Comparing MT Approaches for Text Normalization

Claudia Matos Veliz, Orphée De Clercq and Véronique Hoste
LT^3, Language and Translation Technology Team - Ghent Univeristy
Groot-Brittanniëlaan 45, 9000, Ghent, Belgium
Firstname.Lastname@UGent.be

Abstract

One of the main characteristics of social media data is the use of non-standard language. Since NLP tools have been trained on traditional text material, their performance drops when applied to social media data. One way to overcome this is to first perform text normalization. In this work, we apply text normalization to noisy English and Dutch text coming from different genres: text messages, message board posts and tweets. We consider the normalization task as a Machine Translation problem and test the two leading paradigms: statistical and neural machine translation. For SMT we explore the added value of varying background corpora for training the language model. For NMT we have a look at data augmentation since the parallel datasets we are working with are limited in size. Our results reveal that when relying on SMT to perform the normalization, it is beneficial to use a background corpus that is close to the genre to be normalized. Regarding NMT, we find that the translations - or normalizations - coming out of this model are far from perfect and that for a low-resource language like Dutch adding additional training data works better than artificially augmenting the data.

1 Introduction

Probably one of the most persistent characteristics of social media texts is that they are full of non-standard words (Eisenstein, 2013). Several sources of noise can influence the way people write. For example, the different kind of social media platforms available nowadays provide a diverse range of ways to communicate and particular forms of language variations (Ke et al., 2008). Social variables such as gender, age and race can also influence communication style (Schwartz et al., 2013; Blodgett et al., 2016). Location is also an important variable, since it can lead to the use of dialect and non-standard words. (Vandekerckhove and Nobels, 2010; Blodgett et al., 2016).

Very typical for this User-Generated Content (UGC) is the expression of emotions by the use of symbols or lexical variations of the words (Van Hee et al., 2017). This can be done in the form of flooding or the repetition of characters, capitalization, and the productive use of emoticons. In addition, the use of homophonous graphemic variants of a word, abbreviations, spelling mistakes or letter transpositions are very typical, since people tend to write as they speak and/or write as fast as possible.

One can imagine that all those characteristics contribute to an increased difficulty of automatically processing and analyzing UGC. Since Natural Language Processing (NLP) tools have originally been developed for and trained on standard language, these non-standard forms adversely affect language analysis using these tools. One of the computational approaches which has been suggested to tackle this problem is text normalization (Sproat et al., 2001). This approach envisages transforming the lexical variants to their canonical forms. In this way, standard NLP tools can be applied in a next step, after normalization. Kobus et al. (2008) introduced three metaphors to refer to these normalization approaches: the spell checking, automatic speech recognition and machine translation metaphors. In section 2 these approaches are discussed in more depth.

In this work, we follow the third metaphor and tackle text normalization as a Machine Translation (MT) task. We test the two leading paradigms,

Proceedings of Recent Advances in Natural Language Processing, pages 740–749,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_086

statistical machine translation (SMT) and neural machine translation (NMT), on English and Dutch parallel corpora with data coming from three genres (text messages, message board posts and tweets). For SMT we explore the added value of varying background corpora for training the language model. For NMT we have a look at data augmentation since the parallel datasets we are working with are limited in size. Our results reveal that when relying on SMT to perform the normalization it is beneficial to use a background corpus that is close to the genre to be normalized. Regarding NMT, we find that the translations - or normalizations - coming out of this model are far from perfect and that for a low-resource language, like Dutch, adding additional training data works better than artificially augmenting the data.

In the following section, we discuss related work on text normalization. In section 3, we give more information about our parallel data and describe the methodology we used to perform the SMT and NMT experiments. Section 4 gives an overview of the results, whereas section 5 concludes this work and offers some prospects for future work.

2 Related Works

Previous research on UGC text normalization has been performed on diverse languages using different techniques ranging from hand-crafted rules (Chua et al., 2018) to deep learning approaches (Ikeda et al., 2016; Sproat and Jaitly, 2016; Lusetti et al., 2018). Kobus et al. (2008) introduced three metaphors to refer to these normalization approaches: the spell checking, automatic speech recognition and translation metaphors.

In the **spell checking metaphor**, corrections from noisy to standard words occur at the word level. As in conventional spelling correction one has to deal with both non-word and real-word errors (Clark and Araki, 2011). The disadvantage of this approach is that all non-standard words (NSWs) have to be represented in the dictionary in order to obtain the corresponding normalization. Therefore, the success of this kind of systems highly depends on the dictionary coverage. However, as UGC is a very generative language and new variants of canonical words and phrases appear constantly, it is very difficult and expensive to maintain a high coverage lexicon. Other works have approached the problem using a noisy channel model. In this model, the goal is to find the intended word w given a word x where the letters have been changed in some way. Correct words in the text remain untouched. This model is probably the most popular and successful approach to spelling correction (Dutta et al., 2015; Goot, 2015). Although spelling correction is mostly performed on languages which are morphologically simple and with a fairly strict word order, like English, there has been some progress for normalization applied to other languages as well, such as Russian (Sorokin, 2017) and French (Beaufort and Roekhaut, 2010).

