Phonetic Normalization for Machine Translation of User Generated Content

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

Abstract

We present an approach to correct noisy User Generated Content (UGC) in French aiming to produce a pre-processing pipeline to improve Machine Translation for this kind of non-canonical corpora. Our approach leverages the fact that some errors are due to confusion induced by words with similar pronunciation which can be corrected using a phonetic look-up table to produce normalization candidates. We rely on a character-based neural model phonetizer to produce IPA pronunciations of words and a similarity metric based on the IPA representation of words that allow us to identify words with similar pronunciation. These potential corrections are then encoded in a lattice and ranked using a language model to output the most probable corrected phrase. Compared to other phonetizers, our method boosts a `Transformer`-based machine translation system on UGC.

1 Introduction

In this work we aim to improve the translation quality of User-Generated Content (UGC). This kind of content generally contains many characters repetitions, typographic errors, contractions, jargon or non-canonical syntactic constructions, resulting in a typically high number of Out-of-Vocabulary words (OOVs), which, in turn, significantly decreases MT quality and can introduce noisy artefacts in the output due to rare tokens. Hereby, we propose a normalization pipeline that leverages on the existence of UGC specific noise due to the misuse of words or OOV contractions that have a similar pronunciation to those of the expected correct tokens. This method works without any supervision on noisy UGC corpora, but exploits phonetic similarity to propose normalization token candidates. To explore the capacities of our system, we first assess the performance of our normalizer and then conduct a series of MT experiments to determine if our method improves the translation quality of some Phrase-Based Statistical Machine Translation (`PBSMT`) and Neural Machine Translation (NMT) baselines. Our results show that including a phonetization step in conjunction with a `Transformer` architecture (Vaswani et al., 2017) can improve machine translation over UGC with a minimum impact on in-domain translations. This suggests that phonetic normalization can be a promising research avenue for MT and automatic correction of UGC.

Our contribution in this paper is threefold:

- we propose a pre-processing pipeline to normalize UGC and improve MT quality;
- by quantifying the corrections made by our normalizer in our UGC corpora, we assess the presence of noise due to phonetic writing and demonstrate that this knowledge can be potentially exploited to produce corrections of UGC without any annotated data;
- we explore the performance improvement that can be achieved in machine translation by using a phonetic similarity heuristic to propose different normalization candidates.

2 Related Work

Several works have focused on using lattices to model uncertain inputs or potential processing errors that occur in the early stage of the pipeline. For instance, Su et al. (2017) proposed `lat2seq`, an extension of `seq2seq` models (Sutskever et al., 2014) able to encode several possible input possibilities by conditioning their GRU output to several predecessors' paths. The main issue with this model is that it is unable to predict the score of choosing a certain path by using future scores, i.e, by considering words that come after the current

Proceedings of the 2019 EMNLP Workshop W-NUT: The 5th Workshop on Noisy User-generated Text, pages 407–416
Hong Kong, Nov 4, 2019. ©2019 Association for Computational Linguistics

token to be potentially normalized. Sperber et al. (2017) introduced a model based on `Tree-LSTMs` (Tai et al., 2015), to correct outputs of an Automatic Speech Recognition (ASR) system. On the other hand, Le et al. (2008) use lattices composed of written subword units to improve recognition rate on ASR.

However, none of the aforementioned works have focused on processing noisy UGC corpora and they do not consider our main hypotheses of using phonetizers to recover correct tokens. They aim to correct known tokens such that a neural language model chooses the best output when an uncertain input is present (typically words with similar pronunciation from an ASR output). Instead, our approach calculates the phonetization of the source token and candidates are proposed based on their phonetic similarity to it, where this original observation can be a potential OOV.

On the same trend, (Qin et al., 2012) combined several ASR systems to improve detection of OOVs. More recently, van der Goot and van Noord (2018) achieved state-of-the-art performance on dependency parsing of UGC using lattices.

Closely related to our work, Baranes (2015) explored several normalization techniques on French UGC. In particular, to recover from typographical errors, they considered a rule-based system, SxPipe (Sagot and Boullier, 2008), that produced lattices encoding OOVs alternative spelling and used a language model to select the best correction.

