INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Detecting Mismatch Between Speech and Transcription Using Cross-Modal Attention

Qiang Huang, Thomas Hain

Department of Computer Science, University of Sheffield, Sheffield, UK
{q.huang, t.hain}@sheffield.ac.uk

Abstract

In this paper, we propose to detect mismatches between speech and transcriptions using deep neural networks. Although it is generally assumed there are no mismatches in some speech related applications, it is hard to avoid the errors due to one reason or another. Moreover, the use of mismatched data probably leads to performance reduction when training a model. In our work, instead of detecting the errors by computing the distance between manual transcriptions and text strings obtained using a speech recogniser, we view mismatch detection as a classification task and merge speech and transcription features using deep neural networks. To enhance detection ability, we use cross-modal attention mechanism in our approach by learning the relevance between the features obtained from the two modalities. To evaluate the effectiveness of our approach, we test it on Factored WSJCAM0 by randomly setting three kinds of mismatch, word deletion, insertion or substitution. To test its robustness, we train our models using a small number of samples and detect mismatch with different number of words being removed, inserted, and substituted. In our experiments, the results show the use of our approach for mismatch detection is close to 80% on insertion and deletion and outperforms the baseline.

Index Terms: mismatch detection, deep learning, attention

1. Introduction

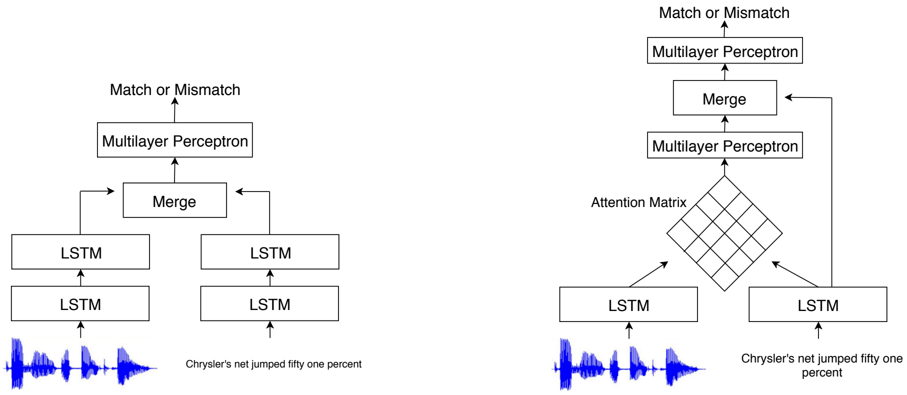
In some speech applications, such as speech recognition [1, 2, 3], speech understanding [4], and retrieval [5, 6], the quality of the transcriptions of speech signals is a very important factor. It is usually assumed that the speech signals are correctly transcribed. However, transcribing a large number of utterances is a very time-consuming and tedious task, and it is hard to avoid the occurrence of differences between speech and transcriptions. We generally consider three types of mismatch. The first type of mismatch is a word spelling error, occurring at character level. The second type is the world-level error caused by word missing, insertion and substitution. The third type occurs at utterance level when mispairing whole utterances and text sequences. In this paper, we focus on tackling the second type of mismatch since this type of mismatch is a common case occurring in manual speech transcriptions and easily leads to performance reduction in some speech related applications. To handle this issue, we view the mismatch detection as a classification task by identifying whether transcriptions contain word deletions, insertions or substitutions.

There have been some studies on tackling word spelling error [7, 8, 9, 10]. In [7], a convolutional neural network was built to correct spelling error using character embeddings. In [8], a word recognition model was trained based on a semi-character level recurrent neural network in order to correct spelling errors by finding links within a word. In [10], Shaona et al. used a model to combine character-level CNN and gated recurrent unit (GRU) encoder along with a word-level GRU attention decoder. In comparison to a word spelling error occurring within a single word, the cause for the second type of mismatch error is complicated. In addition to careless typewriting, in many cases, transcribers might not be able to clearly hear utterances in which voices are corrupted by background noise or there are voice overlaps. In addition, if the transcriber is not a native speaker, the content of the utterance might not be correctly understood. Due to these reasons, the developed methods for the first type of mismatch error might not be very suitable to handle the second case. Some methods [11, 12] tried to handle word level mismatch using language modelling and syntactic parser. In [11], Keisuke et al. designed a dependency parsing scheme to jointly parse a sentence and repair grammatical errors by extending the non-directional transition based formalism. In [12], Park used the whole context to determine proper corrections. However, these methods focused on the application using text only. In [13, 14], Errattahi et al. detected errors in recognition output using word confidence value and features learned from word confusion network. In comparison with the first two types of mismatch, although the mismatch between whole utterances and transcriptions can cause worse effect on speech applications, its occurrence is relatively smaller.

