Analyzing Learned Representations of a Deep ASR Performance Prediction Model

Zied Elloumi, Laurent Besacier, Olivier Galibert, Benjamin Lecouteux

Analyzing Learned Representations of a Deep ASR Performance Prediction Model

Zied Elloumi[1,2] **Laurent Besacier**[2] **Olivier Galibert**[1] **Benjamin Lecouteux**[2]

[1]Laboratoire national de métrologie et d'essais (LNE) , France
[2]Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France
firstname.name@lne.fr
firstname.name@univ-grenoble-alpes.fr

Abstract

This paper addresses a relatively new task: prediction of ASR performance on unseen broadcast programs. In a previous paper, we presented an ASR performance prediction system using CNNs that encode both text (ASR transcript) and speech, in order to predict word error rate. This work is dedicated to the analysis of speech signal embeddings and text embeddings learnt by the CNN while training our prediction model. We try to better understand which information is captured by the deep model and its relation with different conditioning factors. It is shown that hidden layers convey a clear signal about speech style, accent and broadcast type. We then try to leverage these 3 types of information at training time through multi-task learning. Our experiments show that this allows to train slightly more efficient ASR performance prediction systems that - in addition - simultaneously tag the analyzed utterances according to their speech style, accent and broadcast program origin.

1 Introduction

Predicting automatic speech recognition (ASR) performance on unseen speech recordings is an important Grail of speech research. In a previous paper (Elloumi et al., 2018), we presented a framework for modeling and evaluating ASR performance prediction on unseen broadcast programs. CNNs were very efficient encoding both text (ASR transcript) and speech to predict ASR word error rate (WER). However, while achieving state-of-the-art performance prediction results, our CNN approach is more difficult to understand compared to conventional approaches based on engineered features such as *TransRater*[1] for instance. This lack of interpretability of the representations learned by deep neural networks is a general problem in AI. Recent papers started to address this issue and analyzed hidden representations learned during training of different natural language processing models (Mohamed et al., 2012; Wu and King, 2016; Belinkov and Glass, 2017; Shi et al., 2016; Belinkov et al., 2017; Wang et al., 2017).

Contribution. This work is dedicated to the analysis of speech signal embeddings and text embeddings learnt by the CNN during training of our ASR performance prediction model. Our goal is to better understand which information is captured by the deep model and its relation with conditioning factors such as speech style, accent or broadcast program type. For this, we use a data set presented in (Elloumi et al., 2018) which contains a large amount of speech utterances taken from various collections of French broadcast programs. Following a methodology similar to (Belinkov and Glass, 2017), our deep performance prediction model is used to generate utterance level features that are given to a shallow classifier trained to solve secondary classification tasks. It is shown that hidden layers convey a clear signal about speech style, accent and show. We then try to leverage these 3 types of information at training time through multi-task learning. Our experiments show that this allows to train slightly more efficient ASR performance prediction systems that - in addition - simultaneously tag the analyzed utterances according to their speech style, accent and broadcast program origin.

Outline. The paper is organized as follows. In section 2, we present a brief overview of related works and present our ASR performance prediction system in section 3. Then, we detail our methodology to evaluate learned representations in section 4. Our multi-task learning experiments for ASR performance prediction are presented in section 5. Finally, section 6 concludes this work.

[1]https://github.com/hlt-mt/TranscRater

2 Related works

Several works tried to understand learned representations for NLP tasks such as Automatic Speech Recognition (ASR) and Neural Machine Translation (NMT).

(Shi et al., 2016) and (Belinkov et al., 2017) tried to better understand the hidden representations of NMT models which were given to a shallow classifier in order to predict syntactic labels (Shi et al., 2016), part-of-speech labels or semantic ones (Belinkov et al., 2017). It was shown that lower layers are better at POS tagging, while higher layers are better at learning semantics. (Mohamed et al., 2012) and (Belinkov and Glass, 2017) analyzed the feature representations from a deep ASR model using t-SNE visualization (Maaten and Hinton, 2008) and tried to understand which layers better capture the phonemic information by training a shallow phone classifier. Also relevant is the work of (Wang et al., 2017) who proposed an in-depth investigation on three kinds of speaker embeddings learned for a speaker recognition task, i.e. i-vector, d-vector and RNN/LSTM based sequence-vector (s-vector). Classification tasks were designed to facilitate better understanding of the encoded speaker representations. Multi-task learning was also proposed to integrate different speaker embeddings and improve speaker verification performance.

