Priming Neural Machine Translation

MinhQuang Pham[†‡], **Jitao Xu**[†‡], **Josep Crego**[†], **Jean Senellart**[†], **François Yvon**[‡]

[†]SYSTRAN, 5 rue Feydeau, 75002 Paris, France
firstname.lastname@systrangroup.com
[‡]Université Paris-Saclay, CNRS, LIMSI, 91400, Orsay, France
firstname.lastname@limsi.fr

Abstract

Priming is a well known and studied psychology phenomenon based on the prior presentation of one stimulus (cue) to influence the processing of a response. In this paper, we propose a framework to mimic the process of priming in the context of neural machine translation (NMT). We evaluate the effect of using similar translations as priming cues on the NMT network. We propose a method to inject priming cues into the NMT network and compare our framework to other mechanisms that perform micro-adaptation during inference. Overall, experiments conducted in a multi-domain setting confirm that adding priming cues in the NMT decoder can go a long way towards improving the translation accuracy. Besides, we show the suitability of our framework to gather valuable information for an NMT network from monolingual resources.

1 Introduction

Priming is a well studied human cognitive phenomenon, founded on the establishment of associations between a stimulus and a response (Tulving et al., 1982). Multiple studies have shown how external stimuli (cues) may have a profound effect on perception. In the case of language translation, external stimuli having such effects are said to prime language understanding and potentially have a impact the actions of a human translator. Imagine for instance a translator facing the ambiguous sentence *I was in the bank*, and the effect on translation accuracy if primed with the cue *river*. Most likely, the human translator would consider the "edge of river" sense rather than "financial institution" for translation. In the context of human translation, cross-lingual priming is particularly effective as cues in the target language may notably influence the final translation word choice.

Several research works have introduced the priming analogy in deep neural networks. In computer vision priming has been broadly studied: for instance, in Rosenfeld et al. (2018), the authors introduce a cue about the presence of a certain class of object in an image that significantly improves object detection performance. Concerning language generation, Brown et al. (2020) use a combination of prompt and example to guide the GPT-3 network when performing a task, where the prompt is a sentence that describes the task (*i.e. "Translate from English to French"*); and is followed by an example of the task (*i.e. "sea otter ⇝ loutre de mer"*). In the context of NMT, experiments reported (Sennrich et al., 2016a; Kobus et al., 2017; Dinu et al., 2019) aim at influencing translation inference with respectively politeness, domain and terminology constraints. More related to our work, (Bulte and Tezcan, 2019; Xu et al., 2020) introduce a simple and elegant framework where similar translations (cues) are used to prime an NMT model, effectively boosting translation accuracy. In all cases, priming is performed by injecting cues in the input stream prior to inference decoding.

In this paper, we extend a framework that mimics the priming process in neural networks, in the context of machine translation. Following up on previous work (Bulte and Tezcan, 2019; Xu et al., 2020), we consider similar translations as external cues that can influence the translation process. We push this concept further: a) by proposing a novel scheme to integrate similar translation cues into the NMT network. We examine the attention mechanism of the network and confirm that priming stimuli are actually taken into account; b) by extending an efficient network to train distributed representations of sentences that are used to identify accurate translations used as priming cues[1]; c) by analyzing how on-the-fly priming compares to micro-adaptation (fine-tuning). Finally, we

[1] https://github.com/jmcrego/cbon

Proceedings of the 5th Conference on Machine Translation (WMT), pages 516–527
Online, November 19–20, 2020. ©2020 Association for Computational Linguistics

show that our priming approach can also be used with monolingual data, providing a scenario where NMT can be effectively helped by large amounts of available data. Our proposal does not require to change the NMT architectures or algorithms, relying solely on input preprocessing and on prefix (forced) decoding (Santy et al., 2019; Knowles and Koehn, 2016), a feature already implemented in many NMT toolkits.

The remainder of the paper is organized as follows: Section 2 gives details regarding our priming approach. The experimental framework is presented in Section 3. Results and discussion are respectively in Sections 4 and 5. We review related work in Section 6 and conclude in Section 7.

2 NMT Priming On-the-fly

This section describes our framework for priming neural MT with similar translations. We follow the work by (Bulte and Tezcan, 2019; Xu et al., 2020) and build a translation model that incorporates similar translations from a translation memory (TM) to boost translation accuracy. In this work, TMs are parallel corpora containing translations falling in the same domain as test sentences.

We first describe the methods employed in this work to compute sentence similarity. We then introduce various augmentation schemes considered to prime the NMT network with retrieved similar translations. Overall, we pay special attention to efficiency, since retrieval is applied on a sentence-by-sentence basis at inference.

