Inspection of Multilingual Neural Machine Translation

Carlos Mullov, Jan Niehues, Alexander Waibel
Institute for Anthropomatics and Robotics
KIT - Karlsruhe Institute of Technology, Germany
uvdxd@student.kit.edu, jan.niehues@kit.edu, alexander.waibel@kit.edu

Abstract

In this paper we inspect the intermediate sentence representation in the multilingual attention-based NMT system proposed by Ha et al. (2016). We ask the question of how well the NMT system learns a shared representation across multiple languages, as such a shared representation is an important prerequisite for zero-shot translation. To this end we examine whether the sentence representation is independent of the individual languages involved in translation. Having found the sentence representation in our multilingual NMT system to be language dependent, we further inspect the sentence representation for the cause of this dependence. We isolated the language dependent features, and found present a linear correlation between the sentence representation and its source language. Using these isolated features, we describe a method to manipulate these features, and provide a way to eliminate the language specific differences between the sentence representations. This could potentially help to remove noise, which is particularly harmful for zero-shot translation.

Keywords: multilingual, neural machine translation, neural representation analysis

1. Introduction

Since the introduction of neural machine translation (NMT) in recent years, the field of machine translation has made significant progress. However, current NMT systems require large amounts of data for training, while there is a severe lack of data for most language pairs. Therefore, for such language pairs workarounds such as translation using a pivot language are needed. To address this problem Ha et al. (2016) have proposed to extend the originally bilingual attention-based NMT system (Bahdanau et al., 2014) to multilingual translation. Without changes to the network architecture, a single NMT system jointly learns to translate from multiple languages to multiple languages. The attention-based NMT system (Bahdanau et al., 2014) is an encoder-decoder architecture with an attention layer in between the encoder and the decoder, and in the translation process it produces an intermediate sentence representation, the so called *context vectors*. By extracting common semantics across multiple languages, the multilingual NMT system is expected to learn in its sentence representation a shared representation across these languages, as a result significantly reducing the amount of training data needed for each individual language pair, and in the extreme case even enabling zero-shot translation. This shared representation for languages closely resembles an open-domain interlingua, which is a linguistic concept that has often been considered impossible to achieve. Ha et al. (2016) have shown in their experiments, that while zero-shot translation is possible with their approach, there is a significant drop in translation quality.

With the aim to find clues as to what leads to this drop in quality, in this paper we examine how well the NMT system proposed by Ha et al. (2016) learns a shared representation by measuring how independent of the individual languages the sentence representation in the NMT system is. Furthermore, having found in our experiments that the sentence representation is language dependent, we explore the cause for this dependence. We discover a linear correlation between the sentence representation and the individual languages, and find a method to manipulate this correlation in a stable manner. Finally, we demonstrate that through this manipulation we successfully eliminate language dependent linear differences in the sentence representation. This manipulation of the sentence representation could potentially provide a way to manipulate the sentence representation in the process of translation, and effectively reduce noise for zero-shot translation.

2. Related Work

2.1. Multilingual Attention-Based Translation

Based on the attention-based encoder-decoder architecture described by Bahdanau et al. (2014), Ha et al. (2016) have proposed an approach to extend this architecture to multilingual translation. The idea is to train the NMT system using a unified vocabulary and training corpus across all languages, while making no modifications to the architecture. For this, Ha et al. (2016) describe two techniques in the form of input pre-processing steps:

- **Language-specific Coding** words of different languages are distinguished through language codes. This, for example, can look like the following: *bank* → *@**en**@bank*
- **Target Enforcing** A symbol indicating the desired target language of the translation is added to the beginning and at the end of the sentence.

Shared Embedding The approach of using shared vocabularies across multiple languages also results in a shared embedding space. As Ha et al. (2016) have shown in their experiments, the NMT system learns to correlate words of different languages in this shared embedding space in such a way, that words with similar meanings end up closer to each other.

