Noname manuscript No.
(will be inserted by the editor)

Addressing Data Sparsity for Neural Machine Translation Between Morphologically Rich Languages

Mercedes García-Martínez · Loïc Barrault · Fethi Bougares

Received: date / Accepted: date

Abstract Translating between morphologically rich languages is still challenging for actual machine translation systems. In this paper, we experiment with various Neural Machine Translation (NMT) architectures to address the data sparsity problem caused by data availability (quantity), domain shift and the languages involved (Arabic and French). We showed that the Factored NMT (FNMT) model, which uses linguistically motivated factors, is able to outperform standard NMT systems using subword units by more than 1% BLEU point even when a large quantity of data is available. Our work shows the benefits of applying linguistic factors in NMT when faced with low and large resource conditions.

Keywords Neural Machine Translation · Factored models · Deep Learning

1 Introduction

Neural Machine Translation (NMT) [1–3] has been developed very quickly in the recent years. In the last years NMT with attention mechanism [2] achieved better results than existing phrase-based systems [4] for several language pairs. The model consists of a sequence to sequence encoder-decoder which uses as context the full input sentence. Despite the advantages in NMT systems, machine translation is a

Mercedes García-Martínez
LIUM, Le Mans University
E-mail: mercedes.garcia_martinez@univ-lemans.fr

Loïc Barrault
LIUM, Le Mans University
E-mail: loic.barrault@univ-lemans.fr
Tel.: +33 243 833 833

Fethi Bougares
LIUM, Le Mans University
E-mail: fethi.bougares@univ-lemans.fr

complex task and there is still a lot of work ahead to improve it. In this paper we address the following hot topics in NMT:

*Data sparsity:*MT systems primarily rely on the bilingual training corpora which are often available in limited quantity. Moreover, bilingual corpora might not be available for some specific language pairs and domains. The translation of morphologically rich languages requires even more data, and the training set rarely contains all the inflected word forms. In actual systems, words are not linked with all their morphological variations and there is no explicit information about morphological features. These issues can lead to data sparsity.

*Limitation on the target vocabulary size:*due to the computational complexity of the output layer, the target vocabulary size should be limited. Therefore, it is not possible to generate all the words seen in the training dataset. This can lead to the generation of unknown tokens for the words that are not included in the target vocabulary.

Modelling morphological phenomena as inflections for **morphologically rich languages** requires larger vocabulary size compared to other languages. Morphological variation and lexical productivity can cause word forms unseen in training. Increasing the vocabularies partially mitigates these issues but we will face both previously mention issues: (1) data sparsity due to the difficulty of modelling rare seen or unseen inflected forms and (2) a larger target vocabulary increases the computational complexity of the output layer.

In this paper, we translate Arabic into French. Both are morphologically rich languages which do not share morphological roots nor the alphabet. This makes this language pair more difficult to translate. We compare factored NMT models (using linguistically motivated factors) to the state of the art BPE approach and the classic word-level NMT approach. In addition, we apply BPE method on the factored NMT model. We experiment with low resource (LR) conditions and compare it to a scenario with high resource (HR) conditions. Moreover, we translate test files of different sources in order to know which system behaves better in different conditions (low/high resources and same/different data sources).

The rest of this paper is organized in four sections: Section 2 explains previous related works, Section 3 describes the different modelling approaches employed in this work. In Section 4, the experiments are presented and the obtained results are shown. Section 5 presents an analysis of the data sparsity issue. Finally, Section 6 concludes the paper and open some perspectives.

2 Related work

In previous work, in order to avoid the softmax normalization over a large output layer, the batches are organised so that only a subset K of the target vocabulary is possibly generated at raining time [5]. However, the complexity remains the same at test time. In [6], the generation of unknown words issue is addressed using the alignments produced by an unsupervised aligner . The unknown generated words are substituted in a post-process step by the translation of their corresponding aligned source

word (using a bilingual dictionary) or by copying the source word if no translation is found. Another possibility to reduce the vocabulary size is to consider subword units. [7] propose the most successful approach using the Byte Pair Encoding (BPE) method. The output layer can be set to a tractable size. All source and target tokens are encoded with BPE units in order to possibly generate all target words. Some unknown and rare words unseen at training time can be generated by combining several BPE units. The vocabulary can be shared for both languages (joint or bilingual vocabulary) helping to generate, for example, proper names that are already in the source language and are invariable in the target language. As an extreme case of subwords units, some works consider character as translation unit [8–11]. Hybrid systems using mostly word-level and character-level for rare words are implemented [12, 13] finding a good balance of vocabulary between them. They never generate unknown tokens and they are easier to train than fully character-level systems. However, they do not benefit of common lexemes between words. The advantages of character-level NMT is that all the vocabulary can be covered with a small output layer size. It can model morphological variants of a word and avoid problems in preprocessing/tozenization. Moreover, unseen words can be generated similarly to using BPE. The major drawback of character-level NMT is the increase of the sequence sizes which results in longer range dependencies between units. Character-level decoders outperform subwords units using BPE method when processing unknown words [14]. By contrast, character-level systems perform worse than BPE-based systems when extracting morphosyntactic agreement and translating discontinuous units of meaning.

