

Zero-Shot Style Transfer in Text Using Recurrent Neural Networks

Keith Carlson¹, Allen Riddell², and Daniel Rockmore³

^{1,3}Department of Computer Science, Dartmouth College

²School of Informatics and Computing, Indiana University
Bloomington

³Department of Mathematics, Dartmouth College

³The Santa Fe Institute

May 28, 2019

Abstract

Zero-shot translation is the task of translating between a language pair where no aligned data for the pair is provided during training. In this work we employ a model that creates paraphrases which are written in the style of another existing text. Since we provide the model with no paired examples from the source style to the target style during training, we call this task zero-shot style transfer. Herein, we identify a high-quality source of aligned, **stylistically distinct text in Bible versions** and use this data to train an encoder/decoder recurrent neural model. We also train a statistical machine translation system, Moses, for comparison. We find that the neural network outperforms Moses on the established BLEU and PINC metrics for evaluating paraphrase quality. This technique can be widely applied due to the broad definition of style which is used. For example, tasks like text simplification can easily be viewed as style transfer. The corpus itself is highly parallel with 33 distinct Bible Versions used, and human-aligned due to the presence of chapter and verse numbers within the text. This makes the data a rich source of study for other natural language tasks.

1 Introduction

Written text is one way in which we communicate our thoughts to each other. But given any “message” there are many ways to write a sentence capable of conveying the embedded information, even when they are all written in the same language. That’s at the heart of the notion of “style”. The various versions may have the same meaning, or semantic content, and insofar as they use different

words are each “paraphrases” of each other. These paraphrases, while sharing the same semantic content, are not necessarily interchangeable. When writing a sentence we consider not only the semantic content we wish to communicate, but also the manner, or style, in which we express it. A different wording may convey different levels of politeness or familiarity with the reader, display different cultural information about the writer, be easier to understand for certain populations, etc.

The problem of stylistic paraphrasing is clearly relevant for the creation of natural language generation systems. The translations, paraphrases, summarizations, and other language generated a natural language system is only useful if the outputs are understood and accepted by the intended audience. This may require us to target certain levels of simplicity, formality, or other characteristics of style in the language produced.

We approach the problem of stylistic paraphrasing by training a model to produce paraphrases that target the writing style used in some existing work. There are many features of the prose which contribute to the perceived style of a text including sentence length, use of passive or active voice, vocabulary level, tone, and level of formality. Analysis and classification of style focusing on sentiment, usage of stop-words, formality, etc have all been the subject of study [23, 14, 4, 30]. Similarly, generation of language targeting one aspect of style such as simplicity, formality, length, and use of active voice have received attention [31, 29, 40, 41]. Our approach does not consider any of these aspects explicitly, but instead learns to create text similar to the provided target text by learning to preserve content and alter style simultaneously. As mentioned, some work has focused on generating text controlling politeness and simplicity [29, 40], but with only a few recent exceptions both attempting to modernize Shakespearean language [39, 16] there seems to be little work using a broader definition of style. A highly competent system which can preserve meaning but produce text of a desired style could be applied in many ways. For example, we could use text we deem to be simple as a target and use such a system for text simplification, or could even produce literary classics as though they had been written by another literary master. Rather than simply wondering what “Pride and Prejudice” would have looked like if it had been authored by Hemingway, we could have the model generate it.

The main contributions of our work are as follows:

1.1 Identification of a Highly Parallel Corpus.

Herein we identify a novel and highly parallel corpus useful for the style paraphrasing task: thirty-two stylistically distinct versions of the Bible (Old and New Testaments). Each version is understood as embodying a unique writing style. The versions in this corpus embody a wide range of intentions. Versions such as the International Children’s Bible were written to be simple enough to be understood by children. Other versions, like the 21st Century King James

Version, were written to maintain the “the traditional Biblical language”¹ of the original King James Version written in the 1600’s. In addition to being viewed individually, the versions can also be partitioned according to several stylistic criteria, any one of which could be a goal of a paraphrasing. For example, metrics that enable the identification of versions deemed “simple” could identify a subcorpus that trains towards the task of text simplification. Versions identified as using “old” language could be used to train towards the task of “text archaification”. Such richly parallel datasets are difficult to find, but this corpus provides such a wide range of text that it could be used to focus on a wide range of stylistic features already present within the data.