3 ASR performance prediction system

In (Elloumi et al., 2018), we proposed a new approach using convolution neural networks (CNNs) to predict ASR performance from a collection of heterogeneous broadcast programs (both radio and TV). We particularly focused on the combination of text (ASR transcription) and signal (raw speech) inputs which both proved useful for CNN prediction. We also observed that our system remarkably predicts WER distribution on a collection of speech recordings.

To obtain speech transcripts (ASR outputs) for the prediction model, we built our own French ASR system based on the KALDI toolkit (Povey et al., 2011). A hybrid HMM-DNN system was trained using 100 hours of broadcast news from *Quaero*[2], *ETAPE* (Gravier et al., 2012), *ESTER 1 & ESTER 2* (Galliano et al., 2005) and *REPERE* (Kahn et al., 2012) collections. ASR performance was evaluated on the held out corpora presented in table 2 (used to train and evaluate ASR prediction) and its averaged value was 22.29% on the TRAIN set, 22.35% on the DEV set and 31.20% on the TEST set (which contains more challenging broadcast programs).

Figure 1 shows our network architecture. The network input can be either a pure text input, a pure signal input (raw signal) or a dual (text+speech) input. To avoid memory issues, signals are downsampled to 8khz and models are trained on six-second speech turns (shorter speech turns are padded with zeros). For text input, the architecture is inspired from (Kim, 2014) (green in Figure 1): the input is a matrix of dimensions 296x100 (296 is the longest ASR hypothesis length in our corpus ; 100 is the dimension of pre-trained word embeddings on a large held out text corpus of 3.3G words). For speech input, we use the best architecture (*m18*) proposed in (Dai et al., 2017) (colored in red in Figure 1) of dimensions 48000 x 1 (48000 samples correspond to 6s of speech).

For WER prediction, our best approach (called $\text{CNN}_{Softmax}$) used $softmax$ probabilities and an external fixed WER_{Vector} which corresponds to a discretization of the WER output space (see (Elloumi et al., 2018) for more details). The best performance obtained is 19.24% MAE[3] using text+speech input. Our ASR prediction system is built using both *Keras* (Chollet et al., 2015) and *Tensorflow*[4].

In the next section, we analyze the representations learnt in the higher layers (3 blocks colored in yellow and dotted in Figure 1) for pure text (TXT), pure speech (RAW-SIG) and both (TXT+RAW-SIG).

4 Evaluating learned representations

4.1 Methodology

In this section, we attempt to understand what our best ASR performance prediction system (Elloumi et al., 2018) learned. We analyze the text and speech representations obtained by our architecture. Alike (Belinkov and Glass, 2017), the joint text+speech model is used to generate utterance

[2]http://www.quaero.org

[3]Mean Absolute Error (MAE) is a common metric to evaluate WER prediction ; it computes the absolute deviation between the true and predicted WERs, averaged over the number of utterances in the test set.