In order to tackle the **data sparsity** challenge, backtranslated data is incorporated into NMT [15, 16]. Monolingual data is automatic translated with a model trained in the opposite language direction creating a synthetic parallel data. This allows the system to manage a larger quantity of training data boosting the translation performance. Other work uses the WordNet [17] id and POS-tags of the words to add lexical and morphological information, respectively, with the purpose of reducing the data sparsity in a phrased-based system [18].

NMT systems often do not incorporate any additional **linguistic information**, they only rely on the raw text data. Linguistically motivated systems may help to overcome data sparsity, generalize and disambiguate to improve translation when facing the previously described NMT hot topics. When the dataset is small, the morphological information can help the translation process [19].

In the recent years, factors are used as additional information in the source language [20]. *Factors* refer to some linguistic annotations at word-level, *e.g.* the Part of Speech (POS) tag, number, gender, etc. Factors are first introduced for NMT in the target language where two symbols are simultaneously generated in [21]. This work is improved increasing the number of factors in order to translate case sensitive data [22]. The feedback of the model is changed to use tied embeddings [23] with the concatenation of the embeddings of the two output symbols. In other work, Czech and Latvian translation is also performed with this model [24, 25]. This approach which uses factors at target side consists of representing the words using their lemmas and additional factors of the words indicating how to inflect the lemmas. In morphology, a lemma is the dictionary form or headword of a set of words. For example, “are”, “were”, “was”, “being”, “is” are inflections of the same lemma, which is “be”. Two

different sequences are generated synchronously: one for the lemma and the other for the factors. In a second step, the surface form of each word is generated from its predicted lemma and factors. The advantages of using factors as translation unit are two-fold: reducing the output vocabulary size and allowing the model to generate surface forms which are never seen in the training data. Factored system can support larger vocabulary because it can generate words from the lemmas and factors vocabularies, which is an advantage when data is sparse. In standard NMT, the tokens are not linked with all their morphological variations and there is no explicit information about morphological features. By contrast, the use of lemmas directly in the NMT models allows the system to connect all the inflections of a word to the same lemmas and capture lexical correspondence. In addition, factors may help the translation process by providing grammatical information to enrich the output. Knowing the lemmas and their factors, all their inflections can be generated without explicitly seeing them in the training data. Moreover, having the part-of-speech tag can be useful to distinguish polysemic words (*e.g.* book: noun or verb). Unseen words can be also generated using subword segments produced by the BPE method. However, since they are not linguistically informed, they can produce erroneous surface forms by concatenating several incompatible subword units. Recently, another factored NMT system working as well with lemmas and linguistic factors is proposed. The two symbols are predicted interleaving them in a single output sequence with double length [26]. The strategy is applied for translating English into Czech and English into German. They argue that the presence of lemmas allows the system to model inflections and capture lexical correspondence with the source. Unseen words can be generated as well but the better results that they obtain are not mainly due to this reason. They find that the benefit comes from the words decomposition.

This paper studies how factored NMT models behave in different resource conditions and different use cases.

3 Models description

The architectures described in this paper are based on the sequence to sequence encoder-decoder NMT architecture equipped with an attention mechanism [2].

The **encoder** is a bidirectional RNN (see box number 1 of Figure 1). Each input sentence token x_i ($i \in 1 \dots N$ with N the source sequence length) is encoded into an annotation a_i by concatenating the hidden states of a forward and a backward RNN. Each annotation in $\mathbf{a} = a_1 \dots a_N$ is a representation of the whole sentence with a focus on the current token.

The **decoder** is made of a conditional gated recurrent unit (cGRU) [27] consisting of two GRUs interspersed with the attention mechanism (see box number 3 of Figure 1). The first GRU cell of the decoder (GRU_1 in Figure 1) is fed by its previous hidden state and the feedback (*i.e.* the embedding of the previous generated symbol). The second GRU (GRU_2 in Figure 1) is fed by the output of GRU_1 and the context vector $\mathbf{c}_j$. The output layer L_O is connected to the network through a sum operation inside of an hyperbolic tangent function in the hidden to output (*h2o*) layer which takes as input the embedding of the previous generated token as well as the context

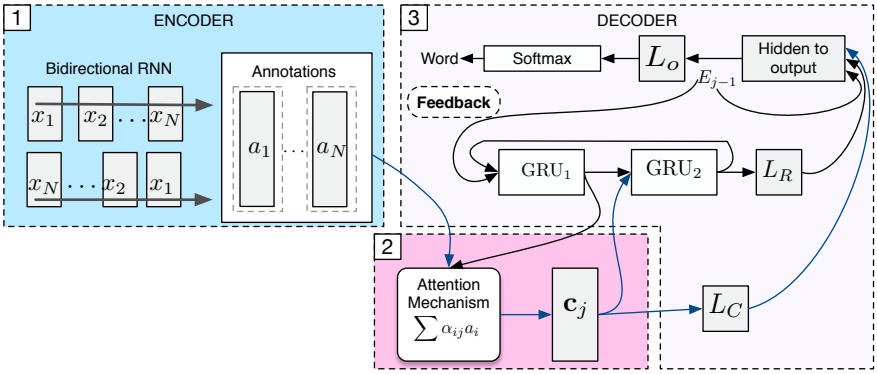

Fig. 1 NMT system with attention mechanism.

vector and the output of the decoder from GRU_2 (both adapted with a linear transformation, respectively, L_C and L_R). Finally, the output probabilities for all tokens in the target vocabulary are computed with a *softmax* function. The token with the highest probability is proposed for translation at each timestep. The encoder and the decoder are trained jointly to maximize the conditional probability of the reference translation.

