Adapting Neural Machine Translation with Parallel Synthetic Data

Mara Chinea-Ríos and **Álvaro Peris** and **Francisco Casacuberta**
Pattern Recognition and Human Language Technology Research Center
Universitat Politècnica de València, València, Spain
{machirio, lvapeab, fcn}@prhlt.upv.es

Abstract

Recent works have shown that the usage of a synthetic parallel corpus can be effectively exploited by a neural machine translation system. In this paper, we propose a new method for adapting a general neural machine translation system to a specific task, by exploiting synthetic data.

The method consists in selecting, from a large monolingual pool of sentences in the source language, those instances that are more related to a given test set. Next, this selection is automatically translated and the general neural machine translation system is fine-tuned with these data.

For evaluating the adaptation method, we first conducted experiments in two controlled domains, with common and well-studied corpora. Then, we evaluated our proposal on a real e-commerce task, yielding consistent improvements in terms of translation quality.

1 Introduction

Neural machine translation (NMT) (Sutskever et al., 2014; Cho et al., 2014a; Bahdanau et al., 2015) has obtained state-of-the art performance in several domains and language pairs (Sennrich et al., 2016b; Wu et al., 2016). Given the nature of NMT paradigms, the limitation for obtaining bilingual corpora—or their availability—has been one of the major obstacles faced when building competitive NMT systems. Recently, the idea of using synthetic corpora in NMT has reported promising results with regard to the data scarcity in NMT. Many different works demonstrated that the combination of real parallel corpora with synthetic bilingual corpus enhances the NMT translation quality (Sennrich et al., 2016a; Zhang and Zong, 2016a; Cheng et al., 2016).

Following these good results, we aim to adapt general NMT models to real, specific tasks by using synthetic parallel data. The core idea is to select the most valuable instances from a large pool of monolingual source sentences, with respect to a given test set. Next, we automatically translate them. Therefore, we obtain a synthetic parallel corpus, related to our test set domain. Such synthetic corpus can be used to fine-tune a NMT system to the domain at hand.

The main contributions of this paper involve the necessary steps required to adapt a NMT system to a specific domain:

- We propose a novel method to create the most adequate synthetic corpus leverages a vector-space representation of sentences, relying on the word embeddings by Mikolov et al. (2013a) and Le and Mikolov (2014).
- We describe the pipeline of our adaptation process, relating the selection, translation and fine-tuning processes.
- We study our adaptation technique on two classical domains. Additionally, we validate our technique on a real e-commerce translation task.
- Results show important improvements over a baseline system.

This paper is structured as follows. NMT technology is briefly described in Section 2. Section 3 summarizes the related work. In Section 4, we present our selection method and we describe the adaptation pipeline. Section 5 presents the experimental set-up and corpora. Results are analyzed and discussed in Section 6. Finally, conclusions and future work are traced in Section 7.

Proceedings of the Conference on Machine Translation (WMT), Volume 1: Research Papers, pages 138–147

2 Neural Machine Translation

Neural machine translation is an instantiation of sequence-to-sequence learning: given a sequence of words in the source language, we must produce the corresponding sequence of words in the target language. This is usually done by means of the encoder–decoder architecture: the encoder computes a representation of the input sequence, while the decoder takes it and generates, word by word, the sentence in the target language (Sutskever et al., 2014). In this work, we use a NMT system featuring long short-term memory (LSTM) units (Hochreiter and Schmidhuber, 1997)—in both the encoder and decoder—and equipped with an attention mechanism (Bahdanau et al., 2015).

The input to the system is a sequence of words in the source language. A word embedding matrix projects each word from the discrete to a continuous space. The sequence of word embeddings is then processed by a bidirectional (Schuster and Paliwal, 1997) LSTM network, which produces a sequence of annotations by concatenating the hidden states from the forward and backward layers.

At each decoding timestep, the attention mechanism computes a weighted mean of the sequence of annotations. The weights are given according to a soft alignment model that weights each annotation with the previous decoding state. This can be seen as a joint, dynamic representation of the input sentence.

The decoder is another LSTM network, conditioned to the representation computed by the attention model and the previously generated word. Finally, a deep output layer (Pascanu et al., 2014) computes a distribution over the target language vocabulary.