Another benefit of this dataset is that the alignment exists within the data itself. While many parallel corpora require alignment before they can be used, here verse numbers immediately identify equivalent pieces of text. Thus, in this data the text has all been aligned by humans already. This eliminates the need to use text alignment algorithms which may not produce alignments that match human judgement.

1.2 Zero-Shot Style Transfer

One of our main technical contributions is our “zero-shot stylistic paraphrasing”, inspired by zero-shot machine translation [17]. In zero-shot translation, the system must translate from one language to another even though it has never seen a translation between the particular language pair. Similarly, we train our model without ever giving examples that pair the source and target versions which we ultimately use for testing. This approach is important, as paired data may not exist for many of the potential applications of style transfer.

1.3 Use of Encoder-Decoder Recurrent Neural Networks for Style Transfer

We treat the task of zero-shot style transfer as a monolingual machine translation problem and train a version of encoder-decoder recurrent neural networks (Seq2Seq) architecture on our dataset. Seq2Seq architectures were first introduced and used for machine translation [6] and has been widely used and adapted [32, 2, 8] since the original publication. One such paper used the model to perform zero-shot machine translation [17]. Despite the success of this architecture on similar problems, application to stylistic paraphrasing has not been thoroughly explored. We are aware of only one existing work[16] which uses such a network in this context on real data. Our work continues the exploration of these models for style transfer. We use some different techniques and modifications, which may allow the Seq2Seq model to be applied to less specialized corpora than in the existing work. We, like the authors[16], find that Seq2Seq is able to outperform existing statistical translation methods on this task, but

¹From the version’s website www.kj21.com

through the use of style tags are able to achieve this result without the use of a domain-specific and human produced dictionary.

2 Related Work

The clearest connection is to work in traditional language-to-language translation. The Seq2Seq model was first created and used in conjunction with statistical methods to perform machine translation[6]. The model consists of a recurrent neural network acting as an encoder which produces an embedding of the full sequence of inputs. This sentence embedding is then used by another recurrent neural network which acts as a decoder and produces a sequence corresponding to the original input sequence.

Long Short-Term Memory(LSTM)[13] was introduced to allow a recurrent neural network to store information for an extended period of time. Using a formulation of LSTM which differs slightly from the original[10], the Seq2Seq model was adapted to use multiple LSTM layers on both the encoding and decoding sides[32]. This model demonstrated near state-of-the-art results on the WMT-14 English-to-French Translation task.² In another modification an attention mechanism was introduced[2] which allowed the decoder network to learn to focus on relevant parts of the encoded source sentence during decoding. This again achieved near state-of-the-art results on English-to-French translation.

Another paper proposed a version of the model which could translate into many languages[2]. They use a single encoder, but a different decoder for each target language. This idea was extended to study multi-way multilingual machine translation with a recurrent network[9] using a model that translates from many different language pairs by training a separate encoder for every source language and a separate decoder for every target language. They use the correct encoder and decoder for the encountered language pair, but have an attention mechanism that is shared across all pairs.

Recently, a neural machine translation model capable of both multilingual translation and zero-shot translation was introduced[17]. The authors make no major changes to the Seq2Seq architecture, but introduce special tokens at the start of each input sentence indicating the target language. The model can learn to translate between two languages which never appeared as a pair in the training data, provided it has seen each of the languages paired with others. They call this task *zero-shot translation*.

With this advance the general idea of using artificially introduced tags in neural machine translation soon found use for encoding data other than target language. Researchers use such tags (at the end of their source sentences rather than beginning) to target the use of active or passive voice in Japanese-to-English translation [40]. Some languages such as German use honourifics to express formality. Tags encoding a target level of formality were employed to

²<http://www.statmt.org/wmt14/translation-task.html>

control the use of the honourifics in English-to-German neural machine translation [29].

This work on machine translation is relevant for paraphrase generation framed as a form of *monolingual translation*. In this context statistical machine translation techniques were used to generate novel paraphrases [26]. More recently, phrase-based statistical machine translation software was used to create paraphrases [36].

Tasks such as text simplification [31, 37] can be viewed as a form of style transfer. Paraphrase targeting a more general interpretation of style was first introduced in 2012 [39]. Therein the authors use a dataset of Shakespearean plays and their modern “translations” and train several models to convert text between the styles.