Text found in social media shares features with spoken language and the **automatic speech recognition metaphor** exploits this similarity. This approach starts by converting the input message into a phone lattice, which is converted to a word lattice using a phoneme-grapheme dictionary. Finally, the word lattice is decoded by applying a language model to it and using a best-path algorithm to recover the most likely original word sequence. This metaphor has mainly been merged with the machine translation (infra) and spell checking (supra) metaphors to improve the quality of the normalization. Kobus et al. (2008), for example, incorporated ideas from speech recognition to text message normalization and combined it with a machine translation system. Beaufort and Roekhaut (2010); Xue et al. (2011) and Han and Baldwin (2011) also combined the automatic speech recognition approach with spell checking and machine translation techniques.

The **machine translation metaphor** treats social media text as the source language and its normalized form as the target language. Several works have tackled the problem of text normalization using this approach. Statistical Machine Translation (SMT) models, especially those trained at the character-level, have proven highly effective for the task because they capture well intra-word transformations. One of the first works following this approach was presented by Aw et al. (2006). They adapted phrase-based SMT to the task of normalizing English SMS producing messages that collated well with manually normalized ones. Besides, they studied the impact of the normalization on the task of SMS translation, showing that SMS normalization, as a preprocessing step of MT, can boost the translation performance. Kaufmann (2010) used a two-step approach for

Source sentence	Target sentence	Translation
iz da muzieksgool vnavnd ? kwt da niemr .	is dat muziekschool vanavond ? ik weet dat niet meer .	is that music school tonight? I don't know that anymore.
wa is je msn k en e nieuwe msn omda k er nie meer op graal . xxx	wat is je msn ik heb een nieuwe msn omdat ik er niet meer op geraak . xx	what is your msn i have a new msn because i can't get it anymore. xx
@renskedemaessc dm me je gsmnummer eens ;-)	<user> doormail me je gsmnummer eens <emoji>	<user> mail me your cellphone number once <emoji>

Table 1: Source and target pairs as parallel data for a machine translation approach.

the normalization of English tweets: he first preprocessed the tweets to remove as much noise as possible and then used a machine translation approach to convert them into standard English. MT approaches when used at the character level, also have the advantage of being effective when small training data is provided, thanks to their small vocabulary size. De Clercq et al. (2013) proposed a phrase-based method to normalize Dutch UGC comprising various genres. They performed experiments at several levels of granularity: character and word level. In a preprocessing step they handled emoticons, hyperlinks, hashtags and so forth. Then they worked in two steps: first at the word level and then at the character level. This approach revealed good results across various genres of UGC; however a high number of phonetic alternations still remained unresolved. Schulz et al. (2016) made a modification to the previous work by combining the three metaphors (machine translation, spell checking and speech recognition) in a multi-modular system and by using a novel approach for decoding. This led to an improvement in the selection of the best normalization option. Furthermore, they showed a performance improvement of state-of-the-art NLP tools on UGC data when normalization is used as a previous step.

Recently, neural networks have proven to outperform many state-of-the-art systems in several NLP tasks (Young et al., 2018). The encoder-decoder model for recurrent neural networks (RNN) was developed in order to address the sequence-to-sequence nature of machine translation and it obtains good results for this task (Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2014; Luong et al., 2015). The model consists of two neural networks: an encoder and a decoder. The encoder extracts a fixed-length representation from a variable-length input sentence, and the decoder generates a correct translation from this representation. Some works on text normalization have followed the same approach. Ikeda et al. (2016) performed text normalization at the character level for Japanese text and proposed a method for data augmentation using hand-crafted rules. They proved that the use of the synthesised corpus improved the performance of Japanese text normalization. Mandal and Nanmaran (2018) presented an architecture for automatic normalization of phonetically transliterated words to their standard forms in a code-mixed scenario improving the accuracy of a pre-existing sentiment analysis system by 1.5%. Lusetti et al. (2018) performed text normalization over Swiss German WhatsApp messages and compared it to a state-of-the-art SMT system. They showed that integrating language models into an encoder-decoder framework can reach and even improve the performance of character-level SMT methods for that language.

In this work, we also consider the normalization task as a MT problem and test both statistical and neural machine translation. For SMT, we explore the added value of varying background corpora for training the language model. For NMT, we investigate whether we can overcome the limited data set size by using data augmentation.