[4]https://www.tensorflow.org

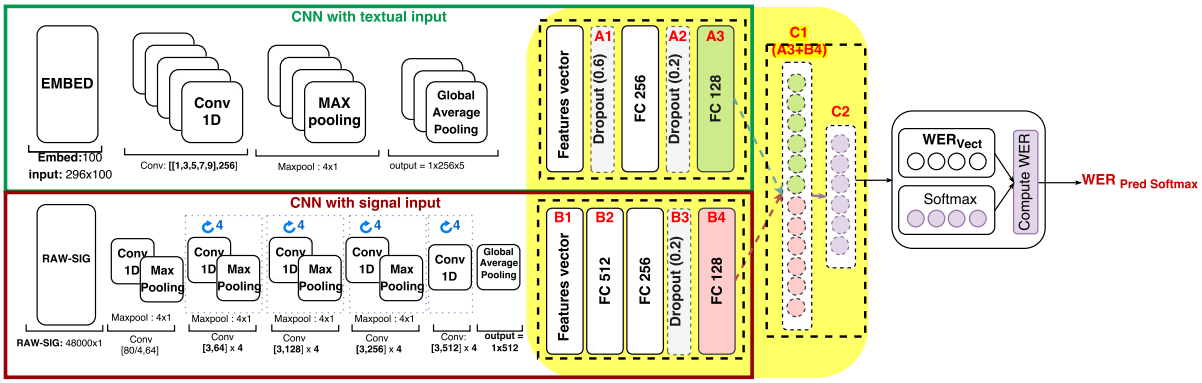


Figure 1: Architecture of our CNN with text (green) and signal (red) inputs for WER prediction

level features (hidden representations of speech turns colored in yellow in Figure 1) that are given to a shallow classifier trained to solve secondary classification tasks such as:

- **STYLE:** classify the utterances between (*spontaneous* and *non spontaneous*) styles (see table 1),
- **ACCENT:** classify the utterances between *native* and *non native* speech (see also table 1, we used the speaker annotations provided with our datasets in order to label our utterances in native/non native speech),
- **SHOW:** classify the utterances in different broadcast programs (as described in table 2, each utterance of our corpus is labeled with a *broadcast program name*).

As a more visual analysis, we also plot an example of hidden representations projected to a 2-D space using t-distributed Stochastic Neighbor Embedding (t-SNE) (Maaten and Hinton, 2008).[5]

4.2 Shallow classifiers

We built three shallow classifiers (SHOW, STYLE, ACCENT) with a similar architecture. The classifier is a feed-forward neural network with one hidden layer (size of the hidden layer is set to 128) followed by dropout (rate of 0.5) and a ReLU non-linearity. Finally, a $softmax$ layer is used for mapping onto the label set size. We chose this simple formulation as we are interested in evaluating the quality of the representations learned by our ASR prediction model, rather than optimizing the secondary classification tasks.

[5] https://lvdmaaten.github.io/tsne/code/tsne_python.zip

The network input size depends on which layer to analyze (see figure 1). Training is performed using *Adam* (Kingma and Ba, 2014) (using default parameters) over shuffled mini-batches in order to minimize the cross-entropy loss. The models are trained for 30 epochs with a batch size of 16 speech utterances. After training, we keep the model with the best performance on DEV set and report its performance on the TEST set. The classifier outputs are evaluated in terms of accuracy.

4.3 Data

A data set from (Elloumi et al., 2018) was employed in our experiments, divided into three subsets: training (TRAIN), development (DEV) and test (TEST). Speech utterances come from various French broadcast collections gathered during projects or shared tasks: *Quaero*, *ETAPE*, *ESTER 1 & ESTER 2* and *REPERE*.

The TEST set contains unseen broadcast programs that are different from those present in TRAIN and DEV (Elloumi et al., 2018).

Category	TRAIN	DEV	TEST
Non Spontaneous	54250	6101	**3109**
Spontaneous	**13277**	**1403**	3728
Native	44487	4945	5298
Non Native	**23040**	**2559**	**1539**

Table 1: Distribution of our utterances between non spontaneous and spontaneous styles, native and non native accents

Tables 1 and 2 show the whole data set in terms of speech turns available for each classification task. We clearly see that the data is unbalanced for the three categories (STYLE, ACCENT, SHOW). Since we are interested in evaluating the discriminative power of our learned representations for

Show	TRAIN	DEV
FINTER-DEBATE	7632	833
FRANCE3-DEBATE	928	77
LCP-PileEtFace	**4487**	525
RFI	25565	2831
RTM	24198	2745
TELSONNE	4717	**493**
Total	67527	7504

Table 2: Number of utterances for each broadcast program

these 3 tasks, we extracted a balanced version of our TRAIN/DEV/TEST sets by filtering among over-represented labels (final number of kept utterances corresponds to bold numbers in table 1 and 2). Table 3 shows the distribution of our final balanced TRAIN/DEV/TEST sets as well as the number of categories for each task.[6]

