Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 2615–2621
Marseille, 11–16 May 2020

Jejueo Datasets for Machine Translation and Speech Synthesis

Kyubyong Park, Yo Joong Choe, Jiyeon Ham
Kakao Brain
20, Pangyoyeok-ro 241, Bundang-gu, Seongnam-si, Gyeonggi-do, Korea
{kyubyong.park, yj.choe, jiyeon.ham}@kakaobrain.com

Abstract

Jejueo was classified as critically endangered by UNESCO in 2010. Although diverse efforts to revitalize it have been made, there have been few computational approaches. Motivated by this, we construct two new Jejueo datasets: *Jejueo Interview Transcripts* (JIT) and *Jejueo Single Speaker Speech* (JSS). The JIT dataset is a parallel corpus containing 170k+ Jejueo-Korean sentences, and the JSS dataset consists of 10k high-quality audio files recorded by a native Jejueo speaker and a transcript file. Subsequently, we build neural systems of machine translation and speech synthesis using them. All resources are publicly available via our GitHub repository. We hope that these datasets will attract interest of both language and machine learning communities.

Keywords: Jejueo, Jeju language

1. Introduction

Jejueo, or the Jeju language, is a minority language used on Jeju Island (O'Grady, 2015). It was classified as critically endangered by UNESCO in 2010.[1] While there have been many academic efforts to preserve the language (Yang et al., 2017; Saltzman, 2017; Yang et al., 2018a; Yang et al., 2018b), data-driven approaches for Jejueo-related language tasks have been rare.

Meanwhile, the natural language processing (NLP) community has observed significant advances in both machine translation (Sutskever et al., 2014; Cho et al., 2014; Vaswani et al., 2017) and speech synthesis (Oord et al., 2016; Wang et al., 2017; Shen et al., 2018), especially driven by deep learning in recent years.

In particular, there has been growing attention towards low-resource scenarios (Zoph et al., 2016; Gu et al., 2018), which pose a unique challenge for existing deep learning methods. The challenge is unique not only in the sense that less data is available but also in the sense that low-resource languages often raise different syntactic, morphological, and semantic challenges that are under-explored by systems optimized on major languages such as English, German, French, and Arabic (Bender, 2019). Tasks relevant to Jejueo often involve having to deal with such challenges.

Motivated by the unique challenges that Jejueo presents as well as the lack of Jejueo resources available for computational approaches, we develop a machine-readable Jejueo-Korean parallel corpus and a clean Jejueo single speaker speech dataset.

Our contributions can be summarized as follows:

- We present the *Jejueo Interview Scripts* (JIT) dataset, a Jejueo-Korean parallel corpus of more than 170k sentences.
- We train neural machine translation models on the JIT dataset so that they can be the baselines for future studies.
- We create the *Jejueo Single Speaker Speech* (JSS) dataset of 10k audio files and their transcripts.
- We build speech synthesis models with the JSS dataset and examine how various tokenisation strategies affect them.

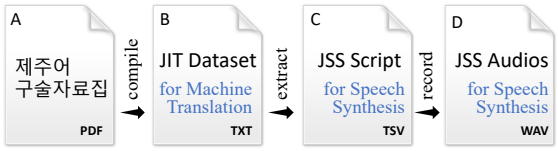

Figure 1: Overview of dataset construction. The original `pdf` files (A) are compiled into the JIT dataset (B). Part of the Jejueo text in the JIT dataset is extracted and saved as the JSS script file (C). Finally, we ask a native Jejueo speaker to record the script (D).

The procedure for constructing the datasets is shown in Figure 1. To the best of our knowledge, they are the first publicly available Jejueo datasets for computational tasks, particularly Jejueo-Korean machine translation and Jejueo speech synthesis.

All the resources are released via our GitHub repository[2].

2. Jejueo

Jejueo (ISO 639-3 language code: jje) is the traditional language used on Jeju Island, located south of the Korean mainland (See Figure 2). Today, there are only 5,000-10,000 fluent speakers, mostly above 70 years of age. The younger generation in Jeju is not learning the language in school, so they show a variety of levels of proficiency.

