Noisy UGC Translation at the Character Level: Revisiting Open-Vocabulary Capabilities and Robustness of Char-Based Models

José Carlos Rosales Núñez[1,2,4] **Guillaume Wisniewski**[3] **Djamé Seddah**[4]
[1] CNRS, LISN [2] Université Paris-Saclay
[3] Université de Paris, LLF, CNRS [4] INRIA Paris
jose.rosales@limsi.fr guillaume.wisniewski@univ-paris-diderot.fr
djame.seddah@inria.fr

Abstract

This work explores the capacities of character-based Neural Machine Translation to translate noisy User-Generated Content (UGC) with a strong focus on exploring the limits of such approaches to handle productive UGC phenomena, which almost by definition, cannot be seen at training time. Within a strict zero-shot scenario, we first study the detrimental impact on translation performance of various user-generated content phenomena on a small annotated dataset we developed, and then show that such models are indeed incapable of handling unknown letters, which leads to catastrophic translation failure once such characters are encountered. We further confirm this behavior with a simple, yet insightful, copy task experiment and highlight the importance of reducing the vocabulary size hyper-parameter to increase the robustness of character-based models for machine translation.

s

1 Introduction

Neural Machine Translation (NMT) models fall far short from being able to translate noisy User-Generated Content (UGC): the quality of their translation is often even worse than that of a traditional phrase-based system (Khayrallah and Koehn, 2018; Rosales Núñez et al., 2019). In addition to ambiguous grammatical constructs and profusion of ellipsis, the main difficulty encountered when translating UGC is the high number of out-of-vocabulary tokens (OOVs) resulting from misspelled words, emoticons, hashtags, mentions, and all specific constructs used in online forums and social medias (Foster, 2010; Seddah et al., 2012; Eisenstein, 2013; Sanguinetti et al., 2020). Some of those phenomena can be perceived as noise while the others are typical markers of language variation among speakers. Moreover, a certain amount of those same phenomena operate at the lexical level (either at the character, subword or word levels) (Sanguinetti et al., 2020). This is why, focusing more on the noise axis, char-based models appear to offer a natural solution to this problem (Luong and Manning, 2016; Ling et al., 2015): indeed these open vocabulary models are designed specifically to address the OOV problem.

In this work, we explore the ability of out-of-the-box character-based NMT models (Lee et al., 2017) to address the challenges raised by UGC translation. While character-based models may seem promising for such task, to the best of our knowledge, they have only been tested either on data sets in which noise has been artificially added through sampling an edited word error data set (Belinkov and Bisk, 2018; Ebrahimi et al., 2018a) and on canonical data set, in which they prove to be very effective for translating morphologically-rich languages with a high number of OOVs (Luong and Manning, 2016).

However, our starting-points experiments show that character-based systems are outperformed by BPE models even when translating noisy UGC. To understand this counter-intuitive result, we conduct several experiments and analyses. In particular, we manually annotated 400 sentences at the token level using a fine-grained typology, to perform our analyses. These sentences correspond to the worst and the best translated utterances of two MT systems (a char-based and a transformer-based model). Our results highlight the extreme sensibility of character-based models to the vocabulary size, a parameter often overlooked in the literature. Using a simple set of experiments, we thus show that these models are unable to perform an *easy* copy task due to their poor handling of unknown and rare characters. By adjusting the vocabulary size parameter, we drastically improve the robustness of our character-based model without causing a large drop in in-domain performance.

Our contributions are as follows:

- we provide an annotated data set[1] that enables in-depth evaluations of the impact of UGC idiosyncrasies;
- we demonstrate that char-based neural machine translation models are extremely sensitive to unknown and rare characters on both synthetic data and noisy user-generated content;
- we show how an overlooked hyper-parameter drastically improve char-based MT models robustness to natural noise while maintaining the in-domain level of performance.

2 Character-level NMT

In this work, we explore two character-based translation models. The first model we consider, `charCNN` is a classic encoder-decoder in which the encoder uses character-based embeddings in combination with convolutional and highway layers to replace the standard look-up based word representations. The model considers, as input, a stream of words (i.e. it assumes the input has been tokenized beforehand) and tries to learn a word representation that is more robust to noise by unveiling regularities at the character level. This architecture was initially proposed by Kim et al. (2016) for language modeling; Costa-jussà and Fonollosa (2016) shows how it can be used in an NMT system and report improvements up to 3 BLEU (Papineni et al., 2002) points when translating from a morphologically-rich language, German, to English.

The second model we consider does not rely on an explicit segmentation into words: Lee et al. (2017) introduce the `char2char` model that directly maps a source characters sequence to a target characters sequence without any segmentation thanks to a character-level convolutional network with max-pooling at the encoder. It can be considered as an open-vocabulary model: it can generate any word made of any of the N most frequent characters of the train set (where N is a model hyper-parameter) and only outputs an `<UNK>` token in the presence of character that is not in this (char-) vocabulary. Lee et al. (2017) show that this model outperforms subword-level (i.e. BPE-based) translation models on two WMT'15 tasks (de-en and cs-en) and gives comparable performance on two tasks (fi-en and ru-en). Lee et al. (2017) additionally report that in a multilingual setting, the character-level encoder significantly outperforms the subword-level encoder on all the language pairs.

