INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Punctuation Prediction in Spontaneous Conversations: Can We Mitigate ASR Errors with Retrofitted Word Embeddings?

Łukasz Augustyniak[12], *Piotr Szymański*[12], *Mikołaj Morzy*[3], *Piotr Żelasko*[4], *Adrian Szymczak*[1], *Jan Mizgajski*[13], *Yishay Carmiel*[1], *Najim Dehak*[4]

[1] Avaya, USA
[2] Wrocław University of Technology, Department of Computational Intelligence, Wrocław, Poland
[3] Poznan University of Technology, Poznan, Poland
[4] Center for Language and Speech Processing, The Johns Hopkins University, Baltimore, MD, USA
{lukasz.augustyniak,piotr.szymanski}@pwr.edu.pl

Abstract

Automatic Speech Recognition (ASR) systems introduce word errors, which often confuse punctuation prediction models, turning punctuation restoration into a challenging task. These errors usually take the form of homophones (words which share exact or almost exact pronunciation but differ in meaning) and oronyms (homophones which consist of multiple words). We show how retrofitting of the word embeddings on the domain-specific data can mitigate ASR errors. Our main contribution is a method for a better alignment of homophone embeddings and the validation of the presented method on the punctuation prediction task. We record the absolute improvement in punctuation prediction accuracy between 6.2% (for question marks) to 9% (for periods) when compared with the state-of-the-art model.

Index Terms: punctuation prediction, punctuation restoration, ASR errors, word embeddings, retrofitting, ASR language models, spontaneous speech, dialogue systems

1. Introduction

Automatic Speech Recognition (ASR) systems are becoming ubiquitous not only in the human-computer interaction systems, such as voice assistants or dictation tools, but also in systems processing human-human conversations. The abundance of the available audio data makes it very tempting to use conversation transcripts as input data for spoken language understanding. Most commercially available ASR systems do not produce any punctuation or capitalization of output transcripts, which is a serious limitation with respect to many downstream tasks. The prerequisite for true spoken language understanding is the ability to comprehend spoken utterances, both at the semantic and syntactic level. The latter requires a robust dependency parsing, which, in turn, partially relies on correct punctuation.

Punctuation is also indispensable for intent annotation. Consider the task of annotating instances of a negative sentiment in call transcripts. The phrase "*nobody came back and I don't like I said, he didn't leave a slip*" would be incorrectly marked as an instance of negative sentiment due to the presence of the utterance "*I don't like*," whereas in reality the phrase should be punctuated as "*nobody came back and I don't, like I said, he didn't leave a slip*." In general, missing punctuation can introduce errors for phrases with personal references ("*let's eat grandma*" vs. "*let's eat, grandma*"), enumerations ("*I love cooking my family and pets*" vs. "*I love cooking, my family, and pets*"), or prepositions at sentence boundaries ("*taken care of the refund*" vs. "*taken care of. The refund*"). The problem of the inherent lack of punctuation is exacerbated by the presence of stochastic ASR errors, which for spontaneous human-human conversations can amount to 15%-20% of transcribed words [1]. Consider the following transcript: "*Hi my name is e agent will do I have pleasure speaking with today*". It may be very challenging to correctly introduce punctuation since the actual utterance is "*Hi my name is Adrian who do I have pleasure speaking with today*."

Spontaneous speech is very different not only from the written language, but from other types of speech as well [2]. Scripted speech and human-computer conversations tend to have well-defined structure with clear demarcation of sentence-like units, correct SVO (subject-verb-object) structures, and limited vocabulary. Spontaneous speech, on the other hand, is filled with all types of disfluencies which can account for 5% of all words and affect more than 30% of utterances. These disfluencies, which include backchannel markers, coordinating conjunctions, discourse markers, or filled pauses, hinder transcript translation, summarization, information extraction, or readability of transcripts. At the same time, the disfluencies are known to play an important role in the management of interactions, for instance, in upholding the turn by a speaker.

An important, yet often overlooked, aspect of spontaneous human-human conversations is the overlap of utterances [3]. As previous research suggests, the overlap in conversational speech is substantial, and it is as frequent in phone conversations between strangers as in face-to-face meetings between close acquaintances. The presence of the overlap makes diarization of speech more difficult, which also affects the ability to model the turn-taking realistically. Interestingly, we can see that the occurrence of the overlap changes the structure of utterances as the speakers react dynamically to the interruptions by repeating certain phrases, correcting, or deleting them. An example of such a change might be the increased number of question marks – speakers use questions much more often than in regular speech, not only due to the conversational nature of the exchange, but also the need to request acknowledgements of comprehension or verify comprehension by paraphrasing.