The model is jointly trained by stochastic gradient descent (SGD), aiming to maximize the log-likelihood over a bilingual parallel corpus. At decoding time, the model approximates the most likely target sentence with beam-search (Sutskever et al., 2014).

3 Related work

Since Kalchbrenner and Blunsom (2013), Sutskever et al. (2014) and Cho et al. (2014b) proposed the first NMT systems, this has been a boiling research topic. A singular effort has been spent into leverage the advantages that this technology brings in. One of them is the ability of NMT systems to rapidly adapt to a given domain, when they are already trained on a general domain. This is useful either for creating domain-dependent NMT systems or for low-resource tasks. Thus, Luong and Manning (2015) tackled the informal speech translation task by starting from a system trained on the WMT data and adapting it to the translation task at hand.

In phrase-based statistical machine translation (SMT), synthetic bilingual corpora have been mainly proposed as a mean to exploit the vast amount of monolingual data available. By applying a self-training scheme, the synthetic parallel data can be obtained by automatically translating a source-side monolingual corpus (Ueffing et al., 2007; Wu et al., 2008). Other works used target-side corpora to build the synthetic parallel corpus (Bertoldi and Federico, 2009; Lambert et al., 2011).

Inspired by these works in SMT, research referring the inclusion of monolingual data in NMT has a growing interest. Different works have tackled the inclusion of monolingual data, either in source (Zhang and Zong, 2016b) and target language (Gulcehre et al., 2015, 2017).

Moreover, Sennrich et al. (2016a) showed that parallel data is not strictly necessary for performing domain adaptation: the usage of synthetic data has positive effects on the NMT system. For obtaining the synthetic data they automatically translated a large monolingual corpus. This synthetic-based approach obtained better results than other methods aimed to exploit monolingual data (e.g. Gulcehre et al. (2015)). Domain adaptation in NMT systems is also integrated in commercial systems, such as SYSTRAN (Crego et al., 2016).

4 Adaptation using synthetic corpus

As described in the previous section, synthetic parallel data have been widely used to boost the translation quality of NMT. In this work, we further extend their application by adapting NMT models with synthetic parallel data. In certain language pairs or domains where parallel corpora are scarce or even non-existent, a model adjusted with synthetic data can improve the performance with respect to a more general model.

The core idea is that, once a model has been trained on a large, general corpus, we can adapt it to a new domain, by fine-tuning it exclusively using the synthetic data. For doing this, we create an

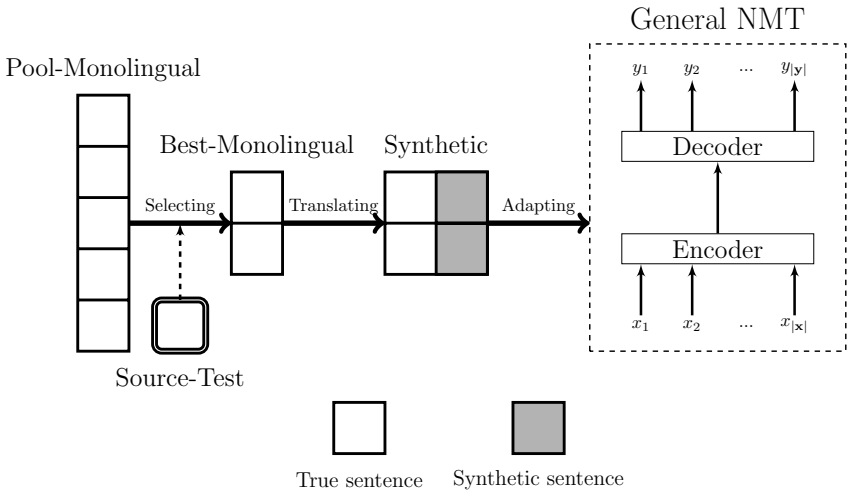

Figure 1: The process of building an adequate synthetic parallel corpus for a given test set.

ad-hoc, specific synthetic corpus in which appear the features from our target-domain data. This corpus is constructed by selecting from a large monolingual pool of sentences—in the source language—those instances that are related with our in-domain dataset. Next, we automatically translate these sentences into the target language. Finally, using this synthetic corpus, we fine-tune a NMT system trained on a more general domain. Figure 1 shows the pipeline of our adaptation process.