These two models have been originally tested on WMT or IWSLT tasks that consider texts that mostly qualify as canonical with very few spelling or grammatical errors. The impact of noise on `charCNN` and `char2char` has been evaluated by Belinkov and Bisk (2018) and Ebrahimi et al. (2018b). By adding noise to canonical texts (the `TEDTalk` dataset). Their results show that the different character levels models fail to translate even moderately noisy texts when trained on 'clean' data and that it is necessary to train a model on noisy data to make it robust. Note that, as explained in Section 3, there is no UGC parallel corpus large enough to train a NMT model and we must rely on the models' ability to learn, from canonical data only, noise- and error-resistant representations of their input that are robust to the noise and errors found in UGCs. This is why, in this work, we are interested in studying MT performance in a zero-shot scenario when translating noisy UGC.

To the best of our knowledge, no work has studied the performance of the character-based NMT architectures on an actual UGC scenario with real-world gathered and fine-grained annotated noisy sentences. This sets the main motivation for the present work.

3 Datasets

Training sets Due to the lack of a large parallel corpus of noisy sentences, we train our systems with 'standard' parallel datasets, namely the corpora used in the `WMT` campaign (Bojar et al., 2016) and the `OpenSubtitles` corpus (Lison et al., 2018). The former contains canonical texts (2.2M sentences), while the latter (9.2M sentences) is made of written dialogues from popular tv series. The reason for using `OpenSubtitles` is the assumed greatest similarity, compared to `WMT`, between its content and the UGC conversational nature that predominates most social media. However, UGC differs significantly from subtitles in many aspects: in UGC, emotions are often denoted with repetitions, there are many typographical and spelling errors, and sentences may contain emojis that can even replace some words (e.g. ♥ can stands for the verb 'love' in sentences such as 'I ♥ you'). UGC productivity limits the pertinence of domain adaptation methods such as fine-tuning, as there will always be new forms that will not have been

[1] `https://github.com/josecar25/char_based_NMT-noisy_UGC`

seen during training (Martínez Alonso et al., 2016). This is why we focus here on zero-shot scenarios, as we believe they provide a clearer experimental protocol when it comes to study the impact of UGC specificities on MT models.

Test sets We consider in our experiments two French-English test sets made of user-generated content. These corpora differ in the domain of their contents, their collection date, and in the way sentences were filtered to ensure they are sufficiently different from canonical data.

The first one, the MTNT corpus (Michel and Neubig, 2018), is a multilingual dataset that contains French sentences collected on `Reddit` and translated into English by professional translators. The second one, the Parallel French Social Media Bank (`PFSMB`)[2], introduced in (Rosales Núñez et al., 2019) and made from a subpart of the French Social Media Bank (Seddah et al., 2012), consists of comments extracted from Facebook, Twitter and Doctissimo, a health-related French forum. Table 9 shows some examples of source sentences and reference translations extracted from these two corpora and illustrates the peculiarities of UGC and difficulties of translating them. User-generated contents raise many challenges for MT systems: they notably contain `char-OOVs`, that is to say characters that have not been seen in the training data (e.g. emojis), rare character sequences (e.g. inconsistent casing or usernames) as well as many `word-OOVs` denoting URL, mentions, hashtags or, more generally, named entities. See (Foster, 2010; Seddah et al., 2012; Eisenstein, 2013; Sanguinetti et al., 2020) for a complete account of UGC idiosyncrasies. We have also selected a pair of blind UGC test sets, corresponding to non-overlapping partitions of the `PFSMB` and `M`TNT datasets, respectively. These were chosen to validate our approaches and models, after performing error analysis and benchmarks with the UGC test sets described above.

Additionally, we also evaluate our NMT systems on a pair of non-noisy, edited, datasets, which are chosen to match both of the train data configuration mentioned above, namely `NeswTest` and `OpenSubTest`, for `WMT` and `OpenSubs` corpora, respectively. These test sets serve a double purpose: evaluate the performance impact due to domain drift and the in-domain performance for both `WMT` and `OpenSubs`. See Tables 1 & 2 for relevant statistics.

It is worth noticing how the noisy UGC corpus, i.e. `PFSMB` and `M`TNT, have remarkably high TTR, high KL-divergence, %OOV rate and perplexity, even when compared to the out-of-domain test set in Table 2 (`NeswTest`).

4 NMT Models

Character-based models In our experiments, we compare the two character-level NMT systems introduced in Section 2, namely `charCNN` and `char2char`. Both models were trained as out-of-the box systems using the implementations provided by Kim et al. (2016) for `charCNN`,[3] and by Lee et al. (2017) for `char2char`.[4]

It must be noted that the `charCNN` extracts character n-grams for each input word and predicts a word contained in the target vocabulary or a special token, `<UNK>`, otherwise, whereas the `char2char` is capable of open-vocabulary translation and does not generate `<UNK>` tokens, unless an out-of-vocabulary character (`char-OOV`) is present in the input.

BPE-based models We use as our baselines two standard NMT models that consider tokenized sentences as input.

The first one is a `seq2seq` bi-LSTM architecture with global attention decoding. The `seq2seq` model was trained using the XNMT toolkit (Neubig et al., 2018).[5] It consists of a 2-layered Bi-LSTM layers encoder and a 2-layered Bi-LSTM decoder. It considers, as input, word embeddings of size 512 and each LSTM units has 1,024 components.

Our second baseline model is a vanilla Transformer model (Vaswani et al., 2017) using the implementation proposed in the OpenNMT framework (Klein et al., 2018). It consists of 6 layers with word embeddings that are 512-dimensional, a feed-forward layers made of 2,048 units and 8 self-attention heads.

Unknown Token Replacement One of the peculiarities of our UGC datasets is that they contain a many OOVs denoting URL, mentions, hashtags, or more generally named entities: for instance several sentences of the `PFSMB` mention the game "Flappy

[2] `https://gitlab.inria.fr/seddah/parallel-french-social-mediabank`

[3] `https://github.com/harvardnlp/seq2seq-attn`

[4] `https://github.com/nyu-dl/dl4mt-c2c`

[5] We decided to use XNMT, instead of OpenNMT in our experiments in order to compare our results to the ones of Michel and Neubig (2018).