In principle, any punctuated text can be used to train a punctuation model. Unfortunately, most of the available textual corpora are not representative of spontaneous speech. Patterns learned from Wikipedia, Web Crawl, or news corpora, hardly generalize to the transcripts of spontaneous conversations. Obtaining new annotated datasets is also very challenging. Raw conversational transcripts are illegible and manual restoration of punctuation marks is both time-consuming and expensive.

The main hypothesis underlying our approach is that it is

http://dx.doi.org/10.21437/Interspeech.2020-1250

possible to mitigate stochastic ASR errors by retrofitting static word embeddings to the application domain. During punctuation prediction we cannot correct the ASR errors, but the retrofitted representation of words allows us to improve the accuracy of punctuation prediction models. As our main contribution, we validate this hypothesis by showing how pre-trained GloVe embeddings can be retrofitted to the domain of call center conversations by using Mittens [4], and how this retrofitting improves the accuracy of punctuation prediction in transcribed calls. Mittens is a method for modifying general-purpose pre-trained word embeddings in such a way that domain-specific word combinations are brought closer in the vector space. General patterns of punctuation encountered in corpora which were used to train GloVe embeddings are visibly different from punctuation patterns characteristic of spontaneous conversations. On the other hand, GloVe embeddings have been trained on very large corpora and they capture impressive amount of general knowledge. Our motivation for using Mittens is to synthesize general knowledge present in the pre-trained GloVe embeddings with domain-specific punctuation data from the Fisher corpus. We compare our approach with two state-of-the-art solutions (a bi-directional LSTM model and a CNN with pre-trained embeddings), showing significant improvements in punctuation prediction accuracy.

2. Related work

The simplest form of punctuation prediction is the discovery of sentence-like unit boundaries, where the problem is the binary classification (with "period" and "space" classes). Historically, many different techniques have been tried, for instance, Word Confusion Matrices [5], Maximum Entropy Models [6], Conditional Random Fields [7], Hidden Markov Models [8], and mixtures of probability models [9]. Features used to detect sentence boundaries included both linguistic information (n-gram language models, turn markers, part of speech annotations) [10] and prosodic features [11, 12, 13].

The advent of modern deep neural networks introduced unprecedented advancements in punctuation prediction. Recurrent neural networks quickly surpassed previous state-of-the-art models. Apart from incorporating word embeddings into punctuation prediction [14], these models initially employed LSTM architectures to predict punctuation marks using longer contexts [15, 16]. More recently, simpler architectures have proven to be sufficiently robust. Character-level convolutional neural networks (CNNs) can restore punctuation marks efficiently and these models do not suffer from out-of-vocabulary tokens or long inference times. At the same time, CNNs struggle to catch longer contexts necessary to restore question marks [17]. In the last two years, transformer-based models have been gaining popularity for both general punctuation prediction [18] and more specific tasks, such as question prediction [19] or disfluency removal [20]. The popularity of transformer-based models can be easily explained by the usefulness of the mechanism of attention in punctuation prediction [16, 21]. Another area of active research is the reframing of the punctuation prediction task in terms of machine translation. These works are mainly driven by real-time translation services [22, 23, 24].

We should stress that most of the previous works focused on punctuation prediction for speech, not for conversations, which makes the results incomparable with our case. Models are usually trained on audio recordings with already available high-quality transcriptions – these are often examples of scripted speech, for instance, TED talk transcripts or transcripts of speeches in the European Parliament. Golden standard conversational transcripts are, as we have already mentioned, very expensive to produce and only a handful of such datasets exists.

3. Methods

3.1. Data

Our primary training data is the Fisher corpus [25] due to its adequately punctuated transcripts. The Fisher corpus creation protocol relied upon a vast number of participants, each making a few short calls. Typically, the speakers would not know each other personally, which maximized inter-speaker variation and vocabulary breadth, although it also increased the formality of speech. The goal was to provide a representative distribution of subjects across a variety of demographic categories, including gender, age, dialect region, and English fluency. Punctuation classes in the Fisher corpus are highly unbalanced (see Table 1), which is typical for conversational speech. Hence, the Fisher corpus is a proper training and evaluation dataset for the punctuation prediction in the ASR transcripts.

