Character vs Subword-level models in Neural Machine Translation

Giulio Zhou
School of Informatics
University of Edinburgh
s1775060@ed.ac.uk

Abstract

Neural Machine Translation (NMT) quickly raised the attention of both academia and industry, obtaining state-of-the-art results despite being introduced only recently. One of the strengths of NMT models is the freedom in the choice of the representation of the sentences in input and output. Although words were used in early NMT models (word-level models), smaller units, like characters or subword (e.g. lexemes and affixes), have become more popular as representation methods due to the limits of word-level models in handling out-of-vocabulary issues.

This paper describes the benefits and problems of the various representation methods, focusing on how different architectures have been designed and the quantitative/qualitative analysis of the NMT translations in relation to the sequence representation method.

1 Introduction

Neural Machine Translation (NMT) is an end-to-end approach to Machine Translation (MT) which developed rapidly as the predominant paradigm in this field. Since the first full neural translation model (Sutskever et al., 2014), impressive improvements have been made. NMT methodologies have outperformed the more established Statistical Machine Translation (SMT) approaches such as the Phrase-Based Machine Translation (PBMT) (Bojar et al., 2016).

Nevertheless, NMT has not overcome several problems (Koehn and Knowles, 2017) one of which is translating rare words like nonce words (terms coined for a single use), morphological variants (e.g. inflections) or really infrequent words. Words in Natural Language Processing (NLP) are generally described by their context. As a consequence, NLP systems such as NMT models cannot process correctly rare words due to the lack of information about such words. To mitigate this problem, a common solution is to split the words into smaller units like characters or subwords (e.g. lemmas and affixes) and use these as input/output of the NMT model (character/subword-level model).

Subword-level models, obtained by applying an adapted version of the Byte Pair Encoding (BPE) algorithm for words (Sennrich et al., 2015), hold the state-of-art results for the majority of the shared translation tasks (Bojar et al., 2017). These models, compared to character-level models, have better performances, and are much more efficient in terms of computational cost and memory usage. Character-level models, instead, are demonstrated to be more robust in translating sentences with highly infrequent words (Chung et al., 2016; Lee et al., 2016; Luong and Manning, 2016). Closing the gap between character and subword-level models for NMT is an important research goal. However, not many attempts have been made to improve character-level models.

This paper explores the benefits and strengths of character and subword-level NMT models by analysing the reasons to prefer one approach to the other, state-of-the-art models and the studies that have been made.

Section 2 provides backgrounds on Neural Machine Translation, in particular, it provides an overview of the core components of NMT architectures. Section 3 describes the problems related to the use of word-level models, how they are solved by character and subword-level models and current challenges. Section 4 presents the main features of recent NMT models architectures. Sec-

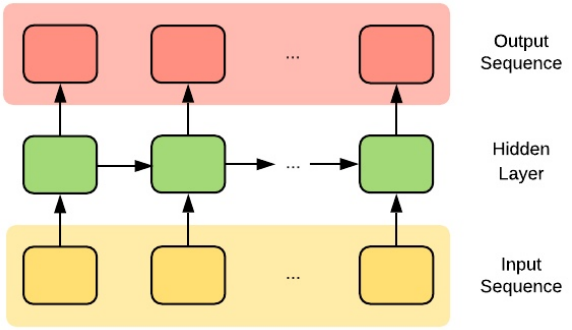

Figure 1: Recurrent Neural Network (RNN) with one hidden layer (shallow RNN).

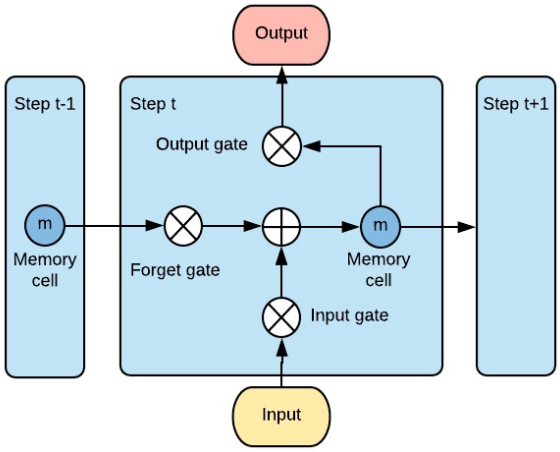

Figure 2: Long Short-Term Memory (LSTM) cell. The cell contains a memory cell m which is updated at each time step t combining the value stored in the previous time step and the input. The hidden state (or output) is obtained with an activation function which takes as input the filtered memory cell.

tion 5 shows the results obtained by the models described in Section 4 in a shared translation task and discusses the discoveries, problems and possible areas of study of character-level models.