Corpus	Split	#sentences	#tokens	avg. sent.len	TTR	Vocab. size	#chars types
WMT	Train	2.2M	64.2M	29.7	0.20	1.037M	335
	NewsDiscussDev	1,500	24,978	16.65	0.22	7,692	
OpenSubs	Train	9.2M	57.7M	6.73	0.18	2.08M	428
PFSMB †	Test	777	13,680	17.6	0.32	3,585	116
MTNT†		1,022	20,169	19.7	0.34	3,467	122
OpenSubsTest (2018)		11,000	66,148	6.01	0.23	15,063	111
NewsTest (2014)		3,003	68,155	22.70	0.23	15,736	111

Table 1: Statistics on the FR side of the corpora used in our experiments. TTR stands for Type-to-Token Ratio. Corpus containing mainly UGC are indicated with a †.

↓ Metric / Test set →	PFSMB †	MTNT †	NeswTest	OpenSubTest
3-gram KL-Div	1.563	0.471	0.406	0.006
%OOVs	12.63	6.78	3.81	0.76
PPL	599.48	318.24	288.83	62.06

Table 2: Domain-related measure on the source side (FR), between Test sets and OpenSubs training set. Dags indicate UGC corpora.

Bird" or the TV show "Teen Wolf". Most of the time, these OOVs are exactly the same in the source and target sentences and consequently, the source and reference share a lot of common words: the BLEU score between the sources and references of the PFSMB being of 15.1,[6] while it is only 2.7 on the WMT test set.

Closed-vocabulary MT systems that are not able to copy OOVs from the source sentence are therefore penalized. That is why, as part of our translation pipeline, we introduce a post-processing step in which the translation hypothesis is aligned with the source sentence and the <UNK> tokens replaced by their corresponding aligned source tokens. For the seq2seq the alignments between the source and translation hypothesis are computed using an IBM2 model.[7]. For the charCNN model, the alignments are deduced from the attention matrix. The char2char model is an open-vocabulary system that is able to generate new words when necessary. The vanilla Transformer implementation we use is able to copy unknown symbols directly.

5 Experiments and Results

We train and evaluate the MT models presented in the previous section using the data train and test configurations in Section 3. We first compare performances for our clean test sets (whether in-domain or our-of-domain, accordingly) and noisy UGC when translated using BPE- and character-based models before exploring into detail our results.

Table 3 reports the BLEU scores, as calculated by Post (2018)'s SacreBleu of the different models we consider, both on canonical and non-canonical test sets. Contrary to the first results of Michel and Neubig (2018), the quality of UGC translation does not appear to be so bad: the drop in performance observed on non-canonical corpora is of the same order of magnitude as the drop observed when translation models are applied to out-of-domain data. For instance, the BLEU score of a Transformer model trained on OpenSubtitles has the same order of magnitude on PFSMB, MTNT and news: on all these datasets, the performance dropped by roughly 4 BLEU points compared to its performance on in-domain data. However, this improvement in translation quality is partly due to our <UNK> replacement strategy, a step that was not considered by Michel and Neubig (2018).

As expected, all models perform better when they are trained on the OpenSubs corpus than when they are trained on WMT, as the former is intuitively more similar to UGC data than WMT is. More surprisingly, it appears that character-based models are largely outperformed by BPE-based models for most train-test combinations and, therefore, that their capacity to learn word representations that are robust to noise can be questioned.

Another interesting observation is that, while the Transformer achieves the best results on all test sets when trained on OpenSubtitles, it is out-

[6] The sources and references of the MTNT corpus are less similar (BLEU score: 5) since all user mentions and URL are removed in the distributed version of the corpus.

[7] The alignments are computed by concatenating the test set to the training set and using FastAlign (Dyer et al., 2013).

	WMT				OpenSubtitles			
	PFSMB	MTNT	News†	Open	PFSMB	MTNT	News	Open†
BPE-based models								
seq2seq	9.9	21.8	27.5	14.7	17.1	27.2	19.6	28.2
+ <UNK> rep.	**17.1**	**24.0**	**29.1**	**16.4**	26.1	28.5	24.5	28.2
Transformer								
+ <UNK> rep.	15.4	21.2	27.4	**16.4**	**27.5**	**28.3**	**26.7**	**31.4**
Character-based models								
charCNN	6.2	12.7	17.2	9.2	13.3	16.3	10.1	21.7
+ <UNK> rep.	16.1	18.2	22.1	11.5	18.6	20.2	14.6	23.9
char2char	7.1	13.9	18.1	8.8	23.8	25.7	17.8	26.3

Table 3: BLEU scores for our models for the different train-test combinations. In-domain test sets are marked with a dag. 'Crap', 'MTNT', 'News' and 'Open' stand, respectively, for the PFSMB, MTNT, WMT and OpenSubtitles test sets. WMT and OpenSubtitles are the training corpora, described in Section 3

performed by seq2seq when the WMT training set is considered. The situation is similar for character-based models. This observation suggests that these models do not capture the same kind of information and do not have the same generalization capacities, as they roughly have the same number of parameters (69M parameters for the char2char and Transformer models, 65M for the charCNN and 49M for the seq2seq model).