2.1 Similarity Computation

We detail the sentence similarity tools evaluated in this work. The first employs discrete word representations, while the rest rely on building distributed representations of sentences to perform similar sentence retrieval:

FM: fuzzy matching is a lexicalized matching method aimed to identify non-exact matches of a given sentence. Following Xu et al. (2020), we use `FuzzyMatch`[2], where the fuzzy match score $\text{FM}(s_i, s_j)$ between two sentences s_i and s_j is:

$$\text{FM}(s_i, s_j) = 1 - \frac{\text{ED}(s_i, s_j)}{max(|s_i|, |s_j|)}$$

with $\text{ED}(s_i, s_j)$ being the Edit Distance between s_i and s_j, and $|s|$ is the length of s.

S2V: we use `sent2vec`[3] (Pagliardini et al., 2018) to generate sentence embeddings. The network implements a simple but efficient unsupervised objective to train distributed representations for sentences. The model is based on efficient matrix factor (bilinear) models (Mikolov et al., 2013a,b; Pennington et al., 2014).

Borrowing the notations of Pagliardini et al. (2018), training the model is formalized as an optimization problem:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \sum_{s \in \mathcal{C}} f_s(\boldsymbol{U}\boldsymbol{V}\iota_s)$$

for two parameter matrices $\boldsymbol{U} \in \mathbb{R}^{|\mathcal{V}| \times d}$ and $\boldsymbol{V} \in \mathbb{R}^{d \times |\mathcal{V}|}$, where $\mathcal{V}$ denotes the vocabulary and d is the embedding dimension. Minimization of the cost function f_s is performed on a training corpus $\mathcal{C}$ of sentences s.

In `sent2vec`, ι_s is a binary vector encoding the bigrams in s (bag of bigrams encoding).

CBON: the Continuous Bag of n-grams (`CBON`) model denotes our re-implementation of the previous `sent2vec` model. In addition to multiple implementation details, the main difference is the use of arbitrary large n-grams to model sentence representations, where `sent2vec` only used bigrams.

Both `sent2vec` and `CBON` learn a source (or context) embedding $\boldsymbol{v}_w$ for each n-gram w in the vocabulary $\mathcal{V}$. Once the model is trained, the embedding of sentence s (h_s) is obtained as the average of its n-gram embeddings:

$$\boldsymbol{h}_s = \frac{1}{|R(s)|} \sum_{w \in R(s)} \boldsymbol{v}_w$$

where $R(s)$ is the list of n-grams (including unigrams) occuring in sentence s and $\boldsymbol{v}_w$ is the target embedding of n-gram w.

The similarity score $\text{EM}(s_i, s_j)$ between two sentences s_i and s_j is then defined via the cosine similarity of their sentence vector representations h_i and h_j:

$$\text{EM}(s_i, s_j) = \frac{h_i \cdot h_j}{||h_i|| \times ||h_j||},$$

where $||h||$ denotes the norm of vector h.

Note that models differ in their vocabularies, which are built selecting the most frequent n-grams.

[2] https://github.com/systran/FuzzyMatch

[3] https://github.com/epfml/sent2vec

Both models implement Negative Sampling to avoid the softmax computation.

2.2 Priming Schemes

We now explore various ways to integrate similar translations for priming NMT:

tgt^k we follow here mostly the work of Bulte and Tezcan (2019), where the input sentence in the source language is augmented with the k translations (in the target language) having the highest matching score (FM or EM) in the TM.

In training, sentence pairs (**s**,**t**) are preprocessed as follows: the source sentence **s** is concatenated with translations t^k of the k most similar sentences (s^k) to **s** found in the TM. Augmented translations are sorted by matching score, with $k = 1$ denoting the most similar. Sentences in the source stream are separated using the special token $\circ$.

$$\texttt{src:}\quad t^k \circ ... \circ t^2 \circ t^1 \circ \mathbf{s}$$
$$\texttt{tgt:}\quad \mathbf{t}$$

In inference, only the source-side is input to the translation network.

In Xu et al. (2020), an issue regarding *unrelated* tokens present in similar translations t^k is raised. The model effectively learns to copy most of the content present in similar translations, but has difficulties to avoid also copying *unrelated* words. Consider for instance the input sentence $s =$ *pertussis vaccin* with similar sentence $s^1 =$ *measles vaccin* and its corresponding translation $t^1 =$ *vaccin contre la rougeole*. Following the **tgt**^{k} scheme, the NMT input consists of:

vaccin contre la rougeole $\circ$ **pertussis vaccin**

yielding the output: **vaccin contre la rougeole**. The word *rougeole* is actually the translation of an unrelated word (*measles*). The model often copies such *unrelated* tokens (Xu et al., 2020), due to the fact that they are present in the input stream as similar translations (t^k) and are usually semantically related to the correct translation choice (here *coqueluche*, the correct translation for *pertussis*).

tgt^k**+STU** adopts the proposal of Xu et al. (2020) to alleviate the *unrelated word* problem. It relies on an additional source stream (factor) to label related/unrelated tokens. Following on our example, in this scheme the input of the NMT model contains two parallel streams:

src$_1$:	vaccin	contre	la	rugeole	$\circ$	**pertussis**	**vaccin**
src$_2$:	T	T	T	U	T	S	S
tgt:	vaccin	contre	la	coqueluche			

Tokens in the second stream are: S for source tokens, U for unrelated and T for related target tokens. *rougeole* is thus tagged as an *unrelated* word that must not be copied in the translation output. Word embeddings are built after concatenating both factor embeddings. Xu et al. (2020) claim achieving a 8% reduction of unrelated tokens when using this scheme.