The goals of this approach are to

- improve translation quality for individual languages, by letting the NMT system learn common semantics across languages, thus helping the system to better generalize

- improve translation quality for language pairs, for which parallel training data are scarce, and in the extreme case even allowing for zero-shot translation, by letting the NMT system find a representation of the sentences, which abstracts from the individual languages
- reduce the amount of translation systems needed for translation between n languages from $n\,(n-1)$ to a single one, thus reducing the amount of training time and the amount of parameters

2.2. Inspection of Neural Sentence Representations

Prior to the introduction of attention to the encoder-decoder architecture Cho et al. (2014), Sutskever et al. (2014) and Shi et al. (2016) among others have inspected the encoder sentence representation. The former two have explored the ability of the encoder to represent sentences at the level of their meaning by comparing the relative positions of sentences close to each other in terms of meaning. In their visualization of selected few sentences by means of a 2-dimensional PCA projection Sutskever et al. (2014) show a discernible additive relation between sentence representations, closely resembling the relation between words in word embeddings. This indicates that the encoder in their NMT system does indeed have the capability of abstracting from language and representing the translated sentence on a semantic level.

Shi et al. (2016) have inspected the sentence representation on a syntactic level and have found that the encoder implicitly learns to store information about the source sentence syntax in the sentence representation.

This sentence level representation, which Cho et al. (2014) call *summary* is the equivalent to the context vectors in the attention-based model, with the difference being that context vectors represent only the part of the sentence it puts attention to.

2.3. Inspection of Context Vectors in the GNMT System

Based on the same principle as (Ha et al., 2016) multilingual NMT system, Johnson et al. (2016) have proposed a multilingual attention-based encoder-decoder NMT system. In the course of their experiments Johnson et al. (2016) have inspected the intermediate sentence representation of the translated sentences in their NMT system in respect to its resemblance of an interlingua representation. This sentence representation they call the *attention vectors*, and is equivalent to what we call context vectors in this paper. Using a t-SNE projection into three-dimensional space, Johnson et al. (2016) observe that attention vectors for semantically identical sentences form clusters. Thus they visually confirm, that their NMT system learns to organize sentence representation by their meaning, which they call "early evidence of shared semantic representations (interlingua) between languages". Furthermore Johnson et al. (2016) have found a correlation between the translation quality for these semantically identical sentences of different languages and the similarity between the attention vectors for these sentences.

2.4. Generative Adversarial Networks

Generative Adversarial Networks (Goodfellow et al., 2014) are a type of neural network, used in an approach for training generative models in an unsupervised fashion. It consists of a generative network G, which tries to generate data and a discriminative network D, which tries to differentiate between data generated by G and the training data. G is then trained to "trick D into thinking" that the data generated by G originates from the training data by maximizing the error for D. In this adversarial manner G is trained to produce data which is indistinguishable from the actual training data.

The approach of letting a discriminating network D classify the output of another network G is similar to the procedure used in this paper: we build a discriminator D on top of the attention mechanism of a NMT system G, while ideally looking for D to fail in its classification task. Unlike with the approach with GANs however, we do not take the next step of adjusting G to maximize the error for D. We describe the potential future work on this matter in Section 5.1..

3. Inspection of Context Vectors

Prior to the introduction of attention to encoder-decoder NMT systems (Cho et al., 2014), a source sentence read by the encoder was encoded into a fixed length vector. The decoder then generated the target sentence having only seen this fixed length vector, forcing the encoder to find a meaningful sentence representation containing all the semantic information in the source sentence. With attention this meaningful representation has moved from this single fixed length vector to a set of multiple vectors, the so called context vectors; the principle however stays the same. The addition of multiple languages to the source and target side of the encoder and decoder as proposed by Ha et al. (2016) increases the problem complexity while keeping the amount of parameters constant, thus compelling the network to generalize by using common semantics between languages. Under such circumstances the NMT system would ideally learn a purely semantic representation of sentences, while abstracting from the individual source and target languages. This principle of translating a sentence into its language-independent meaning is known in linguistics as an interlingua representation, and the idea has been known for many centuries. As the context vectors are strongly reminiscent of such an interlingua representation, this begs the question of how close to an interlingua it is. In other words, we want to know how well the NMT system learns a shared semantic representation of sentences across multiple languages.

One criterion for evaluating how well the NMT system learns this shared representation is to look at the independence of the sentence representation – the context vectors – from the individual languages. In Section 3.1. we describe how we measure this degree of independence for the context vectors in the proposed NMT system.