The advances mentioned previously in neural machine translation have only started to be applied to more general stylistic paraphrasing. One approach proposed the training of a neural model which would “disentangle” stylistic and semantic features, but did not publish any results [18]. Another attempt at text simplification as stylistic paraphrasing is [35]. They generate artificial data and show that the model performs well, but do no experiments with human-produced corpora. The Shakespeare dataset [16] recently was used with a Seq2Seq model [39]. Their results are impressive, showing significant improvement over statistical machine translation methods as measured by automatic metrics. They experiment with many settings, but their best models all require the integration of a human-produced dictionary which translates approximately 1500 Shakespearean words to their modern equivalent. None of the models they explore without using this dictionary are able to outperform the statistical model. Since such dictionaries do not exist for most style pairs we show that Seq2Seq can outperform statistical methods without such specialized data. This generality opens up a much broader collection of text as candidates for style transfer using Seq2Seq.

3 Data

3.1 Data Collection

As stated above, a significant contribution of this paper is the identification of Bible versions as a paraphrase dataset. For our work we collected 32 English translations of the Bible from BibleGateway.com, and also the Bible in Basic English from www.o-bible.com. The complete list of versions used can be seen in Table 1. This data is highly parallel and high-quality, having been produced by human translators. Sentence level alignment of parallel text is needed for many NLP tasks. Work exists on methods to automatically align texts [43, 15, 7], but the alignments produced are imperfect and some have been criticized for issues which decrease their usefulness [38]. The Bible corpus is human-aligned by virtue of the consistent use of book, chapter, and verse number across translations. While many verses are single sentences some are sentence

fragments or several sentences. This is not problematic as we only require the parallel text to be aligned in small parts which have the same meaning, but there is no obvious reason that this must be at a strict sentence level.

Some Bible versions contain instances of several verses combined to one. For example, we may find a Bible version with “Genesis 1:1-4” instead of singular instances of each of the four verses. We remove these aggregations versions from our data to keep the alignment more fine-grained and consistent. Even with this regularization we still have approximately 33.8 million potential source and target verse pairings.

3.2 Data Processing

Data Splitting

We need to split our data into training, testing, and development sets. We begin by randomly selecting 200 verses for the development set. We write out a file containing the “ERV” (Easy-to-Read) and “KJ21” (King James 21st Century) versions of these 200 verses since we will ultimately evaluate performance on the task of paraphrasing “ERV” verses into the “KJ21” style. We wanted to choose versions which were obviously stylistically different in order to discourage the system from simply reproducing the source verses. “ERV” and “KJ21” were identified qualitatively as good candidates and then confirmed to be much less similar than most versions by measuring the BLEU score [24] between them. We then remove all occurrences of these verses from the testing and training datasets. For example, if Genesis 1:1 appears in the development set, then no version of Genesis 1:1 will be included in the testing or training sets. Next a 200-verse “ERV” to “KJ21” testing set is selected from the remaining verses. Again, all other versions of these verses are removed from future consideration. A 1000-verse sample of “KJ21” lines is created for eventual use to fine-tune the language model of the statistical translation baseline. All versions of these 1000 verses are also removed from our active set. Finally, 10 million verse-pairs are selected for the training set. Versions of the same verse can appear in training multiple times as long as they are part of a different source-target pair. In the training set we ensure that we do not select any pairs translating from “ERV” to “KJ21” so we can evaluate the model’s zero-shot performance on this pair.

Style Tags

Existing work does zero-shot neural machine translation by adding to the beginning of each source sentence a special token indicating the language of the target sentence[17]. Similarly for all pairs in our training, testing, and development sets we prepend a tag to the source verse indicating the version of the target. For example if the target style for a verse pair is that of the 21st Century King James Version, we start off the source sentence with a “<KJ21>” token.