For a long time, Jejueo has been treated as a dialect of Korean (ISO 639-3 language code: kor) rather than a distinct language (O'Grady, 2015). In this paper, we do not want to get into the debate of whether to consider Jejueo as a language or a dialect of Korean. Instead, we pay attention to the fact that Jejueo is often incomprehensible to Korean-only speakers (Yang et al., 2018a). This motivates the need

[1] `http://www.unesco.org/languages-atlas/en/atlasmap.html`

[2] `https://github.com/kakaobrain/jejueo`

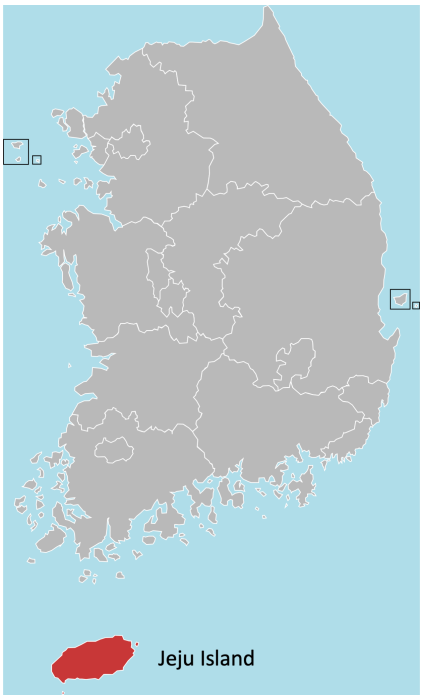

Figure 2: Jeju Island and South Korea.[5]

to consider Jejueo-Korean translation as an important language task. In addition, Jejueo accent is different from standard Korean[3]. So it sounds unnatural when a Korean speaker reads out a Jejueo text. In our preliminary experiment, we found out that our internal speech synthesis model for Korean was not able to generate Jejueo speeches properly.

For further information about Jejueo, we refer readers to the website of the Jejueo Project in University of Hawaii[4]. Here, we highlight one major difference between Jejueo and Korean: Araea (ᄋᆞ). Araea is a mid or low vowel that was used in Middle Korean. It is obsolete in contemporary Korean, but retained in Jejueo. Due to the presence of Araea, although both Jejueo and Korean are written in Hangul, Jejueo text is not easy for Korean speakers to type in digital settings. This will be discussed further in the next section.

3. JIT (Jejueo Interview Transcripts) Dataset

The *Jejueo Interview Transcripts* dataset, or JIT, is Jejueo-Korean parallel data compiled from 제주어구술자료집 1-20 by us. 제주어구술자료집 is the final report of the project performed by Center for Jeju Studies from 2014 until 2018. For the first three years, they interviewed Jeju senior citizens in Jejueo. Afterwards the interviews were carefully transcribed and then translated into standard Korean by experts. Along with additional notes, the results were arranged in 20 `pdf` files and opened to the public via their webpage[6].

제주어구술자료집 is an invaluable Jejueo resource in that it is arguably the largest Jejueo corpus publicly available. Unfortunately, it is not designed for computational use, after all. The `pdf` format is not machine friendly so it is tricky for researchers to work with it. Therefore, we convert the original `pdf` files into plain text files step by step so that they can be used for machine translation or any other computational tasks.

1. Convert `.pdf` files into plain `.txt` files using an online file conversion tool[7].

2. Remove the front and back matters containing metadata. Accordingly, only interview dialogues remain.

3. Parse every line to extract Jejueo text and its Korean translation.

   (a) In each line, Jejueo text is followed by Korean translation enclosed by parentheses. We capture Jejueo and Korean texts separately using simple regular expressions. However, some lines do not conform to the rule. For simplicity, we ignore those irregularities.

   (b) Accidental line breaks frequently occur. We replace the line break with a special symbol, ^. It can take place in the middle of a word or between words. For example, 제주도 날씨 'weather in Jeju Island' can have such forms as 제^주도 날씨 or 제주도^날씨.