Based on our experiments we have found the context vectors in our NMT system to be language dependent. In Section 3.2. we describe how we – while exploring the cause for the dependence – isolated the language dependent features in the context vectors. Furthermore we describe how we use the result to manipulate the context vectors. Finally we describe how we confirm that using this manipulation we can eliminate the linear language specific differences between context vectors. This could potentially be applied in zero-shot translation in order to change context vectors of a language pair which the NMT system never saw during the training to take on the form of context vectors which the NMT system saw during the training, effectively removing noise in the process of translation.

3.1. Measuring Context Vector Independence

We consider the independence of the context vectors from the individual languages to be a good indicator for how well the NMT system learns the shared representation. This is because, assuming that the NMT system perfectly learns a shared representation, then sentences of different languages would arrive at the same representation and would thus be indistinguishable in terms of the languages involved. Considering the fact that showing the independence of the context vectors from the source and target language would require a formal proof, we approach the problem by studying the dependency, which, if present, can be discovered through experimental means. Given a context vector, we try to identify the language pair it was generated from, and declare the dependence in the case of success. As this problem can be formulated as finding a correlation between a vector $c \in \mathbb{R}^n$ and one language from a set of candidates $\{l_1, \ldots, l_k\}$, this calls for classification. Given the recent success of neural networks in discriminative tasks with high dimensional input, we approach this particular classification problem with classification via neural networks.

Neural Classification Due to the nature of our input we believe a simple feed-forward neural network (FFNN) with fully connected layers to be the most appropriate type of network as the basis for the classifier. Using supervised learning, the classifier is trained to predict the correct source-target language pair, by providing context vectors as input and their respective true source-target language pair as the label. Starting with the simplest approach we will first attempt linear classification using a network without any hidden layers. We call this type of network a *single layer perceptron* (SLP), and this type of classification *linear classification*. The capability of such a network to successfully detect the presence of a dependence would suggest a linear separability of the context vectors, allowing the network to partition the feature space into relevant classes. After the linear classification we will attempt a classification using an MLP-classifier with one or more hidden layers to look for nonlinear features.
The labels will be encoded as concatenation of two one-hot vectors, the first vector encoding the source language and the second vector encoding the target language. The classifiers will then be trained using a softmax-layer as the output layer and cross entropy error between the first and second half of the network output and labels:

$$\begin{aligned} o_s &= \mathrm{softmax}(D(x)_{1\ldots5}) & o_t &= \mathrm{softmax}(D(x)_{6\ldots10}) \\ t_s &= t_{1\ldots5} & t_t &= t_{6\ldots10} \\ E_s &= H(o_s, t_s) & E_t &= H(o_t, t_t) \end{aligned}$$

$$E = \frac{E_s + E_t}{2}$$

for the classifier D and the input-label pair (x, t). The network is then trained to minimize E using *adam* (Kingma and Ba, 2014) as optimizer. The classifier predicts the language pair using the $argmax$ of the output first and second half:

$$(L_s, L_t) = (argmax(D(x)_{1\ldots5}), argmax(D(x)_{6\ldots10}))$$

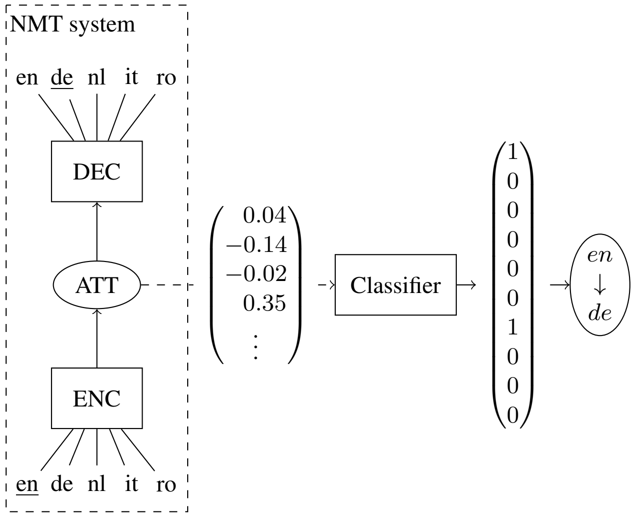

Figure 1: Schematic description of the procedure we use to test for the dependency of the context vectors. During translation the context vectors are extracted from the attention module, and fed to the classifier. The classifier output is a "two-hot" vector representing the predicted source and target language in the its first and second half (every language is assigned a fixed dimension, e.g. English in dimension 0,. . .).

