Improving the Quality of Neural Machine Translation

Pedro Ferreira[1] and André F.T. Martins[2]

***Abstract*—Over the last few years, neural machine translation has become the major approach to the problem of automatic translation. Nonetheless, even though current models are able to output fluent translations, they often lack adequacy. In this thesis we target adequacy issues with two different causes: poor coverage of source words during translation, which lead to unnecessary repetitions and erroneously untranslated words; rare words and out-of-domain sentences.**

In order to mitigate coverage issues we propose a fertility-based approach to neural machine translation, which couples the concept of fertility with sparse and constrained attention transformations. Furthermore, we present two machine translation metrics that allow us to understand how much the problems of over and under-translations are affecting the model.

To deal with rare words and out-of-domain sentences, we implement an existing approach in the literature, that makes use of the concept of translation pieces to guide the decoding step of NMT models. We further extend this method by solving some identified issues, and by applying it to the problem of domain adaptation, something which had not been done in the past.

Finally, we provide an empirical evaluation in three language pairs, presenting an extensive error analysis. This makes it possible to understand the strengths and weaknesses of each of the models, and how they may be improved in the future.

***Index Terms*—Deep learning, natural language processing, neural machine translation, attention transformations, domain adaptation.**

I. Introduction

NEURAL Machine Translation (NMT) has become the de-facto approach to the problem of machine translation in recent years [1]–[3]. Despite the recent improvements in the overall quality of NMT there are still some areas of concern. Namely, current NMT systems are able to output *fluent* sentences that often lack *adequacy*, i.e., are not able to convey the idea of the source sentence [4]. Adequacy errors may be split into three different major types of mistakes [5]:

- *Over-translations*, meaning that some source words are translated more often than they should.
- *Under-translations*, when some source words are erroneously untranslated.
- *Mistranslations*, when a given source word is attended, but the output word is not correct.

Furthermore, the problem of mistranslations may be related with the inability of NMT systems to perform well when translating rare words and out-of-domain sentences [6], [7].

In the current literature it is possible to find different approaches to the problem of improving adequacy. [5], [8] target the problem of adequacy by adapting the concept of coverage to NMT. Namely, both introduce a coverage vector, with the goal of keeping track of which parts of the source sentence have already been attended. In a different approach, [9] proposed using a gating mechanism to decide how much information flows from either the encoder or the decoder, based on the premise that both affect fluency and adequacy differently. Another possible way of ensuring adequacy, is by training the model to be able to reconstruct the original sentence from the obtained translations [10]. All the previous approaches have one thing in common, namely, they introduce some kind of artifact that is used during training. On the contrary, [11] introduces two terms which re-score the candidate translations during beam search: a *coverage penalty*, which penalizes hypotheses that have source words unattended; and *length normalization*, that ensures the model does not produce translations which are too short.

Other works in the literature, that try to include external knowledge into NMT models, are more concerned with rare words, or making sure that the translations are closer to the desired. These approaches range from lexical constraints and biases [6], [12], [13], to strategies that try to find examples similar to the input sentence to re-train the model [14], [15] or that leverage information about the hidden states and the output distribution of the closest examples [16].

Despite the current approaches to the problem of adequacy in NMT, some questions still remain as open problems. Namely,

- How can we avoid the problem of unnecessary repetitions in NMT?
- How can we make sure that every source word is attended during translation?
- How can we make NMT models more robust to domain changes and rare words?

The main contributions of this work, which aim at answering the previously raised questions, are the following:

- We propose two new metrics that aim to measure how much of a problem over-translations and under-translations are in the output sentences. Namely, these metrics are able to capture information that the commonly used automatic metrics miss.
- We introduce a new sparse constrained transformation, *Constrained Sparsemax*, that is able to produce upper bounded sparse probability distributions, while being end-to-end differentiable.
- We introduce a new approach to NMT, *fertility-based NMT*. In particular, it pairs sparse and constrained attention transformations with the fertility concept to promote better coverage of the source words, both during training

[1] Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal
[2] Unbabel & Instituto de Telecomunicações, Lisbon, Portugal

and inference. By enhancing the coverage of the model, fertility-based NMT should decrease the adequacy issues related with over and under-translations.
- We implement a model based on the proposed by [17], with the changes introduced in Section V. We later use this model in the same context of the original paper (in-domain translation), and further evaluate its performance in Domain Adaptation. This approach should help mitigate the identified problem of mistranslations.
- We perform an empirical analysis of both introduced models, focusing also on error analysis.

This work is organized as follows: Section II introduces the necessary background; Section III introduces the two proposed MT metrics; Section IV presents the fertility-based NMT approach; Section V describes Guided-NMT; Section VI summarizes the most important experiments conducted; and finally, Section VII reports the main takeaways.

II. Background

Before delving into the proposed models and the obtained results, it is necessary to introduce the necessary background.

A. Recurrent Neural Network

A Recurrent Neural Network (RNN) is a particular type of neural network, able to capture recurrent dependencies in the input data [18]. Its hidden states are defined as

$$\boldsymbol{h}_t = g(\boldsymbol{W}^x \boldsymbol{x}_t + \boldsymbol{W}^h \boldsymbol{h}_{t-1} + \boldsymbol{b}^h) \tag{1}$$

where $g(.)$ is some non-linear function, $\boldsymbol{W}^x \in \mathbb{R}^{d_x \times d_h}$, $\boldsymbol{W}^h \in \mathbb{R}^{d_h \times d_h}$, $\boldsymbol{b}_h \in \mathbb{R}^{d_h}$, $\boldsymbol{x}_t \in \mathbb{R}^{d_x}$, and $\boldsymbol{h}_t, \boldsymbol{h}_{t-1} \in \mathbb{R}^{d_h}$. The fact that at each time step, the previous hidden state is fed into the calculation of the current hidden state and consequently, of the output, allows information from previous inputs to be incorporated and passed to next states. This leads to the conclusion that RNNs are indeed able to capture long term dependencies in the data. For natural language processing problems it is common to have a discrete output space, thus, a convenient choice for the non-linearity used to predict outputs is the softmax (Equation 3). So, at each time step, it is possible to predict an output as

$$\boldsymbol{y}_t = \text{softmax}(\boldsymbol{W}^y \boldsymbol{h}_t + \boldsymbol{b}^y). \tag{2}$$

where, $\boldsymbol{W}^y \in \mathbb{R}^{d_h \times d_y}$, $\boldsymbol{b}^y \in \mathbb{R}^{d_y}$, $\boldsymbol{h}_t \in \mathbb{R}^{d_h}$, $\boldsymbol{y}_t \in \mathbb{R}^{d_y}$, and the softmax activation function is used to project the logits into a probability space. Softmax is defined as

$$\text{softmax}(z)_n = \frac{\exp(z_n)}{\sum_{j=1}^{N} \exp(z_j)}, \tag{3}$$

where the input vector, z, corresponds to the logits, and N is the dimension of the output.

RNNs are trained as usual for neural networks, in this case making use of the *backpropagation through-time* [19]. One commonly reported issue is that during backpropagation the gradients become increasingly small throughout time steps, making it hard for the model to actually capture long dependencies [20]. To overcome this issue it is common to use Long Short-Term Memory (LSTM) cells as the basic units of RNNs [21].

B. Neural Machine Translation

The goal of NMT is to translate a sequence of words from a source language, $\boldsymbol{X} = x_1, x_2, ..., x_J$ to a sequence of words in a target language, $\boldsymbol{Y} = y_1, y_2, ..., y_T$, by learning to model a conditional probability, $P(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{t=1}^{T} P(y_t|y_1, ..., y_{t-1}, \boldsymbol{X})$.