In this section, we describe our technique for creating adequate synthetic corpora, based on a vector-space representation of sentences, and the NMT adaptation process.

4.1 Continuous vector-space representation

The idea of representing words or sentence in a continuous vector-space employing neuronal networks was initially proposed by Hinton (1986) and Elman (1990). Continuous vector-space representations (CVR) of words or sentences have been widely leveraged in a variety of natural language applications and demonstrated solid results across a variety of tasks, such as speech recognition (Schwenk, 2007), part-of-speech tagging (Socher et al., 2011), sentiment classification and identification (Glorot et al., 2011) or machine translation (Cho et al., 2014a; Mikolov et al., 2013b).

In this paper, we use a sophisticated CVR of the sentences involved in our data selection method. Specifically, we follow the CVR approach presented by Le and Mikolov (2014). In this work, the authors adapted the continuous Skip-Gram model (Mikolov et al., 2013a) to generate representative vectors of sentences and documents. Thus, with this technique, we obtain a particular vector that represents a complete sentence by means of the the Skip-Gram architecture.

4.2 Synthetic creation method

For creating an adequate synthetic corpus for adapting a NMT system, we select from a large pool of monolingual text the most related sentences for our task at hand. We present a novel selection technique, based on the CVR of the sentences.

The intuition is to select sentences whose vector-space representation is similar to the representation of our in-domain instances, assuming that similar sentences will have similar vectors (Le and Mikolov, 2014).

Having a continuous vector space representation of the test sentences allows us to compute a centroid. This can be seen as prototype of the sentences present in the test set.

Provided that similar sentences have similar vector-space representations (Mikolov et al., 2013b), we assume that vectors from the in-domain corpus will be clustered. On the other hand, vectors from the general pool of sentences are likely to be more disperse. The idea of our method is to create a hypersphere in the continuous space, with center in our test set centroid, containing all sentences from the test set. Hopefully, only a selection of the sentences from the general pool will be contained in this hypersphere. The hyper-sphere radius is established according to some similarity metric between the centroid of

the test set, and the furthest of the test sentences.

As similarity metric we consider the cosine similarity, defined as:

$$cos(\mathbf{F}_1, \mathbf{F}_2) = \frac{\mathbf{F}_1 \cdot \mathbf{F}_2}{\|\mathbf{F}_1\| \cdot \|\mathbf{F}_2\|} \quad (1)$$

where $\mathbf{F}_1$ and $\mathbf{F}_2$ are two z-dimensional vectors.

The centroid is defined as an average of the representations of the sentences from our in-domain corpus $\mathcal{T}$ (made up of T sentences):

$$\mathbf{F}_{\mathcal{T}} = \frac{1}{|T|} \sum_{t}^{|T|} \mathbf{F}_{\mathbf{x}_t} \quad (2)$$

where $\mathbf{F}_{\mathbf{x}_t} \in \mathbb{R}^z$ is the z-dimensional representation of the sentence $\mathbf{x}_t$ and $\mathbf{F}_{\mathcal{T}} \in \mathbb{R}^z$ denotes the centroid of our test set.

Data: Pool $\mathcal{P}$; test data $\mathcal{T}$
Result: Source synthetic corpus $\mathcal{S}$
1 $\mathbf{F}_{\mathcal{T}} := centroid(\mathcal{T})$;
2 $\rho := \infty$;
3 $\mathcal{S} := \emptyset$;
4 **forall** $\mathbf{x}_t \in \mathcal{T}$ **do**
5 **if** $cos(\mathbf{F}_{\mathbf{x}_t}, \mathbf{F}_{\mathcal{T}}) \leq \rho$ **then**
6 $\rho := cos(\mathbf{F}_{\mathbf{x}_t}, \mathbf{F}_{\mathcal{T}})$
7 **end**
8 **end**
9 **forall** $\mathbf{x}_p \in \mathcal{P}$ **do**
10 **if** $cos(\mathbf{F}_{\mathbf{x}_p}, \mathbf{F}_{\mathcal{T}}) \geq \rho$ **then**
11 $\mathcal{S} \cup \{\mathbf{x}_p\}$;
12 **end**
13 **end**

Algorithm 1: Pseudo-code for selecting synthetic corpora.