3.2. Investigating Linear Relation of Context Vectors

The results of our experiments have shown that a linear classifier is capable of correctly classifying the context vectors. As described in Section 3.1., this suggests the linear separability of the context vectors. Suspecting the existence of a linear translation, such that a context vector of one language can be obtained when applied to a context vector of another language (similar to the additive relation between words with word embeddings), we decided to further investigate this matter.
Taking context vectors x with s_1 as their source language and t_1 as their target language, and another language s_2 we will try to find a vector b, such that $x + b$ is recognized

as a context vector with s_2 as its source language. In order to find such a vector b for the context vector x we will need a comparable context vector x' which has s_2 as its source language. To obtain this counterpart x' for x, we need to ensure that two matching source sentences f_1 and f_2 in our parallel corpus with s_1 and s_2 as their respective language, are both translated into the same target sentence in the language t_1, allowing for the direct comparison of the generated context vectors. To this end we will adjust the NMT system sampling mechanism to accept a reference target sentence $e = (e_1, \ldots, e_m)$ and use e_{t-1} as the input for the decoding step t, instead of the previously generated target word y_{t-1}. With this method we will obtain a pair of context vectors (x_t, x'_t) for each target word e_t, which can then be used as an input and its label in the training of b.

Using a training set with German-English context vectors for the input, and matching Dutch-English context vectors as labels, we will again use gradient descent to train a translation by randomly initializing a vector b and then minimizing the *summed squared error*

$$sse(x + b, t) = \frac{1}{2} \sum_i (x_i + b_i - t_i)^2$$

for each input-label pair (x, t).

The resulting translated vectors $x + b$ will then be fed to the previously trained classifier, in order to see whether they are recognized as Dutch-English context vectors. Furthermore, in order to see how this translation affects context vectors of different target languages, we will apply this translation b to German-Italian context vectors. We expect the resulting vectors to be classified as Dutch-Italian context vectors.

Eliminating Language Specific Differences After finding the linear translation b_{AB} which translates from the original source language A to the new source language B, we can eliminate the source language specific differences for context vectors c and c' with A and B as their respective source language, by translating c to $c+b_{AB}$. To confirm that the new context vectors are indeed language independent (at least linearly), we can again train a classifier as described in Section 3.1. using the modified context vectors. To this end, we (as illustrated in figure 2) take context vectors of different language pairs and translate each of them to English-German context vectors. We then train classifiers to predict the original language pair for these translated context vectors to see whether the classifiers are still able to differentiate between context vectors of different languages.

4. Evaluation

4.1. Multilingual Translation Models

In order to inspect the context vectors we have trained translation models as described by Ha et al. (2016).

Training Data For training we have used the multilingual WIT^3 (Cettolo et al., 2012) training corpus, which provides high quality multilingual translations. This corpus consists of transcriptions of 200,000 English sentences from TED Talks, and their translations into German, Dutch, Italian and Romanian. We have trained the NMT system using every possible combination of source and target languages, except for language pairs where the source and target language are the same. This gives us 20 language pairs, and therefore 20 valid classes of context vectors and a total of 4,347,886 sentence pairs. For the purpose of evaluation we have used another development dataset set consisting of 900 sentences from TED Talks involving the same five languages.

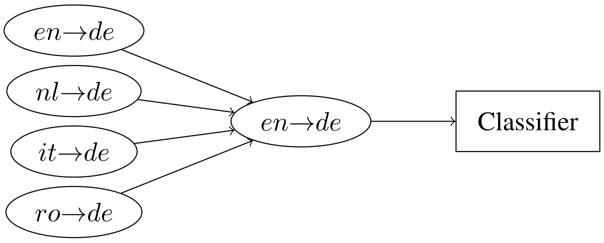

Figure 2: To see whether we can eliminate language specific differences between context vectors, we train classifiers to differentiate between translated context vectors.

