es (Model 1), and a model trained with examples from four different language pairs, En↔Pt and En↔Es (Model 2). As with the previous multilingual models, both of these models perform comparable to or even slightly better than the baseline single models for the language pairs explicitly seen. Additionally, we show that both of these models can generate reasonable quality Pt→Es translations (BLEU scores above 20) without ever having seen Pt→Es data during training. To our knowledge this is the first successful demonstration of true multilingual zero-shot translation.

Table 5 summarizes our results for the Pt→Es translation experiments. Rows (a) and (b) show the performance of the phrase-based machine translation (PBMT) system and the NMT system through explicit bridging (Pt→En, then En→Es). It can be seen that the NMT system outperforms the PBMT system by close to 2 BLEU points. For comparison, we also built a single NMT model on all available Pt→Es parallel sentences (see (c) in Table 5).

Table 5: Portuguese→Spanish BLEU scores using various models.

	Model	Zero-shot	BLEU
(a)	PBMT bridged	no	28.99
(b)	NMT bridged	no	30.91
(c)	NMT Pt→Es	no	31.50
(d)	Model 1 (Pt→En, En→Es)	yes	21.62
(e)	Model 2 (En↔{Es, Pt})	yes	24.75
(f)	Model 2 + incremental training	no	31.77

The most interesting observation is that both Model 1 and Model 2 can perform zero-shot translation with reasonable quality (see (d) and (e)) compared to the initial expectation that this would not work at all. Note that Model 2 outperforms Model 1 by close to 3 BLEU points although Model 2 was trained with four language pairs as opposed to with only two for Model 1 (with both models having the same number of total parameters). In this case the addition of Spanish on the source side and Portuguese on the target side helps Pt→Es zero-shot translation (which is the opposite direction of where we would expect it to help). We believe that this unexpected effect is only possible because our shared architecture enables the model to learn a form of interlingua between all these languages. We explore this hypothesis in more detail in Section 5.

Finally we incrementally train zero-shot Model 2 with a small amount of true Pt→Es parallel data (an order of magnitude less than Table 5 (c)) and obtain the best quality and half the decoding time compared

to explicit bridging (Table 5 (b)). The resulting model cannot be called zero-shot anymore since some true parallel data has been used to improve it. Overall this shows that the proposed approach of implicit bridging using zero-shot translation via multilingual models can serve as a good baseline for further incremental training with relatively small amounts of true parallel data of the zero-shot direction. This result is especially significant for non-English low-resource language pairs where it might be easier to obtain parallel data with English but much harder to obtain parallel data for language pairs where neither the source nor the target language is English. We explore the effect of using parallel data in more detail in Section 4.7.

Since Portuguese and Spanish are of the same language family, an interesting question is how well zero-shot translation works for less related languages. Table 6 shows the results for explicit and implicit bridging from Spanish to Japanese using the large-scale model from Table 4 – Spanish and Japanese can be regarded as quite unrelated. As expected zero-shot translation works worse than explicit bridging and the quality drops relatively more (roughly 50% drop in BLEU score) than for the case of more related languages as shown above. Despite the quality drop, this proves that our approach enables zero-shot translation even between unrelated languages.

Table 6: Spanish→Japanese BLEU scores for explicit and implicit bridging using the 12-language pair large-scale model from Table 4.

Model	BLEU
NMT Es→Ja explicitly bridged	18.00
NMT Es→Ja implicitly bridged	9.14

4.7 Effect of Direct Parallel Data

In this section, we explore two ways of leveraging available parallel data to improve zero-shot translation quality, similar in spirit to what was reported in Firat et al., (2016c). For our multilingual architecture we consider:

- Incrementally training the multilingual model on the additional parallel data for the zero-shot directions.
- Training a new multilingual model with all available parallel data mixed equally.

For our experiments, we use a baseline model which we call "Zero-Shot" trained on a combined parallel corpus of English↔{Belarusian(Be), Russian(Ru), Ukrainian(Uk)}. We trained a second model on the above corpus together with additional Ru↔{Be, Uk} data. We call this model "From-Scratch". Both models support four target languages, and are evaluated on our standard test sets. As done previously we oversample the data such that all language pairs are represented equally. Finally, we take the best checkpoint of the "Zero-Shot" model, and run incremental training on a small portion of the data used to train the "From-Scratch" model for a short period of time until convergence (in this case 3% of "Zero-Shot" model total training time). We call this model "Incremental".