Error analysis In order to find which kind of UGC specificities are the most difficult to translate and can explain the difference in performance between character-based and BPE-based systems, we have conducted a contrastive analysis between the predictions of the Transformer and the char2char models. For each model, we have selected the 100 source sentences with the highest translation quality and the 100 ones with the lowest translation quality.[8] We have manually annotated these 400 sentences, using the typology described in Table 4, to identify which UGC specificities were the hardest to translate. Such a classification is adopted from the fined-grained list of UGC particularities of Sanguinetti et al. (2020). Examples of annotations are given in Table 10, in the Appendix.

Figure 1 shows the number of UGC specificities in the 100 worst and 100 best translations of the two considered models. As expected, there are far fewer specificities in the best translations than in the worst translations except for the high number of named entities that is found in both cases. This is, however, not a surprise given the high number of named entities in UGC (see §4).

code	kind of specificities
1	Letter deletion/addition
2	Missing diacritics
3	Phonetic writing
4	Tokenisation error
5	Wrong verb tense
6	#; @, URL
7	Wrong gender/grammatical number
8	Inconsistent casing
9	Emoji
10	Named Entity
11	Contraction
12	Graphemic/punctuation stretching

Table 4: Typology of UGC specificities used in our manual annotation.

The specificities that are the most problematic to translate[9] appear, as can be seen in Figure 1.a, to be the presence of named-entities (10th category) and the inconsistent casing (8th category) often corresponding to several words written fully uppercased to denote emotions or excitement. Counterintuitively, these two kinds of noise have more impact on the char2char model than on the Transformer model, even if it could be expected that the character embeddings learned by the former would not be sensitive to the case. When manually inspecting its prediction, it appears that, often, the char2char model is not able to copy named entities and replace some of their characters. For instance, the entity *Cllaude468* is translated into *Claudea64666*, and *flappy bird* into *flappe bird*. Interestingly this second entity is correctly translated by the char2char model when it is written *Flappy Bird*. Similarly, the Transformer model often tries to translate part of the hashtags relying

[8] To evaluate translation quality at the sentence level, we consider the edit distance between the translation hypothesis and the reference.

[9] Because we only report statistics on the best and worst translations of each system, our analysis does not allow us to directly compare the performance of the two models and we can only identify general trends in the impact of UGC idiosyncrasies on translation quality.

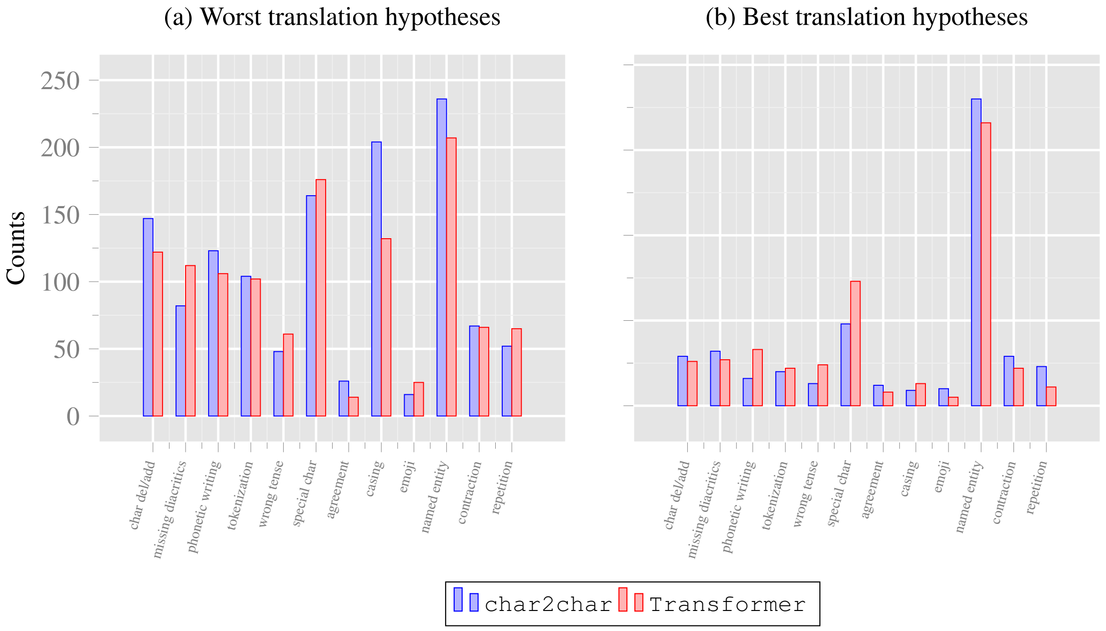

Figure 1: Comparison of the number of UGC specificities in the best and worst translation hypotheses of `char2char` and `Transformer`. Noise categories are defined in Table 4.

④	src	Si au bout de 5 mois **tu rescend** (3) **toujours se** (3) **genre** "d'effet secondaire" c'est vraiment mauvais.
	ref	If after 5 months you're still feeling this kind of "side effect" that's really bad.
	Tx	If after five months you're still re-exciting this whole "special effect" thing is really bad.
	c2c	In the end of five months, you're always responding to secondary effects, that's really bad.
⑤	src	y a ma cousine qui **joue a** (2) **flappy bird** (10) **mdrrrrrrrrrr** (11, 12) **elle et plus nuuul** (12,7) **que moi**
	ref	my cousin plays flappy bird looooooooooool she's more hopeless than me
	Tx	There's my cousin who plays a flappy bird mdrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
	c2c	There's my cousin that plays a flappy bird flappy bird might bring her and nouse than me.

Table 5: Examples from our noisy UGC corpus showing the `Transformer` (denoted as Tx) and `char2char` (denoted as c2c) predictions. Source sentences have been annotated with UGC specificities of Table 1 (in blue). Part of the reference that were correctly generated are underlined.

on the subword units identified by the BPE tokenization rather than simply copying them from the source: for instance "***#ÇaMeVenereQuand***" ("**#It**AnnoysMeWhen" in English) is translated into "*#CaMevenre**When***".