2 Neural Machine Translation

Neural Machine Translation (NMT) is a group of automatic translation approaches that use neural networks as core components. Although different architectures have been proposed (e.g. Kalchbrenner and Blunsom (2013) and Vaswani et al. (2017)), the most largely used so far is the Recurrent Neural Network Encoder-Decoder architecture (Sutskever et al., 2014) with attention mechanism (Bahdanau et al., 2014).

2.1 Recurrent Neural Networks

A Recurrent Neural Network (RNN) is a more general feedforward neural network which takes as input a sequence of vectors $\mathbf{x} = (x_1, \ldots, x_T)$ and returns a sequence of the same length $\mathbf{y} = (y_1, \ldots, y_T)$. In a standard RNN, or shallow RNN (Figure 1), at each time step t the RNN computes the output of the hidden layer considering both the input x_t and the output of the hidden layer at the previous time step:

$$h_t = f(h_{t-1}, x_t) \tag{1}$$

where f is an activation function (e.g. sigmoid, tanh, etc...).

The main advantage of this approach is that each output vector y_i is conditioned not only by the vector x_i but also by all the previous ones. This is particularly evident when the input is a sentence. For instance, in a language model, to predict a word, the contexts of all the preceding words are considered without having to use traditional backoff methods.

2.2 RNN Encoder-Decoder

In Machine Translation (MT), it is not always the case that a sentence and its translation have the same length, thus standard RNN cannot be applied directly.

An RNN architecture that solves this problem is the RNN Encoder-Decoder firstly adopted by Sutskever et al. (2014) and Cho et al. (2014). This model is composed by two RNN: the *encoder* and the *decoder*. The idea is to encode a variable-length sequence into a fixed-sized vector to then decode into to a variable-length sequence. Given a sequence $\mathbf{x}$, the objective is to obtain a sequence $\mathbf{y}$ that maximises

$$p(\mathbf{y}|\mathbf{x}) \propto \prod_{t=1}^{T'} p(y_t|c, y_1, \ldots, y_{t-1}) \tag{2}$$

where c is the context generated by the encoder and T' the length of the target sequence, which can be different from the length of the source sequence.

2.3 RNN cell variants

Short-distance dependencies can be handled by RNNs by using simple activation functions in equation (1). While it is possible that the next output is strictly related to the closer input in terms of time steps, there are no mechanisms to handle long-distance dependencies. This is due to the

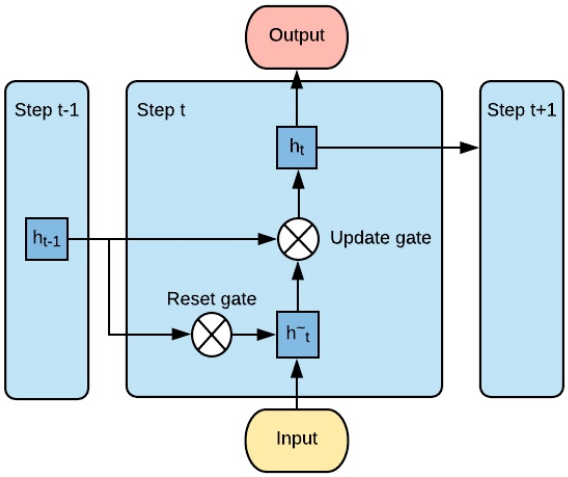

Figure 3: Gated Recurrent Unit (GRU). A new hidden state $\widetilde{h}_t$ is computed by combining the input with the previous hidden state filtered by the reset gate. The update gate decides whether update the hidden state h_t with the newly computed $\widetilde{h}_t$ or not.

fact that at each time step t, the hidden unit is used for both storing the history of the sequence $(h_1, \ldots, h_{t-1})$ and to predict the next output y_t.