The standard approach to NMT is to use the encoder-decoder architecture [1]–[3], presented in Figure 1. The main idea of this approach may be seen in two steps: first, an encoder is used to output a hidden state, i.e., a vector, that is supposed to encode the source language sentence; then, the decoder, which may be seen as a recurrent neural language model, will use that intermediate representation as its initial hidden state and, conditioned on the source sentence, generate target language words. As it is, the intermediate hidden state has to encompass the meaning of the whole source sentence, and consequently, so do the decoder-side hidden states. As a way of mitigating this issue, [22], [23] introduce the concept of attention.

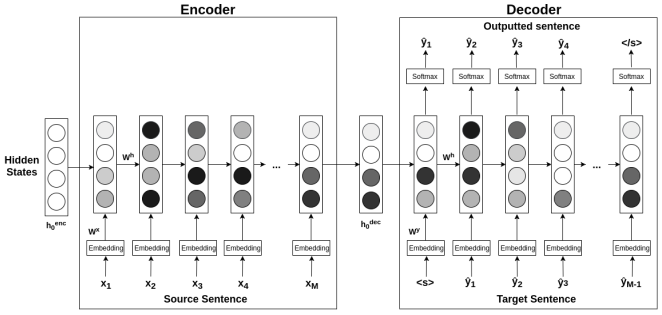

Fig. 1. Simplified representation of the NMT encoder-decoder architecture.

1) Encoder: A common architecture to use as encoder is a bi-directional RNN [24], using LSTMs as units. In practice, this means that we have two RNNs, one processing the source sentence from left to right, and the other on the opposite direction. By concatenating the hidden states of both, it is possible to create a representation that has information about all the context for a given source word. More formally, this is defined as

$$\overrightarrow{\boldsymbol{h}_j^e} = \text{RNN}(\overrightarrow{\boldsymbol{h}_{j-1}^e}, \boldsymbol{E}_{[x_j]}) \tag{4}$$

$$\overleftarrow{\boldsymbol{h}_j^e} = \text{RNN}(\overleftarrow{\boldsymbol{h}_{j+1}^e}, \boldsymbol{E}_{[x_j]}), \tag{5}$$

being $\boldsymbol{E}_{[x_j]}$ the embedded vector of word x_j, and $\boldsymbol{h}_j^e$ the hidden state of the encoder at position j, resulting then in the bi-directional representation, $\boldsymbol{h}_j^e = \left[\overleftarrow{\boldsymbol{h}_j^e}; \overrightarrow{\boldsymbol{h}_j^e}\right]$.

2) Decoder: The goal of the decoder is to produce target language words, until a end of sentence token is output. The common choice for the decoder is to use a recurrent neural network. Each new hidden state $\boldsymbol{h}_t^d$, depends on the previous hidden state $\boldsymbol{h}_{t-1}^d$, the embedding of the previous output word $\boldsymbol{E}_{[y_{t-1}]}$, and also on a context vector $\boldsymbol{c}_t$. Each hidden state of the decoder is therefore given by

$$\boldsymbol{h}_t^d = \mathrm{RNN}(\boldsymbol{h}_{t-1}^d, \boldsymbol{c}_t, \boldsymbol{E}_{[y_{t-1}]}). \quad (6)$$

At each decoding time step t, it is then possible to obtain a probability distribution over the vocabulary by making use of the softmax. This probability distribution allows us to choose the output word at each time step during the translation.

3) Attention Mechanism: As previously stated, the attention mechanism provides a way of enriching the source side information the decoder has available at each time step. This is achieved by the context vector $\boldsymbol{c}_t$, that is used as input to the recurrent neural network. The context vector is obtained as a weighted sum of the hidden states that resulted from the encoding step, formally defined as

$$\boldsymbol{c}_t = \sum_{j=1}^{J} \alpha_{tj} \boldsymbol{h}_j^e, \quad (7)$$

being J the number of words of the source sentence, and $\boldsymbol{\alpha}$ the normalized vector of attention scores. This vector is in fact a probability distribution over the source words, and thus, each of its values may be interpreted as the importance of the corresponding source word for the output at decoding step t. The probability α_{tj}, relative to each encoder hidden state $\boldsymbol{h}_j$, is given by,

$$\alpha_{tj} = \frac{\exp(z_{tj})}{\sum_{k=1}^{J} \exp(z_{tk})}, \quad (8)$$

where z_{tj} is an alignment model that is able to capture how important the source words close to position j are for the current decoding step, t. There are multiple ways of defining this alignment model. The most widely known is the approach by [22], sometimes called *Bahdanau's attention*. In particular, it proposes training a feedforward neural network simultaneously with the rest of the NMT model. The alignment model in this case is defined as $z_{tj} = a(\boldsymbol{h}_{t-1}^d, \boldsymbol{h}_j^e)$.

A posterior work presented a slightly different approach to the attention mechanism, as well as novel alignment models [23]. Before introducing those alignment models, it is necessary to go through some of the subtleties of the attention used by [23] (*Luong's attention*). Namely, in Equation 6, the context vector is no longer provided, i.e., it is calculated as $\boldsymbol{h}_t^d = \mathrm{RNN}(\boldsymbol{h}_{t-1}^d, \boldsymbol{E}_{[y_{t-1}]})$. Instead, the context is used to obtain an intermediate representation, $\boldsymbol{h'} = \tanh(\boldsymbol{W}_c[\boldsymbol{c}_t; \boldsymbol{h}_t^d])$. Notice that Equations 7 and 8 are not affected by this change and still hold true. Finally, the probability distribution at a given decoding step is given by $p(y_t|y_1, ..., y_{t-1}, \boldsymbol{X}) = \mathrm{softmax}(\boldsymbol{W}_s \boldsymbol{h'})$. $\boldsymbol{W}_c$ and $\boldsymbol{W}_s$ are both matrices of weights learned during training.

In the case of *Luong's attention*, three different alignment models, z_{tj}, are introduced [23]:

- *Dot product*: $\boldsymbol{h}_t^{dT} \boldsymbol{h}_j^e$.
- *General*: $\boldsymbol{h}_t^{dT} \boldsymbol{W}_a \boldsymbol{h}_j^e$.
- *Concat*: $\boldsymbol{v}_a^T \tanh\left(\boldsymbol{W}_a^1 [\boldsymbol{h}_t^d; \boldsymbol{h}_j^e]\right)$.

4) Training: A NMT model is trained by maximizing the conditional log-likelihood,

$$C(\theta) = \frac{1}{D} \sum_{d=1}^{D} \sum_{t=1}^{M_d} \log p_\theta(y_t^d | y_1^d, ..., y_{t-1}^d, \boldsymbol{X}^d), \quad (9)$$

via a set of parameters $\boldsymbol{\theta}$, being D the number of sentence pairs in the data set and M_d the length of a target sentence. This will penalize sets of parameters that are not able to output probability distributions at each decoding time step in which the correct words are probable to happen.

5) Translation Generation: After modelling the conditional probability $p(\boldsymbol{Y}|\boldsymbol{X})$, it is necessary to define how to choose the output words at each decoding time step. As discussed previously, for each decoding step it is obtained an output vector with a probability distribution over all possible words. Finding the best translation corresponds to finding $\arg\max_Y p(\boldsymbol{Y}|\boldsymbol{X})$.

Solving this problem would require computing the log probability of every possible translation. Since that is not feasible, an approximate search has to be used as alternative. The most widely used approach is *beam search*, which works by keeping a *beam* of the *n* most probable sequences of words at each time step. This means that at each time step there are temporarily $n \times |V|$ hypothesis, being $|V|$ the number of words in the vocabulary, which are then reduced to the top *n*. When the end sentence token $< /s >$ is predicted by the decoder, a given hypothesis is assumed to be terminated. Whenever this happens, that hypothesis is stored, the width *n* of the beam is reduced by one, and the procedure continues, until that value is reduced to zero. The best yielded hypothesis is chosen as the translation of the source sentence.