3 Methodology

Our objective is to go from noisy to standard text and we tackle this normalization problem using a Machine Translation (MT) approach. As in general MT, a translation model is trained on parallel data consisting of pairs (x, y) of source sentences/words (= noisy text) and their corresponding target equivalents (= standard). Table 1 lists some examples of the noisy data we are dealing with.

3.1 Parallel Corpora

We relied on existing Dutch (Schulz et al., 2016) and English (De Clercq et al., 2014) corpora that were manually normalized[1] (Table 2). Three genres were included for both languages:

Tweets (TWE) which were randomly selected for both languages from the social network.

Message board posts (SNS) which were in both languages sampled from the social network Netlog, which was a Belgian social networking website targeted at youngsters.

Text messages (SMS) which were sampled from the Flemish part of the SoNaR corpus (Treurniet et al., 2012) for the Dutch language and from the NUS SMS corpus (Chen and Kan, 2013) for English.

Genre	# Sent.	# Words		%
English		Ori	Tgt	
TWE	810	13477	13545	0.50
SNS	2592	26881	27713	3.00
SMS	1435	20663	22946	**3.94**
Dutch				
TWE	842	13013	13024	0.08
SNS	770	11670	11913	2.04
SMS	801	13063	13610	**4.19**

Table 2: Parallel corpora data statistics in both languages.

Table 2 shows the number of parallel sentences in each corpus and the number of words before and after normalization. Regarding the level of noise, we observe that the text messages required most normalization operations in both languages (an increase of 3.94% for English and one of 4.19% for Dutch). We also notice that the Dutch tweets required hardly any normalization (0.08%). This can be explained by the fact that this platform has been mainly adopted by professionals in Belgium who write in a more formal style (Schulz et al., 2016).

3.2 SMT Approach

The core idea behind SMT relies on the noisy channel model. In this task, two basic components are integrated:

$$\underset{y \in W}{\operatorname{argmax}}\ P(y|x) = \underset{y \in W}{\operatorname{argmax}}\ P(x|y)P(y)$$

The translation model $P(y|x)$ is responsible for the correctness of the translation from the source $x = x_1, x_2, ..., x_m$ to the target sentence $y = y_1, y_2, ..., y_n$. The language model $P(y)$ is responsible for the fluency of the sentence in the target language. W is the set of all target sentences.

To achieve better context-sensitive source-target mappings, traditional SMT systems rely on phrase-level translation models. These models allow to build a phrase table to store aligned phrase pairs in the source and target language. This is a difficult task since one word in one language may correspond to several words in the other language. However when translating from noisy to standard text we can assume that most of the words have a one-to-one mapping. Figure 1 illustrates the architecture of an SMT system.

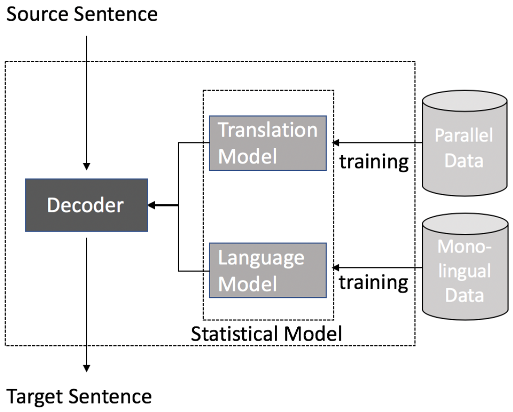

Figure 1: SMT architecture.

For social media translation we suspect that depending on the level of noise of the parallel data, the use of different monolingual corpora for training the language model should lead to better results. Due to the unavailability of a monolingual social media text corpus, we needed to find a resource that somewhat resembles this specific domain. We chose to work with existing corpora comprising two flavors of transcribed speech, namely subtitles and transcriptions of parliamentary debates, because we believe that these can better represent, to some extent, the user-generated content that we can find in social media texts. Table 3 represents the different corpora we used for our experiments. For the English experiments we relied on three different background corpora for constructing our language models: the OpenSubtitles corpus (OPUS) (Tiedemann, 2012b) which is a collection of documents from http://www.opensubtitles.org/; the Europarl corpus (Koehn et al., 2006), extracted from the proceedings of the European Parliament; and the combination of both. A similar approach was

[1] All data was normalized following the procedure described in Schulz et al. (2016)

taken for Dutch, for which we used an in-house subtitles dataset, Europarl, and the combination of both.

Corpus	**Sentences**
English	
OPUS	22,512,649
Europarl	2,005,395
Combined	OPUS+Europarl
Dutch	
Subtitles	8,056,693
Europarl	2,000,113
Combined	OPUS+Europarl

Table 3: Size (expressed in sentences) of the monolingual corpora used for training our LMs.