The **attention mechanism** (see box number 2 of the Figure 1) computes a source context vector $\mathbf{c}_j$ as a convex combination of the annotation vectors, where the weight of each annotation is computed locally using a feed-forward network. These weights can be interpreted as the alignment score between target and source tokens. For each generated token at the target side, the model finds the relevant source context.

The incorporation of the attention mechanism allows the models to discard the unnecessary information of the source sentence. Therefore, using the attention mechanism, long sentences can be translated without remembering the whole source sentence.

We experimented as well with the Factored Neural Machine Translation (**FNMT**) [21, 24] approach. This approach uses the linguistic decomposition of the words only (no surface form) and predicts simultaneous outputs at the target side of the network. For simplicity reasons, only two symbols are simultaneously generated: the lemma and the concatenation of the different factors that are considered. Indeed, each word is represented by its lemma and its linguistic factors. We use six factors for French: POS tag, tense, gender, number, person and the case information (lowercased, uppercased or in capitals). By these means, all the derived forms of the verbs, nouns, adjectives, etc. do not need to be kept in the target vocabulary. The word output vocabulary is reduced into two vocabularies: one for lemmas and a very small vocabulary for the factors (see Equation 1).

$$|V_{words}| > |V_{lemmas}| \gg |V_{factors}| \tag{1}$$

The low frequency words in the training set can benefit from sharing the same lemma with other high frequency words, and also from sharing the factors with other

words. The vocabulary of the target language contains only lemmas and factors but the total number of surface words that can be generated (*i.e.* virtual vocabulary) is larger (see Equation 2). This allows the system to correctly generate words which are considered as unknown in word-level NMT system.

$$|V_{words}| \ll |V_{lemmas}| \times |V_{factors}| \tag{2}$$

Two types of FNMT architectures are used. Both have a second output in contrast to standard NMT system. The first one contains a single hidden to output layer (*h2o*) which is then used by two separated softmax layers (see Figure 2). This model is called **FNMT1**.

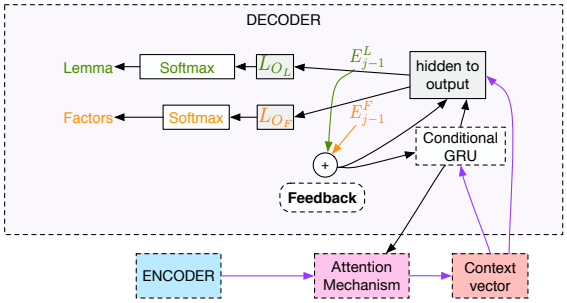

Fig. 2 Factored NMT system with a single *h2o* layer

The second system contains two separated *h2o* layers, each one specialized for a particular output (see Figure 3). This model is called **FNMT2**

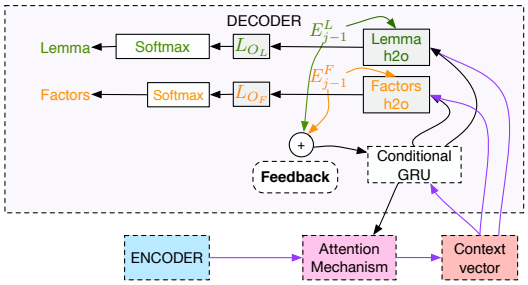

Fig. 3 Factored NMT system with separated *h2o* layers.

The encoder and attention mechanism are similar to standard NMT architecture (see Figure 1) in both FNMT architectures. However, the decoder differs in order to produce multiple outputs. The synchronicity of the two outputs generation is possible because the hidden states are shared between the two of them. The hidden to output layer is a hyperbolic tangent function of the sum of three inputs: (1) hidden

state, (2) source context provided by the encoder and the attention mechanism and (3) feedback. Both FNMT systems are similar excepting that FNMT1 (single *h2o* layer) uses the concatenation of the previously generated lemma and factors embeddings, and FNMT2 (separated *h2o*) uses one *h2o* layer for each output receiving only the embedding of the symbol that is generating. FNMT2 model have more specialized weights for the lemma and factors outputs. The feedback to the hidden state consists of the concatenation of the lemmas and factors embeddings. Finally, in the last part of the model, the output is split into two specialized output layers L_{OL} and L_{OF} which in turn feed a specialized softmax layer, one to calculate the lemmas and the other to calculate the factors.

