Naive Regularizers for Low-Resource Neural Machine Translation

Meriem Beloucif[1], **Ana Valeria Gonzalez**[2], **Marcel Bollmann**[2], **and Anders Søgaard**[2]

[1]Language Technology Group, Universität Hamburg, Hamburg, Germany
[2]Dpt. of Computer Science, University of Copenhagen, Copenhagen, Denmark

Abstract

Neural machine translation models have little inductive bias, which can be a disadvantage in low-resource scenarios. They require large volumes of data and often perform poorly when limited data is available. We show that using naive regularization methods, based on sentence length, punctuation and word frequencies, to penalize translations that are very different from the input sentences, consistently improves the translation quality across multiple low-resource languages. We experiment with 12 language pairs, varying the training data size between 17k to 230k sentence pairs. Our best regularizer achieves an average increase of 1.5 BLEU score and 1.0 TER score across all the language pairs. For example, we achieve a BLEU score of 26.70 on the IWSLT15 English–Vietnamese translation task simply by using relative differences in punctuation as a regularizer.

1 Introduction

One of the major challenges when training neural networks is overfitting. Overfitting is what happens when a neural network in part memorizes the training data rather than learning to generalize from it. To prevent this, neural machine translation (NMT) models are typically trained with an L_1 or L_2 penalty, dropout, momentum, or other general-purpose regularizers. General-purpose regularizers and large volumes of training data have enabled us to train flexible, expressive neural machine translation architectures that have provided a new state of the art in machine translation.

For low-resource language pairs, however, where large volumes of training data are *not* available, neural machine translation has come with diminishing returns (Koehn and Knowles, 2017). The general-purpose regularizers do not provide enough inductive bias to enable generalization, it seems. This is an area of active research, and other work has explored multi-task learning (Firat et al., 2016; Dong et al., 2015), zero-shot learning (Johnson et al., 2016), and unsupervised machine translation (Gehring et al., 2017) to resolve the data bottleneck. In this paper, we consider a fully complementary, but much simpler alternative: naive, linguistically motivated regularizers that penalize the output sentences of translation models departing heavily from simple characteristics of the input sentences.

The proposed regularizers are based on three surface properties of sentences: their length (measured as number of tokens), their amount of punctuation (measured as number of punctuation signs), and the frequencies of their words (as measured on external corpora). While there are languages that do not make use of punctuation (e.g., Lao and Thai), in general these three properties are roughly preserved across translations into most languages. If we translate a sentence such as (1), for example:

(1) That dog is a Chinook.

it is relatively safe to assume that a good translation will be short, contain at most one dot, and contain at least one relatively frequent word (for *dog*) and at least one relatively infrequent word (for *Chinook*). This assumption is the main motivation for our work.

Contributions Our contribution is three-fold: (a) We propose three relatively naïve, yet linguistically motivated, regularization methods for machine translation with low-resource languages.

Two of the regularizers are derived directly from the input, without relying on any additional linguistic resources. This makes them adequate for low-resource settings, where the availability of linguistic resources can generally not assumed. Our third regularizer (frequency) only assumes access to unlabeled data. (b) We show that regularizing a standard NMT architecture using naive regularization methods consistently improves machine translation quality across multiple low-resource languages, also compared to using more standard methods such as dropout. We also show that combining these regularizers leads to further improvements. (c) Finally, we present examples and analysis showing *how* the more linguistically motivated regularizers we propose, help low-resource machine translation.

2 Related Work

End-to-end neural machine translation is based on encoder–decoder architectures (Sutskever et al., 2014; Cho et al., 2014; Luong et al., 2015a, 2017), in which a source sentence $x = (x_1, x_2, ..., x_n)$ is encoded into a vector (or a weighted average over a sequence of vectors) $z = (z_1, z_2, ..., z_n)$. The hidden state representing z is then fed to the transducer (also called decoder) which generates translations, noted as $y = (y_1, y_2, ..., y_m)$.

Neural machine translation has achieved state-of-the-art performance for various language pairs (Luong et al., 2015a; Sennrich et al., 2015; Luong and Manning, 2016; Neubig, 2015; Vaswani et al., 2017), especially when trained on large volumes of parallel data, i.e., millions of parallel sentences (also called bi-sentences), humanly translated or validated. Such amounts of training data, however, are difficult to obtain for low-resource languages such as Slovene or Vietnamese, and in their absence, neural machine translation is known to come with diminishing returns, suffering from overfitting (Koehn and Knowles, 2017).