6) Vocabulary Size: One of the main aspects of a neural machine translation model is its vocabulary, mainly due to the implications of its size. In particular, it implies finding a balance between a large vocabulary size, that leads to the least amount possible out-of-vocabulary words (usually replaced by a *unknown* symbol), and a small vocabulary, which reduces the complexity of the model.

A common way of overcoming this problem is to use the byte pair encoding (BPE) algorithm to create *subwords* [25]. This approach is based on the idea that different words share common smaller units, the so called subwords, whose translation may be concatenated and lead to the correct translation of full words. The benefits from using this approach as opposed to a fixed-vocabulary are twofold: first, the models improve their ability of translating rare and out-of-vocabulary words; second, it shows improvements in BLEU while reducing the vocabulary size, and thus, the complexity of the whole process of NMT [25] .

III. Machine Translation Adequacy Metrics

We present two metrics: $\mathrm{REP}_{\mathrm{Score}}$, and $\mathrm{DROP}_{\mathrm{Score}}$, which measure the problem of over and under-translation, respectively.[1] Both of these metrics were part of our previous work [26], but were further improved in the current work, namely

[1] Available at: https://github.com/PedroMLF/MT-Adequacy-Metrics

by introducing appropriate length penalties, and the possibility of using multiple references.

A. $\mathrm{REP}_{\mathrm{Score}}$

The intuition for $\mathrm{REP}_{\mathrm{Score}}$ is that, if the candidate translation's *n*-grams counts are much larger than the respective counts in the reference translation, then it is very likely that the model is over-translating. Another aspect that is widely seen in machine translation outputs are the cases in which words are generated repeatedly and consecutively. $\mathrm{REP}_{\mathrm{Score}}$ tries to penalize both of these cases according to the following formula,

$$\begin{aligned}\sigma(c,r) = \lambda_1 \sum_{s\in c^n, c(s)\geq 2} \max\{0, c(s) - r(s)\} + \\ \lambda_2 \sum_{w\in c} \max\{0, c(ww) - r(ww)\},\end{aligned} \tag{10}$$

where c and r are the candidate and corresponding reference sentences, respectively, $s \in c^n$ represents a *n*-gram in the candidate translation, $c(s)$ and $r(s)$ are its counts in the given sentences, $w \in c$ is a word in the candidate sentence, and $c(ww)$ and $r(ww)$ are counts of consecutive words appearances in the given pair of sentences.

A posterior work [27], suggested that such metrics could exploit shorter candidate sentences in order to have lower over-translation scores, since the counts of words directly affect the final scores. Hence, we introduce a brevity penalty *BP*, in the final calculation of $\mathrm{REP}_{\mathrm{Score}}$,

$$\mathrm{REP}_{\mathrm{Score}} = 100 \times BP \times \sum_{c\in C} \sigma(c,r) \times \frac{1}{|R|}, \tag{11}$$

where C is the set of all candidate translations for a corpus, c is a candidate translation with r being its corresponding reference, and $|R|$ the number of words of the reference sentences. The normalization by the number of words in the reference corpus makes the metric less sensitive to the reference length. The λ weights are defined as $\lambda_1 = 1$ and $\lambda_2 = 2$, so that consecutive repeated words are more penalized, since they are a better indicative of a problem with over-translations and a common NMT output *hallucination*. The default *n*-grams used are bi-grams. $\mathrm{REP}_{\mathrm{Score}}$ values are non-negative and unbounded, and the lower the better.

B. $\mathrm{DROP}_{\mathrm{Score}}$

Broadly speaking, $\mathrm{DROP}_{\mathrm{Score}}$ finds the amount of source words that aligned with words from the reference translation, but that did not align with any word from the candidate translation. The reasoning behind this is that, if a given source word aligned with some word of the reference, but did not align with any word from the candidate translation, then it is likely that that word was dropped during the process.

In order to do this, fast_align [28] is used to obtain both a set of alignments between the source and the reference corpus, and a set of alignments between the source and the candidate translation. From these set of alignments, the indexes of the aligned source words are stored, for both cases. Obtaining the difference between these two sets and dividing that number by the size of the set of source words that aligned with reference translation words yields the exact percentage of source words that aligned with some word of the reference but with none from the candidate translation. Although this value is interpretable, longer sentences will have more words for the source side to align to, and therefore, may be benefited during the calculation of this metric. Thus, it is included a penalty for these cases (taking note from [27]). $\mathrm{DROP}_{\mathrm{Score}}$'s final value is given by

$$\mathrm{DROP}_{\mathrm{Score}} = 100 \times LP \times \sum_{c\in C} \frac{|S_r \setminus S_c|}{|S_r|}, \tag{12}$$

where c is the candidate translation, C is the set of all candidate translation for a given corpus, S_r is the set of source words indexes that aligned with some word of the reference translation, S_c is the set of source words indexes that aligned with some word of the candidate translation, and LP is a length penalty that penalizes sentences longer than the reference. The final values for the metric are non-negative and unbounded (due to the length penalty), and the lower the value, the better.

C. Correlation with Human Judgment

In order to further validate the proposed metrics, we make use of the Pearson correlation to find whether the proposed metrics correlate with human annotated scores for over and under-translations. The human judgment data was obtained by the authors of [27], and refers to the evaluation of a total of 888 candidate sentences, produced by four different NMT systems, with regard to the presence of repeated or dropped words. The source sentences, and corresponding references, were obtained from the 2002 NIST dataset. Following [27], the corpus level score for each of the models is obtained by averaging the human scores for each system. We refer to these as *over-translation human judgment* (OTHJ), and *under-translation human judgment* (UTHJ). These scores are then compared with the corpus-level values produced by the proposed $\mathrm{REP}_{\mathrm{Score}}$ and $\mathrm{DROP}_{\mathrm{Score}}$. This is a common approach, as similar experiments to validate metrics with regard to human data have been in reported in the literature [27], [29], [30].

The Pearson correlation coefficient yields a value between -1 and 1, being an absolute value close to 1 sign of an high correlation between the values being compared. A value of 0 would mean no correlation between the series of values. We obtain a Pearson correlation of -0.929 between the $\mathrm{REP}_{\mathrm{Score}}$ and OTHJ, and a correlation of -0.882 between $\mathrm{DROP}_{\mathrm{Score}}$ and UTHJ. Even though the number of samples being used does not allow to assess the statistical significance of the results, the reported high correlations show encouraging signs that the proposed metrics are able to perform as expected.

IV. Fertility Based NMT

Previously, we identified some of the shortcomings of NMT that have a negative impact in the output translations.

In particular, two of those shortcoming revolve around the problem of over and under-translating source words [5]. We propose fertility-based NMT, part of our work [26], which addresses the aforementioned cases by using different attention transformations.[2] To introduce these changes, Equation 8 is re-written as $\boldsymbol{\alpha_t} = \rho(\boldsymbol{z_t}, \boldsymbol{u_t})$, where ρ corresponds to the chosen attention transformations, $\boldsymbol{z_t} \in \mathbb{R}^J$ are the scores provided by the attention's chosen alignment model, at a given decoding step t, and $\boldsymbol{u_t} \in \mathbb{R}^J$ is the upper bounds vector. The later is only necessary when ρ is defined as one of the constrained attention transformations and defines an upper bound in the attention each source word can receive. The three alternatives to the traditional softmax are described next.