	#Catg	Turns of speech per category TRAIN	DEV	TEST
SHOW	5	$4487_{\times 5}$	$493_{\times 5}$	-
STYLE	2	$13277_{\times 2}$	$1403_{\times 2}$	$3109_{\times 2}$
ACCENT	2	$23040_{\times 2}$	$2559_{\times 2}$	$1539_{\times 2}$

Table 3: Description of our balanced data set for each category

4.4 Results

For each classification task, we build a shallow classifier using the hidden representations of *TXT*, *RAW-SIG* and *TXT+RAW-SIG* blocks as input. The experimental results are presented in table 4 for both DEV and TEST sets separated by two vertical bars (||).

Classification performance is all above a random baseline accuracy (>50% for STYLE and ACCENT and >20% for SHOW). This shows that training a deep WER prediction system gives representation layers that contain a meaningful amount of information about speech style, speech accent and broadcast program label. Predicting utterance style (spontaneous/non spontaneous) is slightly easier than predicting accent (native/non native) especially from text input. One explanation might be that speech utterances are short ($< 6s$) while accent identification needs probably longer sequences. We also observe that using both text and speech improves the learned representations for the STYLE task while it is less clear for the ACCENT task (for which improvement seen on DEV is not confirmed on TEST). Finally, text input is significantly better than speech input whereas we could have expected better performance from speech for the SHOW task (speech signals convey information about the audio characteristics of a broadcast program). It means that text input contains correlated information with broadcast-program type, speech style and speaker's accent. In case of SHOW task, our performance prediction system is able to capture information (vocabulary, topic, syntax, etc.) about a specific broadcast program type, based on textual features and to differ it from others (radio programs, TV debate programs, phone calls, broadcast news programs, etc.). Likewise, the textual information captured is very different between spontaneous/non-spontaneous speech styles and native/non-native speaker's accents.

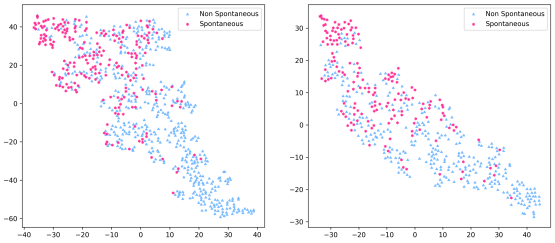
Among the representations analyzed, the outputs of the CNNs (A1,B1) lead to the best classification results, in line with previous findings about convolutions as feature extractors. Performance then drops using the higher (fully connected) layers that do not generate better representations for detecting style, accent or show.

Layer	Dim.	SHOW	STYLE	ACCENT
TXT				
A1	1280	**57.12**||-	**80.72**||68.99	**70.75**||66.54
A2	256	54.89||-	80.01||**69.56**	69.30||69.43
A3	128	51.04||-	79.23||68.27	68.25||**70.89**
RAW-SIG				
B1	512	**42.35**||-	**72.92**||**58.64**	64.60||**55.85**
B2	512	41.22||-	72.20||58.41	64.44||54.84
B3	256	41.22||-	72.38||58.44	64.50||54.65
B4	128	40.77||-	72.38||58.52	**64.74**||54.87
TXT + RAW-SIG				
C1 (A3+B4)	256	**57.04**||-	**81.29**||70.36	**71.41**||**65.98**
C2	128	53.06||-	79.62||**70.55**	70.01||65.20
Random	-	**20.00**	**50.00**	**50.00**

Table 4: Show/Style/Accent classification accuracies using representations from different layers learned during the training of our ASR WER prediction system.

We visualize an example of utterance representations from C2(TXT+RAW-SIG) layer in figure 2 using the t-SNE. For a fixed utterance duration 4s≤D<5s (716 speech turns) and 5s≤D<6s (489 speech turns), non spontaneous utterances are plotted in blue while spontaneous ones are in pink. The C2 layer produces clusters which shows that spontaneous utterances are in the upper-left part

[6]For the *SHOW* classification task, the *FRANCE3-DEBATE* shows were finally removed since they represent a too small amount of speech turns.