In order to avoid overfitting, NMT models are often trained with L_1 or L_2 regularization, as well as other forms of regularization such as momentum training or dropout (Srivastava et al., 2014; Wang et al., 2015; Miceli Barone et al., 2017). However, these regularization methods are very general and do not carry any language specific information.

On the other hand, it has been shown that transfer learning approaches using out of domain data, such as the European Parliament data[1], to regularize the learning helps improve the translation quality (Miceli Barone et al., 2017). This approach produces good results, but it is not applicable in low-resource settings because it requires large amounts of data in the language of interest. To the best of our knowledge, our work is the first to introduce naive, linguistically motivated regularization methods such as sentence length, punctuation and word frequency.

3 Model Description

3.1 Baseline

In order to show the impact that our regularizers have on the translation quality, we use an off-the-shelf NMT system described by Luong et al. (2017) as our baseline. The model consists of two multi-layer recurrent neural networks (RNNs), one that encodes the source language and one that decodes onto the target language. For the encoder cell, we use a single Long Short-Term Memory (LSTM) layer (Hochreiter and Schmidhuber, 1997) and output the hidden state, which then gets passed to the decoder cell.

We train our models to minimize the cross-entropy loss and back-propagate the loss to update the parameters of our model. We update network weights using Adam optimization (Kingma and Ba, 2014), which calculates the exponential moving average of the gradient and squared gradient, and combines the advantages of AdaGrad and RMSProp. For the purpose of comparison, we set the dropout to 0.2, similar to Luong et al. (2015b).

3.2 Regularized NMT

To apply our new regularizers, we add each regularizer to the loss function during the training of the NMT model (Luong et al., 2015a; Luong and Manning, 2016; Luong et al., 2017). Since we aim to minimize the cross-entropy loss, this means that we favor training instances which have a low penalty from the regularizers (e.g., a small length difference). Importantly, we do *not* use dropout in this scenario, as we want to contrast our naive, but linguistically motivated signals with a traditional, but not linguistically motivated regularization method, i.e., dropout.

Furthermore, we do not explore alternative ways for adding regularizers to the loss function here (other alternatives could be to have a

[1] http://www.statmt.org/europarl/

weighted penalty which is then tuned to find the best penalty and added to the loss function for testing). The main purpose of this work is to study the effect of naive linguistically motivated regularizers and show that they can improve translation quality; we leave it to future work to find the optimal configuration of regularizers that maximizes the overall translation quality.

4 Naive Regularizers

4.1 Length-Based Regularizer

NMT models have shown to suffer "the curse of sentence length", and it has been hypothesized that this is due to a lack of representation at the decoder level (Cho et al., 2014; Pouget-Abadie et al., 2014). Our proposed sentence-length-based regularizer penalizes relative differences between the input and the MT output lengths during the training of the NMT model:

$$\text{reg}_{\text{length}} = |l_0 - l_1| \quad (1)$$

Here, l_0 and l_1 represent the input sentence and the MT output sentence lengths, respectively, as measured by the number of words (not to be confused with L_1 and L_2 regularization methods).

Note that this regularizer is different from the word penalty feature in phrase-based machine translation (Zens and Ney, 2004), which only penalizes the target sentence length. The relative difference between the input and the MT output sentence lengths is also used as a feature in Marie and Fujita (2018).

4.2 Punctuation-Based Regularizer

The punctuation-based regularizer penalizes training instances whenever the amount of punctuation marks in the input sentence differs from the amount in the MT output sentence. It is computed as follows:

$$\text{reg}_{\text{punct}} = |p_0 - p_1| \quad (2)$$

Here, p_0 and p_1 is the total number of punctuation marks in the input and the MT output sentence, respectively.

Unfortunately, the only available methods to generate more efficient NMT models have included data intensive methods such as sentence alignment (Bahdanau et al., 2014). Some very early research done in alignment used simple methodologies such as punctuation-based alignment (Chuang et al., 2004). Our second regularizer is based on this simple idea, as it penalizes training instances where the quantities of punctuation marks differ between input and MT output sentences. Example (2) is taken from the training set of the French–English translation task:

(2) IN *Pas parce qu'ils sont moins bons, pas parce qu'ils sont moins travailleurs.*
REF And it's not because they're less smart, and it's not because they're less diligent.
OUT And . . . . . .