Vocabulary Construction

The verse numbers at the beginning of each line were removed and then each of the three sets (development, testing, training) was tokenised using the nltk package in Python. Names and rare words are difficult to handle in NLG

Bible Versions
New Life Version (NLV), Bible in Basic English (BBE), New International Reader’s Version (NIRV), International Children’s Bible (ICB), Easy-To-Read Version (ERV), New Century Version (NCV), Contemporary English Version (CEV), Good News Translation (GNT), God’s Word Translation (GW), Names of God Bible (NOG), World English Bible (WEB), Jubilee Bible 2000 (JUB), New King James Version (NKJV), Young’s Literal Translation (YLT), Modern English Version (MEV), English Standard Version (ESV), 1599 Geneva Bible (GNV), New International Version (NIV), Lexham English Bible (LEB), Douay-Rheims 1899 American Edition (DRA), Holman Christian Standard Bible (HCSB), 21st Century King James Version (KJ21), New Living Translation (NLT), American Standard Version (ASV), New Revised Standard Version (NRSV), Common English Bible (CEB), New English Translation (NET), Darby Translation (DARBY), International Standard Version (ISV), Revised Standard Version (RSV), New American Bible Revised Edition (NABRE), The Living Bible (TLB), The Message (MSG)

Table 1: Names of Bible Versions collected from Biblegateway.com (and BBE from www.o-bible.com) followed by their standard abbreviations in parenthesis.

tasks and often they are replaced with a generic “Unknown” token [34, 25, 2]. Recently, byte pair encoding was used to create a vocabulary of subword units which removes the need for such a token [28]. In the resulting vocabulary rare words are not present but the smaller units which make up the word are. For example, in our data the rarely seen word “caretakers” is replaced by two more frequent subwords “care” and “takers”. We generated a vocabulary of the 40,000 most frequent subword units from all of our Bible versions, a random sampling of modern works from gutenber.org, and articles from Wikipedia. The additional text sources were used to ensure that our vocabulary was diverse and realistic and did not contain only unusual biblical names and terms. This vocabulary was then applied to each of the samples by replacing any word which was not in the vocabulary with its constituent subword units.

6 of the Bible versions we used are available in the public domain. We have created a repository³ which contains these 6 versions as well as the code and a brief walkthrough of how to run a version of our experiment which uses only these 6 versions.

4 Model

We use a multi-layer recurrent neural network encoder and multi-layer recurrent network with attention for decoding. This set-up is similar to those described by Sutskever et al. [32] and Bahdanau et al.[2]. We will refer to this model as

³<https://github.com/keithecarlson/Zero-Shot-Style-Transfer>

“Seq2Seq” when comparing results.

The encoder consists of three layers each with 512 LSTM cells using the formulation from Graves [10]. These LSTM layers have residual connections between them [11] which have previously been applied to paraphrasing [25]. The encoder is bi-directional so it recurrently encodes the words in the sentence both forwards and backwards. Bi-directional recurrent neural networks [27] have been previously used for machine translation [2]. Our encoder first applies a trainable embedding layer to project each word into a 512-dimensional vector before being passed on to the LSTM layers. Dropout is performed on the input to each LSTM layer with probability of dropping set to 0.3. We perform this dropout between each layer, but not between the recurrent connections. This has been found to be most effective [42].

The decoder also has three LSTM layers of 512 cells and first embeds each word into 512 dimensions using a learned embedding layer. At each timestep the next word’s embedding is passed to the LSTM layers. These LSTM layers in the decoder are initialised using the final state of the corresponding layer in the encoder. Dropout is performed between layers as in the encoder. The decoder also uses an attention mechanism as described by Bahdanau et al. [2] to focus on specific parts of the output from each step of the encoder.

During training mini-batches of 64 verse-pairs are randomly selected from the training corpus. Each of the target and source sentences are truncated to 100 tokens if necessary. The tokens are fed into the encoder one-by-one, and once the entire sentence has been encoded the decoder is fed a special token signalling that it should begin generating a paraphrase. During training the decoder is given the correct previous word in the target sentence as input for each timestep other than the first, regardless of the correctness of output it produced. The model’s parameters are then adjusted using the “Adam optimiser” [19]. A checkpoint of the model is saved after every 5000 mini-batches. The checkpoints are used to decode the development set and the checkpoint with the best sliding window of size 3 average BLEU score is used for inference on the testing data.

During inference a single source sentence is fed into the model but the target sentence is not provided. Unlike during training, the decoder is fed its own prediction as input for the next timestep. The decoder performs a beam search [32] with a width of 10 to produce the most likely paraphrase.

5 Experiment

As indicated above, we deploy the Seq2Seq model on a corpus of versions of the Bible using a publicly available library [3]. This library was implemented using the API provided by Tensorflow [1]. The Moses translation system [20] is frequently used to produce paraphrases [5, 39, 36, 16] serves as a useful baseline. See Figure 1 for an overview of our work process.