The Long Short-Term memory (LSTM) architecture was proposed in Hochreiter and Schmidhuber (1997) to separate these two tasks by substituting function f in equation (1) with a block unit (cell) containing an explicit memory state. An LSTM cell (Koehn, 2009) (Figure 2) is composed by a memory state m and three *gates* that perform read,write and reset operations on m, respectively the output, input and forget gates. The memory state is obtained by

$$m_t = gate_{input} \times x_t + gate_{forget} \times m_{t-1} \quad (3)$$

while the output of the cell is

$$h_t = f(gate_{output} \times m_t) \quad (4)$$

where f is an activation function.

To reduce the number of parameters, thus the computational complexity, Cho et al. (2014) proposed a simplification of the LSTM cell: gated recurrent units (GRU). A GRU cell (Figure 3) doesn't have a separate memory cell, and it is composed only by two gates instead of three: the update and the reset.

$$update_t = g(W_{update} x_t + U_{update} h_{t-1}) \quad (5)$$

$$reset_t = g(W_{reset} x_t + U_{reset} h_{t-1}) \quad (6)$$

where g is an activation function, while W and U weight matrices. The hidden state is computed by

$$h_t = update_t h_{t-1} + (1 - update_t)\widetilde{h}_t \quad (7)$$

where

$$\widetilde{h}_t = \phi(W x_t + U(reset_t \circ h_{t-1})) \quad (8)$$

where ϕ is an activation function.

2.4 Attention mechanism

Attention in neural machine translation can be considered as an alignment mechanism in traditional statistic machine translation (although they are not the same thing (Koehn and Knowles, 2017)). Attention mechanisms are used in the decoding process to capture relevant parts of the input sentence in order to generate the next word in the translation (Ghader and Monz, 2017).

The pioneers of attention-based models are Bahdanau et al. (2014). The model proposed is an extension of the one in Sutskever et al. (2014) (Figure 4). The encoder is a bi-directional RNN that encodes the input into a state vector $\mathbf{h} = (h_1, \ldots, h_m)$ where each state h_i is composed by the state obtained by scanning the input sequence backward and forward $h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}]$. The decoder predicts, at each translation step t, the target word y_t based on the hidden state of the decoder s_t, the predicted words y_i with $i < t$ and a context vector c_t also called attention vector. The attention vector is the weighted sum of the state encoder vector

$$c_t = \sum_j a_{tj} h_j \quad (9)$$

$$a_{tj} = \frac{\exp(att(s_t, h_j))}{\sum_k \exp(att(s_t, h_k))} \quad (10)$$

where $att(s_t, h_j)$ is a function that calculates an alignment score between the hidden states (Britz et al., 2017).

Further analysis and development on attention-based models are made in Luong et al. (2015) where they classify these mechanisms as *global* and *local*. The former consider all the hidden states of the encoder when computing the attention vector c_i, the latter focus on a subset of source words per target word.

3 Source and target sequences

RNN Encoder-Decoder architectures for NMT as described in Section 2 are considered sequence-to-sequence models due to the fact that both input