3.3 NMT Approach

Neural Machine Translation incorporates the advantages of newly developed deep learning approaches into the task. Sequence-to-Sequence (seq2seq) models have been used for a variety of NLP tasks including machine translation obtaining state-of-the-art results (Luong et al., 2015; Young et al., 2018). In this approach both input and output sentences are going in and out of the model. As described in the literature overview, the model consists of two neural networks: an encoder and decoder (See Figure 2). The encoder extracts a fixed-length representation from a variable-length input sentence (*A B C D*), and the decoder generates a correct translation from this representation (*X Y Z*). In the figure <eos> marks the end of a sentence. The encoder-decoder model is trained on a parallel corpus consisting of aligned source sentences and their normalized forms (see Table 1).

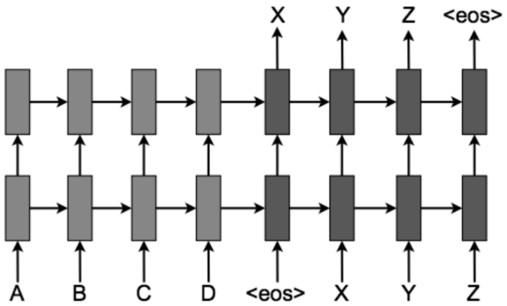

Figure 2: Encoder-decoder architecture. The light-color nodes represent the encoder and the dark-color ones the decoder. Image taken from Luong et al. (2015).

Neural systems, however, require huge amounts of data in order to perform properly. The training data we have available for text normalization amounts to only a few hundred sentences, as can be derived from Table 2. Moreover, manually annotating more training is highly time-consuming. Under these conditions, we decided to experimentally verify whether it is more beneficial to use a data augmentation technique (step A) which possibly resolves the data scarcity problem (Saito et al., 2017) or to annotate more data (step B). We tested this on the Dutch corpus and one particular genre, namely text messages (SMS). For step A, we augmented the parallel data by duplicating monolingual subtitles data on both the source and target side. For step B, we sampled and manually annotated ten thousand extra tokens from the Flemish part of the SoNaR corpus (Treurniet et al., 2012)[2].

We relied on OpenNMT[3] to train our encoder-decoder model. OpenNMT is an open source (MIT) initiative for neural machine translation and neural sequence modeling (Klein et al., 2017). The main system is implemented in the Lua/Torch mathematical framework, and can easily be extended using Torch's internal standard neural network components. We used the version of the system with the basic architecture which consists of an encoder using a simple LSTM recurrent neural network. The decoder applies attention over the source sequence and implements input feeding (Luong et al., 2015).

3.4 Evaluation

For evaluating the results of the normalization we calculated Word Error Rate (WER), a commonly used machine translation evaluation metric. WER is derived from the Levenshtein distance (Levenshtein, 1966), working at the word level instead of the character level. It takes into account the number of insertions (INS), deletions (DEL) and substitutions (SUBS) that are needed to transform the suggested string into the manually normalized string. The metric is computed as follows:

$$WER = \frac{INS + DEL + SUBS}{N}$$

where N in the number of words in the reference.

Table 4 reports WER computed between the original and target parallel sentence pairs that were

[2] Following the same annotation guidelines as Schulz et al. (2016)