The code and data to run a version of our experiment on the publicly available portion of our data is available⁴.

⁴<https://github.com/keithcarlson/Zero-Shot-Style-Transfer>

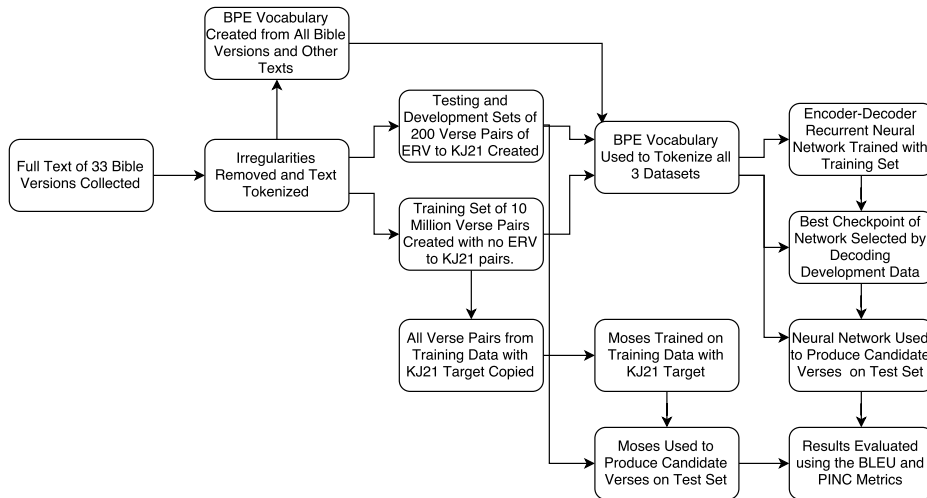


Figure 1: A Diagram of the Experimental Workflow

5.1 Metrics

We find that our Seq2Seq model outperforms Moses on this task. For evaluation we use several established measures. We first calculate BLEU [24] scores for our results. BLEU is a metric for comparing parallel corpora which rewards a candidate sentence for having n -grams which also appear in the target. Although it was created for evaluation of machine translation, it has been found that the scores correlated with human judgement when used to evaluate paraphrase quality [36]. The correlation was especially strong when the source sentence and candidate sentence differed by larger amounts as measured by Levenshtein distance over words.

BLEU gets a some of what a good paraphrase should accomplish (similarity), but a good (i.e., interesting) paraphrase should use different words than the source sentence, as noted by Chen and Dolan [5]. They introduce the PINC score which “computes the percentage of n -grams that appear in the candidate sentence but not in the source sentence”. The PINC score makes no use of target sentence, but rewards a candidate for being dissimilar from the source. To capture a candidate’s similarity to the target and dissimilarity from the source they use both the BLEU and PINC scores together. They find that BLEU scores correlate with human judgement of semantic equivalence and that PINC scores correlated highly with human ratings of lexical dissimilarity. Lexical dissimilarity on its own is important for paraphrasing, but as previously mentioned, a high lexical dissimilarity may also strengthen the correlation of BLEU scores and human judgement of paraphrase quality[36]. The joint use of PINC and BLEU can also be found in previous work on stylistic paraphrasing [39, 16].

ERV (source)	KJ21 (target)	Moses Output	Seq2Seq Output
Now I will set you free from the power of Assyria . I will take the yoke off your neck and tear away the chains holding you . ”	for now will I break his yoke from off thee , and will burst thy bonds asunder . ”	Now I shall make you free from the power of Assyria ; and I will take the yoke from off thy neck and rend away the chains holding thee . ”	And now will I deliver you from the hand of Assyria ; and I will take the yokes from off thy neck , and I will cut off the bonds from thee .
I will make you my faithful bride . Then you will really know the Lord .	I will even betroth thee unto Me in faithfulness , and thou shalt know the Lord .	I will make thee my faithful bride . And ye shall know the Lord .	And I will make thee My covenant , that thou mayest know the Lord .
I have fallen , but enemy , do n’t laugh at me ! I will get up again . I sit in darkness now , but the Lord will be a light for me .	Rejoice not over me , O mine enemy ; when I fall , I shall arise ; when I sit in darkness , the Lord shall be a light unto me .	I have fallen , but enemy , do not laugh at me ! I will rise up again . I sit in darkness now , but the Lord will be a light to me .	I have fallen , but my enemy hath not mocked ; I will rise again ; I shall sit in darkness , and the Lord shall be a light unto me .