   (c) Construct joint vocabulary, or a set of words. The words containing ^ are removed from the vocabulary as they are yet incomplete.

   (d) Check the incomplete words one by one and determine their real form. If the word without the ^ is present in the vocabulary, the ^ is removed. Otherwise, the ^ is replaced by space. In the above examples, 제^주도 날씨 becomes 제주도 날씨 as 제주도 is highly likely to appear somewhere else in the text, while 제주도^날씨 becomes 제주도 날씨 as 제주도날씨 is not a (correct) single word.

4. Split punctuation marks into separate tokens.

5. Change private-use unicode characters into standard ones. Original text makes use of private-use areas in unicode to represent Araea (ᄋᆞ), a letter not used in contemporary Korean any longer. Not only can it cause unexpected issues but it is also against the unicode standard.

6. Shuffle and split the data into train, dev, and test sets. To avoid samples that are too short, the dev set and the test set are to have sentences of five words or more.

[3] Throughout this paper, Korean means standard Korean.

[4] `https://sites.google.com/a/hawaii.edu/jejueo`

[5] `https://en.wikipedia.org/wiki/Jeju_language`

[6] `http://www.jst.re.kr/`

[7] `https://www.zamzar.com/convert/pdf-to-txt/`

	Total	Train	Dev	Test
# sentences	170,356	160,356	5,000	5,000
# jje words	1,421,723	1,298,672	61,448	61,603
# kor words	1,423,836	1,300,489	61,541	61,806
# jje word forms	161,200	151,699	17,828	18,029
# kor word forms	110,774	104,874	14,362	14,595

Table 1: Statistics of JIT dataset. "# words" refers to the number of all tokens in the corpus, and "# word forms" refers to the number of all *unique* tokens (i.e., word types) in the corpus.

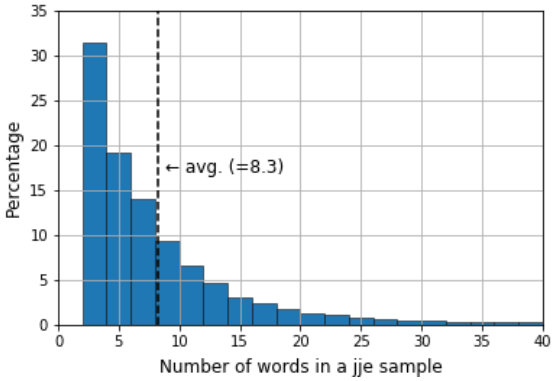

Figure 3: Number of words in a Jejueo sample of JIT dataset.

As a result, we have 160,356, 5,000, and 5,000 Jejueo-Korean sentence pairs for train, dev, and test, respectively, as summarized in Table 1. One thing to note is that the number of word forms in Jejueo are much larger than that in Korean (161,200 $>$ 110,774) although the total number of words in them is almost equal (1.4m). This is likely related to the fact that Jejueo speakers frequently use Korean as well as Jejueo, while Korean speakers do not. This will be further discussed in Section 5.

The length of Jejueo sentences ranges from 1 to 770 words. As can be seen in Figure 3, however, most of them are 15 words or less. The average length is 8.3 words. Korean sentences show similar statistics.

4. JSS (Jejueo Single Speaker Speech) Dataset

4.1. Script

We take the Jejueo text in the JIT dataset as the script for our speech dataset, *Jejueo Single Speaker Speech* dataset (JSS). First, we randomly extract 10,000+ Jejueo sentences from the JIT dataset. To make the dataset more amenable to the training of speech synthesis models, we filter out ones which have more than 35 words or less than 3 words. Then the sentences that include any characters except space, Hangul, and punctuation marks are excluded as well. The final 10,000 sentences with their length information are written to a file in the tab separated format (`tsv`). As in Table 2, the final 10k sentences are 9.4 words long on average. They amount to 94k words, or 335k characters.

	Total	Avg.	Min.	Max.
# samples	10,000	-	-	-
# words	94,415	9.4	3	35
# characters	335,739	33.6	15	105
audio length	13h 47m	5.0s	1.1s	18.4s

Table 2: Statistics of JSS dataset.