(a) 4s≤D<5s (b) 5s≤D<6s

Figure 2: Visualization of utterance representations from C2 layer for different speech styles (S spontaneous - NS non spontaneous) - (a) utt. length is 4s≤D<5s and (b) 5s≤D<6s

of the 2D space. This suggests that C2 hidden representation captures a weak signal about speaking style.

Finally, figure 3 is the confusion matrix produced using C2(TXT+RAW-SIG) layer. The classifiers very well predicted *TELSONNE* category (Accuracy of 82%), which contains many phone calls from the radio listeners. This show is rather different from the 4 other shows in DEV (broadcast debates and news).

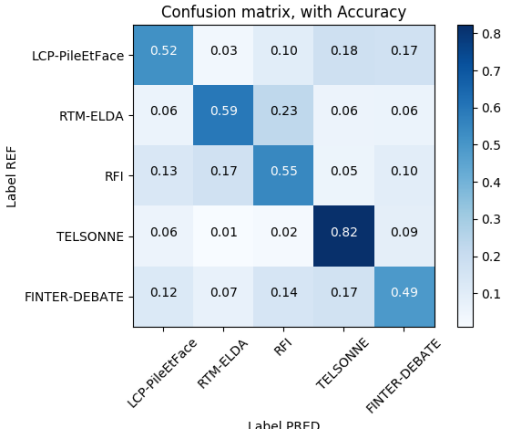


Figure 3: Confusion matrix for SHOW classification using C2(TXT+RAW-SIG) layer as input, evaluated on DEV

5 Multi-task learning

We have seen in the previous section that, while training an ASR performance prediction system, hidden layers convey a clear signal about speech style, accent and show. This suggests that these 3 types of information might be useful to structure the deep ASR performance prediction models. In this section, we investigate the effect of knowledge of these labels (style, accent, show) at training time on prediction systems qualities. For this, we perform multi-task learning providing the additional information about broadcast type, speech style and speaker's accent during training. The architecture of the multi-task model is similar to the single-task WER prediction model of Figure 1 but we add additional outputs: a $softmax$ function is added for each new classification task after the last fully connected layer (C2). The output dimension depends on the task: 6 for SHOW and 2 for STYLE and ACCENT tasks.

We use the full (unbalanced) data set described in tables 1 and 2. Training of the multitask model uses $Adadelta$ update rule and all parameters are initialized from scratch (8.70M). Models are performed for 50 epochs with batch size of 32. MAE is used as the loss function for WER prediction task while cross-entropy loss is used for the classification tasks.

In the composite (multitask) loss, we assign a weight of 1 for MAE loss (main task) and a smaller weight of 0.3 (tuned using a grid search on DEV dataset) for cross-entropy (secondary classification task) loss(es).

After training, we take the model that lead to the best *MAE* on *DEV* set and report its performance on TEST. We build several models that simultaneously address 1, 2, 3 and 4 tasks. The models are evaluated with a specific metric for each task: MAE & Kendall[7] for WER prediction task and Accuracy for classification tasks.

Table 5 summarizes the experimental results on DEV and TEST sets, separated by two vertical bars (||). We considered the mono-task model described in (Elloumi et al., 2018) (and summarized in section 3) as a baseline system.

We recall that we evaluated the SHOW classification task only on the DEV set (TEST broadcast programs are new and were unseen in the TRAIN).

First of all, we notice that performance of classification tasks in muti-task scenarios are very good: we are able to train efficient ASR performance prediction systems that simultaneously tag the analyzed utterances according to their speech style, accent and broadcast program origin. Such multi-task systems might be useful diagnostic tools to analyze and predict ASR on large speech collections. Moreover, our best multi-task systems dis-