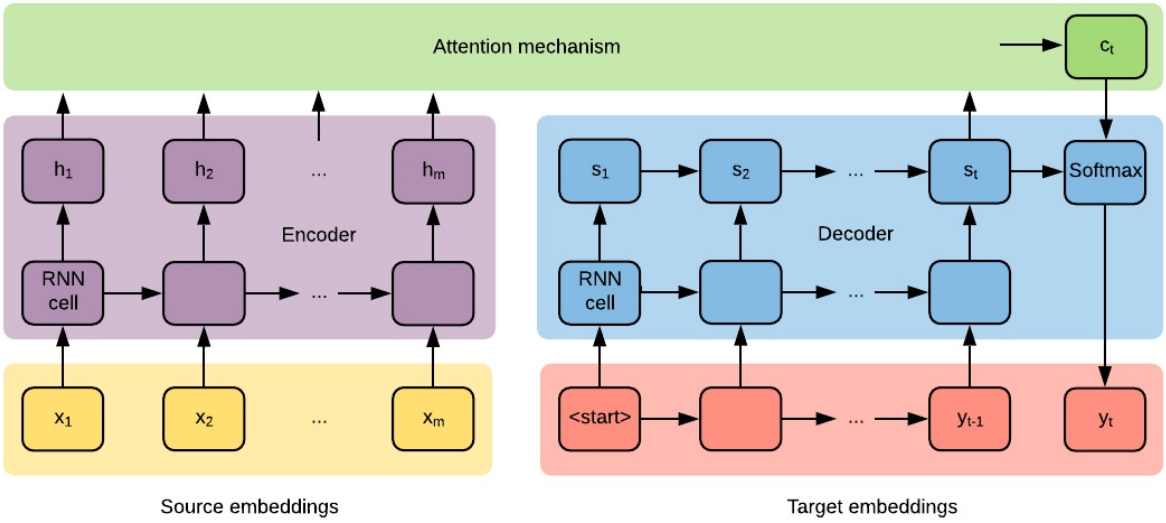

Figure 4: Sequente-to-sequence Encoder-Decoder model for NMT with attention mechanism. The encoder is not bidirectional as described in Bahdanau et al. (2014) to make the image more clear. At time step t the attention mechanism computes a context vector c_t from the encoder hidden states $h_1, \ldots, h_m$ and the decoder state s_t. c_t and s_t are then used to predict the target embedding y_t

and output are sequences. For MT tasks, it was natural to address such sequences as sentences and its elements as words. It is common for NMT to encode each element of the sequence (unit) with a one-hot vector which allows an equal distance between the units. The NMT models are independent from what the one-hot vectors represent. As a consequence, there are many degrees of freedom in the choice of the units' type (words, characters, or any other symbol encodings).

3.1 Character-level translation

Chung et al. (2016) and Lee et al. (2016) provide extensive analysis and discussions about the benefits and challenges of using character-level translation models.

Benefits

The main advantage of character-level translation models is the capability of dealing with out-of-vocabulary issues. Rare words are well-known problems for many NLP tasks and they affect NMT even more. Due to the high computational complexity of attention-based Encoder-Decoder models, it is a common practice to use only a restricted vocabulary composed by a number of high-frequency words, usually 30 000 - 50 000 (Sennrich et al., 2015), and replacing the remaining words with the token "UNK".

Furthermore, character-level models can handle words' morphological variants in an efficient way. It may be the case that a very frequent lexeme has infrequent morphological variants. As previously said, it is hard to translate infrequent words. While word-level models represent each variant as an independent vector, the character-level ones are able to identify shared properties between these variants without having to manually implement linguistics knowledge permitting the model to discover internal structures of the sentences by itself, as well obtaining better translations for sentences containing rare variations.

Challenges

Compared to word-level sequences, character-level sequences are obviously much longer. This raises two issues (Lee et al., 2016).

First of all, the computational cost rises dramatically. Since the character-level softmax is considerably faster than the word-level softmax, a slow down for longer sequences is affordable. However, the attention mechanism's computational cost grows quadratically with respect to the input sequence making naive character-level models prohibitive in terms of translation time. Secondly, it is harder to model long-distance dependencies even when using memory cells like LSTM.

Moreover, building a character-level encoder is not a trivial task, in fact, it has to learn highly non-linear functions to map a long sequence of charac-

ters to the meaning of the words.

3.2 Subword and word-level translation

The popularity of word-level translation model derives from intuitional and technical reasons. As previously stated, it is natural to address a sentence as composed by words, where each word is a unit of meaning (Chung et al., 2016). This means that each word has a specific and unique meaning. For instance, although the meaning of a lexeme modified by morphological processes remains close to the original meaning, the semantic of the obtained word differs from the lexeme's one (e.g. "run" and"runner"). Similarly, words that are similar in their representation may have completely unrelated meanings (e.g. "quit" and "quite"). On the other hand, representing a sentence as a sequence of words has technical advantages as well. Word-level models suffer considerably less of data sparsity compared to character-level models and they have fewer problems in modelling long-distance dependencies, as discussed in Section 3.1.

Another possible solution to the rare words problem, instead of using characters to represent sentences, is to decompose those words into smaller units, subwords, through a process of word segmentation. Recently, the Byte Pair Encoding (BPE) data compression techniques (Gage, 1994) adapted to word segmentation (Sennrich et al., 2015) has become the most popular method. Although current state-of-art models for different translation tasks are based on the BPE segmentation algorithm (Bojar et al., 2017), it is far away from being a perfect segmentation algorithm (Chung et al., 2016). Nevertheless, subword-level models provide a reasonable balance between the advantages and disadvantages of word and character-level models.