Another important category of errors, is the 6th category that corresponds to hashtags, mentions and URLs, for which the `char2char` model is not capable of producing characters or sequence of characters that are rare in the training set (namely `#`, `@` or `http://www`). For instance, the `char2char` model only outputs 8 sharp symbols when translating the test set of the `PFSMB`, while the reference contains 345 hash tags starting with a '#'. While the `Transformer` model is less sensitive to this problem (it produces 105 sharp symbols when translating the `PFSMB` test), its translation quality also suffers from the presence of hashtags as, as for named entities it often translates some of the subword units resulting from the BPE tokenization. Finally, we notice that errors 2 and 12 (diacritization and repetition) are treated somewhat better by the `char2char` model than by the `Transformer` model, being less frequent in the worst translations of the former.

Qualitative Analysis Table 10 reports translation hypotheses predicted by the `Transformer` and `char2char` models. These examples illustrate the difficulty of the `char2char` model to translate named entities: while the simple strategy of copying unknown OOVs from the source implemented by the `Transformer` is very effective, the `char2char` model tend to scramble or add letters to NE. They also illustrate the impact of phonetic writing on translation: for instance "*rescend*" that should be spelt "*resents*" (*feeling*, example 4) and “*joue a* that should be spelt “*joue à*” (*play a* vs *play with* - auxiliary verb instead of a preposition-example 5) both result in a wrong translation: in the former case, the meaning of the sentence is not captured, in the latter the “*a*” is wrongly copied in the translation hypothesis.

5.1 Copy Task Control Experiment

To corroborate our analysis of the special characters impact on `char2char` and quantify the impact of `char-OOVs` and of rare character sequences, we conduct a control experiment in which a `char2char` system with different vocabulary sizes must learn to copy the source, that is to say: we apply a `char2char` on an artificial parallel corpus in which the source and the target sentences are identical. By varying the number of characters considered by the system, we can control the number of rare characters in our corpora (recall that with a char-vocabulary of size N all characters but the N most frequent ones are mapped to a special `<UNK>` character). Note that this task is a simplification of the problem we have highlighted in the previous section: the model has simply to learn to always make an exact copy of the whole source sentence and not to detect some special characters (such as # or @) from the input that trigger the copy of the next characters, while the rest of the sentence should be translated.

More precisely, we built an artificial train set containing 1M random sentences with lengths between 5 and 15 characters long, keeping a 164 fixed-size character vocabulary (this correspond to the size of the extended ASCII alphabet), and whose characters are distributed uniformly, in order to rule out the impact of rare characters and keeping only the effect of `char-OOVs` over the performance. We consider two test sets made of 3,000 sentences each: `in-test` that uses the same 164 characters as the train set and `out-test` that uses 705 different characters. Source and reference are identical for every example of the train and test sets.

Results Table 6 reports the results achieved on the copy task with and without replacing the predicted `<UNK>` symbols.

Note that, in this very simple task, `<UNK>` characters are always replaced by their true value. These results show that this task is not trivial for char-based systems: even when all characters have been observed during training, the system is not able to copy the input perfectly.

Above all, reducing the vocabulary size from 164 to 125 results in an increase of the BLEU score on the two considered conditions, even without replacing the `<UNK>` that the system has started to generate, where ‘%`<UNK>` pred.’ indicates the percentage of `<UNK>` tokens in the prediction. Further reducing the size of the vocabulary artificially improves the quality of the systems: they generate more and more `<UNK>`, which are replaced during post-processing by their true value. These observations suggest that unknown or rare characters are not only difficult to copy, but they also distort the representation of the input sequence built by the system impacting the whole prediction.

	Vocabulary Size				
	164	125	100	80	60
`in-test`					
%`<UNK>` pred.	0	0.2	5	17	29.5
BLEU	92.9	95.8	77.6	24.9	1.9
+`<UNK>` rep.	92.9	96.6	98.4	98.5	98.7
`out-test`					
%`<UNK>` pred.	0	9.2	13.8	25	36
BLEU	54.5	63.7	52.3	15.3	0.9
+`<UNK>` rep.	54.4	96.6	98.7	99.1	99.5

Table 6: Results of the copy task evaluated by the BLEU score before and after `<UNK>` replacement (*+`<UNK>` rep.*) and percentage of `<UNK>` characters in the prediction (*%`<UNK>` pred.*).

6 Impact of the Char-Vocabulary Size on Translation Quality

To validate in a more realistic setting the conclusion we draw from our copy tasks experiments, we present in Figure 2, the impact of unknown characters on translation quality of the `PFSMB` test set for the `Transformer` and `char2char` models. While the translation quality is almost the same for both models when there is no `char-OOVs` in the input, the occurrence of an unknown character hurts the translation quality of the `char2char` model beyond the mistranslation of a single character.