Note that this solution is computationally expensive as it requires to identify related/unrelated tokens in each input sentence and in the corresponding similar translations, based in Xu et al. (2020) on word alignments and edit distance computations.

s+t^k the solution proposed in this paper also addresses the *unrelated word* problem, at a much reduced computational cost. It considers both sides of similar translations (s^k and t^k). Training streams take the form:

$$\texttt{src:}\quad s^k \circ ... \circ s^2 \circ s^1 \circ \mathbf{s}$$
$$\texttt{tgt:}\quad t^k \circ ... \circ t^2 \circ t^1 \circ \mathbf{t}$$

In inference, target-side similar translations t^k are used by the model as a target prefix. The initial steps of the beam search use the given prefix $t^k \circ ... \circ t^2 \circ t^1 \circ$ in forced decoding mode, returning to a regular beam search after the last $\circ$ token is generated. A similar strategy of concatenating previous and current sentences was explored by Tiedemann and Scherrer (2017) in the context of handling discourse phenomena. However, since we use true translation as prefixes, our strategy does not suffer from exposure bias (Ranzato et al., 2016) and the subsequent error propagation problem. Continuing on our running example, during inference the model receives:

$\texttt{input:}$ *measles vaccin* $\circ$ **pertussis vaccin**
$\texttt{prefix:}$ *vaccin contre la rougeole* $\circ$

the encoder embeds the input stream, and force-decodes the target prefix, before starting the translation generation. Note that during beam search, the decoder has thus access both to all input tokens (s^k and s) as well as to similar translations t^k (in the translation prefix).

Following our approach the NMT model learns to attend to priming cues on both source and target streams. Besides, our solution removes the need to mix source and target vocabularies as in previous schemes.

3 Experimental Framework

3.1 Corpora

We experiment with the English-French language pair and data originating from eight domains, corresponding to texts from three European institutions: the European Parliament (EPPS), the European Medicines Agency (EMEA) and the European Central Bank (ECB); Legislative texts of the European Union (JRC); IT-domain corpora corresponding to KDE4 and GNOME; News Commentaries (NEWS); and parallel sentences extracted from Wikipedia (WIKI). Table 1 contains statistics regarding the corpora used in this work[4] (Tiedemann, 2012). Statistics are computed after splitting off punctuation.

Corpus	#Sents (K)	L_{mean}		Vocab (K)	
		English	French	English	French
		Parallel Corpora			
EPPS	1,992.8	27.7	32.0	129.5	149.2
NEWS	315.3	25.3	31.7	90.5	96.7
WIKI	749.0	25.9	23.5	527.5	506.6
ECB	174.1	28.6	33.8	45.3	53.5
EMEA	336.8	16.8	20.3	62.8	68.9
JRC	475.2	30.1	34.5	81.0	83.5
GNOME	51.9	9.6	11.6	19.0	21.6
KDE4	163.9	9.1	12.4	48.7	64.7
		Monolingual Corpora			
WIKI	6,426.8	-	24.1	-	1,626.3
NEWS	83,567.8	-	25.5	-	3,444.1

Table 1: Corpora statistics. Note that K stands for thousands and L_{mean} is the average length in words.

Each corpora is considered as a different domain. Training data sets are also employed as TM of the corresponding domain. This is, similar sentences are mined from the same training set that is used to build the model. Note that we also consider monolingual (French) corpora. For the News domain we use all available monolingual WMT news crawl data[5]. For the Wikipedia domain, we use the French-side of the WikiMatrix data (Schwenk et al., 2019a).

We randomly split the parallel corpora by keeping 500 sentences for validation, $1,000$ sentences for testing and the rest for training. All data is preprocessed using the OpenNMT tokenizer[6] (conservative mode).

3.2 System Configurations

This section gives learning/inference details of the various systems used in this work.