As can be seen from Table 7, for the English↔X directions, all three models show comparable scores. On the Ru↔{Be, Uk} directions, the "Zero-Shot" model already achieves relatively high BLEU scores for all directions except one, without any explicit parallel data. This could be because these languages are linguistically related. In the "From-Scratch" column, we see that training a new model from scratch improves the zero-shot translation directions further. However, this strategy has a slightly negative effect on the En↔X directions because our oversampling strategy will reduce the frequency of the data from these directions. In the final column, we see that incremental training with direct parallel data recovers most of the BLEU score difference between the first two columns on the zero-shot language pairs. In summary, our shared architecture models the zero-shot language pairs quite well and hence enables us to easily improve their quality with a small amount of additional parallel data.

5 Visual Analysis

The results of this paper — that training a model across multiple languages can enhance performance at the individual language level, and that zero-shot translation can be effective — raise a number of questions about how these tasks are handled inside the model. Is the network learning some sort of shared representation, in which sentences with the same meaning are represented in similar ways regardless of language? Does the model operate on zero-shot

Table 7: BLEU scores for English↔{Belarusian, Russian, Ukrainian} models.

	Zero-Shot	From-Scratch	Incremental
En→Be	16.85	17.03	16.99
En→Ru	22.21	22.03	21.92
En→Uk	18.16	17.75	18.27
Be→En	25.44	24.72	25.54
Ru→En	28.36	27.90	28.46
Uk→En	28.60	28.51	28.58
Be→Ru	56.53	82.50	78.63
Ru→Be	58.75	72.06	70.01
Ru→Uk	21.92	25.75	25.34
Uk→Ru	16.73	30.53	29.92

translations in the same way as it treats language pairs it has been trained on?

One way to study the representations used by the network is to look at the activations of the network during translation. A starting point for investigation is the set of *context vectors*, i.e., the sum of internal encoder states weighted by their attention probabilities per step (Eq. (5) in (Bahdanau et al., 2015)).

A translation of a single sentence generates a sequence of context vectors. In this context, our original questions about shared representation can be studied by looking at how the vector sequences of different sentences relate. We could then ask for example: Do sentences cluster together depending on the source or target language? Or instead do sentences with similar meanings cluster, regardless of language? We try to find answers to these questions by looking at lower-dimensional representations of internal embeddings of the network that humans can more easily interpret.

5.1 Evidence for an Interlingua

Several trained networks indeed show strong visual evidence of a shared representation. For example, Figure 1 below was produced from a many-to-many model trained on four language pairs, English↔Japanese and English↔Korean. To visualize the model in action we began with a small corpus of 74 triples of semantically identical cross-language phrases. That is, each triple contained phrases in English, Japanese and Korean with the same underlying meaning. To compile these triples, we searched a ground-truth database for English sentences which were paired with both Japanese and Korean translations.

We then applied the trained model to translate each sentence of each triple into the two other possible languages. Performing this process yielded six new sentences based on each triple, for a total of $74*6 = 444$ total translations with 9,978 steps corresponding to the same number of context vectors. Since context vectors are high-dimensional, we use the TensorFlow Embedding Projector[2] to map them into more accessible 3D space via t-SNE (Maaten and Hinton, 2008). In the following diagrams, each point represents a single decoding step during the translation process. Points that represent steps for a given sentence are connected by line segments.

Figure 1 shows a global view of all 9,978 context vectors. Points produced from the same original sentence triple are all given the same (random) color. Inspection of these clusters shows that each strand represents a single sentence, and clusters of strands generally represent a set of translations of the same underlying sentence, but with different source and target languages.

At right are two close-ups: one of an individual cluster, still coloring based on membership in the same triple, and one where we have colored by source language.

5.2 Partially Separated Representations

Not all models show such clean semantic clustering. Sometimes we observed joint embeddings in some regions of space coexisting with separate large clusters which contained many context vectors from just one language pair.