4 Models

Over the past few years, more and more organisations focused their research on NMT (Bojar et al., 2015; Bojar et al., 2016; Bojar et al., 2017). While most of the models attempted to achieve open vocabulary through some sort of segmentation algorithm, not many advances have been made with character-level models.

This section describes the main characteristics of the most successful character and subword-level models. Details about the hyperparameters settings and the training processes are not covered in this paper.

4.1 Chung et al. (2016)

The objectives of Chung et al. (2016) were to implement a character-level decoder, and to analyse possible limitations and improvements of such approach. In order to achieve character-level translation, they have introduced a new RNN architecture called *bi-scale* RNN. This RNN is composed by hidden layers that operate with the same amount of gated unit, but at different *timescale* (a *faster layer* and a *slower layer*). At each time step t, the faster layer computes the activation given the output of both the faster and the slower layer at time step $t-1$. The main difference between these two layers is that the slower layer has as input the output of the faster layer at time step t, which means that the slower updates only when the faster has finished computing its input. This architecture was designed to capture the timescale differences between processing words and characters. For comparison they have also made tests with an existing RNN decoder which they call *base* decoder. While the decoder operates at character-level, the source sentence is processed and with BPE.

4.2 Lee et al. (2016)

While Chung et al. (2016) have achieved character-level translation only on target side, Lee et al. (2016) proposed a fully character-level model which does not rely on any segmentation algorithm. Chung et al. (2016) pointed out that the main problem of a character-level encoder is its extreme inefficiency. The solution proposed in Lee et al. (2016) is to preprocess the input sentences' embeddings to reduce drastically their dimensionality. The input embedding passes through a single convolutional filter with fixed width, a pooling layer to segment the output of the convolutional layer into segments and finally, a highway layer (Srivastava et al., 2015) to improve the quality of the character-level model. The decoder used was a standard two-layer character-level decoder.

4.3 Luong and Manning (2016)

Luong and Manning (2016) proposed a hybrid word-character RNN architecture in order to take advantage of both word and character-level models' benefits. The base concept of this hybrid model is to translate at word-level while dealing with "UNK" words at character-level. To

deal with such words, two deep LSTM model are trained over characters. The first learns from the source words and replace the "UNK" embedding with the embedding provided by such character-level model. The second is called when the word-level decoder produces a "UNK" word so that it can be converted into a sequence of characters. In addition to the hybrid model, they have developed a naive character-based model.

4.4 Gulcehre et al. (2017)

Gulcehre et al. (2017) claim that although the natural language is generated word-by-word, it is not conceived sequentially. For this reason, they extended the model proposed by Chung et al. (2016) with a planning mechanism which substitutes the classic attention method. The main difference is that the planning mechanism is a predictive model which plans future alignments and decides whether to follow it or not. Hence, it computes an alignment plan matrix with k rows, which stores the current alignment and the alignments for the $k-1$ next time steps, and a commitment plan vector which first element, when discretised, tells the planning mechanism to update the alignment matrix or to shift it.

A more efficient model is to avoid the use of a plan matrix, instead, the model reuses the same alignment until the commitment switch activates. The alignment vector is learned through an implicit unsupervised planning mechanism. This approach reduces both computational cost and memory usage.

4.5 Sennrich et al. (2016)

Sennrich et al. (2016) presented the model that obtained the best result for the shared translation task at WMT 16 (Bojar et al., 2016). The model uses the BPE segmentation described in Sennrich et al. (2015) for both source and target sequences, and as architecture an enhanced model of the attention-based Bahdanau et al. (2014) later called Nematus (Sennrich et al., 2017b). Although Nematus differs from the base Bahdanau et al. (2014) in various implementation aspects like a different initialisation for the decoder hidden state, activation functions and word embeddings, the main difference is the introduction of depth in the decoder using conditional GRU blocks with attention. Each GRU layer is composed of two GRU state transition blocks and an attention mechanism between them. The first blocks process the output of the previous timestep with the encoder context, the seconds receive as input the context vector computed by the attention mechanism.