We note that the punctuation in the French input sentence matches the punctuation of the desired English reference. However, during an early training step, the NMT model translates the input to a sequence containing six times the number of punctuation marks in the input sentence, which is obviously incorrect. Our punctuation regularizer further penalizes examples like this one.

4.3 Frequency-Based Regularizer

Our last regularizer is based on the distribution of word frequencies between the source and the target sentences. Generally speaking, if the source sentence contains an uncommon word, we assume that its translation in the target language is also uncommon. The intuition behind this regularizer is that if the source sentence contains one uncommon word and three common words, then its accurate translation should contain similar word frequencies. The example below is extracted from the English–French translation task:

(3) IN *But now there is a bold new solution to get us out of this mess.*
REF Mais il exist une solution audacieuse pour nous en sortir.
OUT Mais maintenant il y a une solution pour nous en sortir.

The English sentence contains the frequent word *there* and the less frequent word *bold*. The French output sentence is acceptable, but it is not accurate since the English word *bold* (*audacieuse* in the reference translation) was omitted in the output. During training, the frequency regularizer penalizes such cases that have a big divergence between the word frequencies in the input and output sentences.

The purpose of our frequency-based regularizer

Languages	#Words
Czech	1.7M
English	85.57M
French	55.72M
German	35.47M
Russian	2.5M
Slovene	1.45M
Vietnamese	3.5M

Table 1: The size of the Wikipedia dumps (#words) used to calculate word frequencies for each language.

is to calculate how different the MT output sentence is from the source input in terms of vocabulary distribution. For instance, the frequency of using the word *chauve-souris* in French is almost similar to the frequency of using its English translation *bat* in English. The same could be applied for the more frequent words such as *et* in French and its English translation *and*.

We start by computing the frequency vectors $\overrightarrow{v_{\text{in}}}$ and $\overrightarrow{v_{\text{out}}}$, containing the frequency for every word w_i in the input and MT output sentence, respectively:

$$\overrightarrow{v} = \langle f(w_1), \ldots, f(w_n) \rangle \tag{3}$$

To calculate the word frequencies $f(w)$ for each language, we use the Wikipedia database[2] as an external resource. Table 1 contains the size of the datasets (in number of words) used to estimate these. We note that there is considerably more data for English and French than for e.g. Vietnamese (cf. Table. 1); we discuss the effect that this might have on the results in Sec. 6.

We interpret the resulting frequency vectors $\overrightarrow{v}$ as distributions, for which we now calculate the Kullback-Leibler (KL) divergence to obtain our regularization term:

$$\text{reg}_{\text{freq}} = D_{\text{KL}}(\overrightarrow{v_{\text{in}}}, \overrightarrow{v_{\text{out}}}) \tag{4}$$

Essentially, this regularizer penalizes translations if their word frequency distributions diverge too strongly from those of the source sentence.

(4) IN It was a big lady who wore a fur around her neck
REF C'était une dame forte qui portait une fourrure autour du cou
OUT C'était une femme forte portant une fourrure autour du cou

[2]https://en.wikipedia.org/wiki/Wikipedia:Database

Languages	Sentence Pairs		
	Train	Development	Test
Czech	122,382	480	1,327
French	232,825	890	1,210
German	206,112	888	1,080
Russian	178,165	887	1,701
Slovene	17,125	1,144	1,411
Vietnamese	133,317	1,553	1,268

Table 2: The size of the training data in sentence pairs. To test our proposed models, we experiment by translating to/from English for every non-English language.

Example (4) shows an input sentence and its MT output, for which we would compute the frequency vectors as follows:

$$\overrightarrow{v_{\text{in}}} = \langle f(\text{'it'}), \quad f(\text{'was'}), \quad \ldots, f(\text{'neck'}) \rangle$$
$$\overrightarrow{v_{\text{out}}} = \langle f(\text{'c'était'}), \quad f(\text{'une'}), \quad \ldots, f(\text{'cou'}) \rangle$$

5 Experiments

5.1 Data

The purpose of our experiments is to show that signals such as sentence length, punctuation or word frequency help improve the translation quality of a standard neural machine translation architecture. To that effect, we experiment with 12 translation tasks, translating from English to six low-resource languages, and vice versa.