A. Sparsemax

Sparsemax [31] is as activation function that distinguishes itself from the commonly used softmax due to its capacity of producing sparse probability distributions. This means that, contrary to softmax, where the yielded probabilities are strictly positive, sparsemax is able to produce probability distribution where some of the individual values are zero. The sparse probability distribution $\boldsymbol{p}$, is obtained by projecting the vector of logits $\boldsymbol{z} \in \mathbb{R}^J$, onto the probability simplex, $\Delta^{J-1} := \{\boldsymbol{p} \in \mathbb{R}^J \mid \sum_j p_j = 1,\ p_j \geq 0\ \forall_j\}$, as

$$\text{sparsemax}(\boldsymbol{z}) := \underset{\boldsymbol{p} \in \Delta^{J-1}}{\arg\min} \|\boldsymbol{p} - \boldsymbol{z}\|^2. \tag{13}$$

B. Constrained Softmax

Constrained softmax [32] is an activation function that differs from softmax by allowing to define an hard upper bound on the probability that is assigned to each element. The constrained softmax of a vector is obtained by solving the following optimization problem

$$\begin{aligned} &\text{csoftmax}(\boldsymbol{z}; \boldsymbol{u}) := \underset{\boldsymbol{p} \in \Delta^{J-1}}{\arg\min} \text{KL}(\boldsymbol{p}; \text{softmax}(\boldsymbol{z})) \\ &\text{subject to} \quad \boldsymbol{p} \leq \boldsymbol{u}, \end{aligned} \tag{14}$$

in which $\boldsymbol{p} \in \mathbb{R}^J$ is the probability distribution, $\boldsymbol{u} \in \mathbb{R}^J$ is a vector of upper bounds to the individual probabilities and KL refers to the Kullback-Leibler divergence [33]. In words, the optimization problem that leads to the constrained softmax, yields the closest probability distribution to the one produced by softmax, with the constraint of having the probability values capped by a vector of upper bounds.

C. Constrained Sparsemax

Constrained sparsemax, introduced in our work [26], is an activation function based on sparsemax and that differs from it by introducing upper bounds in the attention scores it returns. It is calculated by solving the problem

$$\begin{aligned} &\text{csparsemax}(\boldsymbol{z}; \boldsymbol{u}) := \underset{\boldsymbol{\alpha} \in \Delta^{J-1}}{\arg\min} ||\boldsymbol{\alpha} - \boldsymbol{z}||^2, \\ &\text{subject to} \quad \boldsymbol{\alpha} \leq \boldsymbol{u} \end{aligned} \tag{15}$$

[2]Available at: https://github.com/Unbabel/sparse_constrained_attention

where $\boldsymbol{z} \in \mathbb{R}^J$ is a vector of scores, $\boldsymbol{u} \in \mathbb{R}^J$ is the upper bounds vector, and $\boldsymbol{\alpha}$ are the resulting attention scores. Therefore, obtaining the attention scores at a given decoding step t, may be simply put as $\boldsymbol{\alpha_t} = \text{csparsemax}(\boldsymbol{z_t}; \boldsymbol{u_t})$.

Since it is based on the sparsemax attention transformation, constrained sparsemax also yields sparse probability distributions over the source words, at each decoding step.

To use both of the previously introduced constrained attention transformations it is necessary to better define the vector of upper bounds $\boldsymbol{u_t}$. One way of defining the upper bounds is through the concept of fertility [34]. Namely, it is possible to create a very intuitive abstraction by defining the vector of upper bounds $\boldsymbol{u_t}$, at a given decoding step t as

$$\boldsymbol{u_t} = \boldsymbol{f} - \boldsymbol{\beta_{t-1}}, \tag{16}$$

where $\boldsymbol{f} \in \mathbb{R}^J$ is a vector where each element corresponds to the fertility of a source word, and $\boldsymbol{\beta_{t-1}} \in \mathbb{R}^J$ corresponds to the cumulative attention over source words, more formally, $\sum_{\tau=1}^{t-1} \boldsymbol{\alpha_\tau}$. The element-wise difference between both vectors may be interpreted as the amount of attention a given source word still has available to receive, given that it started with a "credit" of f_j. Consequently, when a source word exhausts its attention "credit" it will stop receiving any probability mass. Thus, constrained attention transformations are able to introduce sparsity over time steps. The goal of improving adequacy is therefore targeted as follows:

- *Over-translations* are tackled by defining a "credit" of attention each source word has available to receive. Once that "credit" is exhausted, the source word will no longer be attended. This should help minimize instance of repetitions, where the same source word is attended several times.
- *Under-translations* should be mitigated by forcing the model to spread the attention over the words which have not had their "credit" of attention exhausted or reduced.

Figure 2 further highlights the behaviour of each of the mentioned attention transformations.

D. Fertility Bounds

One critical aspect in the proposed fertility-based NMT model, is how to define the vector of fertility bounds, $\boldsymbol{f}$. We define three possible approaches:

- *Constant*, by defining the fertility used as upper bound for each source word as a pre-defined constant.
- *MaxAlign*, where the fertility of each source word is defined as the maximum number of target words that aligned with it, using a word aligner like fast_align [28].
- *Predicted*, which uses a bi-LSTM tagger to predict the fertility value for each source word. The fertility values used to train the model are the number of target words that align with a given source word, according to fast_align.

E. Sink Token

At each decoding step it is calculated a probability distribution over the source words of a given input sentence. These

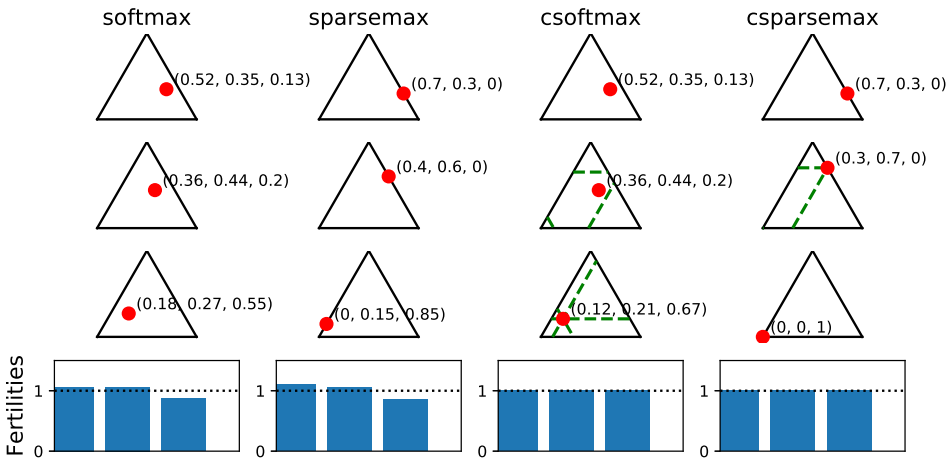

Fig. 2. Behavior of the different attention transformations in a toy problem, with three source words. Each row corresponds to a decoding step, being the first row the first decoding step. The logit values, per row are: $\boldsymbol{z} = (1.2, 0.8, -0.2)$; $\boldsymbol{z} = (0.7, 0.9, 0.1)$; and $\boldsymbol{z} = (-0.2, 0.2, 0.9)$. The red dots show the corresponding attention values on the probability simplex. For the constrained transformations, $\boldsymbol{f} = 1$. Also for those, the green dashed lines represent the upper bounds, $\boldsymbol{u}$. The last row shows the cumulative attention for each of the source words. Retrieved from [26].

are the so called attention scores and, since they correspond to a probability distribution, they sum up to one. Thus, exactly a single "credit" of the upper bounds is exhausted at each decoding step. This leads to one important consideration: the output translation cannot be longer than $\sum_j f_j$. Such a limitation could become a liability during training, since target sentences that exceed the sum of each source word's fertility render any of the constrained transformations infeasible. To overcome this situation, a $<$ SINK $>$ token is appended to every source sentence. This token has unbounded fertility, and therefore solves the previously detected limitation.