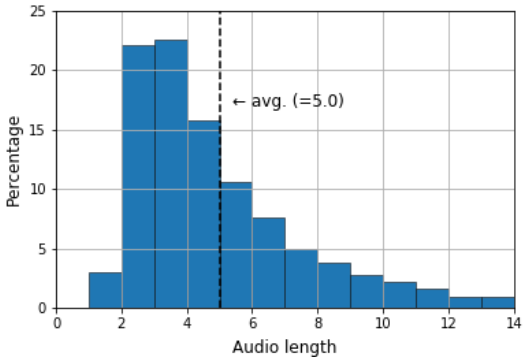

Figure 4: Durations of audio clips in JSS dataset. Most are between 2 and 8 seconds long. The average length of an audio clip is 5 seconds.

4.2. Audio

We have an amateur voice actor record the script. He, in his thirties, was born in a rural area in Jeju and lived there until he was twenty. Although currently he does not stay in Jeju, he regularly visits his family back in Jeju, and speaks with them in Jejueo. He is instructed to read the script line by line as clearly and naturally as possible. Each sentence is saved as a `wav` file sampled at 44100 Hz. He works at his own pace for two months using his home recording devices. We trim the leading and trailing silence in the audio files using librosa[8]. Finally, audio length is added to every line of the script.

The audio files are 13 hours and 47 minutes in total duration (Table 2). Figure 4 shows the distribution of the audio length. The shortest and the longest audio clips are 1.1 and 18.4 seconds long, respectively. Most of them, approximately 80%, are 2 to 8 seconds long. The average length of an audio clip is 5 seconds.

5. Jejueo-Korean Machine Translation

Using the JIT dataset, we train machine translation models between Jejueo and Korean. We consider translation in both directions, kor $\rightarrow$ jje and jje $\rightarrow$ kor, and evaluate the performance of each model by computing the BLEU scores (Papineni et al., 2002) on the dev/test set.

5.1. Model & Setup

Throughout our experiments, we use the Transformer (Vaswani et al., 2017), a state-of-the-art model for neural machine translation. The Transformer is a deep sequence-to-sequence (seq2seq) architecture primarily based on attention mechanisms, including both an encoder-decoder at-

[8] `https://librosa.github.io/librosa/`

tention (Bahdanau et al., 2015; Luong et al., 2015) and self-attention (Lin et al., 2017).

We follow the original parameter settings of the standard Transformer model: 6 encoder and decoder blocks, each with 512-2048 hidden units across 8 attention heads. We run all of our experiments using FAIRSEQ[9] (Ott et al., 2019), a PyTorch-based library for deep sequence models. Details of the training procedure, including all hyperparameters, can be found in our GitHub repository.

5.2. Choosing Optimal Vocabulary Size

Byte Pair Encoding (BPE) is a simple data compression technique that iteratively replaces the most frequent pair of bytes in text with a single, unused byte (Gage, 1994). Since (Sennrich et al., 2016b) successfully applied it to neural machine translation models, it has been a *de facto* standard in the word segmentation for machine translation. Therefore, we also apply BPE to our models.

We first run an experiment to determine the optimal BPE vocabulary size for Jejueo-Korean translation. For various vocabulary size options, we tokenise input text using SentencePiece[10] (Kudo and Richardson, 2018). The vocabulary is shared between the encoder and the decoder.

In Table 3, we summarize our results using five vocabulary sizes: 2k, 4k, 8k, 16k, and 32k. We find that using 4k vocabulary size leads to the best BLEU scores on the dev/test set for both kor → jje (44.85/43.31) and jje → kor (69.35/67.70), although they are within a point difference for 2k and 8k vocabulary sizes. Performance degrades for using larger vocabulary sizes: by approximately 1 point for 16k and another 1 point for 32k.