Table 1: *The distribution of punctuation classes in the Fisher corpus.*

Class	Count	Percentage
ϵ (blank)	1 429 905	79.1%
,	208 289	11.5%
.	148 624	8.2%
?	22 182	1.2%

To fit the Fisher corpus into our model definition, we need to combine information from the time-annotated and punctuated transcripts. The first step is computing the forced alignment of the time-annotated transcripts to obtain word-level information about starting times and durations of words. For that purpose, we use the Kaldi ASR toolkit [26] with an LSTM-TDNN acoustic model trained with lattice-free Maximum Mutual Information (MMI) criterion [27]. In order to minimize the differences between the two transcript versions, we edited the Fisher corpus preparation script not to exclude single-word utterances and the text in parentheses. We retained blanks (no punctuation), periods, commas, and question marks. Other punctuation classes (e.g., exclamation marks or ellipses) were converted to blanks due to their low frequency. Finally, we aligned the time-annotated transcript with the punctuated transcript using the Needleman-Wunsch algorithm [28].

3.2. Features

We represent the conversation $\mathcal{C}$, as a chronologically ordered sequence $\mathcal{C} = [w_1, w_2, \ldots, w_n]$ of words $w_i = \langle t_i, c_i, s_i, d_i, p_i \rangle$, where: t_i is the textual representation of the word w_i; the binary feature $c_i \in \{A, B\}$ represents the conversation side uttering the word w_i; the real number s_i represents the time offset at which the word w_i started; the real number d_i represents the duration of the word w_i; and p_i is the punctuation mark which appears after the word w_i. The use of mixed conversation sides yields to efficient representation of interjections, interruptions, overlap, and simultaneous speech. The punctuation marks are only known at the training time and are being predicted during inference. We treat the punctuation prediction problem as a sequence tagging task.

We use three types of features in our models. First and foremost, we use static word embeddings. We choose 300-dimensional pre-trained GloVe [29] vectors which are being retrofitted (see Section 3). Next, we use the interval (offset) between the start of the current word and the end of the previous word, and the duration of the present word. We standardize both of these features w.r.t. other words uttered by the same speaker in the same conversation. As a result, the pauses are not modelled explicitly as word tokens, but they are inferred by the model based on word timings (offsets and durations).

3.3. GloVe retrofitting with Mittens

Word embeddings have become a widely used transfer learning approach for language processing. While different strategies for training the embeddings exist, the general premise is to encode the probability that a word will occur in the context of other words using dense vectors. These probabilities are estimated within large scale corpora, such as Wikipedia text, product reviews, or social media. In this work, we use pre-trained GloVe embeddings trained on the Common Crawl dataset consisting of 2.6 billion textual documents scraped from the Web. This general written text is a great resource to capture large-scale relations between words and their contexts. It is problematic, however, that the general language model trained on such corpus is not well aligned with the domain-specific tasks.

GloVe trains word representations in such a way that, for a pair of words w_i and w_j, the dot product of their embeddings, $\widehat{w}_i \cdot \widehat{w}_j$, approximates the log-probability of the co-occurrence of these words in the training corpus. Let V denote the available vocabulary and let c_{ij} denote the co-occurrence of words w_i and w_j. The original objective function of GloVe is:

$$J = \sum_{w_i, w_j \in V} f(c_{ij})(\widehat{w}_i^T \widehat{w}_j + b_i + b_j - \log c_{ij})^2$$

where b_i and b_j are the bias terms which represent the frequency of w_i and w_j in the training corpus, respectively. Function $f(c_{ij})$ attenuates the impact of infrequent word pairs, at the same time increasing the impact of frequent word pairs.

In order to tackle the problem of language models in healthcare data, Dingwall and Potts [4] devised a retrofitting model for static embeddings. The authors reformulate the task of training GloVe embeddings in a matrix form and they propose the extension of the objective function to take into account domain-specific data. Given the domain-specific vocabulary D, the authors propose to retrofit the original embeddings by adding the square distance penalty against the new vector $\widehat{d}_i$ for all the words $d_i \in D$. The weight μ can be used to control the retrofitting impact. The new objective function J_{Mittens} becomes:

$$J_{\text{Mittens}} = J + \mu \sum_{d_i \in D} \|\widehat{w}_i - \widehat{d}_i\|^2.$$

In other words, the new objective function exerts two pressures at the same time, by optimizing the original GloVe objective of moving word representations in the direction of the log probabilities of word-pair occurrences (left term), and by imposing a penalty whenever domain-specific word representation strays from the original embedding (right term).