F. Exhaustion Strategies

As previously mentioned, exhausting the attention a given source word has yet to receive forces the attention scores to be spread over the remaining words. This helps the model addressing the issue of under-translations, by covering more source words during decoding. Nonetheless, this will only be verified after some words have their "credit" of attention exhausted. In order to have more attention spread over all source words during the whole process, it is introduced the following change: the scores $\boldsymbol{z_t}$, used in the attention transformation, are now defined as $\boldsymbol{z'_t} = \boldsymbol{z_t} + c\boldsymbol{u_t}$, where c is a constant. The higher the value of c, the more important are the source words which still have a large "credit" of attention. This will force the model to attend the words that are possibly being under-translated.

V. Guided NMT

Besides the problem of over and under-translations, two further issues were identified with the current NMT approaches: translating rare-words [6], and being robust to out-of-domain sentences [7]. To overcome this shortcoming of NMT, we make us of [17]'s approach, which we will name *Guided-NMT*.[3] The central part of this approach is the concept of *translation pieces*. These may be defined as subwords of a given target sentence which are deemed as possible partial translations of a given input sentence. The main advantage of this approach is the fact that it only affects the decoding part of NMT. The overall process may be split into two independent steps: a *retrieving* step, where translation pieces are created using a corpus of extra data; and a *guiding* step, where the translation pieces are used to bias the NMT output layer.

A. Obtaining Translation Pieces

The overall process of retrieving translation pieces is summarized in Algorithm 1, where N is the number of sentences in the extra corpus, the subscript *bpe* indicates that byte pair encoding (BPE) has been applied to the sentence [25], and *IDF* is the inverse document frequency.

B. Guiding NMT with Translation Pieces

The intuition behind translation pieces is that they represent n-grams which are likely to be present in good translation hypotheses of a certain input sentence. In practice, this means that a at each decoding step t, the output layer's log probabilities of subwords corresponding to translation pieces are rewarded. Thus, it is increased the probability of choosing those subwords during beam search. In particular, in translation pieces that correspond to n-grams with $n \geq 2$, only the last subword is rewarded, with the previous ones acting as context. This process may be visualized in Figure 3.

[3] Available at: https://github.com/PedroMLF/guided-nmt

Algorithm 1 Obtaining Translation Pieces

Input (Source Sentence): $\boldsymbol{X}$; $\boldsymbol{X}_{bpe}$;
Input (Extra Data): $\boldsymbol{X}^{1:N}$; $\boldsymbol{X}_{bpe}^{1:N}$; $\boldsymbol{Y}_{bpe}^{1:N}$; $\boldsymbol{A}_{bpe}^{1:N}$;
$E(\boldsymbol{X}) = \frac{1}{|\boldsymbol{X}|} \sum_{w \in \boldsymbol{X}} \mathrm{IDF}(w) E(w)$
for $\boldsymbol{X}^n$, $1 \leq n \leq N$ **do**
  $E(\boldsymbol{X}^n) = \frac{1}{|\boldsymbol{X}^n|} \sum_{w \in \boldsymbol{X}^n} \mathrm{IDF}(w) E(w)$
end for
$\boldsymbol{X}^{1:M}$ = *Faiss*($E(\boldsymbol{X})$, $E(\boldsymbol{X}^{1:N})$)
for X^m, $1 \leq m \leq M$ **do**
  $G_X^m = \emptyset$
  $simi(\boldsymbol{X}, \boldsymbol{X}^m) = 1 - \frac{d(\boldsymbol{X}, \boldsymbol{X}^m)}{\max(|\boldsymbol{X}|, |\boldsymbol{X}^m|)}$
  Obtain $\boldsymbol{W}_{bpe}^m$ using $\boldsymbol{X}_{bpe}$ and $\boldsymbol{X}_{bpe}^m$
  Obtain aligned target subwords from $\boldsymbol{Y}_{bpe}^m$, using $\boldsymbol{W}_{bpe}^m$ and $\boldsymbol{A}_{bpe}^m$
  Obtain translation pieces, i.e., n-grams from aligned target subwords, with $1 \leq n \leq 4$
  for each translation piece, tp_{Xi} **do**
    if tp_{Xi} already in G_X^m **then**
      Choose $\max_{1 \leq m \leq M \wedge tp_{Xi} \in G_X^m} simi(\boldsymbol{X}, \boldsymbol{X}^m)$
    else
      Add tp_{Xi} to G_X^m with value $simi(\boldsymbol{X}, \boldsymbol{X}^m)$
    end if
  end for
end for
Output: $G_X = \bigcup_1^M G_X^m$

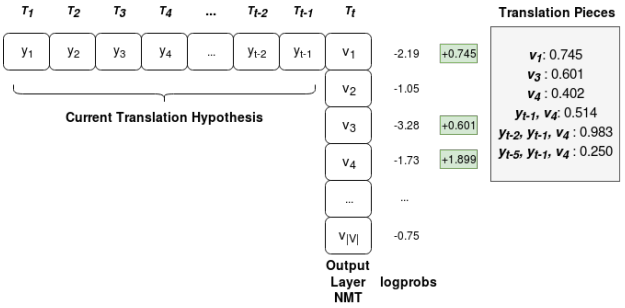

Fig. 3. Visualization of the process of rewarding translation pieces during decoding. Notice that when the translation piece correspond to a n-gram with $n \geq 2$, the last subword is the one to be rewarded, and the previous ones act as context. This context has to be present in the current translation hypothesis for the reward to be given.

C. Proposed Changes

In order to improve the performance of the baseline approach defined in [17], three further changes have been introduced. Looking at the approach described in Subsection V-B, it is noticeable that translation pieces keep being rewarded, even when they are already present in the current translation hypothesis. This may lead to the repetition of certain subwords, since guided-NMT keeps naively rewarding the same subwords repeatedly. In order to overcome this possible issue, we suggest only rewarding translation pieces subwords until they are present in the translation hypothesis.

Another possible source of performance degradation is the fact that the translation pieces are based on every retrieved sentence, even the ones that yield low similarity scores with the input sentence. Furthermore, the bigger the amount of translation pieces available, the slower the decoding process will be. Therefore, it is introduced a similarity threshold γ. Translation pieces from retrieved sentences that yielded a similarity below this threshold will be discarded.

Finally, the original work [17] proposes weighting all the translation pieces with the same weight. This might not be ideal, since translation pieces corresponding to uni-grams are always rewarded regardless of the context. Therefore, it is introduced a new weighting formula, which weights differently the uni-grams and the other translation pieces, given by

$$log_{p_{NMT}}(y_t|y_1, ..., y_{t-1}, \mathbf{X}) + = \lambda_1 tp_{value}(y_t) + \lambda_2 \sum_{n=2}^{4} tp_{value}(y_{t-n+1}, ..., y_t), \quad (17)$$

where $tp_{value}(.)$ corresponds to a mapping between a given n-gram of subwords and its value.

D. Guided-NMT for Domain Adaptation

One possible application for the previous architecture that [17]'s authors mention but do not explore is domain adaptation. A possible issue with this approach lies in Equation 17, which is only able to reward subwords which are part of the vocabulary of the NMT model. The proposed strategy is to incorporate in the vocabulary the translation pieces' subwords that appear during decoding, and that are not part of it. In terms of NMT models this also requires changing the decoder embedding layer and the output layer. The corresponding weights and bias are defined with the values the NMT model has learned for the unknown symbol. Using the same values as the ones obtained for the unknown symbol means that these subwords will be given a low log probability value. It is therefore introduced a new weight value τ, for the translation pieces that were added to the vocabulary. This value should be higher than λ due to the aforementioned log probability deficit for the unknown symbol. To sum up, using translation pieces to perform domain adaptation requires: using the same BPE encodings for the generic and in-domain data; extend the generic vocabulary with the unseen subwords; use a different weight τ, for the added subwords.