Algorithm 1 shows the selection procedure. Here, $\mathbf{x}_t \in \mathcal{T}$, is a sentence from our source test data $\mathcal{T}$; and $\mathbf{F}_{\mathbf{x}_t}$ is the vector-space representation of $\mathbf{x}_t$. Analogously, $\mathcal{P}$ is the pool of candidate sentences, $\mathbf{x}_p \in \mathcal{P}$ is a source candidate sentence, $\mathbf{F}_{\mathbf{x}_p}$ is the vector-space representation of $\mathbf{x}_p$, and $|\mathcal{P}|$ is the number of sentences in $\mathcal{P}$. Then, our objective is to select data from $\mathcal{P}$ such that it is the most suitable for translating data belonging to the source test data $\mathcal{T}$.

Algorithm 1 introduces several functions:

- $centroid(\cdot)$: calculates the centroid (Eq. 2) for the test corpus $\mathcal{T}$.
- $cos(\cdot, \cdot)$: computes the cosine similarity (Eq. 1) between two different vectors.

ρ represents the radius of the hyper-sphere, which is computed in lines 4 to 8 (the first **forall** loop) in Algorithm 1.

4.3 Adapting with the selection

In our adaptation framework, we assume that we have a NMT model trained on a general domain. We also have a large monolingual pool of sentences (in the source language) and the source part of the test set.

As first step, we compute the distributed representation of the sentences in our large pool. Next, we select sentences from the monolingual pool, given the test set, according to Algorithm 1. This subset of sentences are expected to be related with our in-domain test data. We translate them by means of machine translation (see Section 5.3 for further details). Now we have a synthetic parallel corpus, relating our in-domain task. Finally, we fine-tune the general NMT model with these data.

5 Experiments

In this section, we describe the experimental framework employed to assess the performance of the NMT adaptation method described in Section 4. For this purpose, we studied its behavior in three corpora. Two of them refer to controlled tasks; while the last one belongs to a real e-commerce task.

5.1 Corpora

We performed the experiments on English→Spanish translation. Our out-of-domain training data was the Common Crawl (COMMON) corpus which was collected from web sources. We chose the 1 Billion Words corpus (Chelba et al., 2013) as the large pool of monolingual sentences. For validation, we chose the News-commentary test 2013 (dev13) dataset. For testing, we used corpus from three different domains: Xerox printer manuals (XRCE–Test) (Barrachina et al., 2009), Information Technology[1] (IT–Test) and Electronic Commerce (E-Com–Test). This last corpus was obtained from a real e-commerce website (*Cachitos de Plata*[2]). Statistics of all corpora are provided in Table 1.

[1] http://metashare.metanet4u.eu/qtleapcorpus
[2] http://cachitosdeplata.com

Table 1: Corpora main figures, in terms of number of sentences ($|S|$), number of words ($|W|$), vocabulary size ($|V|$) and average sentence length ($\overline{|W|}$).

Corpus		$\|S\|$	$\|W\|$	$\|V\|$	$\overline{\|W\|}$
1 Billion Words	EN	30.3M	800M	800k	26.4
COMMON	EN	1.5M	30M	456k	20.0
	ES		31M	522k	20.0
dev2013	EN	2.7k	48.9k	7.5k	18.1
	ES		52.6k	9.1k	19.5
XRCE – Test	EN	1.1k	8.4k	1.6k	7.6
	ES		10.1k	1.7k	9.2
IT – Test	EN	857	15.6k	2.1k	18.2
	ES		17.4k	2.4k	20.3
E-Com – Test	EN	886	7.3k	874	8.2
	ES		8.6k	973	9.7

5.2 Evaluation

Translation quality was assessed according to the following well-known metrics:

- BLEU (BiLingual Evaluation Understudy) (Papineni et al., 2002), measures n-gram precision with respect to a reference set, with a penalty for sentences that are too short.
- TER (Translation Error Rate) (Snover et al., 2006), is an error metric that computes the minimum number of edits (including swaps) required to modify the system hypotheses so that they match the reference.

For all results, we computed their confidence intervals ($p = 0.05$) by means of bootstrap resampling (Koehn, 2004).