In our case, the punctuation in conversational transcripts is substantially different from the punctuation found in the Common Crawl corpus. Additionally, transcripts suffer from ASR errors, where certain homophone or oronym sequences can be put in place of the correct transcription. However, if these sequences are short (1-2 words), we hypothesize that they can be corrected using the retrofitting, as the ASR errors should happen in the same contexts as correct words. If our hypothesis is correct, it will allow us to overcome the problem of the absence of quality embeddings trained on the domain-specific corpora.

4. Experiments

4.1. Models

All our models share the same convolutional neural network (CNN) architecture. Each model uses several layers of 1D convolutions which can be interpreted as fully-connected layers processing the input in small windows. Each layer is followed by a SELU activation [30], which yielded a small improvement over batch normalization [31] with ReLU [32]. We have experimented with many combinations of different numbers of layers, filter sizes, and other hyper-parameters. The best and most stable results have been achieved with six 1D convolutional layers with the filter size of 128 and zero padding. We use kernel sizes equal to 3 for all layers but the last one, where the kernel size is equal to 20. Note that these kernels could mix words from both sides of the conversation. All hidden layers use the dilation rate equal to 2. We have also experimented with model regularization. Firstly, we have added 0.5 dropout before the softmax layer. Secondly, we have used the weight decay (l2 with 0.001 weight) for the softmax layer. We have added the Gaussian noise with $\sigma = 0.1$ before the last softmax activation and SeLU activations. Finally, we have used SeLU activations that constrain the weights to a $N(0, 1)$ distribution. The final layer of our model is the fully-connected layer with softmax activation. It is applied separately at each time step to retrieve punctuation prediction for a given word.

4.2. Training

The models are implemented in Keras [33] with Tensorflow [34] back-end. During the training, the weights are updated using the Adam optimizer [35]. We use categorical cross-entropy as the loss function. We reduce (by the factor of 0.5) the learning rate (the minimum learning rate is set to $1\text{e}{-5}$) with the patience set to 3 epochs. We use the batch size of 256, and each sample is 200 words long. The Fisher corpus is divided into training, validation, and test sets in proportions 80:10:10. We had to resort to a non-standard split of the Fisher corpus because we are working on a subset of Fisher conversations where the transcript contains punctuation marks.

5. Results

We compare three variants of CNN-based models with CNN and BiLSTM baselines [36]. The BiLSTM model consists of 4 BiLSTM layers with 128 weights in each direction. The CNN model has 6 layers of 1D convolutions, followed by SeLU activations. Convolutional layers use 128 character long filters and the dilation rate of 2. The context is set to 3 words, with the exception of the last layer which uses the context of 20 words. Both models have a fully connected last layer, followed by 0.50 dropout layer and a soft-max layer. Weight decay of 0.001 is applied to the soft-max layer. The inputs, the time feature, and the soft-max activation are distorted with Gaussian noise (0.1 standard deviation). Each model is evaluated using precision, recall, and F1 score for each punctuation class separately. The results are presented in Table 2.

5.1. Model comparison

Table 2: *The per-class precision, recall, and F1-score (in %) achieved by the compared models on the pre-trained GloVe embeddings with additional retrofitting using Mittens.*

Model	Class	Precision	Recall	F1
CNN-baseline	ϵ	92.7	**95.8**	94.2
	.	65.5	58.7	61.9
	?	67.5	49.0	56.8
	,	66.6	55.1	60.3
BiLSTM-baseline	ϵ	***93.5***	94.7	94.1
	.	67.9	66.7	67.3
	?	64.7	54.6	59.2
	,	68.2	**64.1**	**66.1**
CNN-50k	ϵ	92.6	95.5	94.0
	.	70.2	65.0	67.5
	?	70.2	51.7	59.5
	,	69.7	60.8	65.0
CNN-50k-mittens	ϵ	93.3	95.3	94.3
	.	70.7	**68.7**	69.7
	?	**72.8**	53.7	61.8
	,	69.3	62.4	65.7
CNN-100k-mittens	ϵ	93.1	95.6	**94.3**
	.	**71.8**	67.7	**69.7**
	?	71.2	**55.2**	**62.2**
	,	**69.9**	61.9	65.7