VI. Experiments

In order to evaluate the proposed models, we devise three sets of experiments. The first set of experiments concerns the fertility-based NMT model, whereas the last two set of experiments concern the guided-NMT approach. In particular, the latter is first tested in an in-domain setup, and afterwards it is evaluated in the scope of domain adaptation (using a model trained with generic data). For fertility-based NMT, as the main focus is on the problems of over and under-translations, we choose small datasets. Two language pairs are evaluated: German to English, using IWSLT 2014 data; Romanian to English, using WMT 2016 data. The number of parallel sentences in the training corpus are 153,326 and 560,767, respectively. For guided-NMT, the in-domain data is obtained from the UFAL medical corpus, using the *"medical_corpus"* tag to choose the relevant examples. Namely, we extract the top 1M sentences for the German to English data, and all the

available (437,922) for the Spanish to English language pair. As for the large generic corpus, used to perform the domain adaptation experiments, it used Europarlv7 data, for a total of around 2M sentence pairs for both German to English, and Spanish to English.

All data is tokenized and lowercased with Moses scripts, and BPE is applied with 32k merge operations [25], [35]. For the fertility-based NMT experiments it is also appended the $< \mathrm{SINK} >$ token to the end of every source sentence.

Furthermore, every model is trained using the OpenNMT-py framework [36], following the same overall architecture. We train an Encoder-Decoder model, using a 2-layer Bi-LSTM for the encoder and a 2-layer LSTM for the decoder. Furthermore, we use dropout of 0.3 [37], and the default attention mechanism, Luong's General Attention [23]. All the other parameters are kept as default, unless otherwise stated. While training, each model is evaluated on the development set after each epoch. We choose the model that yields the lowest perplexity on these evaluations. As for the decoder, we used beam search, with a beam size of 5.

The models are measured using BLEU [29], METEOR [38], and the aforementioned $\mathrm{REP}_{\mathrm{Score}}$ and $\mathrm{DROP}_{\mathrm{Score}}$.

A. Fertility-Based NMT

Using the development set to perform hyperparameter tuning, it was possible to find that for constrained softmax the optimal fertility strategy is the *constant*, with a value of 1.5 and $c = 0.6$, whereas for constrained sparsemax, it should be used the predicted fertility, with $c = 0.4$.

Besides using the different attention transformations already mentioned throughout this work, two other approaches to mitigate coverage problems during translation are going to be tested. These approaches are:

- *CovPenalty*, using the coverage penalty and length normalization presented in [11]. The corresponding α and β values are tuned in the development set, using grid search on $\{0.2k\}_{k=0}^{5}$.
- *CovVector*, adapted from [5]. To use this in the proposed fertility based NMT, the calculation of pre-attention scores using Luong's general alignment model is changed to $\boldsymbol{z}_{t,j} = \boldsymbol{h}_{t-1}^{d\ T}\left(\boldsymbol{W}\boldsymbol{h}_j^e + \boldsymbol{v}\boldsymbol{\beta}_{t-1,j}\right)$, where $\boldsymbol{v}$ is a matrix of parameters that multiplies the coverage vector, $\boldsymbol{\beta}_{t-1,j}$.

Looking at the results in Table I, it is possible to draw several conclusions. The sparse attention transformations (sparsemax and constrained sparsemax) have a positive impact in both machine translation coverage-related adequacy metrics, as does constrained softmax, which is able to introduce sparsity in the probability distributions over decoding time steps. Also, the proposed fertility-based NMT models are able to, on average, outperform two of the similar approaches existing in the literature, here presented as *CovPenalty* and *CovVector*.

In particular, for the German to English language pair, Constrained Sparsemax with predicted fertility and $c = 0.4$ is able to outperform every other approach across all metrics. When compared with the baseline softmax, it shows an improvement of around 1.5 BLEU points, 0.8 METEOR points and it is able to reduce $\mathrm{REP}_{\mathrm{Score}}$ by 31.1% and $\mathrm{DROP}_{\mathrm{Score}}$ by 19.6%.

For the Romanian to English language pair, even though the difference in performance is not as sound as for the previous language pair, the fertility-based NMT is still able to produce strong BLEU and METEOR scores while effectively reducing the metrics related with over and under-translation issues. When compared with the baseline softmax, using Constrained Softmax with a fixed fertility of 1.5 and $c = 0.6$ is able to reduce $\mathrm{REP}_{\mathrm{Score}}$ by 18.5% and $\mathrm{DROP}_{\mathrm{Score}}$ by 3.5%.

B. Guided-NMT

For guided-NMT we use the in-domain development set to tune the number of retrieved sentences per input sentence M, the similarity threshold γ, and the λ values. The values reported for the test set use $M = 5$, $\gamma = 0.4$, $(\lambda_1, \lambda_2) = (1.0, 1.0)$ for the German to English language pair, and $(\lambda_1, \lambda_2) = (1.1, 1.0)$ for the Spanish to English language pair. The obtained results are reported in Table II. We use the full training corpus as extra data.

1) Test Set Performance: It is possible to see that in terms of BLEU and METEOR, the guided NMT approach always outperforms the baseline, adding around 2 BLEU points in both cases. The reported $\mathrm{DROP}_{\mathrm{Score}}$ is virtually the same in both cases and it is noticeable the increase in $\mathrm{REP}_{\mathrm{Score}}$. This increment is larger for the Spanish to English translations and a similar increase was detected while tuning the λ values in the development set. It is possible to speculate that the $\mathrm{REP}_{\mathrm{Score}}$ reported for the baseline model is so low that, rewarding certain subwords during decoding, even when employing strategies to mitigate repetitions due to the translation pieces, slightly affects the translation in this regard. Nonetheless, the resulting value is still fairly low when compared, for instance, with the value reported for the German to English language pair. Also, the benefit across the other metrics outweighs the increase in $\mathrm{REP}_{\mathrm{Score}}$.

2) Translating Rare Words: One of the goals mentioned for the proposed approach is to be able to translate *n*-grams that appear in a test corpus, but that seldom occur in the data used to train the NMT model. In order to evaluate if the obtained translation is improving the rate of translation of infrequent *n*-grams, we follow an approach similar to the proposed in [17].

The first step to calculate the amount of correctly translated *n*-grams, with $1 \leq n \leq 4$, is to obtain the intersection between the set of *n*-grams present in a candidate translation and the reference, at a corpus level. This yields the *n*-grams that are both present in the output translation and in the reference, i.e., the correctly translated *n*-grams. Then, it is possible to find how many times each of those *n*-grams appeared in the training corpus. This value is referred as ψ. With this information, it is possible to count the number of instances of a given *n*-gram, both in the reference and candidate translations, knowing that it appeared ψ times in the training data. Finally, comparing the counts between the baseline NMT model and the guided-NMT, we can find if adding translation pieces indeed helps translating *n*-grams that seldom appear in the training corpus.

Table III show the *n*-grams counts for the baseline NMT model, the guided-NMT model and the ratio between both,

TABLE I
FERTILITY-BASED NMT - EVALUATION FOR THE DIFFERENT CONSTRAINED ATTENTION TRANSFORMATIONS. † INDICATES TRANSLATIONS WHOSE DIFFERENCE IN BLEU, WITH REGARD TO THE MODEL WITH THE HIGHEST SCORE, IS NOT STATISTICALLY SIGNIFICANT ($p < 0.01$), FOR EACH CONSTRAINED ATTENTION TRANSFORMATION.