Similarity

For fuzzy matching **FM** we follow several works (Koehn and Senellart, 2010; Bulte and Tezcan, 2019; Xu et al., 2020) and keep the n-best matches when $\text{FM}(s_1, s_2) \geq 0.5$ with no approximation. Concerning **S2V**, the model is trained with default options during 20 epochs using all training data. We use an embedding dimension of 300 cells. Regarding **CBON**, we learn models using also the entire training data during one epoch (~50,000 iterations). Similarly to **S2V** we use 10 negative samples per positive word to approximate the softmax, a batch size of $2k$ examples, and embedding size of 300 cells. We build CBON models using 3-grams and 4-grams to enable a comparison with sent2vec which only uses bigrams. All vocabularies are selected keeping the 500,000 most frequent n-grams ($n = 2$ for **S2V** and $n = 3$ and 4 for **CBON**).

For both **CBON** and **S2V** models, we use the 5-best matches when $\text{EM}(s_1, s_2) \geq 0.8$[7]. In all cases, perfect matches are not used for training. Accuracy results on the priming task indicate that 3-grams yield slightly lower accuracy results than those obtained with 4-grams. In the remainder, we always use the 4-gram version of **CBON**.

Sentence Retrieval

To identify similar translations using distributed representations, we use the faiss[8] search toolkit (Johnson et al., 2019) through its Python API with exact *FlatIP* index.

Translation

Our NMT models rely on the Transformer base architecture of Vaswani et al. (2017), implemented in the OpenNMT-tf[9] toolkit (Klein et al., 2017). We use the standard setting of Transformers for all experiments: size of word embedding: 512; size of hidden layers: 512; size of inner feed-forward layer: $2,048$; number of heads: 8; number of layers in the encoder or in the decoder: 6. In the **tgt**^{1}**+STU** scheme, token (508 cells) and **STU** (4

[4] Freely available from http://opus.nlpl.eu

[5] http://data.statmt.org/news-crawl/

[6] https://github.com/OpenNMT/Tokenizer

[7] Optimization experiments on a held-out development set are carried out for both models.

[8] https://github.com/facebookresearch/faiss

[9] https://github.com/OpenNMT/OpenNMT-tf

cells) streams are concatenated, thus using the same number of parameters in all schemes.

For training, we use the Adam (Kingma and Ba, 2015) optimiser with a batch size of $4,096$ tokens. We set the warmup steps to $4,000$ and update the learning rate for every 8 iterations. Models are optimised during $300K$ iterations, using a single NVIDIA V100 GPU. We limit the length of training sentences to 300 BPE tokens (Sennrich et al., 2016c) in both source and target sides to enable the integration of similar sentences. We use a joint BPE-vocabulary of size 32K for both source and target texts. Inference is performed with a beam size of 5 using CTranslate2[10], a custom C++ runtime inference engine for OpenNMT models that enables fast CPU decoding and also implements prefix decoding. For evaluation, we report BLEU (Papineni et al., 2002) scores computed by detokenized case-sensitive `multi-bleu.perl`[11].

We re-implement the work of Farajian et al. (2017) as a contrastive model that we denote μ**adapt**. Note that we only experiment with the basic version of this work, where the closest neighbours of the input sentence are first retrieved from the memory and then used to fine-tune a generic model during 15 additional iterations with a fixed learning rate of 0.0005; the fine-tuned model is then used to produce the translation of the given input sentence. In addition, Farajian et al. (2017) include a variant where learning rate and number of epochs are dynamically adapted considering sentence similarity. Adaptation is run on a sentence-by-sentence basis.

4 Results

Retrieval algorithms employed in this work are significantly faster than NMT Transformer decoding, thus implying a limited decoding overhead.

Table 2 reports efficiency scores (tokens/second) for computing vector representations (Vector), performing sentence retrieval (Retrieval) and translation (NMT) for the WIKI test set according to the similarity model and priming schema used. Results show that the computational cost is dominated by the NMT step. This step, in turn, is affected by the length of the input (and prefix) streams.

Model	Schema	Vector	Retrieval	NMT
Base	-	-	-	806
FM	**tgt**^{1}	-	25K	750
	s+t^{1}			687
S2V	**tgt**^{5}	222K	17K	639
CBON	**tgt**^{5}	59K		
	s+t^{5}			523

Table 2: Efficiency (tokens/second) of each step for different inference configurations. All steps run on CPU (16 cores). K stands for thousands.

Table 3 reports BLEU scores for our various configurations, tested on 8 domain-specific test sets. The last column (avg) reports average results. This table also reports the number of input sentences (out of $1,000$) for which at least one similar sentence was retrieved (in a smaller font).

All NMT models are built using the concatenation of the original parallel corpora in Table 1. Our **Base** configuration does not integrate similar sentences in the training data. All other models extend the original corpora with sentences retrieved following similarity methods (Sim) introduced in Section 2.1 and integration schemes presented in Section 2.2 (Scheme).

The second block of results in Table 3 displays scores obtained when performing translations extended with fuzzy matches **FM**. In line with results presented by Xu et al. (2020), using a second stream to mark related/unrelated tokens (**+STU**) yields a boost in performance of around 1 BLEU points. When the **s+t**^{1} scheme is used, the average improvement reaches 1.25 BLEU points.