Several works have explored different approaches to normalize noisy UGC in various languages. For instance, Stymne (2011) use Approximate String Matching, an algorithm based on a weighted Levenshtein edit distance to generate lattices containing alternative spelling of OOVs. Wang and Ng (2013) employ a Conditional Random Field and a beam-search decoding approach to address missing punctuation and words in Chinese and English social media text. More recently, Watson et al. (2018) proposed a neural sequence-to-sequence embedding enhancing `FastText` (Bojanowski et al., 2017) representations with word-level information, which achieved state-of-the-art on the QALB Arabic normalization task (Mohit et al., 2014).

3 Phonetic Correction Model

To automatically process phonetic writing and map UGC to their correct spelling, we propose a simple model based on finding, for each token of the sentence, words with similar pronunciations and selecting the best spelling alternative, using a language model. More precisely, we propose a four-step process:

1. for each word of the input sentence, we automatically generate its pronunciation. We consider all words in the input sentence as misspelled tokens are not necessarily OOVs (e.g. "*j'ai* ***manger***" — literally "*I have* ***eat***" — which must be corrected to "*j'ai* ***mangé***" — "*I have* ***eaten***", the French words "manger" and "mangé" having both the same pronunciation /mɑ̃.ge/);

2. using these phonetic representations, we look, for each word w of the input sentence, to every word in the training vocabulary with a pronunciation "similar" to w according to an ad-hoc metric we discuss below;

3. we represent each input sentence by a lattice of $n + 1$ nodes (where n is the number of words in the sentence) in which the edge between the i-th and $(i + 1)$-th nodes is labeled with the i-th word of the sentence. Alternative spellings can then be encoded by adding an edge between the i-th and $(i + 1)$-th nodes labeled by a possible correction of the i-th word. Figure 1 gives an example of such a lattice. In these lattices, a path between the initial and final nodes represents a (possible) normalization of the input sentence.

4. using a language model, we compute the probability to observe each alternative spelling of the sentence (note that, by construction, the input sentence is also contained in the lattice) and find the most probable path (and therefore potential normalization) of the input sentence. Note that finding the most probable path in a lattice can be done with a complexity proportional to the size of the sentence even if the lattice encodes a number of paths that grows exponentially with the sentence size (Mohri, 2002). In our experiments we used the `OpenGRM` (Roark et al., 2012) and `OpenFST` (Allauzen et al., 2007) frameworks that provide a very efficient implementation to score a lattice with a language model.

This process can be seen as a naive spellchecker, in which we only consider a reduced set of variations,

tailored to the specificities of UGC texts. We will now detail the first two steps discussed above.

Generating the pronunciation of the input words To predict the pronunciation of an input word, i.e. its representation in the International Phonetic Alphabet (IPA), we use the `gtp-seq2seq` python library[1] to implement a grapheme-to-phoneme conversion tool that relies on a `Transformer` model (Vaswani et al., 2017). We use a 3 layers model with 256 hidden units that is trained on a pronunciation dictionary automatically extracted from `Wiktionary` (see Section 4.1 for a description of our datasets). This vanilla model achieves a word-level accuracy of 94.6%, that is to say it is able to find the exact correct phonetization of almost 95% of the words of our test set.

We also consider, as a baseline, the pronunciation generated by the `Espeak` program.[2] that uses a formant synthesis method to produce phonetizations based on acoustic parameters.

Finding words with similar pronunciation In order to generate alternatives spelling for each input word, we look, in our pronunciation dictionary,[3] for alternate candidates based on phonetic similarity. We define the phonetic similarity of two words as the edit distance between their IPA representations, all edit operations being weighted depending on the articulatory features of the sounds involved. To compute the phonetic similarity we used the implementation (and weights) provided by the `PanPhon` library (Mortensen et al., 2016).