The six languages represent the following language families: Slavic, Romance, Germanic, and Austro-Asian. We further vary the size of the training data to test how our regularization methods affect the quality of the MT output in different setups. Table 2 contains the size of the training, development and test set for every language pair. Note that the training sets vary considerably in size, from 17k sentence pairs for Slovene to almost 233k for French.

The data is from the International Workshop on Spoken Language Translation (IWSLT), except for Russian, Slovene and Vietnamese which are from IWSLT 2015, the data for the remaining translation tasks is from IWSLT 2017 (Cettolo et al., 2012).

Preprocessing The purpose of our experiments is to learn how to efficiently translate low-resource languages. For that purpose, we do not use any advanced preprocessing for any of our translation tasks except tokenization where we use the script from the Moses toolkit (Koehn et al., 2007). We also set the maximum sentence length to 70 tokens and the vocabulary size to 50k.

5.2 Training Details

We use the attention-based model described in Luong et al. (2015b). Our model is composed of two LSTM layers each of which has 512-dimensional units and embeddings; we also use a mini-batch size of 128. Adding an attention mechanism in neural machine translation helps to encode relevant parts of the source sentence when learning the model. We propose to add additional regularizers on top of the attention-based model at each translation step.

We have noticed that the convergence highly depends on the language pairs involved. While our baseline model is identical to the NMT model described by Luong et al. (2015b), we deviate from their training procedure by continuing the training until convergence, which for us took 15 epochs instead of the 12 epochs described by the authors. The convergence in our case is measured by the models having no improvements on the development set over five epochs.

Table 3 shows that our baseline is +1.5 BLEU points better than the scores reported by Luong et al. (2015b). On top of that, our length-based and punctuation-based models produce a statistically significant improvement over the baseline (+0.5 BLEU points).

We train all our models automatically until convergence. In Table 4, we report the number of epochs it took to converge by translation task when translating to/from English. We note that except for Czech and Slovene, which converged the quickest, most of the translation tasks took between 15k and 20k steps to converge.

System	BLEU
Luong et al. (2015)	23.30
Luong et al. (2017) (dropout=0.2)	25.10
Baseline (dropout=0.2)	26.43
+ Length	**26.77**
+ Punct	26.71
+ Frequency	26.12
+ Combined	**27.13**

Table 3: Baseline vs. our proposed models on the English–Vietnamese translation task, using the same dataset as Luong et al. (2015b). The results in bold represent statistically significant results compared to the baseline according to MultEval (Clark et al., 2011).

Translation Task	#Steps
Lang→English	
Czech	12K
French	20K
German	20K
Russian	22K
Slovene	10K
Vietnamese	15K
English→Lang	
Czech	12K
French	22K
German	20K
Russian	18K
Slovene	11K
Vietnamese	15K

Table 4: Number of steps it took until the models stopped improving for all the translation tasks.

6 Evaluation

In order to show that the naive regularizers which we propose in this paper significantly boost the translation quality, we test the machine translation output using the toolkit MultEval defined in Clark et al. (2011). In this paper, we report the results using three commonly used metrics: the *n*-gram based metrics BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005), as well as the error-rate based metric TER (Snover et al., 2006). The evaluation metric BLEU (Papineni et al., 2002) is based on *n*-gram matching between the input and the output, whereas the error-rate based metric TER (Snover et al., 2006) measures how many edits are needed so that the machine translation resembles the man-made reference.