[3] http://opennmt.net

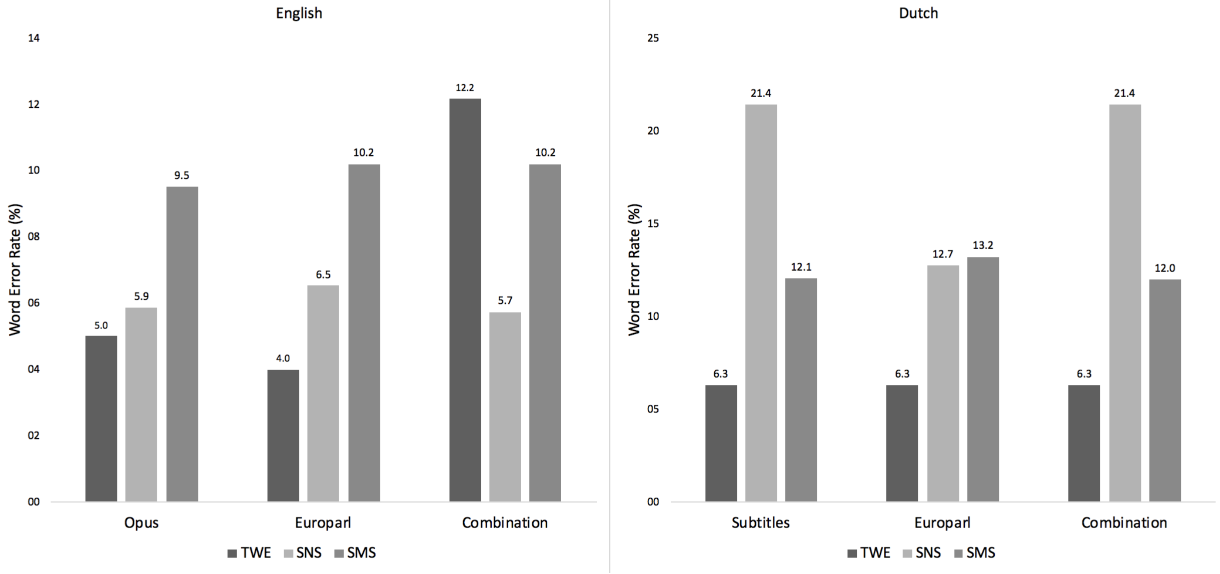

Figure 3: Normalization results at the token level. The left chart presents the results on the English datasets and the right one the results on the Dutch dataset.

used for training our models.

Word Error Rate (%)		
Genre	English	Dutch
TWE	12.160	10.592
SNS	15.400	21.390
SMS	17.190	25.130

Table 4: WER values (in percentage) at the sentence level

WER values were calculated per sentence and averaged within the document. The higher the value, the more operations are needed to obtain the target sentence. Looking at the values, we again notice that genres requiring the most and least normalization are the text messages (SMS) and tweets (TWE), respectively.

4 Experiments

4.1 Varying Background Corpora for SMT

With the first round of experiments we want to research the influence of varying the monolingual data that are used to construct the language models. We trained LMs at the character level using unigrams and bigrams and at the token level. All LMs were built using the SRILM toolkit (Stolcke, 2002) with Witten-Bell discounting which has proven to work well on small data sets (Tiedemann, 2012a; Mahmudul Hasan et al., 2012; De Clercq et al., 2013). To evaluate the performance of each LM, Word Error Rate (WER) was calculated.

The parallel data (Table 2) used for training the translation model were divided using 80% for training the model and 10% for development and testing, respectively. The target sentences from the training set were also added to the monolingual corpus for training the language model.

Despite several works reporting better results when using a character-level approach (De Clercq et al., 2014; Lusetti et al., 2018) our experiments revealed the best performance with SMT at the token level. The bar charts in Figure 3 present the results of SMT at the token level with the different LMs.

Regarding the monolingual background corpora, we notice that Europarl leads to the best results for the tweets (TWE) genre which was actually the genre with the least noise (see Section 3.1). Our experiment shows WERs of 4% and 6.3% for English and Dutch, respectively. This result was to be expected as the word usage in Europarl is mostly standard and therefore close to the word usage in the tweets. The same is true for the genre comprising the most noise, i.e text messages. The word usage in the Subtitles/OPUS dataset is less standard and closer to spoken language and, indeed, also in this case we obtained a WER of 9.5% for English using OPUS, and a WER of 12% for Dutch using a combination of the Subtitles dataset and Europarl.

In addition, we also computed the number of insertions (INS), deletions (DEL) and substitutions

(SUBS) in the original sentence pairs (Ori) and after normalization (Norm) (Table 5).

English						
	INS		DEL		SUBS	
Genre	Ori	Norm	Ori	Norm	Ori	Norm
TWE	91	36	15	**30**	34	22
SNS	354	64	66	43	109	62
SMS	377	63	44	**57**	135	54
Dutch						
TWE	22	19	0	**5**	39	**45**
SNS	386	159	18	**36**	76	47
SMS	0	0	2	**3**	94	85

Table 5: Number of operations required before (Ori) and after (Norm) normalization.

Ideally, the number of operations after normalization should be reduced to zero. As can be derived from the table many operations were strongly reduced; however, many cases still need to be solved. We will have a closer look at some of these cases in the next section.

4.2 Error Analysis of the SMT Results

The number of remaining insertions is mostly linked to the problem of abbreviation expansions. Very common abbreviations like *lol* or *omg* are always corrected, whereas others like *r.e.* and *p.e.* for *religious* and *physical education* or *cum on den* for *come on then* are not corrected since they never appear in the training data.