For example, Figure 2a shows a t-SNE projection of context vectors from a model that was trained on Portuguese→English (blue) and English→Spanish (yellow) and performing zero-shot translation from Portuguese→Spanish (red). This projection shows 153 semantically identical triples translated as described above, yielding 459 total translations. The large red region on the left primarily contains zero-shot Portuguese→Spanish translations. In other words, for a significant number of sentences, the zero-shot translation has a different embedding than the two trained translation directions. On the other hand, some zero-shot translation vectors do seem to

[2] https://www.tensorflow.org/get_started/embedding_viz

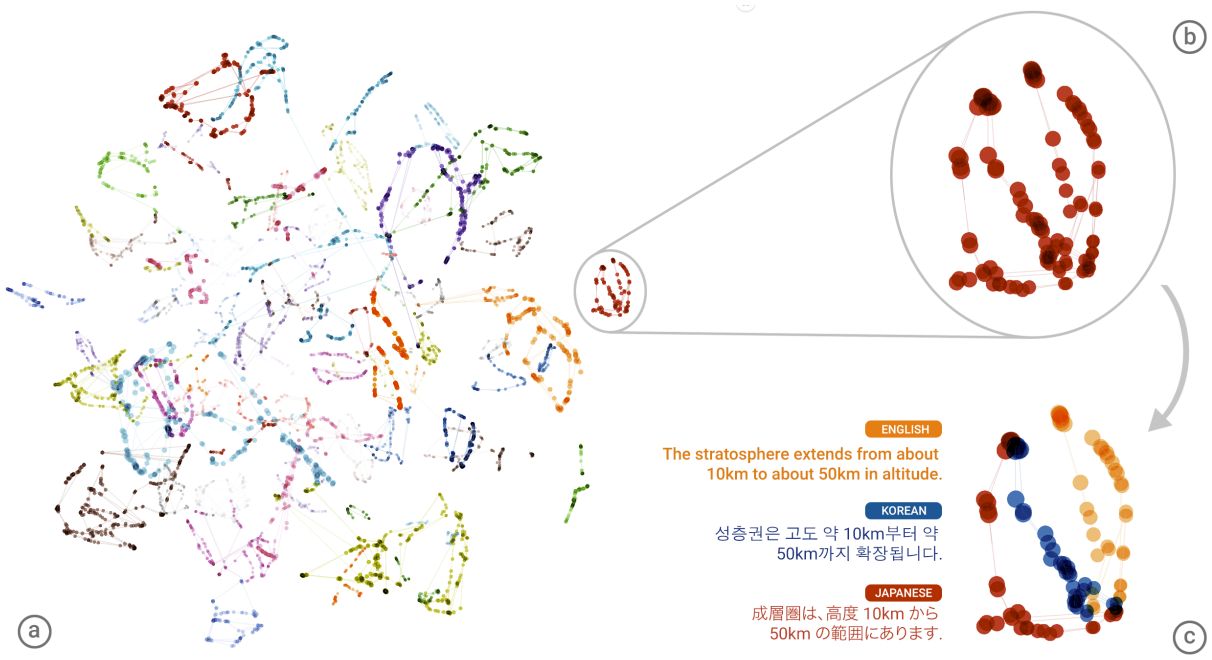

Figure 1: A t-SNE projection of the embedding of 74 semantically identical sentences translated across all 6 possible directions, yielding a total of 9,978 steps (dots in the image), from the model trained on English↔Japanese and English↔Korean examples. (a) A bird's-eye view of the embedding, coloring by the index of the semantic sentence. Well-defined clusters each having a single color are apparent. (b) A zoomed in view of one of the clusters with the same coloring. All of the sentences within this cluster are translations of "The stratosphere extends from about 10km to about 50km in altitude." (c) The same cluster colored by source language. All three source languages can be seen within this cluster.