	De-En				Ro-En			
Model	**BLEU ↑**	**METEOR ↑**	**REP ↓**	**DROP ↓**	**BLEU ↑**	**METEOR ↑**	**REP ↓**	**DROP ↓**
Softmax	29.18	31.21	4.47	6.07	**30.43**†	32.60	2.32	6.01
Softmax + CovPenalty	29.29	31.29	4.53	5.93	**30.44**†	32.63	2.34	5.96
Softmax + CovVector	29.66	31.54	4.03	5.47	**30.68**†	**32.79**	2.09	5.86
Sparsemax	29.56	31.33	4.41	5.77	30.12	32.39	2.11	6.08
Sparsemax + CovPenalty	29.59	31.38	4.43	5.66	30.09	32.46	2.12	5.92
Sparsemax + CovVector	29.98	31.68	3.70	5.65	**30.44**†	32.54	1.91	5.93
CSoftmax-Fixed-1.5-0.6	**30.26**†	31.86	3.49	5.35	**30.87**	32.74	**1.89**	**5.80**
CSparsemax-Predicted-0.4	**30.52**	**32.07**	**3.08**	**4.88**	**30.35**†	32.69	2.08	5.80

TABLE II
IN-DOMAIN GUIDED-NMT - RESULTS FOR THE IN-DOMAIN TEST SET.

	Model	**BLEU ↑**	**METEOR ↑**	**REP ↓**	**DROP ↓**
De-En	Base	52.12	40.89	**5.65**	11.45
	Guided	**54.01**	**41.50**	5.96	**11.41**
Es-En	Base	52.44	40.64	**2.38**	6.54
	Guided	**54.70**	**41.31**	3.77	**6.52**

for *n*-grams that appear ψ times in the training corpus. Even though the ratios are not much higher than one, the guided approach is always able to surpass the baseline. It is also observable that the ratios are larger for *n*-grams that are less frequent in the training corpus. This was to be expected due to the way NMT models are trained, which make it more difficult to output words/patterns that are not seen very often during training, as opposed to more common ones.

TABLE III
IN-DOMAIN GUIDED-NMT - COUNTS OF CORRECTLY TRANSLATED *n*-GRAMS THAT APPEAR ψ TIMES IN THE TRAINING CORPUS.

	ψ	**1**	**2**	**5**	**10**	**20**	**50**	**100**
	Base	7061	4249	2169	1092	586	282	143
De-En	Guided	7895	4823	2353	1184	632	301	151
	Guided/Base	1.12	1.14	1.08	1.08	1.08	1.07	1.06
	Base	3933	2510	1450	843	470	214	103
Es-En	Guided	4542	2835	1650	932	502	228	104
	Guided/Base	1.15	1.13	1.14	1.11	1.07	1.07	1.01

C. Guided-NMT for Domain Adaptation

When using a generic model to translate an in-domain test set, the extra data is even more critical for the improvement of the translations. The set of optimal hyperparameters reflects the larger role of translation pieces. In particular, it is found that optimal set of hyperparameters are: $M = 20$, $\gamma = 0.3$, $(\lambda_1, \lambda_2, \tau) = (3.5, 4.0, 5.0)$ for German to English; and $(\lambda_1, \lambda_2, \tau) = (2.5, 4.5, 5.0)$, for Spanish to English.

1) Test Set Performance: The first step for evaluating the model is to present the results obtained for the in-domain test set (Table IV). Namely, the results for German to English improve around 6 BLEU points, and for Spanish to English, the improvement is slightly more than 10 BLEU points. Also worth mentioning is that all other metrics show improvements, barring the $\mathrm{REP}_{\mathrm{Score}}$ in the Es-En experiments, where a very slight increase is reported. These results highlight the capacity of the translation pieces for domain adaptation, improving the quality of the translation of an in-domain test set across several metrics.

Also noticeable, are the larger improvements reported for the Spanish to English language pair. By obtaining some statistics regarding the collected translation pieces, it is possible to find that the average similarity for the German to English language pair is 0.457, with a median of 0.378, whereas for Spanish to English the average similarity is 0.642 and the median is 0.587. This is a noticeable difference, leading to the conclusion that, even with correctly tuned λs, the similarity of the collected sentences is critical for the success of this approach.

TABLE IV
GUIDED NMT FOR DOMAIN ADAPTATION - RESULTS FOR THE IN-DOMAIN TEST SET.

	Model	**BLEU ↑**	**METEOR ↑**	**REP ↓**	**DROP ↓**
De-En	Base	16.09	20.62	11.41	25.10
	Guided	**22.21**	**23.63**	**9.68**	**24.46**
Es-En	Base	22.03	25.86	**4.81**	12.63
	Guided	**32.20**	**29.95**	4.95	**12.34**

2) Translating Rare Words: We also evaluate how capable the model is of translating *n*-grams that are rare in the training corpus. Looking at Table V it is possible to observe that for lower values of ψ, the guided-NMT approach is more capable of producing the correct *n*-gram than the baseline. In fact, using translation pieces is shown to be able to induce the production of *n*-grams that very rarely appear in the training corpus, close to more than twice as often as the generic NMT model. Therefore, we may conclude that the impact of translation pieces in the production of rare *n*-grams is particularly noticed when using a generic model to translate an in-domain test set.

TABLE V
GUIDED NMT FOR DOMAIN ADAPTATION - COUNTS OF CORRECTLY TRANSLATED *n*-GRAMS THAT APPEAR ψ TIMES IN THE TRAINING CORPUS.

	ψ	**1**	**2**	**5**	**10**	**20**	**50**	**100**
	Base	1011	599	412	257	145	88	59
De-En	Guided	1986	1254	682	379	199	123	78
	Guided/Base	1.96	2.09	1.66	1.47	1.37	1.40	1.32
	Base	1262	778	459	264	141	84	73
Es-En	Guided	2249	1378	871	466	241	134	85
	Guided/Base	1.78	1.77	1.90	1.77	1.71	1.60	1.16

3) Impact of the Number of Extra Sentences: Even though the previous set of experiments is helpful and shows how translation pieces may serve as a possible approach to the problem of domain adaptation, they rely on large sets of extra sentences. When such a big amount of in-domain data is available, it is much more effective to train a model from scratch using that data. This is easily verified when comparing this section's results with the ones from Section VI-B.

A more realistic setting is one in which a NMT model is trained using a large generic corpus and then, as in-domain data is collected, it is used to performed some way of domain adaptation, such as fine-tuning [39]. Since fine-tuning requires re-training the model, which may be a cumbersome task, having the chance of leveraging collected in-domain data using translation pieces, poses as a computationally appealing alternative. Namely, coupling guided-NMT with periodic re-training of the model, is an approach that may conjugate the best of both approaches. In order to test this setup, the following three scenarios are going to be tested:

- *Translation Pieces*, where the baseline generic model uses translation pieces created from an extra corpus of in-domain sentences.
- *Fine-tuning*, where the generic model is fine-tuned using a given amount of in-domain sentences.
- *Latest Fine-tuned + Translation Pieces*, where a model fine-tuned in the previous number of available in-domain sentences has access to translation pieces from the currently available in-domain sentences. This simulates a more real context, where someone fine-tunes a model, later has access to more in-domain data, and wants to avoid re-training the model straightaway.

The obtained results are presented in Figure 4. A first observation to be made is that any of the approaches improves the obtained BLEU score, even when using a small amount of extra 5000 in-domain sentences. Previously, it has been observed that translating an in-domain test set with a model trained with in-domain data yields better results than using the generic model. Therefore it is expected that re-training the model, even if using a small amount of extra in-domain data, further improves BLEU scores. This is clearly observable, as the *Fine-tuning* results are always higher than the *Translation Pieces* ones.