6.1 Results

Table 5 shows the results for all language pairs and all metrics. We observe an improvement over the

	System	Languages					
		Czech	French	German	Russian	Slovene	Vietnamese
EN→Lang	Baseline	14.01	32.13	22.07	12.87	5.60	26.43
	Length	**14.65**	32.32	21.64	12.81	4.98	**26.77**
	Punct	**14.98**	**32.79**	**22.89**	13.06	5.64	26.71
	Frequency	**14.75**	**33.47**	22.14	**13.50**	1.95	26.12
Lang→EN	Baseline	21.32	31.51	24.41	15.39	8.85	24.94
	Length	21.83	31.09	24.56	15.29	9.05	**25.87**
	Punct	**21.96**	**32.43**	**25.17**	**16.36**	**9.63**	25.32
	Frequency	21.88	**32.26**	24.87	15.90	9.18	24.35

(a) BLEU

EN→Lang	Baseline	17.62	51.11	40.47	16.12	26.52	11.46
	Length	**18.41**	51.10	39.93	**16.80**	**27.03**	12.01
	Punct	**18.43**	**51.67**	**41.18**	16.77	27.00	**12.30**
	Frequency	**18.16**	**52.10**	40.57	16.79	26.95	12.29
Lang→EN	Baseline	24.66	31.77	27.23	20.63	16.28	28.11
	Length	**25.07**	31.55	27.11	20.65	15.95	**28.71**
	Punct	**25.10**	**32.31**	**27.75**	**21.45**	**17.05**	**28.48**
	Frequency	**25.27**	**32.16**	27.43	20.80	**16.85**	27.86

(b) METEOR

EN→Lang	Baseline	62.64	49.21	57.17	70.17	77.20	54.29
	Length	62.18	48.96	57.90	70.85	79.51	**53.93**
	Punct	**61.69**	**48.57**	57.24	70.04	**77.02**	54.03
	Frequency	62.46	48.87	57.63	**69.40**	87.20	54.99
Lang→EN	Baseline	57.06	46.42	53.31	63.62	72.46	53.66
	Length	**55.68**	46.44	53.29	63.31	72.54	**52.74**
	Punct	**56.29**	**45.37**	**52.31**	**62.24**	**72.11**	53.51
	Frequency	57.32	**45.55**	52.75	62.10	75.73	54.72

(c) TER

Table 5: Contrasting our three proposed models to the baseline (NMT; Luong et al., 2017) across 12 translation tasks. We evaluate all the models using BLEU, METEOR and TER. The bold values represent the models that show statistically significant improvements over the baseline ($p < 0.001$; Clark et al., 2011). Note that for BLEU and METEOR, higher is better, while for TER, lower is better. All regularization schemes almost consistently lead to improvements, with the punctuation-based regularizer achieving the highest gains.

baseline across almost all language pairs for all models and across all metrics. We obtain statistically significant results for almost all translation tasks for at least one regularization method.

More specifically, the punctuation regularizer outperforms all the other models on all translation tasks except for French–English and English–French. For the latter, we observe that the word frequency regularizer is better than the other systems. This could be explained by the fact that the English vocabulary has many words borrowed from French, which makes the word frequency regularizer a better signal than punctuation or sentence length for this specific task. It also could be due to the fact that both English and French have the largest vocabulary for training the word frequencies (cf. Table 1; English has around 80M words and French has around 50M words, whereas all other languages have much less data).

The most challenging translation tasks are Slovene–English and English–Slovene, especially in terms of error rate. The results show that with 17k sentence pairs as a training set, it becomes more challenging to efficiently learn anything. The results we obtained are between 2 and 5 BLEU points when translating from English. The Slovene output contained many non-translated words. Specifically, this task greatly suffers when using the word frequency regularizer, with an error increase of about 10 TER points from English to Slovene. We do not observe such losses for the Czech–English and English–Czech transla-

tion tasks, even though the vocabulary size for estimating the word frequencies is lower for Czech. We hypothesize that this is due to the Czech training set being seven times larger than the Slovene one. We hypothesize that this is due to the fact that for Slovene we only have 17K sentence pairs for the training step; whereas for Czech, we have 122K sentence pairs, which helped control the model compared to Slovene.

One case where the punctuation regularizer succeeds consistently is on the English–German and German–English translation tasks, with an error reduction of about 1 TER point. This reflects the similarity in punctuation between these languages. Although we also observe improvements using the other regularization methods, e.g. the length-based method, these are not statistically significant here as calculated by MultEval (Clark et al., 2011).

Table 3 shows the BLEU scores of seven different systems including the one where we combine our three regularizers on the English–Vietnamese translation task. The combined regularizer does not only produce a statistically significant improvement of almost 1-BLEU point over the attention based baseline, but it also outperforms all the other regularizers achieving a BLEU score of 27.23.