In contrast to those previous studies that use text processing or the information from recognition outputs, we use only speech and raw transcriptions, and expect that the developed approach can be also used for some low-resourced languages in addition to those popular ones. For low-resource languages, there might be no well-designed lexicon, pre-collected large amounts of text documents, or even an expert-defined phoneme set. These make the construction of a speech recogniser difficult if there are many mismatched samples in the training data. In our framework, we use long short-term memory (LSTM) [15] to extract features from raw speech and manual transcriptions and combine the information learned from the two modalities for mismatch detection. To further enhance detection ability, we employ cross-modal attention in our model. Attention mechanisms have been successfully used in image processing [16, 17], language processing [18, 19], and speech recognition [20, 21] due to their abilities to selectively emphasize and ignore information with respect to their targets or some application conditions required in related tasks. Cross-modal attention refers to the distribution of attention to information from different modalities [22]. The use of cross-modal attention mechanism in our work aims to provide a way to increase the possibility to highlight the relevance between the information learned from speech and transcriptions.

The rest of our paper is organised as follows: Section 2 presents the framework of our approach; Section 3 describes data and experimental setup. The results and analysis are given in Section 4, and finally a conclusion is drawn in Section 5.

http://dx.doi.org/10.21437/Interspeech.2019-2125



(a) *Model_1* (b) *Model_2*

Figure 1: *Models for mismatch detection: Model_1 concatenates features from two modalities and Model_2 uses a cross-modal attention mechanism.*

2. Theoretical Framework

Since the information to be processed in our work is from two different modalities, our model not only needs to extract features from each modality, but also to take into account the dependency between them. Figure 1 shows the structure of proposed models using feature merge (Model_1) and a cross-modal attention mechanism (Model_2), respectively. In this figure, each branch represents a network structure to process information stream from single modality. Before going to fully connected layers connecting to the target, in Model_1, we concatenate the two compact representations learned from each modality. In Model_2, we model the local relevance using a cross-modal attention mechanism to align speech and text features.

2.1. Model with Information Merge (Model_1)

The input of two branches in Figure 1(a) are a whole utterance and word sentence ($\mathbf{w} = [w_1, w_2, \ldots, w_n]$), respectively. For speech input, speech signals are converted into audio vectors ($\mathbf{S} = [s_1, s_2, \ldots, s_m]$). For text input, word embedding $E(w_k)$ with dimension d are built to convert word sentence into word vector sequences $\mathbf{W} = [E(w_1), E(w_2), \ldots, E(w_n)]$.

Given the input, Model_1 employs a two-layer LSTM. The first layer is to obtain context dependent hidden states S_i^1 and W_j^1:

$$S_i^1 = LSTM_1(\mathbf{S}, i) \tag{1}$$

$$W_j^1 = LSTM_1(\mathbf{W}, j) \tag{2}$$

where i and j denote the i-th token of audio stream and the j-th token of text sequence, respectively. The second layer is to yield compact representation S^2 and W^2:

$$S^2 = LSTM_2(\mathbf{S}^1) \tag{3}$$

$$W^2 = LSTM_2(\mathbf{W}^1) \tag{4}$$

where $\mathbf{S}^1$ and $\mathbf{W}^1$ denote a collection of word vectors and speech vectors, respectively. After using a merge layer, we thus obtain:

$$Vec_{\{W^2,S^2\}} = Merge(S^2, W^2) \tag{5}$$

2.2. Cross-Modal Attention (Model_2)

Comparing with Model_1, an attention layer is added in Model_2 after the data vectors passing the low-level LSTM encoder. In this attention layer, a relevance matrix (att) is generated by