To avoid an explosion of the number of alternatives we consider, we have applied a threshold on the phonetic distance and consider only homophones, i.e. alternatives that have the exact same pronunciation as the original word.[4]

To account for peculiarities of French orthography we also systematically consider alternative spellings in which diacritics (acute, grave and circumflex accents) for the letter *"e"* (which is the only one that changes the pronunciation for different accentuation in French) were added wherever possible. Indeed, users often tend to 'forget' diacritics when writing online and this kind of spelling error results in phonetic distances that can be large (e.g. the pronunciation of *bebe* and *bébé* is very different).

We ultimately only keep as candidates those that are present in the train corpus of Section 4.3 to filter out OOV, and nonexistent words.

4 Datasets

In this section, we present the different corpora used in this work. We will first describe the dataset used to train our phonetic normalizer; then, in §4.2, the UGC corpora used both to measure the performance of our normalization step and evaluate the impact of phonetic spelling on machine translation. Finally, in § 4.3 we introduce the (canonical) parallel corpora we used to train our MT system. All our experiments are made on French UGC corpora.[5] Some statistics describing these corpora are listed in Table 1.

4.1 Pronunciation Dictionary

To train our Grapheme-to-Phoneme model we use a dictionary mapping words to their pronunciation (given by their IPA representation). To the best of our knowledge, there is no free pronunciation dictionary for French. In our experiments, we have considered a pronunciation dictionary automatically extracted from `Wiktionary` dumps building on the fact that, at least for French, pronunciation information are identified using special *templates*, which makes their extraction straightforward (Pécheux et al., 2016).[6]

The dictionary extracted from the French `Wiktionary` contains 1,571,090 words. We trained our `G2P` phonetizer on 1,200,000 examples, leaving the rest to evaluate its performance. When looking for words with similar pronunciation (§3), we consider only the word that appear in our parallel training data (described in §4.3) to speed up the search. After filtering, our dictionary contained pronunciation information for roughly 82K French words.

4.2 UGC Corpora

The Parallel `Cr#pbank` corpus The Parallel `Cr#pbank`, introduced in (Rosales Núñez et al.,

[1] https://github.com/cmusphinx/g2p-seq2seq

[2] espeak.sourceforge.net

[3] see § 4.1 for the description of the data we used

[4] We have explored using several values for this parameter but in this work only the most conservative distance (0) is used since higher values add too much candidates and rapidly decreases performance due to the number of ambiguities.

[5] Applying our work to other languages is straightforward and left to future work.

[6] Our French pronunciation dictionary will be made available upon publication.

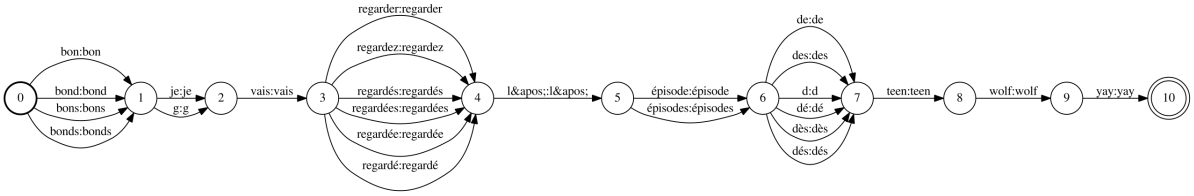

Figure 1: Example of lattice for a segment of a `Cr#pbank` UGC sample.

Corpus	#sentences	#tokens	ASL	TTR
train set				
`WMT`	2.2M	64.2M	29.7	0.20
Small	9.2M	57.7M	6.73	0.18
Large	34M	1.19B	6.86	0.25
test set				
`OpenSubTest`	11,000	66,148	6.01	0.23
`NeswTest`	3,003	68,155	22.70	0.23

Corpus	#sentences	#tokens	ASL	TTR
UGC test set				
`Cr#pbank`	777	13,680	17.60	0.32
`MTNT`	1,022	20,169	19.70	0.34
UGC blind test set				
`Cr#pbank`	777	12,808	16.48	0.37
`MTNT`	599	8,176	13.62	0.38

Table 1: Statistics on the French side of the corpora used in our experiments. *TTR stands for Type-to-Token Ratio, ASL for average sentence length.*