This model has been developed further for the submission for the WMT 17 shared translation task (Sennrich et al., 2017a) with the addition of depth in the encoder, an improved segmentation algorithm, layer normalisation and a better memory usage.

4.6 Huck et al. (2017)

The model proposed by Huck et al. (2017) represent the current state-of-art for En-De news translation (Bojar et al., 2017). They have used Nematus (Sennrich et al., 2017b) as base architecture. The major contribution of Huck et al. (2017) is a linguistically-informed segmentation technique. While on source-side the segmentation algorithm remains BPE, on target-side three splitting techniques are sequentially applied: a German morphological suffix splitter, a compound splitter and BPE to reduce the vocabulary size.

4.7 Wu et al. (2016)

While the scientific community has discordant opinions about the suitability of NMT models in real-world applications (Junczys-Dowmunt et al., 2016; Farajian et al., 2017), Google has presented and deployed[1] its neural translation system (GNMT) (Wu et al., 2016). Although there is no public implementation of such model, the architecture description does not differ drastically from Bahdanau et al. (2014). The RNN encoder and decoder are composed of 8 layers, both with a deeply stacked architecture. The second layer of the encoder has a reversed direction (like the bi-directional encoder in (Bahdanau et al., 2014)). Another difference is the use of *residual connections* to reduce the computational cost. In practice, the input of the layer is summed with the output of the layer itself.

This model has been tested with different sentences representation such as characters, words and subwords obtained by using a segmentation algorithm developed by Google called wordpiece model (WMP) (Schuster and Nakajima, 2012).

[1] https://blog.google/products/translate/found-translation-more-accurate-fluent-sentences-google-translate/

Table 1: BLEU scores on WMT newstest English→German obtained from the respective papers

Model			2014	2015	2016	2017
Chung et al. (2016)	base	bpe2char	21.3	23.5	-	-
Chung et al. (2016)	bi-scale	bpe2char	21.3	23.1	-	-
Lee et al. (2016)	base	char2char	19.7	22.6	26.2	-
Gulcehre et al. (2017)	PAG	char2char	21.9	22.8	-	-
Gulcehre et al. (2017)	rPAG	char2char	21.8	22.7	-	-
Huck et al. (2017)	base	bpe2bpe[2]	-	22.4	26.8	27.1
Huck et al. (2017)	fine-tuned+RL	bpe2bpe	-	**28.6**	33.4	27.1
Sennrich et al. (2017b)	base	bpe2bpe	20.1	23.2	26.7	-
Sennrich et al. (2016)	ensemble+RL	bpe2bpe	25.4	28.1	34.2	-
Sennrich et al. (2017a)	ensemble+RL	bpe2bpe	-	-	**36.2**	**28.3**
Wu et al. (2016)	base	char2char	22.6	-	-	-
Wu et al. (2016)	base	wpm2wpm	24.6	-	-	-
Wu et al. (2016)	ensemble+RL	wpm2wpm	**26.3**	-	-	-

5 Results and Discussion

5.1 Task and Dataset

In this paper, the models described in Section 4 are directly compared by analysing the results obtained in translating news from English to German[3].

The choice of the task was driven by two motivations: first, translating into more morphologically reach languages such as German is a challenging task for machine translation (Sennrich, 2016); second, as previously said, not many character-level translation systems have been developed, so a common shared task was an obvious choice for a better comparison.

The dataset used by the models for training, validation and testing are provided by the Conference[4] on Machine Translation (WMT). Although the training and validation set did not change considerably over the years, the testing sets differ each time with a number of sentences varying between 2100 and 3200 for the English-German language pair.

5.2 BLEU score

BLEU (Papineni et al., 2002) is an automatic evaluation method for MT systems which has become the *de facto* standard metric for such models.

Table 1 shows the BLEU scores obtained by character, subword and mixed models described in Section 4. The results are directly obtained from the original papers, however, for the fully character-level model Lee et al. (2016), the experiments are run by Sennrich (2016) since the original paper does not report the results for translating English to German. The score for this model may also be lower compared to the other scores because it has been evaluated with a case-sensitive method.

Comparing these results, different considerations can be made. First of all, they show clearly that fine-tuning the hyperparameters, enhancing the training data and using ensemble models with RL reranking (Liu et al., 2016) improve drastically the performance of the NMT model, from 1.7 (Wu et al., 2016) up to 7.6 (Sennrich et al., 2016) BLEU score points difference from the base model.