5.3 Machine translation systems

We used NMT-Keras (Peris, 2017) for building the NMT system, as described in Section 2. We applied joint byte pair encoding (BPE) (Sennrich et al., 2016b), learning $32,000$ merge operations, on the out-of-domain dataset. Following the findings from Britz et al. (2017), we used LSTM units. Due to practical reasons, we used single-layered LSTMs. The LSTM, word embedding and attention MLP sizes were 512 each. We applied layer normalization (Ba et al., 2016) and Gaussian noise ($\sigma = 0.01$) to the weights (Graves, 2011). We clipped the L_2 norm of the gradients to 1 (Pascanu et al., 2012). We used Adam (Kingma and Ba, 2014) with a learning rate of 0.0002 (Wu et al., 2016). The size of the beam was set to 6.

We trained further the NMT system using the selected synthetic data. For this training, we used vanilla SGD with an initial learning rate of 0.05. Such hyperparameters were set according the results observed in the development set. From this exploration, we also noticed that the application of more sophisticated SGD optimizers (e.g. Adam) is tricky, as they update the model on a more aggressive way. Therefore, if we apply excessively large updates, the knowledge from the general model is somehow lost.

We also tested our method with ensembles of NMT systems. Ensembles were made up of 4 models sampled at different points of the training process. Such points were evenly chosen (each $2,000$ updates) around the single model which obtained the highest performance on the development set.

Finally, we used Moses toolkit as phrase-based reference (Koehn et al., 2007). We used a 5-gram language model with modified Kneser-Ney smoothing (Kneser and Ney, 1995), built with the SRILM toolkit (Stolcke, 2002). The phrase table was generated employing symmetrised word alignments obtained with GIZA++ (Och and Ney, 2003). The log-lineal combination weights were optimized using MERT (Minimum Error Rate Training) (Och, 2003).

Table 2: Main figures of the selections obtained by Algorithm 1 for each test set ($\mathcal{T}$), employed for adapting the NMT system. $|S|$ denotes number of sentences; $|W|$, number of words; $|V|$, vocabulary size and $\overline{|W|}$, average sentence length.

$\mathcal{T}$		$\|S\|$	$\|W\|$	$\|V\|$	$\overline{\|W\|}$
XRCE – Test	EN	180k	2.2M	54k	9.4
	ES		1.7M	58k	12.2
IT – Test	EN	150k	2.5M	76k	16.7
	ES		3.0M	78k	20.0
E-Com – Test	EN	300k	3.2M	100k	10.6
	ES		4.1M	100k	13.6

5.4 Corpus creation

The process for building synthetic parallel corpora begins with the selection from the monolingual pool. The selection method presented in Section 4.2, requires to set the dimension of the vector-space representation. We set it to 200, according to preliminary research, and it was maintained for all the experiments reported in this paper.

Table 3: Selection examples from each domain.

	Selected sentence
XRCE	id rather send files electronically
	use current antivirus and a firewall
	images are stored on a one terabyte built in hard drive which includes a DVD burner
IT	the technology would also be available to ipod touch users although they would have to buy a microphone and headphones to make calls pc world reported
	if you want to find panorama archive material on delicious the easiest way to search is to use the single word on the right hand column
	my personal have is tweetdeck which although designed for photo uploading amongst other things
E-Com	it is perfect for your collection
	pasta is inexpensive easy and really romantic
	another shows the dust forming into clumps along magnetic lines like pearls on a necklace

Once we obtained the monolingual selections, we translated them. In order to speed up this process, we split the selection and translate it using Moses and NMT. Both systems were trained on the out-of-domain data. In the case of the NMT system, we applied the same BPE subword segmentation to all data. Therefore, the potential vocabulary differences across tasks were effectively leveraged by using subword units.

6 Results and analysis

In this section, we present and discuss the results obtained. We start by analyzing the selection obtained by Algorithm 1. Next, we present the translation results obtained in all tasks. Finally, in order to get some insights of the system behavior, we analyze several representative examples.

6.1 Analysis of the selection

Table 2 shows the features of the selection for each corpus. Note that the average length of the sentences belonging to each selection is tightly related to the sentence length from each test set (Table 1).

Therefore, the selections from XRCE and E-Com had shorter sentences, while the selection obtained from the IT corpus had longer ones. As shown in the following sections, this was a key factor that affected the machine translation systems performance.