2019), consists of 1,554 comments in French, translated from an extension of the French Social Media Bank (Seddah et al., 2012) annotated with the following linguistic information: Part-of-Speech tags, surface syntactic representations, as well as a normalized form whenever necessary. Comments have been translated from French to English by a native French speaker with near-native English speaker capabilities. Typographic and grammar error were corrected in the gold translations but some of the specificities of UGC were kept. For instance, idiomatic expressions were mapped directly to the corresponding ones in English (e.g. "`mdr`" (*mort de rire,* litt. *dying of laughter*) has been translated to "`lol`" and letter repetitions were also kept (e.g. "`ouiii`" has been translated to "`yesss`"). For our experiments, we have divided the `Cr#pbank` into two sets (test and blind) containing 777 comments each. This corpus can be freely downloaded at `https://gitlab.inria.fr/seddah/parsiti`.

The MTNT corpus We also consider in our experiments, the `MTNT` corpus (Michel and Neubig, 2018), a multilingual dataset that contains French sentences collected on `Reddit` and translated into English by professional translators. We used their designated test set and added a blind test set of 599 sentences we sampled from the `MTNT` validation set. The `Cr#pbank` and `MTNT` corpora both differ in the domain they consider, their collection date, and in the way sentences were filtered to ensure they are sufficiently different from canonical data.

4.3 Canonical Parallel Corpora

To train our MT systems, we use the 'standard' parallel data, namely the `Europarl` and `NewsCommentaries` corpora that are used in the WMT evaluation campaign (Bojar et al., 2016) and the `OpenSubtitles` corpus (Lison et al., 2018). We will discuss the different training data configurations for the MT experiments more in detail in Section 5.

We also use the totality of the French part of these corpora to train a 5-gram language model with Knesser-Ney smoothing (Ney et al., 1994) that is used to score possible rewritings of the input sentence and find the best normalization, as we have discussed in Section 3.

5 Machine Translation Experiments

To evaluate whether our approach improve the translation quality of UGC, we have processed all of our test sets, both UGC and canonical ones with our phonetic normalization pipeline (Section 3). The corrected input sentences are then translated by a phrase-based and NMT systems.[7] We evaluate translation quality using SacreBLEU (Post, 2018).

The MT baselines models were trained using the parallel corpora described in Section 4.3. We use 3 training data configurations in our experiments: `WMT`, `Small OpenTest`and `Large`

[7]In our experiments we used `Moses` (Koehn et al., 2007) and `OpenNMT` (Klein et al., 2018).

	PBSMT				Transformer			
	Crap	MTNT	News	Open	Crap	MTNT	News	Open
WMT	20.5	21.2	**22.5**†	13.3	15.4	21.2	27.4†	16.3
Small	28.9	27.3	20.4	26.1†	**27.5**	**28.3**	**26.7**	31.4†
Large	**30.0**	**28.6**	22.3	**27.4**†	26.9	28.3	26.6	**31.5**†

Table 2: BLEU score results for our two benchmark models for the different train-test combinations. None of the test sets are normalized. The best result for each test set is marked in bold, in-domain scores with a dag. *Crap, News and Open respectively stand for the* `Cr#pbank`, `NeswTest` *and* `OpenSubTest`.

	PBSMT				Transformer			
	Crap	MTNT	News	Open	Crap	MTNT	News	Open
WMT	20.4	20.2	**21.9**†	13.4	15.0	20.4	**26.7**†	16.2
Small	28.4	26.2	19.9	26.1†	**29.0**	28.3	25.7	31.4†
Large	**29.0**	27.6	21.8	**27.4**†	28.5	28.2	25.9	**31.5**†

(a) **(`G2P`)** phonetizer.

	PBSMT				Transformer			
	Crap	MTNT	News	Open	Crap	MTNT	News	Open
WMT	20.4	20.4	21.7†	13.4	14.6	20.7	26.5†	16.1
Small	28.0	26.3	19.8	26.2†	28.5	**28.8**	25.6	31.4†
Large	28.3	**27.7**	21.6	**27.4**†	27.5	28.6	25.8	**31.5**†

(b) **(`Espeak`)** phonetizer.