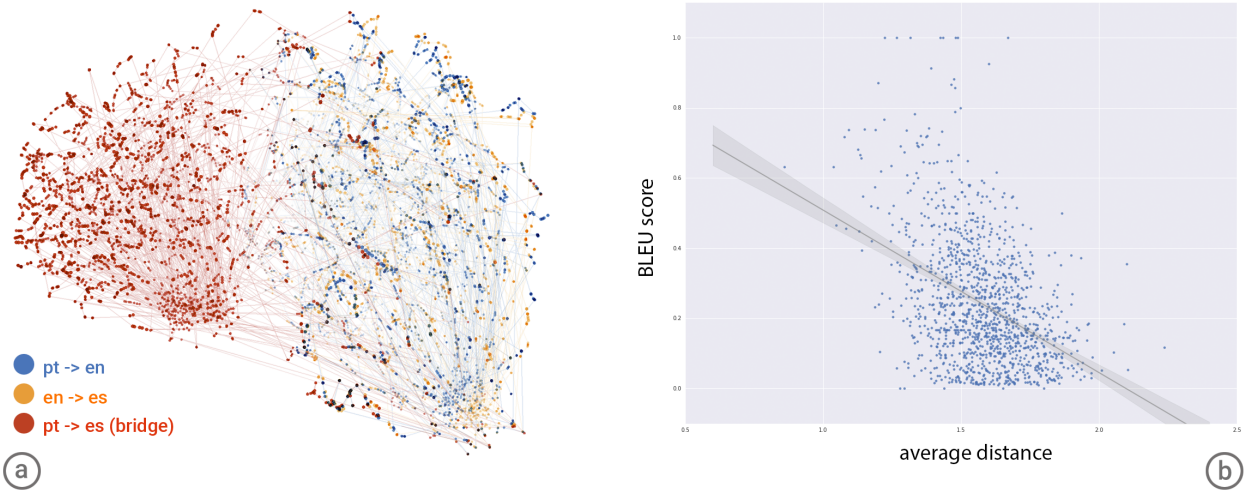

Figure 2: (a) A bird's-eye view of a t-SNE projection of an embedding of the model trained on Portuguese→English (blue) and English→Spanish (yellow) examples with a Portuguese→Spanish zero-shot bridge (red). The large red region on the left primarily contains the zero-shot Portuguese→Spanish translations. (b) A scatter plot of BLEU scores of zero-shot translations versus the average point-wise distance between the zero-shot translation and a non-bridged translation. The Pearson correlation coefficient is -0.42.

fall near the embeddings found in other languages, as on the large region on the right.

It is natural to ask whether the large cluster of "separated" zero-shot translations has any significance. A definitive answer requires further investigation, but in this case zero-shot translations in the separated area do tend to have lower BLEU scores.

Figure 2b shows a plot of BLEU scores of a zero-shot translation versus the average pointwise distance between it and the same translation from a trained language pair. An interesting area for future research is to find a more reliable correspondence between embedding geometry and model performance to predict the quality of a zero-shot translation during decoding by comparing it to the embedding of the translation through a trained language pair.

6 Mixing Languages

Having a mechanism to translate from a random source language to a single chosen target language using an additional source token made us think about what happens when languages are mixed on the source or target side. In particular, we were interested in the following two experiments: 1) Can a multilingual model successfully handle multi-language input (code-switching) in the middle of a sentence?; 2) What happens when a multilingual model is triggered with a linear mix of two target language tokens?

6.1 Source Language Code-Switching

Here we show how multilingual models deal with source language code-switching – an example from a multilingual {Ja,Ko}→En model is below. Mixing Japanese and Korean in the source produces in many cases correct English translations, showing that code-switching can be handled by this model, although no such code-switching samples were present in the training data. Note that the model can effectively handle the different typographic scripts since the individual characters/wordpieces are present in the shared vocabulary.

- **Japanese:** 私は東京大学の学生です。 → I am a student at Tokyo University.
- **Korean:** 나는 도쿄 대학의 학생입니다. → I am a student at Tokyo University.
- **Japanese/Korean:** 私は東京大学학생입니다. → I am a student of Tokyo University.

Interestingly, the mixed-language translation is slightly different from both single source language translations.

6.2 Weighted Target Language Selection

Here we test what happens when we mix target languages. Using a multilingual En→{Ja, Ko} model, we feed a linear combination $(1-w)$<2ja>$+w$<2ko> of the embedding vectors for "<2ja>" and "<2ko>". Clearly, for $w = 0$ the model should produce Japanese, for $w = 1$ it should produce Korean, but what happens in between?

The model may produce some sort of intermediate language ("Japarean"), but the results turn out to be less surprising. Most of the time the output just switches from one language to another around $w = 0.5$. In some cases, for intermediate values of w, the model switches languages mid-sentence.

A possible explanation for this behavior is that the target language model, implicitly learned by the decoder LSTM, may make it very hard to mix words from different languages, especially when they use different scripts.

Table 8 shows an example of mixed target languages (Ja/Ko), where we can observe an interesting transition in the script and grammar. At $w_{ko} = 0.58$, the model translates the source sentence into a mix of Japanese and Korean. At $w_{ko} = 0.60$, the sentence is translated into full Korean, where all of the source words are captured, however, the ordering of the words is not natural. When w_{ko} is increased to 0.7, the model starts to translate the source sentence into a Korean sentence that sounds more natural.[3]

7 Conclusion

We present a simple solution to multilingual NMT. We show that we can train multilingual NMT models that can be used to translate between a number of different languages using a single model where all parameters are shared, which as a positive side-effect also improves the translation quality of low-resource languages in the mix. We also show that zero-shot translation without explicit bridging is possible, which is the first time to our knowledge that a

[3]The Korean translation does not contain spaces and uses '。' as punctuation symbol, and these are all artifacts of applying a Japanese postprocessor.