When we consider the deletions, we can observe that flooding or repetition of characters is often not solved with SMT. For example, tokens like *okkk*, *awwwww* or sentences like *immaaa dooiin fiiine !* remained unchanged. A straightforward way to overcome this problem could be to reduce the number of repetitions to two or three in some cases as a pre-normalization step. The second factor that affects the number of deletions is the hypernormalization of some words. This leads to an increase in the number of operations since sometimes we will have to perform more deletion operations on the predicted sentences than on the original ones (these instances are indicated in bold in Table 5). This is for example the case with the name *al* which was incorrectly normalized to *all* or the normalization of *i can 't really think...* into *i can not really not think*. These problems also affect the number of substitutions. In general, we also notice that the normalization of Dutch presents a higher number of errors.

4.3 NMT Approach to UGC Normalization

As we explained before, neural approaches have obtained state-of-the-art results for the task of Machine Translation. Neural approaches however, are well-known to require big amounts of parallel data in order to perform properly. Especially for Dutch, which can be considered a low-resource language, it is difficult to find freely available parallel data and annotating new data is both money and time-consuming.

Under these conditions we decided to experimentally verify whether it is more beneficial to use a data augmentation technique (step A) which possibly resolves the data scarcity problem (Saito et al., 2017) or annotate more data (step B). We tested this on the Dutch corpus and one particular genre, the most noisy one, namely text messages (SMS).

For Step A, our idea was to make use of the monolingual subtitles corpus in both sides of the parallel data in order to augment the number of sentences available for training our model. Doing this, we obtain a bigger dataset consisting of the Dutch SMS parallel corpus and the Dutch Subtitles dataset which is duplicated in the source and target data. That is a total of 8,057,334 parallel sentences for training.

src sent.	wa gaat je doen ? xxx
norm sent.	wat gaat je doen ? xxx
src sent.	oeiiii misterieus <emoji> xxx
norm sent.	oeiiii misterieus <emoji> xxx
src sent.	dne dvd vn is ni goe ze . ge kunt nx zien . mt betale . x
norm sent.	het dvd hem is niet niet het maar wat wat in ik . niet betale . x x x x x

Table 6: Original (src) and predicted (norm) sentences using the NMT approach.

Unfortunately, as can be derived from the table above, results following this approach are very poor. It is common in the output to find repetition of words like in the third sentence (*niet, wat* and *x*). Besides, some sentences that needed normalization like the second sentence in the table, were not normalized at all. For example, the words *wat* and *niet* in the first and last sentence respectively, were correctly normalized. These results may have been determined to a great extent by the unbalance in the parallel sentences. However, we could see a slight improvement compared to the results using only the small parallel data in this architecture. For that case, the system output consisted of sentences of the type <*emoji*> , *de* , , , . . . <*emoji*>. These are random repetitions of the most represented tokens like *ik* (*I* in English), punctuation marks or <*emoji*> labels.

In order to corroborate our other hypothesis, we collected and manually normalized more data for step B. In order to check how this system works at different levels of granularity, we also performed experiments using bigram and unigrams of characters. Regarding the results, also for NMT the results are better at the word level than at the character level, with WERs of 15% instead of 29% and 26% for bigram and unigram, respectively.

4.4 Error Analysis of the NMT Results

Using the new data configuration the system is capable to correctly translate sentences like the first one in Table 7.

src sent.	aahn , ok ma cva dan kzal dan wel wa zoeken xp merci eh x
norm sent.	ah , oké maar ça va dan ik zal dan wel wat zoeken <emoji> merci h x
tgt sent.	ah , oké maar ça va dan ik zal dan wel wat zoeken <emoji> merci h x
src sent.	zal dan eentje **v** mezelf sturen . zorgen we morgenavond dan voor verrassing **v** kareltje ?
norm sent.	zal dan eentje van mag sturen . zorgen we morgenavond dan voor cocktail van droomt ?
tgt sent.	zal dan eentje van mezelf sturen . zorgen we morgenavond dan voor verrassing voor kareltje ?

Table 7: Original (src), predicted (norm) and target (tgt) sentences using the NMT approach trained on the extended dataset.

However, the system still produces a large number of odd normalizations. In the second sentence in Table 7, for example, only the bold words should have been normalized. However, only one of those two words was correctly normalized, the other one was normalized but not into its correct form. On the other hand, the system also produces odd translations of words that already were in their standard form. For example the word *mezelf* is changed to *mag* and we got *cocktail van droomt* instead of the desired normalization, ie. *verrassing voor kareltje*.

In general, while the results using this approach are very poor, the experiments revealed that having a bigger parallel training corpus could improve the performance of this system.

5 Conclusions and Future Work

In this article, we have presented two different approaches to text normalization of social media text: statistical and neural machine translation. We applied text normalization to English and Dutch text from different genres: text messages, message board posts and tweets. Best results were achieved at the token level for all genres and for both SMT and NMT.