Table 3: BLEU score results for our three benchmark models on normalized test sets. The best result for each test set is marked in bold, in-domain scores with a dag.

`OpenTest`, for which Table 1 reports some statistics. We will denote `Small` and `Large` the two `OpenSubtitles` training sets used in the MT experiments. For every model, we tokenize the training data using byte-pair encoding (Sennrich et al., 2016) with a 16K vocabulary size.

BLEU scores for our normalized test sets are reported in Table 3a and Table 3b, for the `G2P` and `Espeak` phonetizers. Results of the unprocessed test sets are reported in Table 2. We present some UGC examples of positive and negative results along with their normalization and translation in Table 6.

6 Results Discussion

We noticed significant improvement in results for the UGC test corpora when using the `Transformer` architecture trained with the `Small OpenTest`training set. Specifically, a BLEU score improvement for the `Cr#pbank` and `MTNT` test corpora in Tables 3a and 3b, compared to the baseline translation in Table 2. Interestingly, these improvements only hold for the `Transformer` model, whereas we consistently obtain a slight decrease of BLEU scores when the normalized text is translated using the `PBSMT` model. Moreover, our trained `G2P` phonetizer achieved the best improvement over the `Cr#pbank` corpus, attaining +1.5 BLEU points compared to the baseline. On the other hand, the `Espeak` phonetizer produces the highest translation improvement on the `MTNT` corpus (+0.5 BLEU).

Regarding the performance decrease on our non-UGC test corpora, `newstest'14` and `OpenSubtitles`, we observe that there is usually a considerable under-performance on the latter (-0.65 BLEU averaging over our 6 model and training set configurations), that is not as noticeable in the former (-0.1 BLEU in the worst case). This could be explained by the substantially longer sentences in `newstest'14` compared to `OpenSubtitles`, which have roughly 6 times more words in average according to Table 1. When sentences are longer, the number of possible lattices paths grow exponentially, thus adding confusion to our language model decisions that will ultimately produce the most probable normalization. Such observation strongly suggests that our normalizing method performances is somewhat dependent

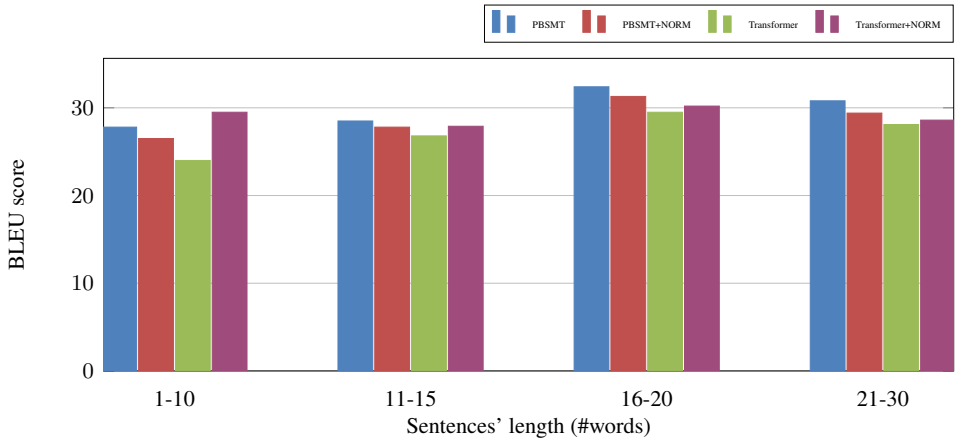

Figure 2: Bar plot of the BLEU score for the Cr#pbank test set translation divided in sentences' length groups.

on the length of the target sentence that are to be normalized.

In addition, we display the number of replacements performed by our normalizer over the Cr#pbank test set for several values of the phonetic distance threshold in Figure 3. We can notice that the higher this threshold is, the higher the number of replacements. In our experience, the normalization candidates proposed by our method do not share a close pronunciation for threshold values above 0.2, thus adding a substantial quantity of spurious ambiguities.