The decoder of the FNMT architecture may lead to sequences with different length since lemmas and factors are simultaneously generated but in separated outputs. Indeed, each sequence of symbols ends when the end-of-sequence (<eos>) symbol is generated and nothing prevents the lemma generator to output the <eos> symbol before or after the factors generator. To avoid this scenario, the length of the factors sequence is constricted to be equal to the length of the lemma sequence. This implies to ignore the <eos> symbol for factors (to avoid shorter factors sequence) and stop the generation of factors when the lemma sequence has ended (to avoid longer factors sequence). This is motivated by the fact that the lemmas are closer to the final objective (a sequence of **words**) and they are the symbols carrying most of the meaning.

In order to extract the lemmas and factors, a linguistic tool is necessary. The morphological and grammatical analysis is performed with the MACAON toolkit [28]. MACAON POS-tagger outputs the lemma and factors for each word taking into account its context. The Lefff dataset [29], a large-coverage morphological and syntactic lexicon for French, is used by MACAON to build the models.

Once the factored representation outputs are obtained from the neural network, the post-process to fall back to the surface form is performed. This step is not trivial. For that purpose, a lookup table is built to match the lemmas and factors as keys with the surface forms as values. This knowledge is also extracted from the MACAON tool for French language, which given a lemma and some factors, provides the word candidate.

For the sake of simplicity, the first proposition is taken for the very few cases when there are several proposals of surface forms for the same pair of lemma and factors. In French, in most of the cases when several words are proposed for the same pair of lemma and factors, all the proposals are correct and their choice only depends on the situation. For example, for written or spoken versions or formal or informal situations. In other cases (*e.g.* name entities) where the surface form corresponding to the lemma and factors is not found, the system outputs the lemma itself.

Table 1 presents the different model approaches used in this work describing the outputs of each of them.

We interleaved the lemmas and factors in a single sequence, as done in [26], this model is named **IFNMT1**. Additionally, we introduced a new representation called **IFNMT2** where the first output predicts lemmas and factors as IFNMT1 but we add a new second output generating also factors at the same positions of the factors in the first output. When lemmas are generated in the first output, the second output

Model	*h2o* layer	output 1st	output 2nd
NMT	-	word	-
BPE	-	wo+ rd	-
FNMT1	single	lemma	factors
FNMT2	sep.	lemma	factors
IFNMT1	single	lemma *factors*	-
IFNMT2	single	lemma *factors*	null *factors*
FBPE	single	lem+ *ma*	factors *factors*

Table 1 Model approaches at target side.

generates $null$. IFNMT2 pretends to better model the factors having the advantages of the two factorized architectures: (1) as FNMT model, the model learns specialized embeddings only for factors in the second output and (2) as IFNMT1 model, factors are also included jointly with the lemmas in the embeddings of the first output and factors output receives as feedback its corresponding lemma generated in the previous timestep which can help the generation of factors. For FNMT systems, BPE is applied on the lemma sequence and the corresponding factors are repeated when a split occurs. We call this system Factored BPE (**FBPE**).

Standard NMT architecture generating only one output (see Figure 1) has been used for the word-level NMT model, BPE model and IFNMT1 model. The FNMT architecture containing a single *h2o* layer (see Figure 2) is used for FNMT1, IFNMT2 and FBPE models. Lastly, the FNMT architecture with separated *h2o* layers (see Figure 3) is used for FNMT2 model.

4 Experiments

In the experimental framework, we use Arabic in order to translate from a morphologically rich language. The target language is French which is a moderately inflected language.

Training details

For the training of the models, we used `NMTpy` toolkit [30], a Python toolkit based on Theano [31] and available as open-source software[1]. The feedback embeddings (input to the decoder) and the output embeddings are tied [23] to enforce learning a single target representation and decrease the number of total parameters. In order to avoid exploding gradients, we clipped the norm of the gradient to be no more than 1 [32]. The optimizer used is Adadelta [33] with an initial learning rate of 1. We use Xavier [34] weights initialization.

The WEB test set has been used as development set in order to apply early stopping, validating from the 2nd epoch every 1k updates with a patience of 10. The same vocabulary size has been used for Arabic at input side and French at output side

[1] `https://github.com/lium-lst/nmtpy`

for words and lemmas, which is 30k. All the factors vocabulary is covered by the network.

For BPE method, we applied the formula $30k - \#characters$ to obtain the number of BPE units comparable with the vocabulary of the rest of the systems. The vocabulary size is equal to the size of the initial vocabulary (number of characters) plus the number of merge operations (BPE units) as mentioned in work [7]. The joint vocabulary sharing the BPE tokens for source and target language is not beneficial when the languages use different alphabets. Therefore, we have not trained a joint BPE model. Note that we could have been used a method to unify the alphabets using transliteration in order to avoid the problem. On the other hand, we think that French and Arabic are languages that do not share the same roots and the benefit is harder to glimpse. The same procedure is applied for FBPE model, source words and target lemmas are segmented in subwords. Factors are repeated for each subword to synchronize lemmas and factors sequences.