Motivated by these observations, we carried out experiments on our `char2char` models, trained on `OpenSubtitles`, with different character vocabulary size. Results in Table 7 show that ‘correctly’ choosing the vocabulary size can improve the translation performance on both canonical and non-canonical corpora. We did not observe any noticeable improvements in in-domain data as those test sets contain very few `char-OOVs` compared to their UGC counterparts, *e.g.,* 0.76% for the

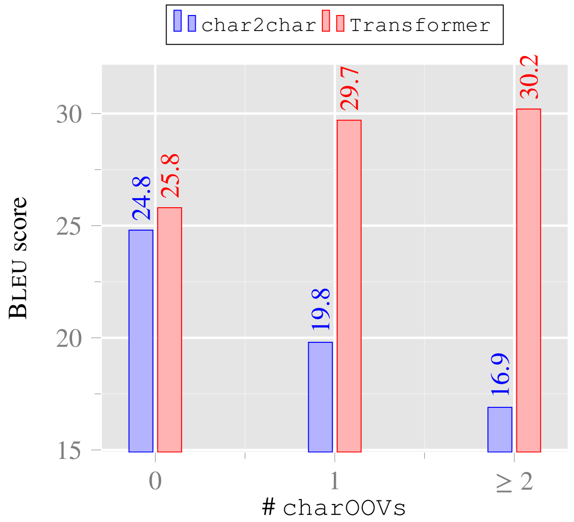

Figure 2: Impact of the number of char-OOVs in the input sentence on the BLEU score. Systems trained on OpenSubs.

OpenSubtitles test set. However, overall stability in in-domain performance is observed, unlike what has been reported when domain-adaptation methods were used (Michel and Neubig, 2018; Belinkov and Bisk, 2018).

For vocabulary sizes larger than 90 characters, the char2char system achieves the same BLEU scores as with 90 characters. Note that in our first experiments (reported in Table 3), we used the 'standard' character vocabulary size (namely 300 characters) that was used in (Lee et al., 2017) and, to the best of our knowledge, in all following works.

vocab. size	PFSMB	MTNT	News	OpenTest†
90	23.9	25.8	18.7	26.6
85	23.9	25.3	**19.9**	**26.9**
80	23.9	25.8	18.3	26.6
75	24.5	**25.9**	17.8	26.3
70	**24.6**	25.4	17.8	26.3
65	22.7	25.5	18.0	26.4

Table 7: BLEU results for MT of char2char with reduced vocabulary size. Systems trained on OpenSubs. † *marks in-domain test set.*

As shown by an evaluation done on the blind tests of our UGC dataset, in Table 8, setting the vocabulary size parameters based on the improvement they bring reduces the difference in performance between the char2char and Seq2seq systems. With the 'correct' vocabulary size the former is outperformed by only 0.3 (resp. 1) BLEU points while, with the default value, the difference is 1.2 (resp. 3.1) BLEU points on the PFSMB (resp. MTNT).

	Cr#pbank blind	MTNT blind
char2char-base	17.8	20.9
vocab-75	18.3	24.0
vocab-70	18.7	22.8
Transformer	*19.0*	*25.0*
seq2seq	*22.1*	*20.4*

Table 8: BLEU results for blind MT test sets of char2char with reduced vocabulary size. Systems trained on OpenSubs. *The Transformer and Seq2seq are presented as comparison data points.*

By manually comparing the system outputs for different vocabulary sizes, it seems that the observed differences in performance can be explained by drop- and over-translation phenomena, two well-identified limits of NMT (Mi et al., 2016; Koehn and Knowles, 2017; Le et al., 2017). An analysis of Figure 3, which displays the ratio between the hypothesis and reference lengths for the different vocabulary sizes, seems to confirm this hypothesis as it appears that the vocabulary size parameter provides control over the translation hypothesis length and consequently, a way to limit these drop- and over-translation phenomena.

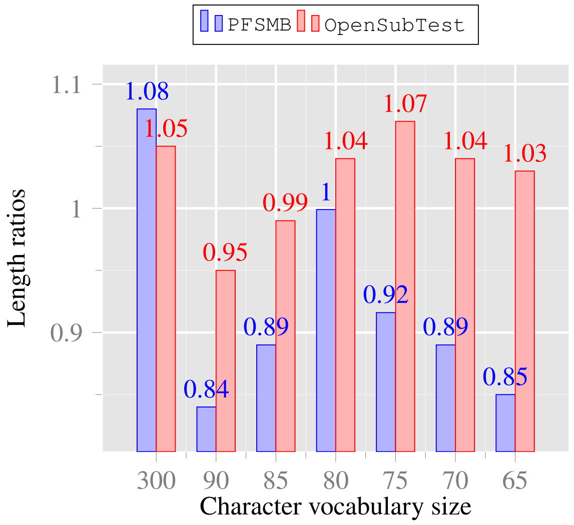

Figure 3: Reference/hypothesis length ratios on the PFSMB test set produced by char2char for different vocabulary sizes.

7 Discussion

Our work follows many studies on the robustness of character-based models, most notably from Belinkov and Bisk (2018) who compared different translation unit levels for NMT and showed that, overall, MT systems trained on clean data only presented considerable performance decrease when processing synthetic character-level noise and word-level noise. Here, we also perform comparisons with other mainstream strong BPE-based baselines and because of our annotated UGC data set are able to provide a more precise view on the phenomena at stake when processing *natural* noisy-input.

Similarly to Durrani et al. (2019) what concluded for morpho-syntactic classification tasks, our results show that a more fined-grained granularity of learning representations, characters over BPEs, provides a higher robustness to certain types of noise. Contrary to the both aforementioned works, our study is performed using annotated real-world noisy UGC, which proved crucial for our study.

In line to Fujii et al. (2020) findings, where fine-grained NMT granularity provided robustness advantages when processing misspelling, our results show that the best and worst translations' specificities distribution point to a better performance of char2char for the *missing diacritics* category, giving insights on more specific types of misspellings that affect performance.

Continuing this research track, we broaden the spectrum of studying UGC specificities, exploring the effects of the vocabulary training size and show that tuning it can achieve better results when translating noisy UGC. This simple hyper-parameter choice proved effective, providing an alternative to using fine-tuning methods (Michel and Neubig, 2018) or back-translation (Vaibhav et al., 2019).