As our baseline we choose the convolutional neural network and standard pre-trained GloVe embeddings for 50 000 most frequent words in GloVe training data. This is a strong baseline which has proven to be a viable solution for a production-ready, real-time system [36]. Next, we constrain the selection of the 50 000 words to those which appear in the ASR vocabulary (CNN-50k model). This model language covers words that appear in the ASR vocabulary twice as frequently as the baseline model, and it achieves approx. 60% coverage of the vocabulary. We observe the improvement of both precision and recall across all punctuation classes. The final model, with embeddings covering almost the entire ASR vocabulary (approximately 20% of the ASR vocabulary is missing from GloVe), outperforms the baseline with respect to all three considered metrics while not imposing significantly increased resource requirements or incurring computational costs. Finally, we compare our CNN-based model with the BiLSTM model based on the GloVe embeddings. Our model outperforms this baseline with regards to precision and recall for all punctuation classes, except full stops. The additional advantage of using a CNN-based model in the production environment is a more straightforward parallelization compared to the BiLSTM architecture.

5.2. Confusion matrix analysis

Table 3 presents the comparison of the confusion matrices for the CNN-baseline and the CNN-100k-mittens models. The latter model improves the accuracy of all three punctuation classes, with the most pronounced improvement for the "period" class (9% absolute improvement). The improvement for "question mark" and "comma" classes is 6.2% and 6.8%, respectively. The improvements stem mostly from the fact that the CNN-100k-mittens predicts punctuation marks missed by the baseline model. In other words, our model inserts punctuation marks in places where the baseline model predicts blanks.

Table 3: *Confusion matrix comparison for the CNN-baseline and CNN-100k-mittens models.*

		CNN-baseline				CNN-100k-mittens			
		ϵ	.	?	,	ϵ	.	?	,
Actual	ϵ	95.8	1.4	0.2	2.6	95.6	1.1	0.1	2.8
	.	32.2	58.7	1.5	7.6	21.2	67.7	1.5	9.6
	?	23.7	21.0	49.0	6.3	19.9	18.4	55.2	6.4
	,	37.8	6.5	0.6	55.0	31.1	6.6	0.4	61.9

5.3. GloVe similarity analysis

To see how the retrofitting of word embeddings moves homophones and oronyms closer in the embedded space, we compute the cosine similarity between selected pairs of confusing utterances in the original and the retrofitted space. As can be seen in Table 4, the vectors are much closer in the retrofitted space than in the original GloVe space.

Table 4: *Similarity between homophones/oronyms in the original GloVe space and in the retrofitted space.*

Homophone/Oronym	Cosine similarity	
	original	retrofitted
I have to cancel I have to cancer	0.74	0.91
by buy	0.16	0.4
go to the court got the card	0.69	0.78
thank you think you	0.84	0.9

6. Conclusions

Punctuation prediction is an important topic for improving the readability and the segmentation of audio transcripts produced by ASR. However, ASR introduces inherent errors in word recognition and a significant divergence of punctuation mark distribution due to the dynamics of spontaneous speech. These phenomena limit the usefulness of word embeddings, since most of the static word embeddings are trained on the correctly segmented sentences from written text.

We have hypothesized that aligning language models present in pre-trained word embeddings with the word co-occurrence structure visible in transcribed calls would allow us to overcome some of the challenges of ASR and to improve the quality of punctuation prediction. We have used a recently published method – Mittens – to retrofit the GloVe embeddings with call transcripts obtained from ASR. Retrofitted embeddings yield an consistent improvement of 6%-9% over the original GloVe embeddings for all punctuation classes. Furthermore, the retrofitted embeddings allow us to outperform the BiLSTM model with a faster CNN-100k-mittens model. This is a very important practical result, as CNN-based models are much more suitable for the deployment in production environments due to the inference time constraints.

7. References

[1] V. Manohar, S.-J. Chen, Z. Wang, Y. Fujita, S. Watanabe, and S. Khudanpur, "Acoustic modeling for overlapping speech recognition: Jhu chime-5 challenge system," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 6665–6669.