Translation System We have trained the translation models using Nematus (Sennrich et al., 2017), which provides an attention-based encoder-decoder NMT system as proposed by Bahdanau et al. (2014). Nematus uses a special type of RNN, which Sennrich et al. (2017) call *conditional GRU with attention* in its decoder. As alignment model Nematus uses a feed-forward *tanh*-layer, which is jointly trained together with the rest of the system. We have used subword translation units.

In order to achieve multilinguality as described by Ha et al. (2016), all 20 parallel corpora have been merged into one multilingual parallel corpus through the concatenation of the source and target texts in the same respective order. The source and target vocabularies have been built from the merged corpus after applying every pre-processing step. The pre-processing steps include:

1. tokenization
2. true casing
3. byte pair encoding
4. language specific encoding
5. target enforcing

As we assume, that a smaller hidden layer size will force the network to abstract from the source language even more, we have trained models with different sizes of hidden layers in order to test whether this holds true. Furthermore, in order to compare how classification results evolve with the ongoing training of the translation system, we also evaluate one particular model at different checkpoints in the course of training.

Model Parameters For the evaluation task we have trained three translation models. We trained all models using the same setup and network parameters, differing only in the amount of training time and the hidden layer sizes, which is 1024 for the first model, and 512 for the second and third models. We used English, German, Dutch, Italian and Romanian as source as well as target languages, whereby words were pre-processed into subword units with BPE[1], using a BPE merging parameter of 39,500 trained on the merged corpus before applying language specific encoding, resulting in a total vocabulary size of 88,000 words. We used a maximum target sentence length of 50 and a word embedding size of 500.

Training We have trained all models using a batch size of 40, *adam* as optimizer and dropout training (Srivastava et al., 2014), with a dropout ratio of 0.2 for the embedding layers and hidden layers and 0.1 for the source and target layers. Since we use a concatenation of the bidirectional encoder forward and backward hidden states as annotation vectors, this resulted in context vectors of size 2048 for the first model and 1024 for the second and third models.
The first model, with the hidden layer size of 1024 was trained for 110,000 iterations, until coming to an early stop. This resulted in a final BLEU score of 11.94. Another model, with a hidden layer size of 512 was trained for 160,000 iterations, resulting in the second model. Training the second model for another 100,000 iterations, resulted in the third model after a total of 260,000 iterations. These models achieved BLEU scores of 11.07 after 160,000 iterations and 14.94 after 260,000 iterations (see Table 1).
All translations used for calculating BLEU scores were generated using beam search decoding, with beam size 5.

trg / src	en	de	nl	it	ro	avg
en		17.82	16.17	17.27	14.75	16.50
de	23.21		13.84	12.07	9.65	14.69
nl	20.42	12.46		12.11	9.44	13.61
it	22.84	12.12	12.35		11.16	14.62
ro	22.85	11.87	12.23	14.10		15.26
avg	22.33	13.56	13.64	13.88	11.25	14.94

Table 1: BLEU scores for the third NMT model (hidden layer size 512 and 260,000 training iterations). The distribution of scores for specific language pairs is also representative for the other trained models.

4.2. Classification of Language Pairs

Using the development dataset as translation source, we generated and extracted the context vectors and the correct language pairs from *Nematus* using the previously trained models. This resulted in 459,878 context vectors for the first model, 457,054 vectors for the second model, and 444,570 vectors for the third model, with overall 20 classes. The number of context vectors matches the number of translation symbols in the generated target sentence, and thus differs for each translation model, since they produce different translations. We merged the data from each class into one sequence by alternating between single sentences of each class, ensuring that each mini batch of size 1000 to contained samples of all 20 classes, considering the sentence maximum length of 50.
Using the *TensorFlow* (Abadi et al., 2016) low level API we built for each model SLP-classifiers, and MLP-classifiers with one hidden layer containing 64 hidden neurons. As the hidden layer activation function we use *ReLU*. All classification accuracies were calculated as ratio of correctly predicted language pairs, using 25% of the context vectors as the validation set.

Results As illustrated in Table 2 all classifiers have achieved significantly high classification accuracies, with linear classification achieving 86-96% correct classification rates after 50 epochs of training, and slightly higher rates for their nonlinear counterparts. These values are also representative for the classification rates of the source languages alone, as the target languages have been correctly classified with near 100% accuracy by all the classifiers.