Secondly, not considering the tuned and ensemble models, the scores obtained by character-level models do not perform much worse than the subword-level ones, in some cases, their scores are even higher. In fact, all the analysed character-level models' performances can be compared to the actual state-of-art base model (Sennrich et al., 2017b).

Lastly, more complex architecture does not always result in a better model. In Chung et al. (2016) the proposed bi-scale RNN performed worse than more basic RNN Encoder-Decoder. While (Gulcehre et al., 2017) managed to drastically simplify their first model without lowering down much their BLEU score.

5.3 Human evaluation

Despite the fact that the limitations of automatic evaluation methods are well known to the scien-

[3]Luong and Manning (2016) will not be included in this direct comparison because they did not attempt this task

[4]Previously Workshop

tific community (Callison-Burch, 2009), it is not rare to find models evaluated only with BLEU score. This is due to the conception that human evaluation methods are expensive and time-consuming, while an automatic method like BLEU score is immediate and provides a fair approximation of human judgements. Lee et al. (2016) pointed out that "BLEU encourage reference-like translations and do not fully capture true translation quality", for this reason, it is still not possible to determine the quality of a translation (its adequacy or fluency) trough automatic evaluation methods. In addition, the experiments made by Wu et al. (2016) have shown that improvements in the BLEU score are not always reflected in the human evaluation.

A mismatch between the BLEU score and a human assessment is shown in the translation shared task WMT 17 (Bojar et al., 2017) English→German. As shown in Table 1, the model that obtained the highest BLEU score is Sennrich et al. (2017a). However, the Direct Assessment (DA) made by human assessors in Bojar et al. (2017) ranked the adequacy of the translation made by Huck et al. (2017) superior to Sennrich et al. (2017a).

About character-level models, the human assessment made by Lee et al. (2016) has shown that the fully character-level model is more robust than (Chung et al., 2016) in four different scenarios such as spelling mistakes, rare words, morphology and nonce words. While it is still not possible to determine firmly if a char2char model is better (or worse) than other models, it is interesting to notice that despite the fact that char2char models generate the target sequence character by character, the resulting sequence remains a long and coherent sentence.

5.4 Grammatical quality

Recently, Sennrich (2016) has conducted an extensive evaluation of three different models: Lee et al. (2016) (char2char), Chung et al. (2016) (bpe2bpe) and Sennrich et al. (2017b) (bpe2bpe). The objective of this evaluation is to determine the ability of bpe-based model to handle unseen names and to assess the grammatical quality of character-generated translations and how the length of the target sequence influence it.

To do so, it is necessary to capture how well a model handles linguistic phenomena. Sennrich (2016) propose a new evaluation method which consists in comparing the probability of generating two sentences: the reference sentence and a contrastive sentence (a sentence with a specific translation error). For this purpose, it has been created a test set containing 97 000 contrastive translation pairs making possible to compute the accuracy of the models with five linguistic phenomena: noun phrase agreement, subject-verb agreement, separable verb particle, polarity and transliteration.

The results obtained with such method has shown that character-decoder models outperform in terms of generalisation to unseen words. However, BPE-decoder generates more grammatically correct sentences, especially when long-distance dependencies are involved.

5.5 Further work

Since the introduction of a fully neural network-based translation system, more and more organisation shifted their research into this field. This is reflected in the number of NMT models submitted for the WMT shared translation task over the past few years. However, among these models only a few are character-based.

Despite the fact that character-level models have important benefits and strengths as discussed in the previous sections, not much attention has been put on these models, although several improvements and studies can be made.

- **Computational cost**: the main factor that prevents researchers using character-level models is the prohibitive computational cost needed for training such models. An extreme example is the naive char2char model developed by Luong and Manning (2016) which took three months to be trained. Although an attempt to improve the training process of char2char model has been made (Zhao and Zhang, 2016), the resulting architecture was overly complicated.

- **Deep architecture**: almost the totality of character-level models are based on shallow RNN with at most two layers. It is known that deeper neural networks are able to capture more complex structures, while in NMT depth improves the translation quality resulting in an increase in both BLEU scores and cross-entropy (Barone et al., 2017).