We have also calculated and proceed to display the BLEU score of the Cr#pbank corpus by groups of sentences length in Figure 2 in order to further investigate why our method enhances the Transformer MT systems output, whereas this is not the case for the PBSMT models, as seen in Table 3. In this way, in Figure 2, we can notice that the highest improvement caused by our phonetic normalization pipeline is present in short sentences (between 1 and 10 words). It is worth noting that this is the only case where the Transformer outperforms PBSMT in this Figure. Hence, the higher overall Transformer BLEU score over PBSMT is certainly due to a relatively high successful normalization over the shortest sentences of the Cr#pbank test set. This agrees with the documented fact that NMT is consistently better than PBSMT on short sentences (Bentivogli et al., 2016) and, in this concrete example, it seems that the Transformer can take advantage of this when we apply our normalization pipeline. Additionally, these results could be regarded as evidence supporting that our proposed method performs generally better for short sentences, as observed in Table 3 results' discussion.

System	Blind Tests MTNT	Cr#pbank
Large - PBSMT Raw	**29.3**	**30.5**
Large - PBSMT Phon. Norm	26.7	26.9
Small - Transformer Raw	**25.0**	**19.0**
Small - Transformer Phon. Norm	24.5	18.3
M&N18 Raw	19.3	13.3
M&N18 UNK rep. Raw	21.9	15.4

Table 4: BLEU score results comparison on the MTNT and Cr#pbank blind test sets. The G2P phonetizer has been used for normalization.*M&N18 stands for (Michel and Neubig, 2018)'s baseline system.*

Furthermore, we have applied our method over a blind test set of the UGC corpora MTNT and Cr#pbank. These results are displayed in Table 4, we also show the performance of the (Michel and Neubig, 2018)'s baseline system on such test sets. The translation system is selected as the best for each of the UGC sets from Table 3. For such test corpus, we noticed a 0.5 and a 3 BLEU points decrease for Transformer and PBSMT systems, respectively, when our normalizer is used over the MTNT blind test. On the other hand, we obtained a 0.7 BLUE point loss for the Transformer and a 3.6 point drop for PBSMT, both on the Cr#pbank blind test. These results suggest that, when we do not tune looking for the best translation system, and for certain UGC, our approach introduces too much noise and MT performance can therefore be detrimentally impacted.

Normalization	a→à	sa→ça	et→est	la→là	à→a	tous→tout	des→de	regarder→regardé	ils→il	prend→prends
Number of app.	87	16	15	13	12	11	8	7	6	6

Table 5: Most frequent normalization replacements on the `Cr#pbank` test corpus.

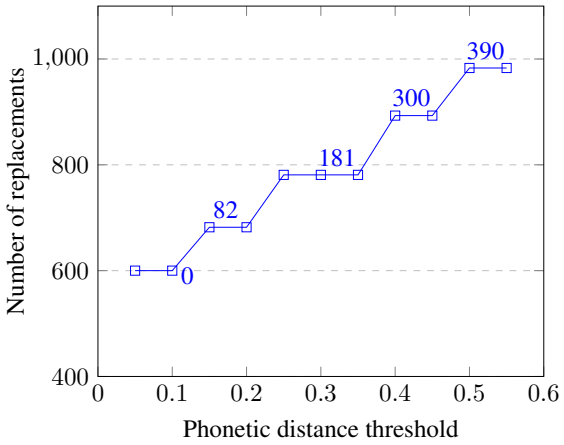

Figure 3: Number of replacement operations of our normalizer over the `Cr#pbank` test set. The quantity of non-homophones normalizations are displayed as point labels.

7 Qualitative Analysis

We display the most frequent normalization changes in the `Cr#pbank` test set, along with their phonetic distance in Table 5. We notice that the 20 most frequent normalization changes are homophones, i.e. they have a 0.0 phonetic distance even when the threshold is set to 0.2.[8] Replacements with a phonetic distance of 0.1 to 0.2, appear at most twice in this test set, except for *"apres"* →*"après"* and *"tt"* →*"td"* that appear, respectively, 6 and 4 times.

Table 6 reports some examples of the output of our method along with their translation before and after correction.