Test sets

We evaluate the models with three test files in order to compare different use cases. These test sets are provided with multiple references to better evaluate with automatic metrics such as BLEU [35]. Table 2 provides information about the different test sets used to evaluate the models.

Test set	#Sentences	#Tokens (AR/FR)	#Unique words (AR/FR)	#References
WEB*	409	10k/∼18k	4.2k/3.7k	4
TEXT	352	10k/∼18k	4.1k/3.6k	2
BROADCAST (2h)	466	14k/∼23k	4.7k/4.1k	4

Table 2 Test sets for Arabic to French translation. Information about number of sentences and references in second and last columns, respectively. Number of tokens and number of unique words for each language are shown in the third and forth columns. *WEB test set has been used for development purposes. French unique words and number of tokens are average numbers of the references.

4.1 Low resource conditions

The first experiment consists of translating under LR conditions using a small training dataset. The hyperparameters chosen for this experiment have been adapted to the small size of the dataset. Therefore, we used reduced dimensions for the layers: 512 for the recurrent layers and 300 for the embeddings.

Data and preprocessing

The datasets used for training are News Commentary version 9 and 80 hours of broadcast news. Arabic data has been tokenized in Arabic Tree Bank (ATB) scheme with

the morphological analyser tool MADA [36,37] separating prefixes and suffixes from stems. French data is tokenized with Moses and the morphological analysis is performed by MACAON. After filtering sentences longer than 100 tokens, the training dataset only contains 150k sentences. Table 3 shows the full vocabulary and number of words for Arabic and French languages. The number of words is small, 4.6M for Arabic and 4.7M for French. However, the full vocabulary is still large which is a challenge in machine translation, all the unique words in the training vocabulary are 72k for Arabic ATB tokens separating stems and affixes in source side, 73k for French words and 43k for French lemmas in target side. The vocabulary size of French factors is 282. The factored model can possible generates 148k words from the 30k size lemma vocabulary and the factors vocabulary.

	AR	**FR**
#Unique words	72k	73k
#Tokens	4.6M	4.7M

Table 3 Number of unique words and tokens in the training dataset for Arabic to French translation under LR conditions.

Results

The results for the Arabic to French translation under LR conditions are presented in Table 4.

Model	**WEB***	**TEXT**	**BROADCAST**
NMT	13.52	10.15	19.05
BPE	14.49	9.40	18.27
FNMT1	**16.99**	12.27	**25.93**
FNMT2	14.60	11.06	24.07
IFNMT1	15.25	11.81	**26.00**
IFNMT2	15.89	**12.90**	24.06
FBPE	**17.04**	10.39	23.63

Table 4 Results for Arabic to French translation under LR conditions. Scores are measured in BLEU. *WEB test file is used for development set.

We observe that factored models (last 5 models in Table 4) obtained the highest values for all the test sets. This means that factored models are a good option for LR conditions.

BPE compared to NMT and FBPE compared to FNMT perform well only with the development test (WEB) but not translating the other test files (TEXT and BROADCAST). The development set is used together with the training dataset to build the BPE units dictionary, as a consequence, BPE models can easier translate it. Moreover, it seems that under LR conditions, BPE models do not have enough samples to

well create the BPE dictionary. This means that models using BPE units (BPE and FBPE) do not generalize well and FNMT is more robust under LR conditions.

The separated *h2o* layers for FNMT2 model seems not to help in LR conditions, the higher number of parameters compared to FNMT1 is not necessary to learn a small dataset.

We see that FNMT1 performs better than interleaved models (IFNMT1 and IFNMT2) when translating WEB and similar to IFNMT1 with BROADCAST. This fact suggests that FNMT architecture, where lemmas and factors are separated in two outputs, can benefit in some use cases. IFNMT2 model obtained the best score for TEXT and better score than IFNMT1 for WEB. The addition of the 2nd output modelling only factors helps the translation in some use cases.

4.2 High resource conditions

In this set of experiments, we used the same language pair (Arabic→French) increasing the training dataset in order to observe the behaviour of the systems when resource conditions change. For the sake of simplicity, interleaved models (IFNMT1 and IFNMT2) are not included in this set of experiments. The training options are the same except that the size of the recurrent layer is increased to 1024 dimensions and the size of the embedding layer to 512 dimensions.

Data and preprocessing

The dataset added to the previous presented data (news-commentary and broadcast news) is the United Nations (UN) corpus which is out-of-domain. Adding UN corpus, the training dataset has a total of 14M of sentences, which is almost 100 times bigger than the previous training dataset. The full vocabulary is large, all the unique words in the training vocabulary are 881k for Arabic ATB tokenization separating stems and affixes in source side, 674k for French words and 511k for French lemmas in target side. Table 5 presents the full vocabulary size and number of words in the training dataset.

	AR	**FR**
#Unique words	881k	674k
#Tokens	315M	350M

Table 5 Number of unique words and tokens in the training dataset for Arabic to French translation under HR conditions.

The preprocessing of the data was performed similarly to previous experiment. The vocabulary size remains 30k. Factors French vocabulary size is 388. From the 30k lemmas and the 388 factors, the total vocabulary that factored model can generate is 137k. For BPE systems, we use 30k BPE units not using joint vocabularies. BPE method is applied as well for FBPE systems.