Table 8: Gradually mixing target languages Ja/Ko.

w_{ko}	I must be getting somewhere near the centre of the earth.
0.00	私は地球の中心の近くにどこかに行っているに違いない。
0.40	私は地球の中心近くのどこかに着いているに違いない。
0.56	私は地球の中心の近くのどこかになっているに違いない。
0.58	私は지구の中心의가까이에어딘가에도착하고있어야한다。
0.60	나는지구의센터의가까이에어딘가에도착하고있어야한다。
0.70	나는지구의중심근처어딘가에도착해야합니다。
0.90	나는어딘가지구의중심근처에도착해야합니다。
1.00	나는어딘가지구의중심근처에도착해야합니다。

form of true transfer learning has been shown to work for machine translation. To explicitly improve the zero-shot translation quality, we explore two ways of adding available parallel data and find that small additional amounts are sufficient to reach satisfactory results. In our largest experiment we merge 12 language pairs into a single model and achieve only slightly lower translation quality as for the single language pair baselines despite the drastically reduced amount of modeling capacity per language in the multilingual model. Visual interpretation of the results shows that these models learn a form of interlingua representation between all involved language pairs. The simple architecture makes it possible to mix languages on the source or target side to yield some interesting translation examples. Our approach has been shown to work reliably in a Google-scale production setting and enables us to scale to a large number of languages quickly.

Acknowledgements

We would like to thank the entire Google Brain Team and Google Translate Team for their foundational contributions to this project. In particular, we thank Junyoung Chung for his insights on the topic and Alex Rudnick and Otavio Good for helpful suggestions. We would also like to thank the TACL Action Editor and the reviewers for their feedback.

References

Martin Abadi and Paul Barham et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Ozan Caglayan and Walid Aransa et al. 2016. Does multimodality help human and machine for translation and image captioning? In *Proceedings of the First Conference on Machine Translation*, pages 627–633, Berlin, Germany, August. Association for Computational Linguistics.

Rich Caruana. 1998. Multitask learning. In *Learning to learn*, pages 95–133. Springer.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*.

Josep Crego and Jungi Kim et al. 2016. Systran's pure neural machine translation systems. *arXiv preprint arXiv:1610.05540*.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 1723–1732.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 866–875.

Orhan Firat, Kyunghyun Cho, Baskaran Sankaran, Fatos T. Yarman Vural, and Yoshua Bengio. 2016b. Multi-way, multilingual neural machine translation. *Computer Speech and Language*.

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016c. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas, November. Association for Computational Linguistics.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38, February.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. Multilingual language processing from bytes. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1296–1306, San Diego, California, June. Association for Computational Linguistics.

William John Hutchins and Harold L. Somers. 1992. *An introduction to machine translation*, volume 362. Academic Press London.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Conference on Empirical Methods in Natural Language Processing*.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2016. Fully character-level neural machine translation without explicit segmentation. *arXiv preprint arXiv:1610.03017*.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015a. Multi-task sequence to sequence learning. In *International Conference on Learning Representations*.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015b. Effective approaches to attention-based neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*.

Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2015c. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Laurens Van Der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9.

Richard H Richens. 1958. Interlingual machine translation. *The Computer Journal*, 1(3):144–147.

Tanja Schultz and Katrin Kirchhoff. 2006. *Multilingual speech processing*. Elsevier Academic Press, Amsterdam, Boston, Paris.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing*.

Jean Sébastien, Cho Kyunghyun, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 35–40.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Yulia Tsvetkov, Sunayana Sitaram, Manaal Faruqui, Guillaume Lample, Patrick Littell, David Mortensen, Alan W Black, Lori Levin, and Chris Dyer. 2016. Polyglot neural language models: A case study in cross-lingual phonetic representation learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1357–1366, San Diego, California, June. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, and Zhifeng Chen et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144v2*.

Hayahide Yamagishi, Shin Kanouchi, and Mamoru Komachi. 2016. Controlling the voice of a sentence in Japanese-to-English neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation*, pages 203–210, Osaka, Japan, December.

Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016. Deep recurrent models with fast-forward connections for neural machine translation. *Transactions of the Association for Computational Linguistics*, 4:371–383.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 30–34.