[7]Correlation between true ASR values and predicted ASR values

Models	Performance prediction task MAE	Performance prediction task Kendall	Classification tasks SHOW	Classification tasks STYLE	Classification tasks ACCENT
Baseline: Mono-task					
WER (Elloumi et al., 2018)	15.24||19.24	45.00||46.83	-	-	-
2-task					
WER SHOW	**14.83**||19.15	**47.25**||47.05	**99.29**||-	-	-
WER STYLE	15.07||19.66	45.92||45.49	-	99.01||65.24	-
WER ACCENT	15.05||19.60	46.17||45.60	-	-	91.72|| 75.30
3-task					
WER STYLE ACCENT	15.12||20.23	45.75||44.09	-	98.63||69.07	88.99|| **77.46**
WER SHOW ACCENT	14.94||19.76	46.19||43.61	98.38||-	-	89.87||71.44
WER SHOW STYLE	14.90||**19.14**	45.87||**47.28**	99.12||-	**99.47**||**81.98**	-
4-task					
WER SHOW STYLE ACCENT	15.15||19.64	45.59||45.42	99.04||-	99.29||81.55	**91.92**||73.60
WER ALL COMBINED OUTPUTS	**14.50** ||**18.87**	**48.16**||**48.63**	-	-	-

Table 5: Evaluation of ASR performance prediction with multi-tasks models ($DEV||TEST$) computed with MAE and Kendall - secondary classification tasks accuracy is also reported

play a better performance (MAE, Kendall) than the baseline system, which means that the implicit information given about style, accent and broadcast program type can be helpful to structure the system's predictions. For example, in 2-task case, the best model is obtained on WER+SHOW tasks with a difference of +0.41%, +2.25% for MAE and Kendall respectively (on DEV) compared to the baseline on WER prediction task. However, it is also important to mention that the impact of multi-task learning on the main task (ASR performance prediction) is limited: only slight improvements on the test set are observed for MAE and Kendall metrics. Anyway, the systems trained seem complementary since their combination (averaging, over all multi-task systems, predicted WERs at utterance level) leads to significant performance improvement (MAE and Kendall).

6 Conclusion

This paper presented an analysis of learned representations of our deep ASR performance prediction system. Experiments show that hidden layers convey a clear signal about speech style, accent, and broadcast type. We also proposed a multi-task learning approach to simultaneously predict WER and classify utterances according to style, accent and broadcast program origin.

References

Yonatan Belinkov and James Glass. 2017. Analyzing hidden representations in end-to-end automatic speech recognition systems. In *Advances in Neural Information Processing Systems*, pages 2438–2448.

Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1–10.

François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das. 2017. Very deep convolutional neural networks for raw waveforms. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 421–425. IEEE.

Zied Elloumi, Laurent Besacier, Olivier Galibert, Juliette Kahn, and Benjamin Lecouteux. 2018. Asr performance prediction on unseen broadcast programs using convolutional neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Sylvain Galliano, Edouard Geoffrois, Djamel Mostefa, Khalid Choukri, Jean-François Bonastre, and Guillaume Gravier. 2005. The ester phase ii evaluation campaign for the rich transcription of french broadcast news. In *Interspeech*, pages 1149–1152.

Guillaume Gravier, Gilles Adda, Niklas Paulson, Matthieu Carré, Aude Giraudel, and Olivier Galibert. 2012. The etape corpus for the evaluation of speech-based tv content processing in the french language. In *LREC-Eighth international conference on Language Resources and Evaluation*, page na.

Juliette Kahn, Olivier Galibert, Ludovic Quintard, Matthieu Carré, Aude Giraudel, and Philippe Joly. 2012. A presentation of the repere challenge. In *Content-Based Multimedia Indexing (CBMI), 2012 10th International Workshop on*, pages 1–6. IEEE.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

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

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Abdel-rahman Mohamed, Geoffrey Hinton, and Gerald Penn. 2012. Understanding how deep belief networks perform acoustic modelling. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4273–4276. IEEE.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, EPFL-CONF-192584. IEEE Signal Processing Society.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural mt learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534.

Shuai Wang, Yanmin Qian, and Kai Yu. 2017. What does the speaker embedding encode? In *Interspeech*, volume 2017, pages 1497–1501.

Zhizheng Wu and Simon King. 2016. Investigating gated recurrent neural networks for speech synthesis. *CoRR*, abs/1601.02539.