Results

Table 6 presents the results of adding UN corpus.

Model	WEB*	TEXT	BROADCAST
NMT	29.33	20.86	35.77
BPE	28.32	20.15	32.47
FNMT1	28.99	18.36	34.26
FNMT2	27.00	18.74	33.70
FBPE	**29.53**	**21.05**	**36.98**

Table 6 Results for Arabic to French translation under HR conditions. Scores are measured in BLEU. *WEB has been used for development file.

The results show that FBPE system obtains the best performance for all the test sets: WEB, TEXT and BROADCAST.

BPE system does not perform well again probably because joint vocabularies option is not used (the vocabularies are separated for source and target due to the different scripts of the languages). On the other hand, FBPE is benefiting from the BPE units to handle the increase of training data.

FNMT systems without BPE units obtain low scores confirming the hypothesis that they perform better when they are applied in LR conditions. FNMT2 system performs better than FNMT1 system when translating TEXT. This confirms that FNMT2 model, which includes more parameters, can be better option when the training dataset is big. On the other hand, translations of WEB and BROADCAST test sets still obtained better scores by FNMT1 system than FNMT2 system.

FNMT1 improves over BPE showing the benefits of using factors in some use cases (WEB and BROADCAST).

5 Analysis of data sparsity

We computed the coverage of the models comparing word and factored level models. Results for training dataset under LR and HR conditions are shown in Table 7. We measured the expressivity of the model dividing the number of covered unique words by the total number of unique words (vocabulary used). We also measured the number of covered tokens by the total number of tokens in order to know the percentage

	LR conditions		HR conditions	
Coverage	**FR**$_{word}$	**FR**$_{factored}$	**FR**$_{word}$	**FR**$_{factored}$
Unique words	40.89%	83.50%	4.45%	20.14%
Tokens	98.66%	99.74%	98.28%	99.31%

Table 7 Comparison of the training datasets in terms of unique words coverage (number of covered unique words / total number of unique words) and tokens coverage (number of covered tokens / total number of tokens).

Test set	Coverage	LR conditions		HR conditions	
		FR$_{word}$	**FR**$_{factored}$	**FR**$_{word}$	**FR**$_{factored}$
WEB	Unique words	86.55%	95.01%	84.44%	94.56%
	Tokens	96.13%	98.21%	95.43%	98.39%
TEXT	Unique words	90.07%	96.08%	87.09%	95.59%
	Tokens	97.02%	98.60%	96.19%	98.68%
BROADCAST	Unique words	86.90%	93.98%	84.41%	93.02%
	Tokens	96.63%	98.35%	95.85%	98.10%

Table 8 Comparison of the test sets in terms of unique words coverage and tokens coverage. This is the average of multiple references.

of tokens that are covered by each model. Finally, the third measure is the average frequency of a token where we divide the number of covered tokens by the total number of unique words at word level or unique lemmas at factored level. With this last measure, we know how well a token is represented in the corpus by counting its average frequency.

Results in Table 7 show that factored model can cover more vocabulary in both LR and HR conditions. We can see that for LR conditions, factored representation can model twice the number of words (83.50%) than word representation model (40.89%). But this 40% increase only represents 1.1% of the total number of tokens ($\sim$47k tokens). For HR conditions, factored representation can model four times more words (20.14%) than word representation model (4.45%), still representing $\sim$1% (3.5M tokens).

We measured in the same way the coverage in each test file (see Table 8). We observe that the unique words coverage for LR conditions is greater than for HR conditions. This is the case at the word and the factored level. This can be due to the fact that the 30k tokens selected for LR conditions are extracted from in-domain data, despite that the dataset is small. Training data in HR conditions contains the UN dataset which is big but out-of-domain. Consequently, there is a decrease of the tokens coverage at word level. But we can see that this is not the case for factored level (except a very little drop for the BROADCAST test file). This tells us that factored representation makes the model more robust to domain shift.

Figure 4 compares NMT, BPE, FNMT1 and FBPE systems in terms of BLEU for LR and HR conditions.

We observe that in HR conditions, in spite of covering slightly less unique words, the model better learns the token representations because of their higher frequency. Consequently, the results in terms of BLEU are better in HR conditions than in LR conditions.