The third block compares translation results obtained when identifying similar translations by **S2V** and **CBON**. In both cases, the **s+t**^{5} scheme is used. The choice for 5-best similar translations and $\text{EM}(s_i, s_j) \geq 0.8$ threshold is made after running optimization work on a held out development set. Sentences identified by **CBON** outperform those selected by **S2V**. The idiosyncrasy of fuzzy matching does not enable to find multiple similar sentences for a given input sentence. Overall best results are obtained by the **CBON** **s+t**^{5} configuration. Note that as expected, the number of similar translations found using distributed representations is larger than those found by fuzzy matching.

Finally, the last block in Table 3 gives results for a system that retrieves similar sentences to dynamically adapt the model on a sentence-per-

[10] https://github.com/OpenNMT/CTranslate2

[11] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

Sim	Scheme	ECB	EMEA	EPPS	GNOME	JRC	KDE4	NEWS	WIKI	avg
Base	-	49.23	49.53	42.83	49.99	59.05	49.52	**36.66**	35.15	46.50
FM	**tgt**^{1}	56.21	59.34	42.08	60.95	65.86	53.49	35.80	34.54	51.03
(Bulte and Tezcan, 2019)		585	765	195	686	612	575	54	184	457
FM	**tgt**^{1}**+STU**	**57.30**	61.03	42.95	62.68	67.24	54.68	35.54	35.16	52.07
(Xu et al., 2020)		585	765	195	686	612	575	54	184	457
FM	**s+t**^{1}	56.16	60.88	43.18	62.50	67.58	55.25	36.55	36.94	52.38
		585	765	195	686	612	575	54	184	457
S2V	**s+t**^{5}	57.16	60.44	**43.19**	62.44	65.39	51.32	35.98	35.82	51.47
		740	840	161	639	735	623	39	297	509
CBON	**s+t**^{5}	56.50	**61.04**	42.22	**63.76**	**68.75**	**55.83**	35.41	36.38	**52.49**
		710	896	195	854	733	862	63	378	586
FM	μ**adapt**	53.09	55.02	43.04	53.88	62.99	48.70	36.48	35.81	48.63
(Farajian et al., 2017)		585	765	195	686	612	575	54	184	457
CBON	μ**adapt**	53.41	53.32	43.20	54.77	63.37	52.06	36.47	36.39	49.12
(Farajian et al., 2017)		710	896	195	854	733	862	63	378	586

Table 3: BLEU scores for various model configurations and 8 test domains. Smaller numbers correspond to the number of input sentences in each domain for which at least one similar sentence is found.

sentence basis (Farajian et al., 2017; Li et al., 2018). We show micro-adaptation results when similar sentences are found by **CBON** and **FM** models (μ**adapt**). In our experiments, micro-adaptation does not yield the gains observed with priming methods. As previously stated, the best performing variants of the adaptation method presented in Farajian et al. (2017) were not included in our comparison. Variants employ a dynamically adapted learning rate and number of epochs.

Monolingual Corpora

Retrieval results shown in Table 3 (small font numbers) indicate a reduced number of similar sentences found for some domains (NEWS, EPPS and WIKI). In the context of scarce similar sentences, the boost in translation quality observed for most domains is subsequently reduced. The case of the **NEWS** domain is particularly harmful since worst results are always obtained when compared to our **Base** system.

However, very large monolingual collections of texts exist, far exceeding the amount of available parallel corpora. The latter are more expensive to collect and typically only exist for a limited number of domains and language pairs. With the objective to enhance NMT with monolingual corpora, we now apply the methods presented above to monolingual corpora.

We collect monolingual corpora in the target language (French in this work) and translate each sentence back into English to obtain synthetic parallel data. Similar to back-translation experiments in Sennrich et al. (2016b), we only use original (human-crafted) target-language data. We expect this to add less noise than incorporating synthetic target-language data into the NMT input. Once translated into English, the various priming approaches identify similar synthetic sentences and injects both the synthetic source and original target in the NMT input stream. Note that cross-lingual sentence embedding models exist (Sabet et al., 2019; Schwenk and Douze, 2017; Conneau and Lample, 2019) but our preliminary experiments using these tools did not show satisfactory results.

Thus, we exploit large collections of French texts for the News and Wikipedia domains (as detailed in Table 1) that we translate into English to enable similarity retrieval. Table 4 reports BLEU scores obtained by our best performing network **CBON** following the **s+t**^{5} scheme.

The supplementary number of similar sentences (468 input sentences have similar translations) collected for the WIKI domain over parallel and monolingual[12] corpora (par+mon) yields an improvement of 2 BLEU points. However, very few (97) similar sentences are identified[13] over near 95 million sentences (par+mon), showing a small gain when compared to using only parallel sentences (par). The network does not succeed to outperform the accuracy of the **base** system. As outlined by Bulte and Tezcan (2019) and Xu et al. (2020) the accuracy of networks implementing priming may slightly drop in performance when no similar translations are integrated.