Table 2: Example Verses from the Easy-To-Read and 21st Century King James Versions and the Outputs from Seq2Seq and Moses

5.2 Baseline

The statistical machine translation system Moses [20] is an established baseline for testing new paraphrasing corpora and models. It recasts paraphrase as a monolingual translation task on the paired data. Previous such uses include the work of Chen and Dolan[5], Xu et al.[39], and Wubben et al.[36] who found that it outperformed paraphrasing based on word substitution. It was also used as a baseline in [16] for stylistic paraphrasing of Shakespeare into present day English, and was outperformed by a Seq2Seq model supplemented by an external dictionary of Shakespearean words and their approximate modern day equivalents was utilized. Our work uses no additional reference information.

In our case about we need to be careful about the data that we provide Moses with for training. Moses is equipped only to handle translation from one language to one other language, and has no simple way to perform zero-shot translation. We could give Moses all the training data that we give Seq2Seq and then it would learn to produce good paraphrases, but we want the paraphrases produced by Moses to be in the style of 21st Century King James Version. So, we instead provide Moses with only the training pairs where the target sentence is from KJ21. This reduces the 10 million pairs given to the Seq2Seq model to about 300,000 pairs. These pairs provided to Moses have not had the subword unit vocabulary applied to them, nor do the source sentences have the special target tokens at their beginnings. We train Moses using mgiza for word alignment[22], Implz for the language model[12], and mert[22] to fine-tune the model parameters to the development data set. All of these tools are provided with Moses. The language model is order 5 and built on all of the KJ21 targets in the training data that have 100 or fewer tokens.

In addition to the results from Moses and the Seq2Seq model, we show the BLEU score between the unaltered source and target sentences. Since both the source and target translations are written in English, albeit different styles, this comparison also serves as a meaningful baseline.

5.3 Results

After training we decoded the 200-verse test set with both Moses and Seq2Seq and calculated the PINC and BLEU scores. Recall that neither model has seen any version of the verses provided during testing. Table 2 shows the original versions and the outputs of each system for a few samples from the testing set. The results of the evaluation metrics can be seen in Table 3.

	BLEU	PINC
Unmodified ERV	9.61	-
MOSES	16.53	48.53
Seq2Seq	20.09	74.04

Table 3: Comparison of BLEU and PINC Scores for the Original Text, MOSES Output, and Seq2Seq Output

Seq2Seq outperforms Moses on the test set on both metrics. The BLEU score indicates that the neural model’s output is closer to the KJ21 target than the output of Moses. The difference in PINC score is even greater, indicating that the Seq2Seq output differs more from the ERV source than that of Moses. This matches our qualitative observation that Moses is much more conservative in its paraphrasing, as can be seen in the examples in Table 2. In the text produced by Moses, many words or phrases from the source are left unchanged.

The Seq2Seq model appears to have learned some subtleties about the KJ21 style that Moses is not able to capture. The 21st Century King James Version capitalises pronouns such as “Me” or “My” when they refer to the divine. In the second example in Table 2 the Seq2Seq output correctly capitalises “My”. In the third example the output of Seq2Seq contains “my” but is correctly lowercase since the speaker is human. Since ERV does not capitalise these pronouns when they do not appear at the start of the sentence the Seq2Seq model would need some “understanding” of who is speaking to capitalise these pronouns correctly and to only do so when targeting a version, such as KJ21, which capitalises these pronouns.

In the 200 verses used for testing, Seq2Seq produced a capitalised “My” in 12 occurrences not at the start of a sentence and produced a lowercase “my” 28 times. By the authors’ evaluation all 12 of the capitalised occurrences are correctly capitalised and 27 of the 28 lowercase occurrences are correctly lowercase. The only failure of Seq2Seq is the output *“And He answered and said unto them , “ Do not forbid him , for he that uses my name in power shall not quickly curse me .”* In the KJ21 version of this verse the pronoun is capitalised: *“But Jesus said , “ Forbid him not , for there is no man who shall do a miracle in My name that can lightly speak evil of Me .”*

As previously mentioned, one advantage of this framework is that the same model is able to paraphrase into a variety of different styles. Table 4 shows the output of Seq2Seq when provided with an ERV verse from the training set, and asked to produce text mimicking several other styles. The outputs qualitatively appear to be quite different, indicating that the model is learning that different versions should have different paraphrases. Once again, Seq2Seq seems to have learned some intricacies which differ across the styles. It has learned that ASV frequently uses “Jehovah”, for example, and that neither BBE or ASV uses quotation marks.

