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.
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 means 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) or 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
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 sentences 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 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 hypersphere 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:

$$
\cos(F_1, F_2) = \frac{F_1 \cdot F_2}{\|F_1\| \cdot \|F_2\|}
$$

(1)

where $F_1$ and $F_2$ are two $z$-dimensional vectors.

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

$$
F_T = \frac{1}{|T|} \sum_{t} F_{x_t}
$$

(2)

where $F_{x_t} \in \mathbb{R}^z$ is the $z$-dimensional representation of the sentence $x_t$ and $F_T \in \mathbb{R}^z$ denotes the centroid of our test set.

**Data:** Pool $\mathcal{P}$; test data $\mathcal{T}$

**Result:** Source synthetic corpus $\mathcal{S}$

1. $F_T := \text{centroid}(\mathcal{T})$;
2. $\rho := \infty$;
3. $S := \emptyset$;
4. forall $x_t \in \mathcal{T}$ do
5.  if $\cos(F_{x_t}, F_T) \leq \rho$ then
6.  $\rho := \cos(F_{x_t}, F_T)$
7.  end
8. end
9. forall $x_p \in \mathcal{P}$ do
10. if $\cos(F_{x_p}, F_T) \geq \rho$ then
11.  $S \cup \{x_p\}$;
12. end
13. end

Algorithm 1: Pseudo-code for selecting synthetic corpora.

Algorithm 1 shows the selection procedure. Here, $x_t \in \mathcal{T}$, is a sentence from our source test data $\mathcal{T}$; and $F_{x_t}$ is the vector-space representation of $x_t$. Analogously, $\mathcal{P}$ is the pool of candidate sentences, $x_p \in \mathcal{P}$ is a source candidate sentence, $F_{x_p}$ is the vector-space representation of $x_p$, 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:

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

$\rho$ 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 Technology¹ (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.

¹http://metashare.metanet4u.eu/qtleapcorpus
²http://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 (|\bar{W}|).

| Corpus         | EN  | |W| | V | |W| |
|----------------|-----|-----|-----|-----|-----|
| 1 Billion Words | 30.3M | 800M | 800k | 26.4 |
| COMMON         | 1.5M | 30M | 456k | 20.0 |
| dev2013        | 2.7k | 48.9k | 7.5k | 18.1 |
| XRCE – Test    | 1.1k | 8.4k | 1.6k | 7.6  |
| IT – Test      | 857  | 15.6k | 2.1k | 18.2 |
| E-Com – Test   | 886  | 7.3k | 874  | 8.2  |

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 (σ = 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-linear 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 (T), employed for adapting the NMT system. |S| denotes number of sentences; |W|, number of words; |V|, vocabulary size and |\bar{W}|, average sentence length.

| T             | |S| | |W| | |V| | |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.
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
NMT system was very fast. The system only required to be trained on ~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. Σ denotes an ensemble of 4 neural models. $|W|$ is the average number of words per sentence.

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

The NMT systems behaved 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 that 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 NMTΣ 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 4: Translation results for the XRCE and IT tasks. BLEU and TER results given in percentage. Σ denotes an ensemble of 4 neural models. $|W|$ is the average number of words per sentence.

<table>
<thead>
<tr>
<th>System</th>
<th>XRCE BLEU</th>
<th>XRCE TER</th>
<th>IT BLEU</th>
<th>IT TER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moses</td>
<td>26.2 ± 0.8</td>
<td>59.0 ± 0.8</td>
<td>33.4 ± 0.6</td>
<td>45.6 ± 0.6</td>
</tr>
<tr>
<td>NMT</td>
<td>20.4 ± 1</td>
<td>94.5 ± 5.1</td>
<td>29.0 ± 0.8</td>
<td>53.5 ± 0.8</td>
</tr>
<tr>
<td>NMTΣ</td>
<td>25.5 ± 0.8</td>
<td>76.8 ± 2.0</td>
<td>31.4 ± 0.8</td>
<td>51.2 ± 0.8</td>
</tr>
<tr>
<td>NMT + Synthetic</td>
<td>27.5 ± 0.8</td>
<td>56.7 ± 0.8</td>
<td>34.1 ± 0.7</td>
<td>45.7 ± 0.7</td>
</tr>
<tr>
<td>NMTΣ + Synthetic</td>
<td>27.3 ± 0.8</td>
<td>56.3 ± 0.8</td>
<td>33.8 ± 0.7</td>
<td>46.3 ± 0.7</td>
</tr>
</tbody>
</table>
Table 6: Translation examples for each domain with the MT systems built: Src (source sentence), Moses (moses system), NMT (NMT system), NMT* (NMT system with ensemble), NMT + Synth (NMT using synthetic corpus) and Ref (reference).

<table>
<thead>
<tr>
<th>Domain</th>
<th>Source</th>
<th>Moses</th>
<th>NMT</th>
<th>NMT*</th>
<th>NMT + Synth</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>XRCE</td>
<td>specifying the output file format 2-29</td>
<td>especificando el formato de salida 2-29</td>
<td>especificar el formato de archivo de salida 2-29</td>
<td>especificar el formato de archivo de salida 29 a 29</td>
<td>especificar el formato de archivo de salida 2-29</td>
<td>especificación del formato de archivo de salida 2-29</td>
</tr>
<tr>
<td>IT</td>
<td>almost all apps installed on windows 8 should work correctly in windows 8.1</td>
<td>casi todas las aplicaciones instaladas en las ventanas 8 deberán funcionar correctamente en ventanas 8.1</td>
<td>casi todas las aplicaciones instaladas en windows 8 debería funcionar correctamente en windows 8.1</td>
<td>casi todas las aplicaciones instaladas en windows 8 deberían funcionar correctamente en windows 8.1</td>
<td>casi todas las aplicaciones instaladas en windows 8 deberán funcionar correctamente en windows 8.1</td>
<td></td>
</tr>
<tr>
<td>E-Com</td>
<td>they are a lovely set of small and thin strips silver intertwined</td>
<td>son un conjunto de pequeñas y encantadoras tiras finas plata entrelazado</td>
<td>son un precioso conjunto de tiras de película pequeña y delgada</td>
<td>son un hermoso conjunto de pequeñas y finas tiras de plata</td>
<td>son un precioso conjunto de pequeñas y finas tiras de plata entrelazada</td>
<td></td>
</tr>
</tbody>
</table>

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


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