$$r_{i,j} = F(S_i, W_j) \tag{6}$$

where F denotes a vector dot product and is used to calculate the relevance of any two vectors learned from the two modalities. Once some anomaly cases, such as word deletion, insertion or substitution, occur in transcriptions, it is possible that the change of relevance can be detected with respect to the features learned from the lower layer. To highlight certain features that allow better prediction of the system's internal state, the relevance matrix is normalised by using $softmax$ along one modality axis, and then is used to weight the feature vectors of the other modality:

$$\alpha_{ij} = \frac{\exp(r_{ij})}{\sum_{k=1}^{n}(\exp(r_{ik}))} \qquad W'_j = \sum_{i=1}^{m} \beta_{ij} S_i \tag{7}$$

$$\beta_{ij} = \frac{\exp(r_{ij})}{\sum_{k=1}^{n}(\exp(r_{kj}))} \qquad S'_i = \sum_{j=1}^{n} \alpha_{ij} W_j \tag{8}$$

where α and β are the weight matrices normalised along speech and text axis, respectively. W'_j and S'_i denote the speech output and text output of the attention layer. In order to avoid vanishing gradient problem, we follow the step done by the residual network [23] to concatenate the features learned from previous layer with the output of this attention layer:

$$Vec = Merge(W_j, W'_j) \tag{9}$$

As in our work, anomalies only occur in transcriptions, we only merge the output (W'_j) of the attention layer with the learned text vectors (W_j) before going to the latter fully-connected dense layers.

3. Data and Experimental Setup

3.1. Data

To evaluate our approach, we use the dataset of Factored WSJCAM0 [24], and its data statistics are summarized in Table 1. This version of the WSJCAM0 corpus [25] has augmented variability in 4 factors: speaker, channel, background and Signal-to-Noise Ratio (SNR). It has been created by the University of

Sheffield for experiments in robustness and factorisation under non-stationary noise conditions [24]. Files are single-channel WAVE format, sampled at 16kHz and with a bit depth of 16 bits. In the dataset, each sentence contains 16.9 words and 102 letters in average. We make mismatched samples by randomly deleting , inserting, and substituting words in original training and test data. The training data contains 100 samples, randomly selected from si_tr. The test data for evaluation contains 331 matched samples and 331 mismatched ones.

Table 1: *Data statistics of Factored WSJCAM0*

Set	Description	Speakers	#Sample
si_tr	speaker independent training data	86	7387
si_dt5a	Primary 5K task development set	19	331

3.2. Experimental Setup

In our experiments, word-level and sentence-level mismatches, namely the second and third types of mismatch as mentioned in Section 1, are detected. For the second type of mismatch, the detection performance are tested in the conditions of different number (N$\in \{1, 2, 3, 4\}$) of words being randomly deleted, inserted or substituted. For the detection of sentence-level mismatch, some utterances are paired with noncorresponding transcriptions.

In the experiments of sentence-level mismatch detection, three different input representations are used for word embedding generation [26, 27]. The first is pre-trained vector obtained from Word2Vec [28] and the vector dimension is 300. The second is 26 English letters, and the third is 44 phonemes. The maximum number of words in each sentence is set as 50. All utterances are pre-processed and speech signals are converted into 13 dimensional MFCCs.

The proposed models are optimized using Adam [29] with an initial learning rate of 0.0001. The batch size is 64, with half mismatch and half non-mismatch. The number of epoch is set as 100. As mismatch detection is defined as a binary classification task, we use categorical cross-entropy as a loss function and detection accuracy as an evaluation metric.

3.3. Baseline

As a comparison, the baseline for mismatch detection is done by computing the edit distance (ED)[30] between manual transcriptions and text sequences obtained using a speech recogniser constructed with Kaldi [31]. The speech recogniser consists of a triphone acoustic model and a unigram word based language model trained on 100 samples randomly selected from the training data. The word error rate (WER) is 22% on the training data and 90% the test data. The errors in recognised word sequence (Z_r) is mainly caused by out of vocabulary. Phoneme output (Ph_r) were also obtained using Kaldi in the same conditions. The phone error rate is 15.12% on the training data and 67.8% on the test data. To find out whether there exit a mismatch, we compute $D_{r,g}$ and $D_{r,t}$, the distance between Z_r and the other two word sequences, i.e. word based ground truth Z_g and word mismatch transcriptions Z_t.

$$D_{r,g} = \frac{ED(Z_r, Z_g)}{N_g} \qquad D_{r,t} = \frac{ED(Z_r, Z_t)}{N_t} \tag{10}$$

where N_g and N_t denote the number of word in Z_g and Z_t, respectively. The decision is made by:

$$\begin{cases} mismatch, & \text{if } D_{r,t} \geq threshold \\ no\_mismatch, & \text{if } D_{r,g} < threshold \end{cases}$$

where the value of threshold is determined according to the maximum accuracy obtained on the training data. The same steps can be also used to detect mismatch on phoneme strings.