Lang. Pair	# Vocab.	Dev	Test
kor → jje	2k	44.80	43.26
	4k	**44.85**	**43.31**
	8k	44.40	43.03
	16k	43.33	42.08
	32k	42.57	41.07
jje → kor	2k	69.05	67.63
	4k	**69.35**	**67.70**
	8k	69.02	67.46
	16k	67.61	66.30
	32k	66.32	65.08

Table 3: BLEU scores of models according to the different BPE vocabulary size. SentencePiece is used for BPE segmentation. All hyperparameters except the vocabulary size are identical.

Lang. Pair	Model	Dev	Test
kor → jje	Copy	24.06	24.44
	JIT	44.85	43.31
	JIT + KorWiki	**45.25**	**44.19**
jje → kor	Copy	24.07	24.45
	JIT	69.35	67.70
	JIT + KorWiki	**69.59**	**67.94**

Table 4: BLEU scores of various translation models. In the Copy model, translation outputs are copied from the source. For the JIT + KorWiki model, 160,356 Korean sentences extracted from a Wikidump are added to both source and target sides of the JIT dataset. The vocabulary size is fixed to 4k.

5.3. Comparison with Copy Models

Using the Transformer model with 4k vocabulary size, we present our main baselines in Table 4. As a simple baseline, we include a copying model that predicts its input as its output ("Copy"). The copying model already achieves 24.44 and 24.45 BLEU scores on the kor → jje and jje → kor test sets respectively. By training a Transformer model on the JIT dataset ("JIT"), the scores significantly improve to 43.31 and 67.70 respectively, as we illustrated in Section 5.2.

We remark that the BLEU scores of the jje → kor models (65-67) are much higher than those of the kor → jje models (41-43). One possible explanation is that Korean as well as Jejueo is frequently used in Jejueo dialogues as we discussed in Section 2. For example, when 아버지 'father' appears in the Korean text of the JIT dataset, 아버지 co-occurs 530 times in the paired Jejueo text, while the Jejueo equivalent, 아방, does only 332 times. In short, that a Korean word can correspond to either a Jejueo counterpart or itself makes the kor → jje translation harder than the other direction.

The fact that copying models without training achieve non-trivial BLEU scores implies that the JIT model may benefit from additional training on a copying task. To test this idea, we follow the approach taken by (Sennrich et al., 2016a) and augment both the source and target sides of the training set with the same number of randomly sampled Korean sentences from a Wikidump[11] ("JIT + KorWiki"). This further improves the dev/test set BLEU scores by up to 0.88 points: 44.19 for kor → jje and 67.94 for jje → kor.

6. Jejueo Speech Synthesis

6.1. Model

We train a Jejueo Text-To-Speech (TTS) model called DCTTS (Tachibana et al., 2018), on the JSS dataset. For the past years there have been many neural TTS models such as WaveNet (Oord et al., 2016), Tacotron 1 & 2 (Wang et al., 2017; Shen et al., 2018), Char2Wav (Sotelo et al., 2017), DeepVoice 1-3 (Arik et al., 2017; Gibiansky et al., 2017; Ping et al., 2017), and VoiceLoop (Taigman et al., 2017). Among them, DCTTS is lightweight and fast because it is made up of convolution layers only. Besides, thanks to several tricks such as guided attention and incremental attention, its training is stable. With our working implementation which were already successfully used in (Park and Mulc, 2019), we conduct experiments on the JSS dataset.

For training each model, we mostly adopt the hyperparameters in (Tachibana et al., 2018). Compared to the original implementation, we additionally add dropout (Srivastava et

[9] https://github.com/pytorch/fairseq

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

[11] https://dumps.wikimedia.org/kowiki/20190601/

Token Type	Unicode Range	# Vocab.	Length	Example 1	Example 2	**MCD** (mean / std.)
character	U+AC00-D7AF	1,412	34	국	쉐똥	14.47/0.59
Hangul Jamo	U+1100-11FF	74	64	ᄀ$^{\text{onset}}$ᅮᄀ$^{\text{coda}}$	쉐ᄠ노ᄋ	**14.32/0.38**
Hangul Jamo (S)	U+1100-11FF	59	65	ᄀ$^{\text{onset}}$ᅮᄀ$^{\text{coda}}$	쉐ᄃᄃ노ᄋ	14.34/0.43
HCJ	U+3130-318F	57	64	ㄱㅜㄱ	ㅅㅞㅃㄴㅗㅇ	14.46/0.62
HCJ (S)	U+3130-318F	44	65	ㄱㅜㄱ	ㅅㅞㄷㄷㄴㅗㅇ	14.48/0.44