Moreover, Table 3 shows some samples from each domain, selected by our selection technique. We can notice that such samples are related to the correspondent test set domain. Thus, sentences from XRCE and IT domains refer to a technological field. As illustrated in Table 2, sentences selected from the IT corpus were notoriously longer than those selected from XRCE. Sentences selected from the E-Com task are related to jewelry or economy. Given the E-Com domain—an electronic shop of silver jewelry—these sentences are also coherent.

6.2 Quantitative results

Table 4 shows the results on the XRCE and IT tasks. The general NMT model performed worse than Moses in out-of-domain tasks. The use of a 4-model ensemble was very helpful. Nevertheless, it still had a lower performance than Moses.

The TER values of the general NMT system in the XRCE task were unusually high. This is due to the corpus features: As shown in Table 1, the XRCE–Test set has an average sentence length of 9 words. The general NMT model generated sentences with an average of 13 words, because it was trained on general-domain data. The TER metric greatly penalizes this behavior, because it must delete the exceeding words. Therefore, TER results of the NMT system were surprisingly high. In the case of Moses, the average sentence length of the sentences generated by Moses was 9.5. Because the generation was bounded by the phrase and language models.

The addition of synthetic data significantly improved the NMT systems, in all cases. Taking the reference of a single NMT model, the gains ranged from 5 to 7 BLEU points. The performance of a single fine-tuned NMT model was also clearly better than fine-tuned ensembles.

Especially critical were the enhancements in terms of TER. In the XRCE task, the synthetic data improved TER by almost 40 and 20 points, for single model and ensembles, respectively. Due to the addition of synthetic data, the system learned to produce shorter translations (around 5 words shorter, in average), and therefore, greatly diminishing TER. In the IT task, the synthetic data also improved TER, but to a lower extent. This is because the IT task is closer to the out-of-domain corpus. Therefore, the adaptation benefits brought by the synthetic data were less crucial than in the XRCE task.

It is worth noting that the adaptation of the

Table 4: Translation results for the XRCE and IT tasks. BLEU and TER results given in percentage. $^{\Sigma}$ denotes an ensemble of 4 neural models. $\overline{|W|}$ is the average number of words per sentence.

	XRCE			IT		
System	BLEU	TER	$\overline{\|W\|}$	BLEU	TER	$\overline{\|W\|}$
Moses	26.2 ± 0.8	59.0 ± 0.8	9.1	33.4 ± 0.6	45.6 ± 0.6	20.4
NMT	20.4 ± 1	94.5 ± 5.1	12.8	29.0 ± 0.8	53.5 ± 0.8	15.3
NMT$^{\Sigma}$	25.5 ± 0.8	76.8 ± 2.0	11.3	31.4 ± 0.8	51.2 ± 0.8	15.3
NMT + Synthetic	27.5 ± 0.8	56.7 ± 0.8	8.6	34.1 ± 0.7	45.7 ± 0.7	17.8
NMT$^{\Sigma}$ + Synthetic	27.3 ± 0.8	56.3 ± 0.8	8.4	33.8 ± 0.7	46.3 ± 0.7	18.1

NMT system was very fast. The system only required to be trained on $\sim 15,000$ samples in order to achieve the best results. Using a GPU, the fine-tuning of the NMT model can be done in minutes.

Table 5 shows the results on the real E-Com task. This was a very specific task. In these cases, the single NMT model also yielded worse performance in terms of BLEU than Moses, but when applying an ensemble, the results were significantly enhanced. In terms of BLEU, even beating Moses.

Table 5: E-Com – Test set results. BLEU and TER results given in percentage. $^{\Sigma}$ denotes an ensemble of 4 neural models. $\overline{|W|}$ is the average number of words per sentence.

	E-Com		
System	BLEU	TER	$\overline{\|W\|}$
Moses	21.1 ± 0.8	56.7 ± 0.7	9.4
NMT	16.9 ± 1.0	104.7 ± 6.3	14.1
NMT$^{\Sigma}$	23.0 ± 1.0	80.8 ± 2.9	12.0
NMT + Synthetic	25.5 ± 1.0	59.1 ± 1.0	8.7
NMT$^{\Sigma}$ + Synthetic	25.8 ± 1.0	61.1 ± 2.6	8.7

The NMT systems behaviored similarly to the XRCE task in terms of TER. The E-Com corpus had similar features than XRCE–Test (in this case, 9.7 words per sentence). Therefore, we observed the same phenomenon: as we introduced in-domain-related sentences, the system learned to produce shorter sentences, diminishing TER consequently.