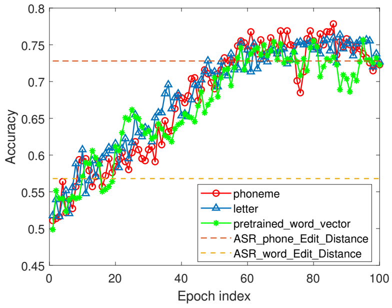
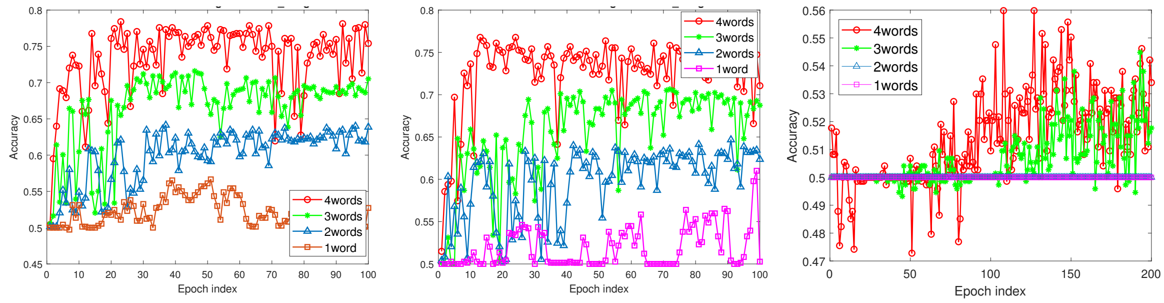


Figure 2: *Accuracy of sentence-level mismatch detection on the test data over 100 epochs.*

4. Results and Analysis

Our experiments starts with sentence-level mismatch detection. Model_1 and three different input representations, i.e. phoneme, letter and pre-trained word vector, were used to tackle sentence-level mismatch. Figure 2 shows utterance-level mismatch detection performance on the test data when using the three different inputs for word embedding generations [26]. This figure shows that the detection accuracy obtained using phoneme is slightly better than the other two. So, in the following experiments phonemes are only input representation. As a comparison, the detection accuracy using word-based and phoneme-based baselines mentioned in Section 3.3 are also shown in this figure. The top dashed line shows the detection accuracy using recognised phoneme sequence, and the bottom dashed line represents the performance using recognised word sequence. It is clear that the baseline using phoneme sequence outperforms the word based one but its accuracy is still lower than that obtained using Model_1.

Figure 3 shows the performances when using Model_1 to test three kinds of word mismatches occurring in manual transcriptions. Further experiments tested four cases when different number of words are deleted, inserted, or substituted. Figure 3(a) shows the detection accuracy can reach 74% and 70% when four and three words are deleted from manual transcriptions. If only one word is deleted, the detection accuracy degrades sharply and its convergence is also worse than the other three curves. This case is probably caused by three factors. The first is the quality of speech data is not good enough. From a speech recognition point of view, one word deletion in a sentence containing 17 words indicates the WER is less than 6%. This means the speech quality should be very good in order to provide enough information to support the match between speech and text. However, Factored WSJCAM0 contains many utterances corrupted by different-level noise and cannot meet this requirement. The second factor is that the number of one word deletion is relatively small in comparison with three or four words being deleted. The third is the number of training



(a) *Del.* (b) *Ins.* (c) *Sub.*

Figure 3: *Mismatch detection accuracy obtained using **Model_1** on the test data with 1,2,3,4 words deletion, insertion and substitution.*