NMT model	linear	MLP
first	95.75%	96.09%
second	85.93%	89.94%
third	91.93%	94.29%

Table 2: Classification rates for the linear classifiers and the MLP-classifiers after 50 epochs of training. *NMT model* refers to the model which was used for generating the context vectors.

These results strongly suggest that the extracted context vectors are not independent of the source language. The high classification rates of the linear classifiers further suggest a linear relation between context vectors of different languages. In view of the fact, that the classification rates for the linear classifier increase with ongoing training of the NMT system, it is apparent that the linear features which the classifier makes use of become more distinctive with the progression of the training.
The confusion matrix (see Table 3) shows a discernible correlation between the language specific BLEU scores and classification errors for that language, Romanian being the language with lowest BLEU scores, as well as with most classification errors. Furthermore, there is also a noticeable correlation between language similarity and the classifiers tendency to confuse them with each other, as can be seen with Dutch being commonly misclassified as German and

label / pred	en	de	nl	it	ro
en	22398	25	21	15	42
de	32	21818	129	19	24
nl	24	326	24784	43	89
it	171	46	74	19728	2490
ro	85	54	94	628	20841

Table 3: The Confusion matrix shows a comparison between predicted source languages and labels

[1] for BPE we used the script from the subword-nmt repository (`https://github.com/rsennrich/subword-nmt`). For tokenization and true casing we used the scripts provided by the Moses framework (`http://www.statmt.org/moses/`)

source language	original	translation
en	0	0
de	1268	133
nl	100	1246
it	32	16
ro	13	18

Table 4: Comparison of predicted source language for German to English context vectors, after training and applying translation of source language to Dutch

source language	original	translation
en	4	1
de	1233	172
nl	18	1081
it	0	0
ro	2	3

Table 5: Comparison of predicted source language after applying the same translation to German to Italian context vectors

Romanian misclassified as Italian.

For the classification with the MLP we can observe slight increases in classification rates.

4.3. Linear Relation between Context Vectors

To investigate the supposed linear relation between the context vectors of different languages, we successfully trained a translation as described in Section 3.2.. For this we first modified *Nematus* to accept a reference target sentence for translating a source sentence.

Using the German and Dutch translation of the development dataset as translation source and the English translation as the reference, we generated a training set with the German-English context vectors as the input and the Dutch-English context vectors as the labels, using the third translation model. Using *adam* as optimizer we then trained a vector b, by minimizing the *summed squared error*. The training of such a translation for 20 epochs resulted in a vector b with a norm of 0.710 and a mean distance of 6.912 between the translations and their labels, the mean distance between untranslated German-English vectors and their Dutch-English counterparts being 6.939 (see Figure 3).

Applying this trained translation to a validation set unseen in training, we further classified the originally German-English context vectors with the help of the previously trained linear classifier. As seen in Table 4, 88% of the translated vectors were classified as Dutch-English. Furthermore, the application of this translation to German-Italian context vectors, resulted in 86% of these to be classified as Dutch-Italian (see Table 5). Applying this same procedure for all 180 valid four-tuples of languages[2], from a total of 752,841 context vectors for 90.9% the source language was successfully translated as intended, while not affecting the predicted target language for these context vectors.

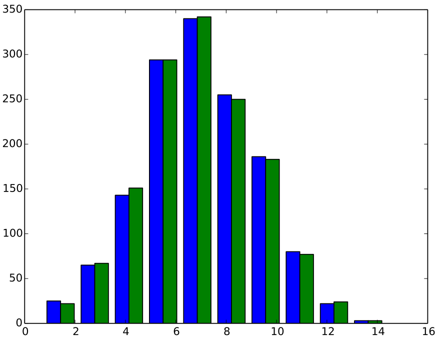

Figure 3: The distribution of distances between original (blue) and translated (green) German to English context vectors to their correspondent Dutch to English context vectors shows, that the translation has negligible impact on the distances.