The use of synthetic data again greatly improved the system. The results were coherent with the previous experiments: A single, fine-tuned model, significantly outperformed the general system (+9 BLEU points). A sole adapted system was even better than a general model ensemble. With respect to Moses, we also found major enhancements in terms of BLEU.

It is also noticeable the ensemble of systems trained with synthetic data did not improve the performance of a single fine-tuned system. This is probably due to the fact that the adaptation was performed from an already trained model and with few data. Therefore, the systems belonging to the ensemble were quite similar, all of them around the same local minimum. Therefore, potential enhancements from the ensembles were diluted.

Finally, we should remark than the E-Com task belongs to a real-world scenario. This corpus is not designed for experimental purposes. It contains elements that distort the experiment, and therefore yield to unpredictable results. In such open scenarios, a human evaluation should be the next step to take.

6.3 Qualitative results

Some translation examples from each corpus are shown in Table 6. In the first example, all the systems presented the similar error at the beginning of the translation (*especificación del*). This is because that was the most likely translation in our corpora, both the real and synthetic ones.

In the second example, Moses was not able to correctly identify the right meaning of the word (*windows*) in the sentence to translate. It should be left untranslated, as it is a proper noun. The NMT systems were able to detect it. Also, Moses, NMT and NMT+Synth systems presented the same lexical choice error at the word (*deberían*).

Finally, we show the translation examples for the e-commerce domain. Moses obtained the worst translation. The NMT$^{\Sigma}$ method was not able to obtain the word (*precioso*), as provided in the reference, but instead it a synonym (*hermoso*). Nevertheless, note that, although this may not be an actual mistake in translation terms, it will be penalized by BLEU and TER. The NMT+Synth ob-

Table 6: Translation examples for each domain with the MT systems built: Src (source sentence), Moses (moses system), NMT (NMT system), NMT$^{\Sigma}$ (NMT system with ensemble), NMT + Synth (NMT using synthetic corpus) and Ref (reference).

XRCE	Src	specifying the output file format 2-29
	Moses	*especificando el* formato de salida 2-29
	NMT	*especificar el* formato de archivo de salida 2-29 .
	NMT$^{\Sigma}$	*especificar el* formato de archivo de salida *de 29 a 29* .
	NMT+Synth	*especificar el* formato de archivo de salida 2-29
	Ref	especificación del formato de archivo de salida 2-29
IT	Src	almost all apps installed on windows 8 should work correctly in windows 8.1 .
	Moses	casi todas las aplicaciones instaladas en *las ventanas* 8 *debería* funcionar correctamente en *ventanas* 8.1 .
	NMT	casi todas las aplicaciones instaladas en windows 8 *deben* funcionar correctamente en windows 8.1 .
	NMT$^{\Sigma}$	casi todas las aplicaciones instaladas en windows 8 deberían funcionar correctamente en windows 8.1 .
	NMT+Synth	casi todas las aplicaciones instaladas en windows 8 *debería* funcionar correctamente en windows 8.1 .
	Ref	casi todas las aplicaciones instaladas en windows 8 deberían funcionar correctamente en windows 8.1 .
E-Com	Src	they are a lovely set of small and thin strips silver intertwined .
	Moses	son un *conjunto* de pequeñas y *encantadoras tiras finas plata interrelacionado* .
	NMT	son un precioso conjunto de *tiras de película pequeña y delgada* .
	NMT$^{\Sigma}$	son un *hermoso* conjunto de pequeñas y finas tiras de plata .
	NMT+Synth	son un precioso conjunto de pequeñas y finas tiras de plata .
	Ref	son un precioso conjunto de pequeñas y finas tiras de plata entrelazada .

tained the closer translation to the reference. Even though, the system was unable to obtain a translation for the word (*intertwined*).