For the SMT experiments, regarding the different corpora that were used to construct the LM, we found that Europarl gave the best results for the least noisy genre (tweets). The same is true for the noisiest genre (text messages). Considering our results, it seems to be important to make variations in the background data for building the LM, depending on the amount of noise and vocabulary that is present in the genre. With respect to the remaining errors we believe that following a modular approach instead of only using SMT could lead to a much better performance.

Our NMT approach performs poorly due to the scarcity of the data, although we did find that for a low-resource language like Dutch adding additional training data works better than artificially augmenting the data. The data augmentation technique used, however, was very basic and we believe that other techniques could lead to better results, such as hand-crafted rules for the production of abbreviations or the use of previously trained embedding in order to build similar sentences helping to generalize better.

Exploring those other data augmentation techniques is a first avenue for future work. Besides we also want to test the benefits of the integration of a neural LM in the encoder-decoder model to help with the translation of out-of-vocabulary words.

References

AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for SMS text normalization. *Proceedings of the COLING/ACL on Main conference poster sessions* - (July):33–40. https://doi.org/10.1109/TCST.2005.854339.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint* pages 1–15. http://arxiv.org/abs/1409.0473.

Richard Beaufort and Sophie Roekhaut. 2010. A hybrid rule/model-based finite-state framework for normalizing SMS messages. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics* 1(July):770–779. https://doi.org/10.1097/ICO.0b013e318245c02a.

Su Lin Blodgett, Green Lisa, and OĆonnor Brendan. 2016. Demographic Dialectal Variation in Social Media: A Case Study of African-American English. *arXiv preprint* .

Tao Chen and Min Yen Kan. 2013. *Creating a live, public short message service corpus: The NUS SMS corpus*, volume 47. https://doi.org/10.1007/s10579-012-9197-9.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *International Conference on Learning Representations ICLR* http://arxiv.org/abs/1406.1078.

Mason Chua, Daan Van Esch, Noah Coccaro, Eunjoon Cho, Sujeet Bhandari, and Libin Jia. 2018. Text Normalization Infrastructure that Scales to Hundreds of Language Varieties. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*. European Language Resource Association, Miyazaki, Japan, pages 1353–1356. http://www.lrec-conf.org/proceedings/lrec2018/pdf/8883.pdf.

Eleanor Clark and Kenji Araki. 2011. Text normalization in social media: Progress, problems and applications for a pre-processing system of casual English. *Procedia - Social and Behavioral Sciences* 27(Pacling):2–11. https://doi.org/10.1016/j.sbspro.2011.10.577.

Orphée De Clercq, Sarah Schulz, Bart Desmet, and Véronique Hoste. 2014. Towards Shared Datasets for Normalization Research. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* pages 1218–1223.

Orphée De Clercq, Sarah Schulz, Bart Desmet, Els Lefever, and Véronique Hoste. 2013. Normalization of Dutch User-Generated Content. *Proceedings of the 9th International Conference on Recent Advances in Natural Language Processing (RANLP 2013)* pages 179–188.

Sukanya Dutta, Tista Saha, Somnath Banerjee, and Sudip Kumar Naskar. 2015. Text Normalization in Social Media. In *2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS)*. IEEE, Kolkata, India, c, pages 378–382. https://doi.org/0.1109/ReTIS.2015.7232908.

Jacob Eisenstein. 2013. What to do about bad language on the internet. *NAACL HLT 2013 - 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Main Conference* (June):359–369.

Rob Van Der Goot. 2015. Normalizing Social Media Texts by Combining Word Embeddings and Edit Distances in a Random Forest Regressor. In *Normalisation and Analysis of Social Media Texts (NormSoMe)*. 1.

Bo Han and Timothy Baldwin. 2011. Lexical Normalisation of Short Text Messages : Makn Sens a #twitter. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics* pages 368–378.

Taishi Ikeda, Hiroyuki Shindo, and Yuji Matsumoto. 2016. Japanese Text Normalization with Encoder-Decoder Model. *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)* pages 129–137.

Max Kaufmann. 2010. Syntactic Normalization of Twitter Messages. *International conference on natural language processing* 2:1–7.

Jinyun Ke, Tao Gong, and William S-y Wang. 2008. Language Change and Social Networks. *Communications in Computational Physics* 3(4):935–949.

Guillaume Klein, Yoon Kim, Yuntian Deng, Josep Crego, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source Toolkit for Neural Machine Translation. *arXiv preprint* http://arxiv.org/abs/1709.03815.

Catherine Kobus, Francois Yvon, and Geéraldine Damnati. 2008. Normalizing SMS: are two metaphors better than one? *Proceedings of the 22nd International Conference on Computational Linguistics* 1(August):441–448. https://doi.org/10.14288/1.0064682.