Table 5: Mel Cepstrum Distortion values on the 100 test samples. (S) denotes single consonants only. Note that although the examples of Hangul Jamo and HCJ may look the same, actually they are different in code point. The MCD values of Jamo are the lowest. The lower, the better.

al., 2014) of 0.05 to every layer for regularization. We train all models for 200k steps. Among the 10k JSS samples, the last 100 samples are held out for test.

6.2. Finding the Best Token Type

In most neural TTS systems, either graphemes (spelling) or phonemes (pronunciation) are taken as input. For a script that is not phonetic, e.g., Chinese characters, grapheme-to-phoneme conversion is considered compulsory. However, as Hangul is phonetic, in other words, text in Hangul sounds as it is written, we stick with graphemes rather than converting them into phonemes.

Throughout our experiments, we examine which token unit works the best for Jejueo speech synthesis. In truth, a Hangul character is a syllable, and can be decomposed into its constituent vowels and consonants. They are called *Jamo* in Korean. This strategy is helpful for readability in practice, but brings about the following question: do we have to break Hangul syllables into *Jamo* in Jejueo speech synthesis?

Jamo has two character blocks in unicode: Hangul Jamo (U+1100-11FF) and Hangul Compatibility Jamo (HCJ) (U+3130-318F). Their major difference is that, in HCJ, syllable-initial consonants (onset) are reused as syllable-final consonants (coda), whereas in Hangul Jamo onset and coda are two separate sets. In Example 1 of Table 5, the character 국 is decomposed into a vowel (ㅜ) and consonants (both ㄱ) in the Hangul Jamo and HCJ rows. Note that in Hangul Jamo, the onset ㄱ and the coda ㄱ are treated as separate characters unlike in HCJ. A further distinction can be made according to whether or not we break consonant clusters in *Jamo* such as ㅟ, ㅠ, or ㅙ into a sequence of letters, i.e., ㄱㄱ, ㄷㄷ, and ㄴㅈ. In Example 2 of Table 5, the ㅠ in Hangul Jamo and HCJ is segmented into ㄷㄷ in their (S) versions.

In the first five columns of Table 5, we summarize various tokenisation strategies we compare in our experiments.

6.3. Evaluation & Results

TTS systems are commonly evaluated with Mean Opinion Score (MOS), the arithmetic mean over all values in the range 1-5 given by individuals. Although the MOS is widely used, it is inherently weak to biases as it is subjective. Besides, it is costly so scalablity is low. For these reasons, we evaluate the performance of each model using a Mel Cepstral Distortion (MCD) measure in this study. It is the average Euclidean distance between the mel cepstral feature vectors of reference and synthesized audio files. So generally speaking, the lower the MCD value is, the better the audio quality is.

For each model, we synthesize 100 audio samples based on the last 100 lines of the script that were not used for training. As shown in Table 5, we find that the MCD mean values of the Hangul Jamo model are the lowest of all. In other words, the audios synthesized by the Hangul Jamo model are most similar to the original ones. We believe it is because the Hangul Jamo model has more granular information than all the others. In terms of granularity, the Hangul Jamo model performs better than the character model, possibly because in the latter vowels and consonants are hidden in the syllable. It also outperforms the HCJ models, as the former has two different sets for a consonant, unlike the latter. Finally, the Hangul Jamo model has more information than its single consonants only version, which replaces consonant clusters with a sequence of single consonants.

7. Conclusion

In this paper, we presented two new Jejueo datasets, JIT and JSS, and explained why and how we developed them. The JIT dataset is bilingual data where 170k+ Jejueo sentences are paired with their Korean translations. The JSS dataset consists of 10k high-quality audio files recorded by a Jejueo speaker and a transcript file. We carried out two follow-up tasks: Jejueo-Korean machine translation and Jejueo speech synthesis using those datasets. In our experiments, neural machine translation models of 4k shared BPE vocabulary and a neural speech synthesis model based on Hangul Jamo tokens showed the best performance. We hope that our datasets will attract a lot of attention from both language and machine learning communities.