7 Conclusions

In this work we presented an instance selection method and applied it to collect the most adequate sentences for translating a corpus from a specific domain. We selected domain-related instances from a large monolingual corpus, automatically translated them and fine-tuned a NMT system, originally trained on a more general domain. Results showed significant improvements in terms of BLEU and TER with respect to the original model. Moreover, we found that it is preferable to use a single fine-tuned model than an ensemble of general models. It is also worth mentioning that, once the selection was performed, the adaptation of NMT systems to new domains was very fast (few minutes).

As byproduct of the evaluation carried out in this work, we can also conclude two main points. First, to use a single automatic metric for evaluating machine translation is risky, as every automatic metric is likely to be distorted. In order to have more confidence about the performance of a machine translation system, it should be tested on more metrics. Second, when applying NMT systems to tasks with different features than the training data, we should control the length of the output sentences. This can be achieved either with some heuristics or adapting with an in-domain corpus. We leave the study of this control as future work.

As additional future work, we intend to prove our methods in more domains and different language pairs in order to establish its robustness. Moreover, we want to observe the influence of the quality and nature of the synthetic data in our pipeline. Therefore, we aim to study the influence of different translation methods or technologies when translating the monolingual corpus. We should also study if adding source synthetic data instead of target synthetic data affects the system. Finally, given the good results obtained, we want to leverage the bondages of the synthetic data, using it in different applications.

Acknowledgments

The research leading to these results has received funding from the Generalitat Valenciana under grant PROMETEOII/2014/030 and the FPI (2014) grant by Universitat Politècnica de València. We also acknowledge NVIDIA for the donation of a GPU used in this work.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi,

Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan-Miguel Vilar. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics* 35:3–28.

Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Workshop on Statistical Machine Translation*. pages 182–189.

Denny Britz, Anna Goldie, Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. *arXiv:1703.03906*.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv:1312.3005*.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-supervised learning for neural machine translation. *arXiv:1606.04596*.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv:1409.1259*.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014b. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of the Workshop on Syntax, Semantic and Structure in Statistical Translation*. pages 103–111.

Josep Maria Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou, and Peter Zoldan. 2016. SYSTRAN's pure neural machine translation systems. *arXiv:1610.05540*.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14(2):179–211.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the International Conference on Machine Learning*. pages 513–520.

Alex Graves. 2011. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*. pages 2348–2356.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv:1503.03535*.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, and Yoshua Bengio. 2017. On integrating a language model into neural machine translation. *Computer Speech & Language* 45:137 – 148.

Geoffrey E Hinton. 1986. Learning distributed representations of concepts. In *Proceedings of the Annual Meeting of the Cognitive Science Society*. pages 12–24.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. pages 1700–1709.

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

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*. pages 181–184.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. pages 388–395.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. pages 177–180.

Patrik Lambert, Holger Schwenk, Christophe Servan, and Sadaf Abdul-Rauf. 2011. Investigations on translation model adaptation using monolingual data. In *Proceedings of the Workshop on Statistical Machine Translation*. pages 284–293.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv:1405.4053*.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*. pages 76–79.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv:1301.3781*.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *arXiv:1309.4168*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. pages 160–167.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics* 29(1):19–51.

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

Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. How to construct deep recurrent neural networks. *arXiv:1312.6026*.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. On the difficulty of training recurrent neural networks. *arXiv:1211.5063*.

Álvaro Peris. 2017. NMT-Keras. `https://github.com/lvapeab/nmt-keras`. GitHub repository.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

Holger Schwenk. 2007. Continuous space language models. *Computer Speech & Language* 21(3):492–518.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. pages 1715–1725.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the Association for Machine Translation in the Americas*. pages 223–231.

Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the International Conference on Machine Learning*. pages 129–136.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*. pages 901–904.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*. volume 27, pages 3104–3112.

Nicola Ueffing, Gholamreza Haffari, Anoop Sarkar, et al. 2007. Transductive learning for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. pages 25–35.

Hua Wu, Haifeng Wang, and Chengqing Zong. 2008. Domain adaptation for statistical machine translation with domain dictionary and monolingual corpora. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. pages 993–1000.

Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv:1609.08144*.

Jiajun Zhang and Chengqing Zong. 2016a. Bridging neural machine translation and bilingual dictionaries. *arXiv:1610.07272*.

Jiajun Zhang and Chengqing Zong. 2016b. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. pages 1535–1545.