The results of using fine-tuned models together with translation pieces are also particularly interesting. Namely, as long as we have fine-tuned models (corresponding to the last three measurements of the green line), this approach always outperforms simply using the fine-tuned model. This means that, not only the translations pieces are beneficial when using fine-tuned models, but also that the extra 5000 in-domain sentences have a larger impact when used as translation pieces (paired with a fine-tuned model) than when used to further fine-tune the generic model. This may be due to the fact that such a small amount of extra sentences might not be enough to push the model parameters that closer to the optimal for the in-domain scenario, when compared with the previously fine-tuned model, while being enough for the translation pieces algorithm to find much similar sentences and therefore, much more reliable *n*-grams to reward during decoding. The observed effect should be diluted over time as the fine-tuned model learns to give the correct probability to the necessary subwords.

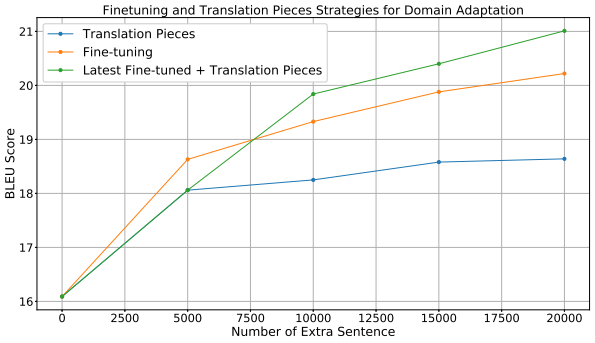

Fig. 4. Domain Adaptation Experiments.

VII. Conclusion

In Section I we raised three open questions regarding the problem of adequacy in neural machine translation. Our contributions in this work aim to answer those same questions. Namely, three major sources of poor adequacy were identified: over-translations, under-translations, and mistranslations.

We started by introducing $\mathrm{REP}_{\mathrm{Score}}$ and $\mathrm{DROP}_{\mathrm{Score}}$ as a way of measuring how problematic over and under-translations are in the produced translations. Furthermore, these metrics were shown to correlate well with the human judgment regarding those same identified problems.

Then, we proposed a fertility-based approach to NMT, which leverages both fertility, and sparse and constrained attention transformations. Using fertility-based NMT it was possible to improve the obtained translation with regard to, not only the commonly used MT metrics, such as BLEU and METEOR, but also also with regard to the two proposed metrics.

In order to improve the performance of NMT with regard to mistranslations, we followed [17], and further enhanced it by addressing some of the identified shortcomings of the original work. To evaluate its performance, first we used an in-domain setup, where it was able to yield improvements in terms of the overall MT metrics, as well as capable of translating more often *n*-grams that are rare in the training corpus. Then, we used it as a way of performing domain adaptation, something that had not been done in the past. In this case it was possible to obtain an even larger increase in terms of performance, with regard to the baseline. Furthermore, we have shown that a common domain adaptation technique, fine-tuning, is able to leverage translation pieces, implying an interesting synergy between both approaches. The mentioned conclusions are part of a comprehensive evaluation and error analysis we carried for all the proposed metrics on three language pairs, with overall encouraging results.

REFERENCES

[1] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models." in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013.

[2] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 2014 Advances in Neural Information Processing Systems*, 2014.

[4] M. Snover, N. Madnani, B. J. Dorr, and R. Schwartz, "Fluency, adequacy, or hter?: exploring different human judgments with a tunable mt metric," in *Proceedings of the Fourth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2009.

[5] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," in *Proceedings of the 2016 Annual Meeting of the Association for Computational Linguistics*, 2016.

[6] P. Arthur, G. Neubig, and S. Nakamura, "Incorporating discrete translation lexicon into neural machine translation," in *Proceedings of the 2016 Empirical Methods in Natural Language Processing*, 2016.

[7] P. Koehn and R. Knowles, "Six challenges for neural machine translation," in *Proceedings of the 2017 Workshop on Neural Machine Translation*, 2017.

[8] H. Mi, B. Sankaran, Z. Wang, and A. Ittycheriah, "Coverage embedding models for neural machine translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

[9] Z. Tu, Y. Liu, Z. Lu, X. Liu, and H. Li, "Context gates for neural machine translation," in *Transactions of the Association for Computational Linguistics*, vol. 5, 2017, pp. 87–99.

[10] Z. Tu, Y. Liu, L. Shang, X. Liu, and H. Li, "Neural machine translation with reconstruction." in *Proceedings of the 2017 AAAI Conference on Artificial Intelligence*, 2017.

[11] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," Google Brain Team, Tech. Rep., 2016.

[12] C. Hokamp and Q. Liu, "Lexically constrained decoding for sequence generation using grid beam search," in *Proceedings of the 2017 Annual Meeting of the Association for Computational Linguistics*, 2017.

[13] R. Chatterjee, M. Negri, M. Turchi, M. Federico, L. Specia, and F. Blain, "Guiding neural machine translation decoding with external knowledge," in *Proceedings of the 2017 Conference on Machine Translation*, 2017.

[14] M. A. Farajian, M. Turchi, M. Negri, and M. Federico, "Multi-domain neural machine translation through unsupervised adaptation," in *Proceedings of the 2017 Conference on Machine Translation*, 2017, pp. 127–137.

[15] X. Li, J. Zhang, and C. Zong, "One sentence one model for neural machine translation," *arXiv preprint arXiv:1609.06490*, 2016.

[16] J. Gu, Y. Wang, K. Cho, and V. O. Li, "Search engine guided non-parametric neural machine translation," in *Proceedings of the 2018 AAAI Conference on Artificial Intelligence*, 2018.

[17] J. Zhang, M. Utiyama, E. Sumita, G. Neubig, and S. Nakamura, "Guiding neural machine translation with retrieved translation pieces," in *Proceedings of the 2018 North American Chapter of the Association for Computational Linguistics*, 2018.

[18] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[19] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[20] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the 2015 International Conference on Learning Representations*, 2015.

[23] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[24] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[25] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 2016 Annual Meeting of the Association for Computational Linguistics*, 2016.

[26] C. Malaviya, P. Ferreira, and A. Martins, "Sparse and constrained attention for neural machine translation," in *Proceedings of the 2018 Annual Meeting of the Association for Computational Linguistics*, 2018.

[27] J. Yang, B. Zhang, Y. Qin, X. Zhang, Q. Lin, and J. Su, "Otem&utem: Over-and under-translation evaluation metric for nmt," in *Proceedings of the 2018 CCF International Conference on Natural Language Processing and Chinese Computing*, 2018.

[28] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of ibm model 2," in *Proceedings of the 2013 North American Chapter of the Association for Computational Linguistics*, 2013.

[29] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 2002 Annual Meeting of the Association for Computational Linguistics*, 2002.

[30] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the Association for Computational Linguistics 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005.

[31] A. Martins and R. Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *Proceedings of the 2016 International Conference on Machine Learning*, 2016.

[32] A. Martins and J. Kreutzer, "Learning what's easy: Fully differentiable neural easy-first taggers," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

[33] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[34] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[35] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 2007 Annual Meeting of the Association for Computational Linguistics on Interactive Poster and Demonstration Sessions*, 2007.

[36] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, "Opennmt: Open-source toolkit for neural machine translation," *Proceedings of the 2017 Association for Computational Linguistics: System Demonstrations*, 2017.

[37] N. Srivastava, "Improving neural networks with dropout," Ph.D. dissertation, University of Toronto, 2013.

[38] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proceedings of the European Chapter of the Association for Computational Linguistics 2014 Workshop on Statistical Machine Translation*, 2014.

[39] M.-T. Luong and C. D. Manning, "Stanford neural machine translation systems for spoken language domains," in *Proceedings of the 2015 International Workshop on Spoken Language Translation*, 2015.