Comparing the systems, we see better results for factored models (FNMT and FBPE) than NMT and BPE under LR conditions due to the data sparsity. We showed that factored models better behave under LR conditions facing data sparsity issues. For HR conditions, FNMT obtains lower BLEU than NMT. An explanation is that on the one hand, since the training dataset is huge, the NMT model has enough examples to perform well. On the other hand, the more complex architecture of the FNMT model does not benefit from that, resulting in lower scores. FBPE system using also

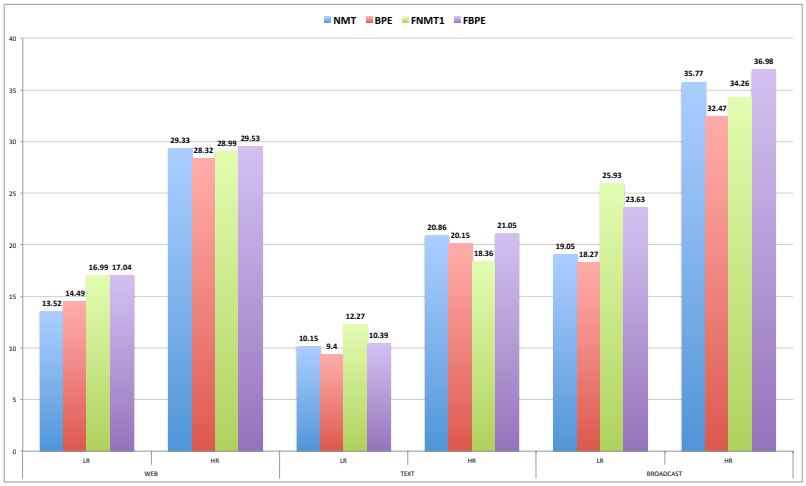

Fig. 4 Comparison of LR and HR conditions in terms of BLEU by NMT, BPE, FNMT1 and FBPE systems for all the use cases.

BPE units performs the best showing the advantages of combining factored and BPE models.

To explain this, we computed the BLEU scores at factors level for FNMT and FBPE systems (see Table 9). FBPE repeats the factors for each BPE unit, which seems to help for better factors modelling. We can see that the behaviour changes completely when faced with LR or HR conditions. Using BPE units leads to process longer sequences, which makes it difficult to model the long-distance dependencies. This is essentially true when the model is trained with a small amount of data, as we can see in Table 9 (LR conditions).

Model	**WEB** (dev.)		**TEXT**		**BROADCAST**	
Cond.	LR	HR	LR	HR	LR	HR
FNMT1	27.80	34.80	25.04	28.72	31.24	36.94
FBPE	27.87	36.59	21.35	31.87	30.05	37.91

Table 9 Results for Arabic to French translation in terms of BLEU at factors level for LR and HR conditions.

For HR conditions, repeating the factors multiple times result in a better modelling, leading to performances similar or better than word level NMT.

6 Conclusion

In this paper, we have compared various NMT models at word-level, BPE-level and factored-level including linguistics to decompose the target words. We compared several ways of using linguistic factors in an NMT system (FNMT with single and sep-

arated *h2o* layers, interleaved FNMT and Factored BPE). Arabic to French translation has been carried out in two different conditions, using a small or large training dataset. The systems have been evaluated with different test sets of different domains in order to compare the behaviour of the systems. The analysis of the vocabulary coverage showed that factored-level NMT is more robust to domain shift than other approaches.

We have demonstrated that factored NMT models applied in low resource conditions obtain better results than the rest of the models. By combining factors and subword units (BPE), the systems are able to achieve best performance when trained with a large training corpus, surpassing the other NMT systems by more than 1.2% BLEU.

For future work, instead of generating all the factors in the same sequence, the architecture can be extended to produce each factor, independently, in different sequences. This would solve the current limitation that the systems would only model factors combinations which are seen in the training set. If we build the factors vocabulary with each factor separately, the generalization power of the model will be increased. In addition, more types of factors can be included without being necessarily linguistically motivated like the domain.

Finally, FNMT approach can be explored for other tasks where several related sequences are required. For example, PoS tagging jointly with spoken language understanding tasks. Additionally, multimodal or multilingual machine translation models can be extended with the factored approach adding linguistic information to help the generalization performance.

Acknowledgements This work was partially funded by the French National Research Agency (ANR) through the CHIST-ERA M2CR project, under the contract number ANR-15-CHR2-0006-01.

References

1. K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, CoRR **abs/1406.1078** (2014)
2. D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, CoRR **abs/1409.0473** (2014)
3. I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, CoRR **abs/1409.3215** (2014)
4. 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, E. Herbst, in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions* (Association for Computational Linguistics, Stroudsburg, PA, USA, 2007), ACL '07, pp. 177–180
5. S. Jean, K. Cho, R. Memisevic, Y. Bengio, On using very large target vocabulary for neural machine translation, CoRR **abs/1412.2007** (2014)
6. T. Luong, I. Sutskever, Q.V. Le, O. Vinyals, W. Zaremba, Addressing the rare word problem in neural machine translation, CoRR **abs/1410.8206** (2014). URL `http://arxiv.org/abs/1410.8206`
7. R. Sennrich, B. Haddow, A. Birch, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Association for Computational Linguistics, 2016), pp. 1715–1725
8. J. Chung, K. Cho, Y. Bengio, A character-level decoder without explicit segmentation for neural machine translation, CoRR **abs/1603.06147** (2016)