7 Translation Examples

The punctuation regularizer outperforms the baseline in most cases, and all of our regularization methods show statistically significant improvements in at least one language. Below we present examples, extracted from the test data, of how each of the regularization methods affects the output in comparison to the baseline model. The purpose of the examples is to show how each objective function in the learning component affects the performance component.

7.1 Frequency-Based Regularizer

The frequency-based regularization method penalizes cases where the distribution of the target vocabulary greatly differs from the source vocabulary. We have noted a significant improvement for this specific regularizer when translating from French to English and vice-versa. Examples (5) and (6) show how this regularizer is improving the translation output.

(5) IN 90 % de notre temps entourés par l'architecture .
REF That's 90 percent of our time surrounded by architecture .
BASE <unk> percent of our time **via** architecture .
FREQ <unk> percent of our time **surrounded** by architecture .

(6) IN Débloquer ce potentiel est dans **l'intérêt** de chacun d'entre nous .
REF Unlocking this potential is in the **interest** of every single one of us .
BASE <unk> that potential is in all of us .
FREQ <unk> that potential is in the **interest** of all of us .

More precisely, *entourés* in French is almost as frequent as *surrounded* in English, which is a word that our model with frequency-based regularization translates correctly, while the baseline does not. Additionally, in Example (6), our model has a better fluency and adequacy than the baseline since it not only correctly translates *l'intérêt* to *interest*, but also correctly produces *of all* instead of *in all*, as in the baseline output.

7.2 Punctuation-Based Regularizer

The punctuation-based regularization performs best in the German–English and English–German translation tasks. This regularizer penalizes cases where the difference in the number of punctuation between the source and the target sentences is particularly large. As seen in Example (7), simply introducing this bias into a translation model leads to an output which more closely matches the punctuation of the source and target sentences.

(7) IN Und die Antwort , glaube ich , ist ja . [" F = T ∇ Sτ "] . Was Sie gerade sehen , ist wahrscheinlich die beste Entsprechung zu $E = mc^2$ für Intelligenz , die ich gesehen habe .
REF And the answer , I believe , is yes . [" F = T ∇ Sτ "] What you're seeing is probably the closest equivalent to an $E = mc^2$ for intelligence that I've seen .
BASE And the answer , I think , is yes .
PUNC And the answer , I think , is yes . [**" R = T T <unk> "**] **What you're looking at is probably the best <unk> <unk> <unk> of intelligence that I've seen .**

The baseline MT output completely fails to cap-

ture anything from the input except for the first part up to "...is yes." Our punctuation-based model, however, manages to capture most parts of the sentence.

7.3 Length-Based Regularizer

Finally, the length-based regularization method leads to noticeable improvements in the Czech–English and English–Czech translation tasks. Example (8) shows that introducing an input sentence length bias led to an MT output that is much closer to the reference than the baseline. The input sentence consists of 12 tokens (including punctuation), the baseline output consists of 10 tokens, while our length based regularization model preserves the length of 12 tokens.

(8) IN V roce 2009 jsem ztratila někoho , koho jsem velmi milovala .
REF In 2009 , I lost someone I loved very much .
BASE In 2009 , I lost somebody who I loved .
LEN In 2009 , I lost somebody who I loved **very much .**

7.4 General Improvements

The Slovene dataset is our smallest with about 17k sentence pairs for training. Despite the low amount of resources available in Slovene, we found that introducing very naive linguistic biases into our machine translation models actually leads to subtle differences that result in an output closer to the reference, not only lexically, but also semantically. In Example (9), we compare the output of the frequency based system against the baseline for the Slovene to English translation:

(9) IN In kaj potem ?
REF And so , what after that ?
BASE And then **then** ?
FREQ And then **, what** ?

In this particular case, the frequency based regularization model takes care of the translation of the word *what*, and although the word *so* is not translated, the overall meaning of the source is preserved.

(10) IN Imeti moraš otroke , da preživiš .
REF You need to have children to survive .
BASE **Well you have the kids that you need** to educate .
FREQ You **have to have kids** to educate .

Example (10) shows another case of how the output of the frequency-based regularization system actually shows overall improvements in an extremely low-resource language. The output of our system is semantically closer to the reference than the baseline output, up to the word *educate*. In addition, the system preserves a similar length as the source sentence.