8 Conclusion

We showed that in zero-shot scenarios, char-based models are not robust to UGC idiosyncrasies. We presented several experiments that explained this counter-intuitive result by an over-sensibility of these models to the vocabulary size. We demonstrated that drastically lowering this parameter increased the robustness of our char-based models when facing noisy UGC while maintaining almost the same level of performance for in-domain and out-of-domain canonical datasets. Interestingly, we noticed a lack in the literature regarding the importance of vocabulary considerations when training character-based NMT models. However, our results show that a path for improvement exists, leading to more generic and stable models and making us believe that further research on the matter can bring promising alternatives to domain-adaptation via target domain data addition. Our data set is available at `https://github.com/josecar25/char_based_NMT-noisy_UGC`.

References

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana L. Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin M. Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, pages 131–198.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*.

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, Yonatan Belinkov, and Preslav Nakov. 2019. One size does not fit all: Comparing NMT representations of different granularities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1504–1516. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018a. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 653–663, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018b. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 653–663.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *HLT-NAACL*, Atlanta, USA.

Jennifer Foster. 2010. "cba to check the spelling": Investigating parser performance on discussion forum posts. In *NAACL*, Los Angeles, California.

Ryo Fujii, Masato Mita, Kaori Abe, Kazuaki Hanawa, Makoto Morishita, Jun Suzuki, and Kentaro Inui. 2020. Phemt: A phenomenon-wise dataset for machine translation robustness on user-generated contents. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5929–5943. International Committee on Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, NMT@ACL 2018, Melbourne, Australia, July 20, 2018*, pages 74–83. Association for Computational Linguistics.

Yoon Kim, Yacine Jernite, David A. Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2741–2749.

Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander M. Rush. 2018. Opennmt: Neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas, AMTA 2018, Boston, MA, USA, March 17-21, 2018 - Volume 1: Research Papers*, pages 177–184.

P. Koehn and R. Knowles. 2017. Six challenges for neural machine translation. In *NMT@ACL*.

An Nguyen Le, Ander Martinez, Akifumi Yoshimoto, and Yuji Matsumoto. 2017. Improving sequence to sequence neural machine translation by utilizing syntactic dependency information. In *IJCNLP*.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *TACL*, 5:365–378.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2015. Character-based neural machine translation. *arXiv preprint arXiv:1511.04586*.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. Opensubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1063, Berlin, Germany. Association for Computational Linguistics.

Héctor Martínez Alonso, Djamé Seddah, and Benoît Sagot. 2016. From noisy questions to Minecraft texts: Annotation challenges in extreme syntax scenario. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 13–23, Osaka, Japan. The COLING 2016 Organizing Committee.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage embedding models for neural machine translation. In *EMNLP*.

Paul Michel and Graham Neubig. 2018. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 543–553.

Graham Neubig, Matthias Sperber, Xinyi Wang, Matthieu Felix, Austin Matthews, Sarguna Padmanabhan, Ye Qi, Devendra Singh Sachan, Philip Arthur, Pierre Godard, John Hewitt, Rachid Riad, and Liming Wang. 2018. XNMT: the extensible neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas, AMTA 2018, Boston, MA, USA, March 17-21, 2018 - Volume 1: Research Papers*, pages 185–192.

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, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

José Carlos Rosales Núñez, Djamé Seddah, and Guillaume Wisniewski. 2019. Comparison between NMT and PBSMT performance for translating noisy user-generated content. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 2–14, Turku, Finland. Linköping University Electronic Press.

Manuela Sanguinetti, Lauren Cassidy, Cristina Bosco, Özlem Çetinoglu, Alessandra Teresa Cignarella, Teresa Lynn, Ines Rehbein, Josef Ruppenhofer, Djamé Seddah, and Amir Zeldes. 2020. Treebanking user-generated content: a UD based overview of guidelines, corpora and unified recommendations. *CoRR*, abs/2011.02063.

Djamé Seddah, Benoît Sagot, Marie Candito, Virginie Mouilleron, and Vanessa Combet. 2012. The french social media bank: a treebank of noisy user generated content. In *COLING 2012, 24th International*

Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India, pages 2441–2458.

Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving robustness of machine translation with synthetic noise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1916–1920. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz 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.

A Appendix

Cr#pbank	src	ma **TL** se vide après **j'me fais chier** puis **jme** sens seule mais **c** surtout **pdt les vacs mais c pas le cas dc** ça compte pas trop
	ref	my TL is emptying then I get bored then I feel alone but mostly during holidays but it's not the case so it's not so important
	google	my TL is empty after I'm pissing then I feel alone but c especially pdt vacs but it does not matter that it does not count too much
	src	Dans Flappy Bird quand mon Bird il fait 10 jsuis trop contente #TeamNul Mdddr
	ref	At Flappy Bird when my Bird has 10 Im so happy #TeamFail Loool
	google	In Flappy Bird when my Bird is 10, I'm too happy #TeamNul Mdddr
	src	**Booooooooooooooooooooooonnne Annniversaure Ma viiiiiiiiiiiiie jtm** plus que tout profite bien de tes **22ans moaaaaaaaaa**
	ref	Haaaaaaaaaaaaaaaaaaaaaapppy Biiirthday My liiiiiiiiiiiiife luv you more than anything enjoy your 22years mwaaaaaaaaaah
	google	Booooooooooooooooooooooonnne Annniversaure My viiiiiiiiiiiiie jtm more than anything enjoy your 22 years moaaaaaaaaaa
MTNT	src	**AJA** que le XV de France féminin est **un train de faire** un grand **chelem, OKLM**
	ref	TIL that the XV of Feminine France is doing a grand chelem, FRESH
	google	AJA that the XV of France feminine is a train of a grand slam, OKLM
	src	Je pensais mais c'est le même **ident et mdp** que mon compte "normal", et il détecte même la **profile pic** et le nom
	ref	I thought so but it's the same username and password as my 'normal' account, and it detects even the profile pic and the name
	google	I thought but it's the same ident and mdp as my "normal" account, and it even detects the pic profile and the name
	src	**Aaaaaaaah.... 8 ans après, je viens de percuter.... :o 'tai** je me disais bien que je passais à côté d'un truc vu les **upvotes.**
	ref	Aaaaaaaah.... 8 years later, I've just realized.... :o damn I had the feeling that I was missing something considering the upvotes.
	google	Aaaaaaaah .... 8 years later, I just collided ....: oh well I was telling myself that I was missing something seen the upvotes.