8. Acknowledgements

We express our deep respect to Center for Jeju Studies for their devotion to 제주어구술자료집 where this project began. We also thank Soo Kyung Lee of Kakao Brain for her support.

9. References

Arik, S. Ö., Chrzanowski, M., Coates, A., Diamos, G., Gibiansky, A., Kang, Y., Li, X., Miller, J., Ng, A., Raiman, J., et al. (2017). Deep voice: Real-time neural text-to-speech. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 195–204. JMLR. org.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *ICLR*.

Bender, E. M. (2019). The #benderrule: On naming the languages we study and why it matters. https://thegradient.pub/the-benderrule-on-naming-the-languages-we-study-and-why-it-matters/.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*.

Gage, P. (1994). A new algorithm for data compression. *C Users J.*, 12(2):23–38, February.

Gibiansky, A., Arik, S., Diamos, G., Miller, J., Peng, K., Ping, W., Raiman, J., and Zhou, Y. (2017). Deep voice 2: Multi-speaker neural text-to-speech. In *Advances in neural information processing systems*, pages 2962–2970.

Gu, J., Hassan, H., Devlin, J., and Li, V. O. (2018). Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354, New Orleans, Louisiana, June. Association for Computational Linguistics.

Kudo, T. and Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B., and Bengio, Y. (2017). A structured self-attentive sentence embedding. In *ICLR*.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *EMNLP*.

O'Grady, W. (2015). Jejueo: Korea's other language. In *World Congress Of Korean Studies*, pages 1–10.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). 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.

Park, K. and Mulc, T. (2019). Css10: A collection of single speaker speech datasets for 10 languages. *Interspeech*.

Ping, W., Peng, K., Gibiansky, A., Arik, S. O., Kannan, A., Narang, S., Raiman, J., and Miller, J. (2017). Deep voice 3: Scaling text-to-speech with convolutional sequence learning. *ICLR*.

Saltzman, M. (2017). Jejueo talking dictionary: A collaborative online database for language revitalization. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 122–129, Honolulu, March. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Neural machine translation of rare words with subword units. In *ACL*.

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerrv-Ryan, R., et al. (2018). Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE.

Sotelo, J., Mehri, S., Kumar, K., Santos, J. F., Kastner, K., Courville, A., and Bengio, Y. (2017). Char2wav: End-to-end speech synthesis. *ICLR*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *NIPS*.

Tachibana, H., Uenoyama, K., and Aihara, S. (2018). Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4784–4788. IEEE.

Taigman, Y., Wolf, L., Polyak, A., and Nachmani, E. (2017). Voiceloop: Voice fitting and synthesis via a phonological loop. *ICLR*.

Vaswani, A., Shazeer, N., Parmar, N., Jones, L., Uszkoreit, J., Gomez, A. N., and Kaiser, L. u. (2017). Attention is all you need. In *NIPS*.

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agiomyrgiannakis, Y., Clark, R., and Saurous, R. A. (2017). Tacotron: Towards end-to-end speech synthesis. In *Interspeech*.

Yang, C., O'Grady, W., and Yang, S. (2017). Toward a linguistically realistic assessment of language vitality: The case of jejueo. *Language Documentation and Conservation*, 11:103–113, 01.

Yang, C., O'Grady, W., Yang, S., Hilton, N. H., Kang, S.-G., and Kim, S.-Y. (2018a). Revising the language map of korea. *Handbook of the Changing World Language Map*, pages 215–229.

Yang, C., Yang, S., and O'Grady, W. (2018b). Integrating analysis and pedagogy in the revitalization of jejueo. *Japanese-Korean Linguistics*.

Zoph, B., Yuret, D., May, J., and Knight, K. (2016). Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–

1575, Austin, Texas, November. Association for Computational Linguistics.