- **Ensemble**: recent ensemble subword-level models outperform drastically the performance of single models. While different settings have been tried with subword-level models (e.g. layer normalisation, RL reranking etc...), more complex settings have to be made for character-level models.

- **Long-distance dependencies**: due to the length of a sequence of characters, long-distance dependencies are not well captured by char2char models. Although Gulcehre et al. (2017) did not evaluate their method in relation to long-distance dependencies, planning future alignments could possibly improve the quality of such translations. Another challenge could be improving the memorisation ability of gated units without drastically increment the number of parameters.

- **Morphology**: Huck et al. (2017) have shown that the introduction of little morphological information improves the quality of the translations. Character-level models have been proven to be able to capture more morphological information by itself than other types of model. However, it is believed that learning jointly translation and morphology can result in a better model (Belinkov et al., 2017).

6 Conclusion

Due to the high computational cost of NMT models, their vocabularies are strictly limited compared to SMT models. This caused severe problems with highly infrequent words and rare morphological variants. To mitigate this problem three different approaches have been developed: divide the sentences into subwords (word segmentation), consider the sentence as a sequence of characters or use a mixed model (a word-level model that handle the problematic words at character-level).

Although subwords do not completely solve this problem due to non-optimal segmentation methods, they have become the most popular sentence representation method. While character-level models outperform subword-level models when translating infrequent words and morphological variants, it cannot handle too long-distance dependencies and their extremely high computational cost made them unappealing for practical uses.

While different attempts have been made to improve translations with long-distance dependencies and the efficiency of character-level models, many margins of improvements are left to enhance overall performances of character-level models.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Antonio Valerio Miceli Barone, Jindřich Helcl, Rico Sennrich, Barry Haddow, and Alexandra Birch. 2017. Deep architectures for neural machine translation. *arXiv preprint arXiv:1707.07631*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? *arXiv preprint arXiv:1704.03471*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal, September. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 131–198.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, et al. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214.

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

Chris Callison-Burch. 2009. Fast, cheap, and creative: evaluating translation quality using amazon's mechanical turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 286–295. Association for Computational Linguistics.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. *arXiv preprint arXiv:1603.06147*.

M Amin Farajian, Marco Turchi, Matteo Negri, Nicola Bertoldi, and Marcello Federico. 2017. Neural vs. phrase-based machine translation in a multi-domain scenario. *EACL 2017*, page 280.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Hamidreza Ghader and Christof Monz. 2017. What does attention in neural machine translation pay attention to? *arXiv preprint arXiv:1710.03348*.

Caglar Gulcehre, Francis Dutil, Adam Trischler, and Yoshua Bengio. 2017. Plan, attend, generate: Character-level neural machine translation with planning. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 228–234.

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

Matthias Huck, Fabienne Braune, and Alexander Fraser. 2017. Lmu munich's neural machine translation systems for news articles and health information texts. In *Proceedings of the Second Conference on Machine Translation*, pages 315–322.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is neural machine translation ready for deployment? a case study on 30 translation directions. *arXiv preprint arXiv:1610.01108*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP*, volume 3, page 413.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2016. Fully character-level neural machine translation without explicit segmentation. *arXiv preprint arXiv:1610.03017*.

Lemao Liu, Masao Utiyama, Andrew M Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *HLT-NAACL*, pages 411–416.

Minh-Thang Luong and Christopher D Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. *arXiv preprint arXiv:1604.00788*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 5149–5152. IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh Neural Machine Translation Systems for WMT 16. In *Proc. of the First Conference on Machine Translation (WMT16)*, Berlin, Germany.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017a. The University of Edinburgh's Neural MT Systems for WMT17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, Copenhagen, Denmark.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, et al. 2017b. Nematus: a toolkit for neural machine translation. *arXiv preprint arXiv:1703.04357*.

Rico Sennrich. 2016. How grammatical is character-level neural machine translation? assessing mt quality with contrastive translation pairs. *arXiv preprint arXiv:1612.04629*.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training very deep networks. In *Advances in neural information processing systems*, pages 2377–2385.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Shenjian Zhao and Zhihua Zhang. 2016. An efficient character-level neural machine translation. *arXiv preprint arXiv:1608.04738*.