Sim	Scheme	Data	NEWS	WIKI
Base	-	-	**36.66**	35.15
CBON	**s+t**^{5}	par	35.41 63	36.38 378
CBON	**s+t**^{5}	par+mon	36.05 97	**38.20** 468

Table 4: Translation performance for the NEWS and WIKI domain test sets using similar sentences retrieved from parallel data (par) and from both parallel and monolingual (par+mon) data. The first two rows correspond to experiments already shown in Table 3.

5 Discussion

Unrelated Words

As previously outlined in Section 2, Xu et al. (2020) raised a problem regarding *unrelated* words. It concerns those words that, even through they appear in similar translations, must not be used to translate input sentences. An example of translation with unrelated word is given in Section 2.2 where the input sentence with similar translation:

vaccin contre la rougeole ∘ **pertussis vaccin**

is translated as: **vaccin contre la rougeole**, the right translation being: **vaccin contre la coqueluche**. The error is due to the fact that word *rougeole* is present in the input stream and is semantically related to *coqueluche*. The problem is particularly hurting when it involves keywords (like the proper noun in our example) which convey essential information regarding the meaning of sentences.

The work by Xu et al. (2020), that we denoted **tgt**^{1}**+STU**, obtains an average reduction of these erroneous words in the translation hypotheses of 8%. We conduct the same experiment to analyse the performance of the new scheme **s+t**^{1} introduced in this work. Table 5 reports the total number of unrelated words in 1-best similar sentences obtained by fuzzy matching[14]. As can be seen, the scheme **s+t**^{1} further mitigates the apparition of unrelated words in translations, with a drop of -8.3%.

NMT Attention

We analyse the Encoder and Decoder self-attention layers, aiming to better understand how our **CBON s+t** model configuration makes use of similar translations.

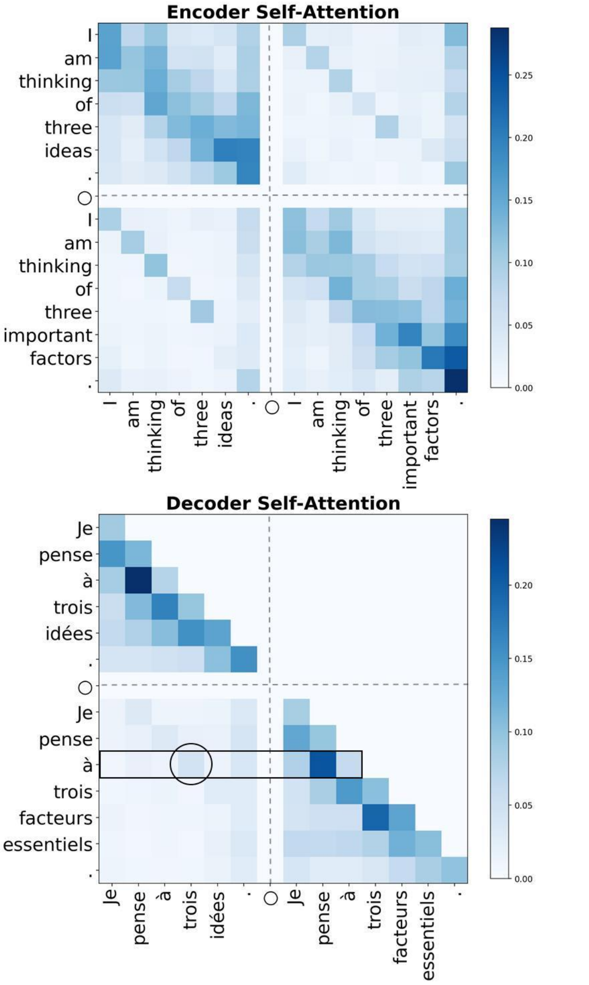

Figure 1: Average attention values of all heads through all layers for the encoder (top) and decoder (bottom). Dashed lines are used to separate similar and input sentences.

Figure 1 displays the attention[15] values for sen-

[12] Test French sentences entirely found in monolingual WIKI corpora are not considered as similar translations.

[13] In all cases we consider similar sentences s_i and s_j when (EM$(s_i, s_j) \geq 0.8$)

[14] We follow the procedure detailed in Xu et al. (2020) to identify related/unrelated words.

[15] We use the average of all heads through all layers.