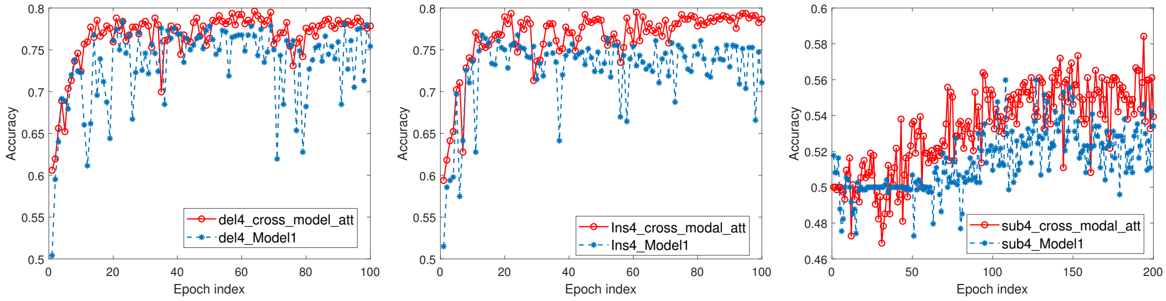


Figure 4: *Mismatch detection accuracy obtained using **Model_2** on the test data with 4 words deletion, insertion and substitution.*

samples is small. These factors also suit to explain insertion mismatch detection. In comparison with word deletion and insertion, the detection performance on word substitution is relatively poor. This may be the case is related to two additional factors besides the reasons mentioned above. The first is some phonemes occurring in original words probably also occur in some words used as substitutions. This reduces the possibility to find out whether there exist mismatches. The second factor is, in comparison with deletions and insertions, the change of total number of phonemes in a transcription sentence is not noticeable after word substitution. This means the length of phoneme sequence can still well match the duration of the corresponding utterance. This probably leads to some features, like the length of a sentence, cannot be well learned.

The three sub-figures in Figure 4 show performance when using cross-modal attention (Model_2) to detect a four-word deletion, insertion and substitution mismatch, respectively. For comparison, these figures also show the results obtained using Model_1 in the same condition. The use of cross-modal attention can yield better and more stable performances on deletion and insertion mismatch detection than Model_1, although the performance improvement on substitution mismatch is slight. In comparison to the use of simple concatenation merge used in Model_1, the improvement means that the cross-modal attention enables the network to learn features relevant to target more effectively by using $softmax$ over a cross-modal relevance matrix.

In Table 2, 4-word mismatch detection performances were compared on the training and test data using baseline, Model_1 and Model_2, respectively. Although the baseline can generate good detection performances on the training data for three kinds of mismatch, the learned threshold on the training data does not help to determine a satisfying performance on the test data. The use of cross-modal attention shows good improvements on the test data over the three kinds of mismatch.

Table 2: *Comparison of 4-word detection accuracy for deletion, insertion and substitutions.*

	Deletion	Insertion	Substitution	Averg.
		Baseline		
train	0.93	0.90	0.927	0.919
test	0.52	0.5	0.5	0.507
(threshold)	(0.35)	(0.25)	(0.3)	
		Model_1		
train	1.0	0.995	0.981	0.992
test	0.761	0.739	0.535	0.678
		Model_2		
train	0.992	0.995	0.984	0.99
test	0.781	0.792	0.558	0.7103

5. Conclusion and Future Work

In this paper an approach was presented to detect sentence-level and word-level mismatch between speech and manual transcriptions without using speech recognition. Two models using deep neural networks were designed to learn features from two modality streams. The first model uses concatenation to combine the features, and the second model uses cross-modal attention to make a soft alignment and highlights the information which is relevant to target. Our experiments were run on Factored WSJCAM0 and the results show the use of our approach gains a good improvement in effectiveness and robustness over the baseline.

In our future work, the study in three aspects will be taken into account. More robust deep neural networks will be employed to handle different types of mismatch between speech and transcriptions, instead of those occurring only in text. The performance of large-sized data will be evaluated in different conditions. Visual information will be used in order to enhance the effectiveness and robustness of our system.

6. References

[1] Y. Gaur, W. S. Lasecki, F. Metze, and J. P. Bigham, "The effects of automatic speech recognition quality on human transcription latency," in *Proceedings of the 13th Web for All Conference*, ser. W4A '16. New York, NY, USA: ACM, 2016, pp. 23:1–23:8.