(11) IN Mi smo tu na vrhu .
REF We are here on top .
BASE **What** we are at the top .
FREQ We are **here** at the top .

Finally, Example (11) shows a low-resource case where our system manages to make subtle changes in order to reach the correct translation, whereas the baseline system does not.

8 Conclusion

We have shown that using naive regularization methods based on sentence length, punctuation, and word frequency consistently improves the translation quality in twelve low-resource translation tasks. The improvement is consistent across multiple language pairs and is not dependent on the language family. We have reported and discussed examples demonstrating why and how each regularizer is improving the translation quality.

Our proposed approach shows that even naive, but linguistically motivated, regularizers help improve the translation quality when training NMT models. We believe this shows the usefulness of using task-related regularizers for improving neural models, and opens the door for future work to exploit these regularization methods in an even more efficient manner by experimenting with different ways of combining the regularizers with the loss function.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473. http://arxiv.org/abs/1409.0473.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Association for Computational Linguistics, pages 65–72. https://www.aclweb.org/anthology/W05-0909.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*. Trento, Italy, pages 261–268.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Association for Computational Linguistics, pages 103–111. https://www.aclweb.org/anthology/W14-4012.

Thomas C Chuang, Jian-Cheng Wu, Tracy Lin, Wen-Chie Shei, and Jason S Chang. 2004. Bilingual sentence alignment based on punctuation statistics and lexicon. In *International Conference on Natural Language Processing*. Springer, pages 224–232.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 176–181. https://www.aclweb.org/anthology/P11-2031.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 1723–1732. https://doi.org/10.3115/v1/P15-1166.

Orhan Firat, KyungHyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. *CoRR* abs/1601.01073. http://arxiv.org/abs/1601.01073.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *CoRR* abs/1705.03122. http://arxiv.org/abs/1705.03122.

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

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR* abs/1611.04558. http://arxiv.org/abs/1611.04558.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980. http://arxiv.org/abs/1412.6980.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Association for Computational Linguistics, Prague, Czech Republic, pages 177–180. https://www.aclweb.org/anthology/P07-2045.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*. Association for Computational Linguistics, pages 28–39. https://www.aclweb.org/anthology/W17-3204.

Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. 2017. Neural machine translation (seq2seq) tutorial. https://github.com/tensorflow/nmt.

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)*. Association for Computational Linguistics, pages 1054–1063. https://doi.org/10.18653/v1/P16-1100.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015a. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1412–1421. https://doi.org/10.18653/v1/D15-1166.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015b. Effective approaches to attention-based neural machine translation. *CoRR* abs/1508.04025. http://arxiv.org/abs/1508.04025.

Benjamin Marie and Atsushi Fujita. 2018. A smorgasbord of features to combine phrase-based and neural machine translation. In *AMTA*.

Antonio Valerio Miceli Barone, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 1490–1495. https://www.aclweb.org/anthology/D17-1156.

Graham Neubig. 2015. lamtram: A toolkit for language and translation modeling using neural networks. https://github.com/neubig/lamtram.

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. https://www.aclweb.org/anthology/P02-1040.

Jean Pouget-Abadie, Dzmitry Bahdanau, Bart van Merriënboer, Kyunghyun Cho, and Yoshua Bengio. 2014. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. *Syntax, Semantics and Structure in Statistical Translation* page 78.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR* abs/1508.07909. http://arxiv.org/abs/1508.07909.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*. The Association for Machine Translation in the Americas, pages 223–231. http://mt-archive.info/AMTA-2006-Snover.pdf.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958. http://jmlr.org/papers/v15/srivastava14a.html.

Ilya Sutskever, Oriol Vinyals, and Quoc Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR* abs/1706.03762. http://arxiv.org/abs/1706.03762.

Zhen Wang, Tingsong Jiang, Baobao Chang, and Zhifang Sui. 2015. Chinese semantic role labeling with bidirectional recurrent neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1626–1631. https://doi.org/10.18653/v1/D15-1186.

Richard Zens and Hermann Ney. 2004. Improvements in phrase-based statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*. Association for Computational Linguistics, Boston, Massachusetts, USA, pages 257–264. https://www.aclweb.org/anthology/N04-1033.