For Example 6.1, we can notice that our normalizer enables the MT system to produce the first part of the translation ("*When I get to the taff*"). This is a result of correctly changing the French homophones "*arriver*" → "*arrivé*", i.e. from the infinitive to the past form. It is very interesting to notice that the robustness of the Transformer using subword units seems to be good enough to correctly translate the typographical error "*ce met a battre*", thus, the correct proposed normalization ("*se met à battre*") does not impact the MT result but it certainly does impact the correctness of the French phrase.

Regarding Example 6.2, we can notice that our normalized proposition significantly improves MT translation, producing an output closer to the reference translation, when compared to the raw MT output. The key normalization change is the misused French token "*faite*" (pronounced /fɛt/) — "*does*" in English — by its correct homophone "*fête*" — "*celebrates*" in English —. It is worth noting that the MT system robustness is once again capable of correctly translating a phonetic contraction "*c*" as the two correct tokens "*c'est*".

Example 6.3 shows how semantically different can be a misused French word due to homophones confusion. We can observe that the normalization replacement *"nez"* ("nose" in English) → *"né"* ("born" in English), which are French homophones, drastically changes the meaning of the output translation. Additionally, the correction *"marqué"* → *"marquer"*[9] (changing to correct verb tense) also causes the translation to be closer to the reference.

Finally, in Example 6 we display some inconveniences for our method, where the correct original plural "*Carte***s** *bancaire***s** ... *restrouvé***s**" was changed to the singular form "*Carte bancaire ... retrouvé*". This is due to the homophonic property of most French singular and plural pronunciations. Whenever there is no discriminant token with different pronunciation, such as an irregular verb, the language model has trouble choosing the correct final normalized phrase since both plural and singular propositions are proposed as candidates and can be indistinctly kept as final normalization since both forms are correct and theoretically very similar in their perplexity measure.

8 Conclusions

In this work, we have proposed a pre-processing method that relies on phonetic similarity to normalize UGC. Our method is able to improve the translation quality of UGC of a state-of-the-art NMT system. Conversely, we have performed error analysis showing that the MT system achieves to correctly translate phonetic-related errors with its increased robustness. However, it must be noted that

[8] This is the highest value for which we consider a related pronunciation, according to our preliminary trials.

[9] *marked* vs *mark*-INFINITIVE in English.

①	src	**arriver au taff, des que j'ouvre le magasin** je commence **a** avoir le vertige mon coeur **ce met a battre a 200** et je sens que je vais faire un malaise,
	ref	**once at work, as soon as I open the store** I'm starting to feel dizzy my heart **starts racing at 200** and I feel I'm gonna faint,
	raw MT	I start to get dizzy. My heart starts to beat at 200 and I feel like I'm going to faint.
	norm	**arrivé au taff, dès que j'ouvre le magasin** je commence **à** avoir le vertige mon coeur **se met à battre à 200** et je sens que je vais faire un malaise,
	norm MT	**When I get to the taff, as soon as I open the store**, I start to get dizzy. My heart starts pounding at 200 and I feel like I'm gonna get dizzy.
②	src	c un peu plus que mon ami qui **faite son annif**,
	ref	it's a bit more than a friend to me who **celebrate his birthday**,
	raw MT	It's a little more than my friend **doing his birthday**,
	norm	c un peu plus que mon amie qui **fête son annif**
	norm MT	It's a little more than my friend **celebrating her birthday**,
③	src	zlatan est **nez** pour **marqué**
	ref	Zlatan was **born** to **score**
	raw MT	Zlatan's **nose** is for **marking**
	norm	zlatan est **né** pour **marquer**
	norm MT	Zlatan was **born** to **score**
④	src	**Cartes bancaires** de Zlatan retrouvés dans un taxi... On en parle ou pas WWW44
	ref	Zlatan's **bank cards** found in a cab... we talk about it or not WW44
	raw MT	Zlatan's **bank cards** found in a cab... we talk about it or not WW44
	norm	**Carte bancaire** de Zlatan retrouvé dans un taxi... On en parle ou pas WWW44
	norm MT	Zlatan **bank card** found in a taxi... we talk about it or not WWW44