9. W. Ling, I. Trancoso, C. Dyer, A.W. Black, Character-based neural machine translation, CoRR **abs/1511.04586** (2015)
10. M.R. Costa-Jussà, J.A.R. Fonollosa, Character-based neural machine translation, CoRR **abs/1603.00810** (2016)
11. J. Lee, K. Cho, T. Hofmann, Fully character-level neural machine translation without explicit segmentation, CoRR **abs/1610.03017** (2016). URL http://arxiv.org/abs/1610.03017
12. M.T. Luong, D.C. Manning, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2016), pp. 1054–1063
13. Y. Wu, G.R. Team, Google's neural machine translation system: Bridging the gap between human and machine translation, CoRR **abs/1609.08144** (2016). URL http://arxiv.org/abs/1609.08144
14. R. Sennrich, How grammatical is character-level neural machine translation? assessing mt quality with contrastive translation pairs, CoRR **abs/1612.04629** (2016)
15. R. Sennrich, B. Haddow, A. Birch, Edinburgh neural machine translation systems for WMT 16, CoRR **abs/1606.02891** (2016). URL http://arxiv.org/abs/1606.02891
16. R. Sennrich, B. Haddow, A. Birch, in *Proc. ACL* (2016)
17. C. Fellbaum (ed.), *WordNet: an electronic lexical database* (MIT Press, 1998)
18. K. Singla, K. Sachdeva, D. Yadav, S. Bangalore, D.M. Sharma, in *Procedings of the Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation* (2014)
19. J. Niehues, E. Cho, in *Proceedings of the Second Conference on Machine Translation, Volume 1: Research Papers* (Association for Computational Linguistics, Copenhagen, Denmark, 2017), pp. 80–89. URL http://www.aclweb.org/anthology/W17-4708
20. R. Sennrich, B. Haddow, Linguistic input features improve neural machine translation, CoRR **abs/1606.02892** (2016)
21. M. García-Martínez, L. Barrault, F. Bougares, in *Proceedings of the International Workshop on Spoken Language Translation* (Seattle, USA, 2016), IWSLT'16. URL http://workshop2016.iwslt.org/downloads/IWSLT_2016_paper_2.pdf
22. M. García-Martínez, L. Barrault, F. Bougares, in *Statistical Language and Speech Processing*, ed. by N. Camelin, Y. Estève, C. Martín-Vide (Springer International Publishing, Cham, 2017), pp. 21–31
23. O. Press, L. Wolf, Using the output embedding to improve language models, CoRR **abs/1608.05859** (2016). URL http://arxiv.org/abs/1608.05859
24. F. Burlot, M. García-Martínez, L. Barrault, F. Bougares, F. Yvon, in *Proceedings of the Second Conference on Machine Translation, Volume 1: Research Paper* (Association for Computational Linguistics, Copenhagen, Denmark, 2017), pp. 20–31. URL http://www.aclweb.org/anthology/W17-4703
25. M. García-Martínez, O. Caglayan, W. Aransa, A. Bardet, F. Bougares, L. Barrault, in *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers* (Association for Computational Linguistics, Copenhagen, Denmark, 2017), pp. 288–295. URL http://www.aclweb.org/anthology/W17-4726
26. A. Tamchyna, M.W.D. Marco, A. Fraser, in *Proceedings of the Second Conference on Machine Translation (WMT)* (Copenhagen, Denmark, 2017). URL http://arxiv.org/abs/1707.06012
27. O. Firat, K. Cho. Conditional gated recurrent unit with attention mechanism. github.com/nyu-dl/dl4mt-tutorial/blob/master/docs/cgru.pdf (2016)
28. A. Nasr, F. Béchet, J.F. Rey, B. Favre, J.L. Roux, in *Proceedings of the ACL-HLT 2011 System Demonstrations* (2011), pp. 86–91
29. B. Sagot, in *7th international conference on Language Resources and Evaluation (LREC 2010)* (Valletta, Malta, 2010). URL https://hal.inria.fr/inria-00521242
30. O. Caglayan, M. García-Martínez, A. Bardet, W. Aransa, F. Bougares, L. Barrault, Nmtpy: A flexible toolkit for advanced neural machine translation systems, arXiv preprint arXiv:1706.00457 (2017). URL http://arxiv.org/abs/1706.00457
31. R. Al-Rfou, M. development team, Theano: A python framework for fast computation of mathematical expressions, CoRR **abs/1605.02688** (2016). URL http://arxiv.org/abs/1605.02688
32. R. Pascanu, T. Mikolov, Y. Bengio, Understanding the exploding gradient problem, CoRR **abs/1211.5063** (2012)
33. M.D. Zeiler, ADADELTA: an adaptive learning rate method, CoRR **abs/1212.5701** (2012)
34. X. Glorot, Y. Bengio, in *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics* (2010)
35. K. Papineni, S. Roukos, T. Ward, W.J. Zhu, in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (Stroudsburg, PA, USA, 2002), ACL '02, pp. 311–318

36. N. Habash, R. Roth, O. Rambow, R. Esk, N. Tomeh, in *In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT* (2013)
37. A. Pasha, M. Elbadrashiny, M. Diab, A. Elkholy, R. Eskandar, N. Habash, M. Pooleery, O. Rambow, R. Roth, in *Proceedings of the 9th International Conference on Language Resources and Evaluation* (2014), pp. 1094–1101