6 Conclusions and Future Work

In this paper we collected a previously untapped dataset of already aligned parallel text in the form of Bible translations. We view stylistic paraphrase generation as a monolingual machine translation task and attempt to do zero-shot style transfer by considering each Bible version to be a unique writing style. We train a sequence-to-sequence recurrent neural network to do this zero-shot translation and also train the statistical machine translation software Moses as a baseline for comparison.

Original (ERV)	NRSV Target	ASV Target	BBE Target
On this day you should tell your children , 'We are having this festival because the Lord took me out of Egypt . '	On that day you shall say to your children , 'The Lord brought me out of Egypt . '	And it shall come to pass in that day that ye shall show your children , saying , We have a solemn assembly , because Jehovah brought me out of Egypt .	And on this day you will say to your children , The Lord took me out of Egypt .
I will make you my faithful bride . Then you will really know the Lord .	I will make you my beloved , and you will know the Lord .	And I will make thee my beloved , that thou mayest know Jehovah .	And I will make you my true bride , and you will have knowledge of the Lord .

Table 4: Examples of Seq2Seq Output When Targeting Different Styles for Verses in Testing Set

The application of neural networks to the task of stylistic paraphrasing using a corpus of “real-world” data has only started to be explored. Previous work [16] used a similar model and found that Seq2Seq was able to outperform the statistical machine translation software Moses when paraphrasing Shakespeare into modern English style. Their work and ours show that Seq2Seq models are a promising alternative to statistical methods for style transfer. Their model requires a human-produced dictionary of Shakespearean words and their modern equivalent to beat Moses, and their results without this dictionary, while still impressive, fall short of Moses. In distinction, we find that using style tags in the source sentence allows the Seq2Seq model to outperform Moses on this task as measured by both BLEU and PINC scores without the use of such specialized external data. We suspect that this gain is because the network is able to improve its translations from ERV to KJ21 by generalising what it learns about translating between unrelated style pairs, just as translation quality between two languages can improve by allowing the network to see unrelated language pairs [17]. Our result reinforces the finding [16] that neural networks are suitable for the task of style-transfer on natural data and should continue to be explored.

It is possible that the Seq2Seq model could be improved with existing techniques and achieve even better results on this task. Some potential improvements are the use of coverage modelling [33] to help track which parts of the source sentence have already been paraphrased, or a pointer network [21] to allow copying of words directly from the source sentence. These pointer networks

have already been used for stylistic paraphrasing [16], and may also prove useful in a Seq2Seq model with style tags.

Additional future work could also revolve around the data we introduce. For example, due to the large number of already aligned human-produced translations this data could be used for training towards the traditional paraphrasing task in which a specific style is not targeted. Alternatively, you could choose some aspect of style, such as simplicity or formality, and partition the corpus based on that criteria. The partitioned corpus could then be used to train models which produce text with the desired characteristic.

The task of zero-shot style transfer itself could be explored more fully as well. We would like to experiment with how Seq2Seq does on other corpora, especially when trained on pairs which are quite stylistically distinct. Since we treat each text source as a unique style we believe that additional “styles” could be added to the training data and effectively transferred without major change to the architecture.

Even without these improvements our results show that Seq2Seq models appear to be well-suited to style transfer. The use of target style tags in these networks enable zero-shot translation and our experiments indicate that this formulation is able to beat established phrase-based statistical methods without the use of any specialized external data.

Data accessibility: Data, code, and a walk-through to run a version of our experiment on the public domain portion of our data is available at <https://github.com/keithecarlson/Zero-Shot-Style-Transfer>

Authors’ contributions: K.C., A.R. and D.R. designed the experiment, interpreted results and reviewed, edited, and approved the final paper. K.C. collected the data, wrote the code, ran the experiment and wrote the paper.

Competing interests: We have no competing interests.