Table 9: Examples from both noisy UGC showing the source phrase, reference translation and Google Translate output. UGC idiosyncrasies are highlighted using bold characters.

①	src	**JohnDoe389** (10) qui n'arrive pas **a** (2) **depasser** (2) 1 a (2) **FlappyBird** 10 ... **ptddddr** (11, 12)
	ref	JohnDoe389 who can't score more than 1 at FlappyBird ... lmaooooo
	Tx	John Doe389 who can't pass one to Flappy Bird... ptdddr.
	c2c	Johndeigh doesn't happen to pass one on Flappyrib... please.
②	src	**JohnDoe665** (10) **viens** (5) de regarder **Teen Wolf (2011) S03E17 [Silverfinger]** (10) et s'en va apprendre l'humanité à **Castiel** (10).
	ref	JohnDoe665 just watched Teen Wolf (2011) S03E17 [Silverfinger] and he's on his way to teach Castiel humanity.
	Tx	John Doe665 just watched Teen Wolf (2011) S03E17 (Silverfinger) and is going to teach humanity at Castiel.
	c2c	Johndoedsoids is looking at Teen Wolf 2011, and learn about humanity in Castiel.
③	src	**Jai** (1,4) **fait 10 a** (2) **flappy bird** (10) **mddr** (11, 12) **# JeMeLaDonneMaisJavancePas** (6)
	ref	I did 10 at flappy bird lool # JeMeLaDonneMaisJavancePas
	Tx	I did 10 a flappy bird mdr # I'm not moving
	c2c	♪I've made 10 flappy birdd ♪

Table 10: Examples from our noisy UGC corpus showing the Transformer (denoted as Tx) and char2char (denoted as c2c) predictions. Source sentences have been annotated with UGC specificities of Table 1 (in blue). Part of the reference that were correctly generated are underlined.

B Reproducibility

Data All the UGC test sets and source code for our experiments are provided in the supplementary materials. For training data, we let the reader refer to each project's website for WMT[10] (consisting of Europarlv7, Newcommentariesv10 and Open Subtitles[11], both accessed on November, 2019. Regarding clean test sets, we used newstest15 from WMT and a subset of 11.000 unique phrases from Open Subtitles. We make the former test available in the link provided above for exact performance comparison.

Computation The NMT systems were trained using 1 Tesla V100, during an average of 72 hours to converge to the final solution for the char2char model and 56 hours for the BPE-based baselines.

B.1 NMT Models

Character-based models Both charCNN and char2char models were trained as out-of-the box systems using the implementations provided by (Kim et al., 2016) for charCNN,[12] an by (Lee et al., 2017) for char2char.[13]

BPE-based models We consider, as baseline, two standard NMT models that take, as input, tokenized sentences. The first one is a seq2seq bi-LSTM architecture with global attention decoding. The seq2seq model was trained using the XNMT toolkit (Neubig et al., 2018).[14] It consists of a 2-layered Bi-LSTM layers encoder and 2-layered Bi-LSTM decoder. It considers, as input, word embeddings of 512 components and each LSTM units has 1,024 components.

We also study a vanilla Transformer model using the implementation proposed in the OpenNMT framework (Klein et al., 2018). It consists of 6 layers with word embeddings of 512 components, a feed-forward layers made of 2,048 units and 8 self-attention heads.

The vocabulary parameter for experiments in § 6 were obtained through a first exploratory uniform segmentation of the possible vocabulary sizes, and then, discovering the threshold (vocabulary with 90 characters) for which the char2char model started producing <UNK> during evaluation. We then proceeded to obtain the results displayed above by sampling this parameter by decreasing 5 characters.

Hyper-parameters In Table 11, we list the training variables set for our experiments. They match their corresponding default hyper-parameters.

Batch size	64
Optimizer	Adam
Learning rate	1e-4
Epochs	10 (best of)
Patience	2 epochs
Gradient clip	[-1.0, 1.0]

Table 11: Hyper-parameters used for training the NMT systems.

Pre-processing For the BPE models, we used a 16K merging operations tokenization employing sentencepiece[15]. For word-level statistics we segmented the corpora using the Moses tokenizer [16].

[10] https://www.statmt.org/wmt15/translation-task.html

[11] http://opus.nlpl.eu/download.php?f=OpenSubtitles/v2018/moses/en-fr.txt.zip

[12] https://github.com/harvardnlp/seq2seq-attn

[13] https://github.com/nyu-dl/dl4mt-c2c

[14] We decided to use XNMT, instead of OpenNMT in our experiments in order to compare our results to the ones of Michel and Neubig (2018).

[15] https://github.com/google/sentencepiece

[16] https://github.com/moses-smt/mosesdecoder