Table 6: Examples from our noisy UGC corpus.

we obtained negative results on a blind test evaluation, suggesting that the phonetic normalization approach introduced more noise than useful corrections on totally unseen data. This highlights the importance of holding out data so that the real efficiency of an MT system can be verified. In addition, we have applied our normalizer to clean canonical test data and have shown that it slightly hurts MT quality. Further study is needed to assess whether our proposed normalization pipeline can correct phonetic-related errors on UGC for other languages and other difficult UGC scenarios, such as video-games chat logs (Martínez Alonso et al., 2016) while maintaining the level of the performance on cleanly edited text steady.

Acknowledgments

We thank our anonymous reviewers for providing insightful comments and suggestions. This work was funded by the ANR projects ParSiTi (ANR-16-CE33-0021).

References

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. Openfst: A general and efficient weighted finite-state transducer library. In *Proceedings of the 12th International Conference on Implementation and Application of Automata*, CIAA'07, pages 11–23, Berlin, Heidelberg. Springer-Verlag.

Marion Baranes. 2015. *Spelling Normalisation of Noisy Text*. Theses, Université Paris-Diderot - Paris VII.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: a case study. In *EMNLP*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*, 5:135–146.

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.

Rob van der Goot and Gertjan van Noord. 2018. Modeling input uncertainty in neural network dependency parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4984–4991.

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.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*.

Viet Bac Le, Sopheap Seng, Laurent Besacier, and Brigitte Bigi. 2008. Word/sub-word lattices decomposition and combination for speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2008, March 30 - April 4, 2008, Caesars Palace, Las Vegas, Nevada, USA*, pages 4321–4324.

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.*

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.

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.

Behrang Mohit, Alla Rozovskaya, Nizar Habash, Wajdi Zaghouani, and Ossama Obeid. 2014. The first QALB shared task on automatic text correction for arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing, ANLP@EMNLP 20104, Doha, Qatar, October 25, 2014*, pages 39–47.

Mehryar Mohri. 2002. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350.

David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori S. Levin. 2016. Panphon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484. ACL.

Hermann Ney, Ute Essen, and Reinhard Kneser. 1994. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech & Language*, 8(1):1–38.

Nicolas Pécheux, Guillaume Wisniewski, and François Yvon. 2016. Reassessing the value of resources for cross-lingual transfer of pos tagging models. *Language Resources and Evaluation*, pages 1–34.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 186–191.

Long Qin, Ming Sun, and Alexander I. Rudnicky. 2012. System combination for out-of-vocabulary word detection. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012*, pages 4817–4820.

Brian Roark, Richard Sproat, Cyril Allauzen, Michael Riley, Jeffrey Sorensen, and Terry Tai. 2012. The OpenGrm open-source finite-state grammar software libraries. In *Proceedings of the ACL 2012 System Demonstrations*, pages 61–66, Jeju Island, Korea. 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.

Benoît Sagot and Pierre Boullier. 2008. SxPipe 2: architecture pour le traitement pré-syntaxique de corpus bruts. *Traitement Automatique des Langues*, 49(2):155–188.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel. 2017. Neural lattice-to-sequence models for uncertain inputs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1380–1389.

Sara Stymne. 2011. Spell checking techniques for replacement of unknown words and data cleaning for haitian creole SMS translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT@EMNLP 2011, Edinburgh, Scotland, UK, July 30-31, 2011*, pages 470–477.

Jinsong Su, Zhixing Tan, Deyi Xiong, Rongrong Ji, Xiaodong Shi, and Yang Liu. 2017. Lattice-based recurrent neural network encoders for neural machine translation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3302–3308.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1556–1566.

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.

Pidong Wang and Hwee Tou Ng. 2013. A beam-search decoder for normalization of social media text with application to machine translation. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 471–481.

Daniel Watson, Nasser Zalmout, and Nizar Habash. 2018. Utilizing character and word embeddings for text normalization with sequence-to-sequence models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 837–843.