Scheme	ECB	EMEA	EPPS	GNOME	JRC	KDE4	NEWS	WIKI	avg
tgt^1**+STU**	3,555	2,320	312	1,285	3,515	940	39	344	1,538
s+t^1	3,199	1,985	306	1,195	3,413	845	31	310	1,410
unrelated	6,310	4,405	4,405	2,473	6,309	2,358	236	1,591	3,510

Table 5: Number of unrelated words appearing in test sets according to different augmentation schemes. The last row indicates the total number of unrelated words included in 1-best **FM** similar sentences.

tence s = [**I am thinking of three important factors .**] when translated into t = [**Je pense à trois facteurs essentiels .**] using the similar translation example s^1 = [*I am thinking of three ideas .*] and t^1 = [*Je pense à trois idées .*]. For visualization purposes we mask the attention of the sentence separator token ∘.

Concerning the encoder self-attention (top), we can clearly observe that the encoder pays attention to the words in the similar sentence (down-left) when embedding the input sentence (down-right). Equivalently, the decoder self-attention (bottom) also attends to the similar translation (down-left: prefix words generated in forced mode) when producing the translation of sentence s. Note that when the decoder is about to generate the French word *trois [three]*, attention weights (rectangle) are the highest for the preceding words (in particular to *pense [think]*), with *trois* (circle in the similar translation) also receiving a substantial weight. This suggests that the model has learned to use similar translations passed in the form of a target prefix to help generating translations.

Priming Model

The priming network leverages similar sentences from a TM so as to yield more accurate translations. From a mathematical perspective, the search for the best translation $\bar{t}$ is conditioned to the input sentence s as well as to similar pairs of translations s^1 and t^1:

$$\bar{t} = \arg\max_{t} P(t|s, s^1, t^1)$$

to facilitate reading we use one single similar translation (s^1 and t^1) rather than k-best translations.

To evaluate the intuition that $P(t|s, s^1, t^1)$ gives better translations than $P(t|s)$, we report the average of $\log P(t|s, s^1, t^1)$ computed by **CBON s+t**^{5} and of $\log P(t|s)$ computed by **Base** over test sets sentences with similar sentences translations.

Table 6 reports the difference between the token average of $\log P(t|s, s^1, t^1)$ and the token average of $\log P(t|s)$. More precisely, for each test sentence s, we compute the log probability of predicting reference t, we then sum all the calculated log probabilities and divide the sum by the total number of tokens in the references. For each test set, we computed the average log probability of model **CBON s+t**^{5} and **Base**. We report the difference in the average of both models. Results indicate that $\log P_{\texttt{CBON s+t}^5}(t|s^1, s, t^1)$ are actually greater than $\log P_{\texttt{Base}}(t|s)$ in most cases, with the exception of EPPS and NEWS for which the base system yields higher probabilities. We observe a strong correlation between values reported and the gap in BLEU score for the same model configurations.

Domain	**CBON s+t**^{5} − **Base**
ECB	0.222
EMEA	0.231
EPPS	-0.039
GNOME	0.248
JRC	0.165
KDE4	0.252
NEWS	-0.173
WIKI	0.009

Table 6: Differences of token average log probability between **CBON s+t**^{5} and **Base** model.

Similarity over Synthetic Sentences

Results in Table 4 show a clear boost in performance (∼2 BLEU points) when making use of synthetic translations of the WIKI monolingual data set. We now want to measure the noise introduced by synthetic translations when compared to human translations. Thus, we consider the input sentences of the WIKI test set for which we found similar sentences in both the parallel (human translation) and monolingual (synthetic translation) corpus (279 sentences).

Results in Table 7 show a clear drop in BLEU scores when using synthetic matches. As expected, machine translation quality degrades the results of similarity search which in turns provides less valuable similar translations.

Priming sentences	WIKI
par (human)	52.50
mon (synthetic)	49.94

Table 7: Results for a reduced test set (279 sentences) using **CBON** when priming with human and synthetic (back-translated) translations.

6 Related Work

Our work relates to the ideas introduced in Bulte and Tezcan (2019) and Xu et al. (2020). Both of them leverage similar translations from parallel corpora and inject similar sentences in the NMT network. While Bulte and Tezcan (2019) integrates fuzzy matches into the NMT model by concatenating similar translations to source sentences, Xu et al. (2020) extended the framework by adding additional source side features to distinguish between related and unrelated words, employed distributed sentence representations. A similar idea is also explored in Schwenk et al. (2019b), where the authors use multilingual sentence embeddings to retrieve pairs of similar sentences and train models uniquely with such sentences.

Previously, Niehues et al. (2016) augmented input sentences with pre-translations generated by a phrase-based MT system. Our work, in contrast, integrates similar sentences in both source and target sides and employs similar translations found in parallel as well as monolingual data sets.

A similar strategy of concatenating previous and current sentences was explored by Tiedemann and Scherrer (2017) further evaluated by Bawden et al. (2018) in the context of tackling discourse phenomena. Our work employs force decoding to allow including true translations in the decoder target-side. Thus, avoiding the error propagation problem (Ranzato et al., 2016) of longer sequences in auto-regressive models.