Philipp Koehn, Wade Shen, Marcello Federico, Nicola Bertoldi, Chris Callison-Burch, Brooke Cowan, Chris Dyer, Hieu Hoang, Ondrej Bojar, Richard Zens, Alexandra Constantin, Evan Herbst, and Christine Moran. 2006. Moses: Open Source Toolkit for Statistical Machine Translation. *Proceedings of ACL* (June):177–180. https://doi.org/10.3115/1557769.1557821.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions. *Soviet physics doklady* 10(8):707–710.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. *arXiv preprint* http://arxiv.org/abs/1508.04025.

Massimo Lusetti, Tatyana Ruzsics, Anne Göhring, Tanja Samardi Samardžic, and Elisabeth Stark. 2018. Encoder-Decoder Methods for Text Normalization. In *Proceedings of the Fifth*

Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial). pages 18–28. http://www.aclweb.org/anthology/W18-3902.

A. S. M. Mahmudul Hasan, Saria Islam, and M. Mahmudul Rahman. 2012. A Comparative Study of Witten Bell and Kneser-Ney Smoothing Methods for Statistical Machine Translation. *Journal of Information Technology (JIT)* 1(June):1–6.

Soumil Mandal and Karthick Nanmaran. 2018. Normalization of Transliterated Words in Code-Mixed Data Using Seq2Seq Model & Levenshtein Distance. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*. Association for Computational Linguistics, Brussels, Belgium, pages 49–53. https://en.wikipedia.org/wiki/ISO http://arxiv.org/abs/1805.08701.

Itsumi Saito, Jun Suzuki, Kyosuke Nishida, and Kugatsu Sadamitsu. 2017. Improving Neural Text Normalization with Data Augmentation at Character- and Morphological Levels. *Proceedings of the The 8th International Joint Conference on Natural Language Processing* pages 257–262. http://aclweb.org/anthology/I17-2044.

Sarah Schulz, Guy De Pauw, Orphée De Clercq, Bart Desmet, Véronique Hoste, Walter Daelemans, and Lieve Macken. 2016. Multimodular Text Normalization of Dutch User-Generated Content. *ACM Transactions on Intelligent Systems and Technology* 7(4):1–22. https://doi.org/10.1145/2850422.

Andrew H. Schwartz, Johannes C. Eichstaedt, Margaret L. Kern, Lukasz Dziurzynski, Stephanie M. Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin E. P. Seligman, and Lyle H. Ungar. 2013. Personality, Gender, and Age in the Language of Social Media : The Open-Vocabulary Approach. *PLoS ONE* 8(9). https://doi.org/10.1371/journal.pone.0073791.

Alexey Sorokin. 2017. Spelling Correction for Morphologically Rich Language: a Case Study of Russian. *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing* (April):45–53. https://doi.org/10.18653/v1/w17-1408.

Richard Sproat, Alan W. Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech and Language* 15(3):287–333. https://doi.org/10.1006/csla.2001.0169.

Richard Sproat and Navdeep Jaitly. 2016. RNN Approaches to Text Normalization: A Challenge. *Computing Research Repository (CoRR)* abs/1611.0. http://arxiv.org/abs/1611.00068.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *7th International Conference on Spoken Language Processing.*. Denver, Colorado, pages 901–904. https://doi.org/10.1.1.157.2429.

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. http://arxiv.org/abs/1409.3215.

Jörg Tiedemann. 2012a. Character-Based Pivot Translation for Under-Resourced Languages and Domains. *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* pages 141–151.

Jörg Tiedemann. 2012b. Parallel Data, Tools and Interfaces in OPUS. *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)* pages 2214–2218.

Maaske Treurniet, Henk van den Heuvel, Nelleke Oostdijk, and Orphée De Clercq. 2012. Collection of a corpus of Dutch SMS. *Proceedings of the Eight Conference of International Language Resources and Evaluation.* pages 2268–2273.

Cynthia Van Hee, Marjan Van De Kauter, Orphe De Clercq, Els Lefever, Bart Desmet, and Vronique Hoste. 2017. Noise or music? Investigating the usefulness of normalisation for robust sentiment analysis on social media data. *Revue Traitement Automatique des Langues* 58(1):63–87.

Reinhild Vandekerckhove and Judith Nobels. 2010. Code eclecticism : Linguistic variation and code alternation in the chat language of Flemish teenagers. *Journal of Sociolinguistics* 14(5):657–677.

Zhenzhen Xue, Dawei Yin, and Bd Davison. 2011. Normalizing Microtext. *Analyzing Microtext* pages 74–79.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine* 13(3):55–75. https://ieeexplore.ieee.org/abstract/document/8416973/.