Funding: We have received no funding for this work.

References

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.
- [3] Denny Britz, Anna Goldie, Thang Luong, and Quoc Le. Massive Exploration of Neural Machine Translation Architectures. *ArXiv e-prints*, March 2017.
- [4] Keith Carlson, Michael A Livermore, and Daniel Rockmore. A quantitative analysis of writing style on the us supreme court. 2015.
- [5] David L Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 190–200. Association for Computational Linguistics, 2011.
- [6] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.
- [7] William Coster and David Kauchak. Simple english wikipedia: a new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 665–669. Association for Computational Linguistics, 2011.
- [8] Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-task learning for multiple language translation. In *ACL*, 2015.
- [9] Orhan Firat, KyungHyun Cho, and Yoshua Bengio. Multi-way, multilingual neural machine translation with a shared attention mechanism. *CoRR*, abs/1601.01073, 2016.
- [10] Alex Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013.
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
- [12] Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics, 2011.
- [13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [14] James M Hughes, Nicholas J Foti, David C Krakauer, and Daniel N Rockmore. Quantitative patterns of stylistic influence in the evolution of literature. *Proceedings of the National Academy of Sciences*, 109(20):7682–7686, 2012.

- [15] William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. Aligning sentences from standard wikipedia to simple wikipedia. In *HLT-NAACL*, pages 211–217, 2015.
- [16] Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. Shakespearizing modern language using copy-enriched sequence-to-sequence models. *arXiv preprint arXiv:1707.01161*, 2017.
- [17] 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. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558, 2016.
- [18] Jad Kabbara and Jackie Chi Kit Cheung. Stylistic transfer in natural language generation systems using recurrent neural networks. *EMNLP 2016*, page 43, 2016.
- [19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [20] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics, 2007.
- [21] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.
- [22] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics, 2003.
- [23] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.
- [24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.
- [25] Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek V. Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. Neural paraphrase generation with stacked residual LSTM networks. *CoRR*, abs/1610.03098, 2016.
- [26] Chris Quirk, Chris Brockett, and William B Dolan. Monolingual machine translation for paraphrase generation. In *EMNLP*, pages 142–149, 2004.

- [27] Mike Schuster and Kuldeep K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.
- [28] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909, 2015.
- [29] Rico Sennrich, Barry Haddow, and Alexandra Birch. Controlling politeness in neural machine translation via side constraints. In *HLT-NAACL*, pages 35–40, 2016.
- [30] Fadi Abu Sheikha and Diana Inkpen. Automatic classification of documents by formality. In *Natural Language Processing and Knowledge Engineering (NLP-KE), 2010 International Conference on*, pages 1–5. IEEE, 2010.
- [31] Lucia Specia. Translating from complex to simplified sentences. *Computational Processing of the Portuguese Language*, pages 30–39, 2010.
- [32] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.
- [33] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. *arXiv preprint arXiv:1601.04811*, 2016.
- [34] Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey E. Hinton. Grammar as a foreign language. *CoRR*, abs/1412.7449, 2014.
- [35] Tong Wang, Ping Chen, Kevin Michael Amaral, and Jipeng Qiang. An experimental study of LSTM encoder-decoder model for text simplification. *CoRR*, abs/1609.03663, 2016.
- [36] Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. Paraphrase generation as monolingual translation: Data and evaluation. In *Proceedings of the 6th International Natural Language Generation Conference*, pages 203–207. Association for Computational Linguistics, 2010.
- [37] Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1015–1024. Association for Computational Linguistics, 2012.
- [38] Wei Xu, Chris Callison-Burch, and Courtney Napoles. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297, 2015.
- [39] Wei Xu, Alan Ritter, William B Dolan, Ralph Grishman, and Colin Cherry. Paraphrasing for style. In *24th International Conference on Computational Linguistics, COLING 2012*, 2012.

- [40] Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato, and Mamoru Komachi. Controlling the voice of a sentence in japanese-to-english neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 203–210, 2016.
- [41] Mark Yatskar, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. For the sake of simplicity: Unsupervised extraction of lexical simplifications from wikipedia. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 365–368. Association for Computational Linguistics, 2010.
- [42] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *CoRR*, abs/1409.2329, 2014.
- [43] Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd international conference on computational linguistics*, pages 1353–1361. Association for Computational Linguistics, 2010.