4.3.1. Eliminating Linear Dependence

As described in Section 3.2. in order to confirm that the found linear translations can be used to eliminate the linear dependence of the context vectors, we have retrained our classifiers using modified context vectors. For the training set we took all the context vectors with German as their target language, and translated them to English as the new source language. Analogous to the procedure used in Section 3.1. we then trained classifiers to predict the original target language. For the linear classifiers this resulted in classification rates around 25% for 4 classes, thus failing at the classification task. This indicates that we successfully eliminated the linear dependence on the source language.

The MLP-classifiers achieved classification rates of up to 83.3%, showing the presence of purely non-linear language dependent features.

5. Conclusion

In this paper we have explored the ability of the multilingual NMT system proposed by Ha et al. (2016) to produce a shared representation across multiple languages. To this end we have inspected the intermediate sentence representation of the NMT system, the context vectors. We took as criterion for how well the NMT system learns a shared representation the degree of independence of the context vectors from the source and target languages involved in translation. In order to measure the dependence, we have trained classifiers based on feed-forward neural networks which, given a context vector, predict the language pair involved in its generation. Using the context vectors generated by our trained multilingual NMT systems, our linear

[2] For the original source language s, the translated source language s', the target language used in training the translation t, and the target language t' of the context vectors which the trained translation was applied to, a four-tuple is considered valid if s, s', t are pairwise different and s, s', t' are pairwise different, resulting in 60 different translations, which are each applied to context vectors of 3 different target languages.

classifiers have achieved rates of correct classification of up to 95.75% for 25 possible classes. This suggests that our NMT system does not successfully produce a shared representation.

Having explored the underlying cause of success in classification, we have found present a linear relation in Euclidean space between context vectors of different languages. More precisely, for a pair of languages (A, B) we have found a vector b_{AB}, such that for a context vector c with A as its source language, $c + b_{AB}$ is classified as having B as the source language in 90.9% of the cases. This translation of the source language does not affect the target language. We have found this translation to have negligible impact on the distance between the context vectors, which leads us to the belief that these language dependent differences in context vectors are merely noise, and particularly harmful for zero-shot translation. Finally we have demonstrated that our linear classifiers, which we trained on context vectors with modified source language fail in their task to classify the original source language. This shows that we can use translations found to effectively eliminate the language specific linear differences between context vectors.

5.1. Future Work

Adversarial Training of NMT System As described in Section 2.4., the approach used in this paper is similar to the first stage in the procedure to train generative models with GANs. As GANs have shown great success, the remaining steps of this procedure could be applied to the training of the NMT system as well. By training the NMT system G to produce context vectors for which a discriminating network D is unable to predict the correct language pair in an adversarial manner, the NMT system could learn to produce indistinguishable context vectors. The NMT system would then be alternatingly be trained in supervised learning and adversarial unsupervised learning, potentially learning a language independent representation.

Zero-Shot Translation The linear translation which we have found to be present between the context vectors could be potentially applied in order to improve zero-shot translation. Zero-shot translation is the task of producing translations for a language pair which the NMT system never saw during the training. For a language pair (A, C), which the NMT system never saw during the training, and a language pair (B, C), which the NMT system saw during training an attempt at improving translation quality for the unseen language pair could be made by translating the context vectors c_{AC} to $c_{BC} = c_{AC} + b_{AB}$ for the previously described translation b_{AB}. This produces context vectors for a language pair which the NMT system is more familiar with.

6. Bibliographical References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I. J., Harp, A., Irving, G., Isard, M., Jia, Y., Józefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D. G., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P. A., Vanhoucke, V., Vasudevan, V., Viégas, F. B., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Networks. *ArXiv e-prints*, June.

Ha, T., Niehues, J., and Waibel, A. H. (2016). Toward multilingual neural machine translation with universal encoder and decoder. *CoRR*, abs/1611.04798.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F. B., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2016). Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Barone, A. V. M., Mokry, J., and Nadejde, M. (2017). Nematus: a toolkit for neural machine translation. *CoRR*, abs/1703.04357.

Shi, X., Padhi, I., and Knight, K. (2016). Does string-based neural mt learn source syntax? In *Proceedings of EMNLP 2016*, pages 1526–1534, 01.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

7. Language Resource References

Mauro Cettolo and Christian Girardi and Marcello Federico. (2012). *WIT^3: Web Inventory of Transcribed and Translated Talks.*