Bapna and Firat (2019) propose a neural MT model that incorporates retrieved neighbours relying on local phrase level similarities. Using deep pre-trained models (Peters et al., 2018; Radford et al., 2019; Devlin et al., 2019; Le et al., 2020; Conneau and Lample, 2019) to compute contextualized sentence representations has become common fashion in recent works (Feng et al., 2020; Chang et al., 2020). However, deep models suffer from computation complexity when applied on-the-fly for inference. We propose an extension of sent2vec (Pagliardini et al., 2018) to compute sentence representations that also inherits from the computationally efficient bilinear models (Mikolov et al., 2013a,b; Pennington et al., 2014).

Similar to our work, Farajian et al. (2017) and Li et al. (2018) retrieve similar sentence to dynamically adapt each individual input sentence. Farajian et al. (2017) obtains best performance when tuning the adaptation learning rate and number of epochs according to level of similarity between the input and retrieved sentences. In Xu et al. (2019) the model is dynamically adapted to a entire test set to reduce adaptation time.

In computer vision, priming network has been recently studied. For the object detection task , Rosenfeld et al. (2018) primed the network via an external information that affects all the processing layers. Upon processing each image in the network, Rosenfeld et al. (2018) also presented the network with the category of the object in the image; this information is injected at all layers.

7 Conclusions

Inspired by the human psychological phenomenon of priming, we have presented a simple framework for priming NMT networks. Following other research works, we used similar translations as priming cues to influence the NMT network. We presented a novel method that injects similar translations in the NMT network as prefixes of the decoder. The proposed method obtains higher translation accuracy results and reduces the undesirable effect observed in previous methods of copying unrelated words when performing translations.

We also proposed an extension to sent2vec that considers larger n-gram orders. It allows us to identify similar sentences (cues) that yield higher accuracy rates as measured on translation test sets.

We evaluate results on a multi-domain setting using a single model trained on a heterogeneous data set, built from multiple corpora and domains, achieving better results when compared to previous micro-adaptation approaches. In addition, we showed the suitability of our approach to gather valuable information from large monolingual corpora.

In our future work, we would like to explore alternative algorithms to compute distributed sentence representations from word embeddings, such as TF-IDF. Furthermore, we would like to consider source sentence coverage when selecting n-best similar translations. As regards distributed representations we plan to experiment with cross-lingual networks to retrieve similar translations directly from human-crafted monolingual data in order to eliminate the noise introduced by synthetic translations.

Acknowledgments

This work was granted access to the HPC resources of [TGCC/CINES/IDRIS] under the allocation 2020- [AD011011270] made by GENCI (Grand Equipement National de Calcul Intensif). We also would like to thank the anonymous reviewers for their valuable suggestions.

References

Ankur Bapna and Orhan Firat. 2019. Non-parametric adaptation for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1921–1931, Minneapolis, Minnesota. Association for Computational Linguistics.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313, New Orleans, Louisiana. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Bram Bulte and Arda Tezcan. 2019. Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1800–1809, Florence, Italy. Association for Computational Linguistics.

Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7059–7069. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT sentence embedding.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Rebecca Knowles and Philipp Koehn. 2016. Neural interactive translation prediction. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria. INCOMA Ltd.

Philipp Koehn and Jean Senellart. 2010. Convergence of Translation Memory and Statistical Machine Translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pages 21–31, Denver.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020. Flaubert: Unsupervised language model pre-training for french. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France. European Language Resources Association.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2018. One sentence one model for neural machine translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2016. Pre-translation for neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1828–1836, Osaka, Japan. The COLING 2016 Organizing Committee.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540, New Orleans, Louisiana. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Amir Rosenfeld, Mahdi Biparva, and John K. Tsotsos. 2018. Priming neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2092–209209.

Ali Sabet, Prakhar Gupta, Jean-Baptiste Cordonnier, Robert West, and Martin Jaggi. 2019. Robust cross-lingual embeddings from parallel sentences.

Sebastin Santy, Sandipan Dandapat, Monojit Choudhury, and Kalika Bali. 2019. INMT: Interactive neural machine translation prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 103–108, Hong Kong, China. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019a. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *CoRR*, abs/1907.05791.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019b. Ccmatrix: Mining billions of high-quality parallel sentences on the web. *arXiv preprint arXiv:1911.04944*.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Languages Resources Association (ELRA).

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

E Tulving, D.L. Schacter, and H.A. Stark. 1982. Priming effects in word-fragment completion are independent of recognition memory. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, 8:336–342.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Jitao Xu, Josep Crego, and Jean Senellart. 2019. Lexical micro-adaptation for neural machine translation. In *International Workshop on Spoken Language Translation*, Honk Kong, China.

Jitao Xu, Josep Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1580–1590, Online. Association for Computational Linguistics.