[2] S. Novotney and C. Callison-Burch, "Cheap, fast and good enough: Automatic speech recognition with non-expert transcription," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, Jun. 2010, pp. 207–215. [Online]. Available: https://www.aclweb.org/anthology/N10-1024

[3] V. H. Do, N. F. Chen, B. P. Lim, and M. Hasegawa-Johnson, "Multi-task learning using mismatched transcription for under-resourced speech recognition," in *InterSpeech'2017*, 2017, pp. 734–738.

[4] A. Stolcke and J. Droppo, "Comparing human and machine errors in conversational speech transcription," in *InterSpeech'2017*, 2017, pp. 127–141.

[5] P. C. Woodland, S. E. Johnson, P. Jourlin, and K. S. Jones, "Effects of out of vocabulary words in spoken document retrieval (poster session)," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2000, pp. 372–374.

[6] J. S. Garofolo, C. G. P. Auzanne, and E. M. Voorhees, "The trec spoken document retrieval track: A success story," in *Content-Based Multimedia Information Access - Volume 1*, Paris, France, 2000, pp. 1–20.

[7] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," *Proceedings of the 30st AAAI Conference on Artificial Intelligence*, pp. 2741–2749, 2016.

[8] K. Sakaguchi, K. Duh, M. Post, and B. V. Durme, "Robsut wrod reocginiton via semi-character recurrent neural network," *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pp. 3281–3287, 2017.

[9] P. Etoori, M. Chinnakotla, and R. Mamidi, "Automatic spelling correction for resource-scarce languagesusing deep learning," *Proceedings of ACL 2018, Student Research Workshop*, pp. 146–152, 2018.

[10] S. Ghosh and P. O. Kristensson, "Neural networks for text correction and completion in keyboard decoding," *https://arxiv.org/pdf/1709.06429.pdf*, 2017.

[11] K. Sakaguchi, M. Post, and B. V. Durme, "Error-repair dependency parsing for ungrammatical texts," *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 189–195, 2017.

[12] P. Etoori, M. Chinnakotla, and R. Mamidi, "Automated whole sentence grammar correction using a noisy channel model," *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pp. 934–944, 2011.

[13] R. Errattahi, A. E. Hannani, H. Ouahmane, and T. Hain, "Automatic speech recognition errors detection using supervised learningtechniques," *the 13th International Conference of Computer Systems and Applications*, 2016.

[14] R. Errattahi, S. Deena, A. E. Hannani, H. Ouahmane, and T. Hain, "Improving asr error detection with rnnlm adaptation," *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 190–196, 2018.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–1780, 12 1997.

[16] X. Wang, Y.-F. Wang, and W. Y. Wang, "Watch, listen, and describe: Globally and locally aligned cross-modalattentions for video captioning," in *Proceedings of NAACL-HLT*. Association for Computational Linguistics, 2018, pp. 795–801.

[17] V. Mnih, N. Heess, A. Graves, and k. kavukcuoglu, "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 2204–2212. [Online]. Available: http://papers.nips.cc/paper/5542-recurrent-models-of-visual-attention.pdf

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[19] A. Ambartsoumian and F. Popowich, "Self-attention: A better building block for sentiment analysis neuralnetwork classifiers," in *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, 2018, pp. 130–139.

[20] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, 2016, pp. 4945–4949.

[21] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015, pp. 577–585. [Online]. Available: http://papers.nips.cc/paper/5847-attention-based-models-for-speech-recognition.pdf

[22] C. W. Robinson and V. Sloutsky, "When audition dominates vision: Evidence from cross-modal statistical learning," *Experimental Psychology*, vol. 60, no. 2, pp. 113–121, 10 2011.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[24] T. Hain and O. Saz, "Factored wsj-cam0 speech corpus," 2013. [Online]. Available: https://mini.dcs.shef.ac.uk/resources/wsjcam0

[25] T. Robinson, J. Fransen, D. Pye, J. Foote, S. Renals, P. Woodland, and S. Young, "Wsjcam0 cambridge read news," 1995. [Online]. Available: https://catalog.ldc.upenn.edu/LDC95S24

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[27] S. Bengio and G. Heigold, "Word embeddings for speech recognition," pp. 1053–1057, 2014.

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," pp. 3111–3119, 2013.

[29] D. P. Kingma and J. L. Ba, "Adam : A method for stochastic optimization," 2014. [Online]. Available: arXiv:1412.6980v9

[30] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. New York: Pearson Education International, 1966.

[31] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, Dec. 2011.