[2] E. Shriberg, "Spontaneous speech: How people really talk and why engineers should care," in *9th European Conference on Speech Communication and Technology*, 2005, pp. 1781–1784.

[3] E. Shriberg, A. Stolcke, and D. Baron, "Observations on overlap: Findings and implications for automatic processing of multi-party conversation," in *7th European Conference on Speech Communication and Technology*, 2001, pp. 1359–1362.

[4] N. Dingwall and C. Potts, "Mittens: an extension of GloVe for learning domain-specialized representations," *arXiv preprint arXiv:1803.09901*, 2018.

[5] D. Hillard, M. Ostendorf, A. Stolcke, Y. Liu, and E. Shriberg, "Improving automatic sentence boundary detection with confusion networks," Washington University, Tech. Rep., 2004.

[6] J. Huang and G. Zweig, "Maximum entropy model for punctuation annotation from speech," in *7th International Conference on Spoken Language Processing*, 2002.

[7] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 177–186.

[8] J. P. Yamron, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt, "A hidden markov model approach to text segmentation and event tracking," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1. IEEE, 1998, pp. 333–336.

[9] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, "Comparing and combining generative and posterior probability models: Some advances in sentence boundary detection in speech," in *Conference on Empirical Methods in Natural Language Processing*, 2004, pp. 64–71.

[10] A. Stolcke and E. Shriberg, "Automatic linguistic segmentation of conversational speech," in *4th International Conference on Spoken Language Processing*, vol. 2. IEEE, 1996, pp. 1005–1008.

[11] H. Christensen, Y. Gotoh, and S. Renals, "Punctuation annotation using statistical prosody models," in *ISCA Tutorial and Research Workshop (ITRW) on Prosody in Speech Recognition and Understanding*, 2001.

[12] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tür, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech Communication*, vol. 32, no. 1-2, pp. 127–154, 2000.

[13] D. Wang and S. S. Narayanan, "A multi-pass linear fold algorithm for sentence boundary detection using prosodic cues," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 2004, pp. I–525.

[14] X. Che, C. Wang, H. Yang, and C. Meinel, "Punctuation prediction for unsegmented transcript based on word vector," in *10th International Conference on Language Resources and Evaluation*, 2016, pp. 654–658.

[15] O. Tilk and T. Alumäe, "LSTM for punctuation restoration in speech transcripts," in *16th Annual Conference of the International Speech Communication Association*, 2015.

[16] O. Tilk and T. Alumae, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration." in *Interspeech*, 2016, pp. 3047–3051.

[17] W. Gale and S. Parthasarathy, "Experiments in character-level neural network models for punctuation." in *Interspeech*, 2017, pp. 2794–2798.

[18] B. Nguyen, V. B. H. Nguyen, H. Nguyen, P. N. Phuong, T.-L. Nguyen, Q. T. Do, and L. C. Mai, "Fast and accurate capitalization and punctuation for automatic speech recognition using transformer and chunk merging," *arXiv preprint arXiv:1908.02404*, 2019.

[19] Y. Cai and D. Wang, "Question mark prediction by BERT," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. IEEE, 2019, pp. 363–367.

[20] Q. Chen, M. Chen, B. Li, and W. Wang, "Controllable time-delay transformer for real-time punctuation prediction and disfluency detection," *arXiv preprint arXiv:2003.01309*, 2020.

[21] A. Öktem, M. Farrús, and L. Wanner, "Attentional parallel RNNs for generating punctuation in transcribed speech," in *International Conference on Statistical Language and Speech Processing*. Springer, 2017, pp. 131–142.

[22] E. Cho, J. Niehues, K. Kilgour, and A. Waibel, "Punctuation insertion for real-time spoken language translation," in *11th International Workshop on Spoken Language Translation*, 2015.

[23] E. Cho, J. Niehues, and A. Waibel, "NMT-based segmentation and punctuation insertion for real-time spoken language translation." in *Interspeech*, 2017, pp. 2645–2649.

[24] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling punctuation prediction as machine translation," in *International Workshop on Spoken Language Translation*, 2011.

[25] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: a resource for the next generations of speech-to-text." in *International Conference on Language Resources and Evaluation*, vol. 4, 2004, pp. 69–71.

[26] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[27] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free mmi." in *Interspeech*, 2016, pp. 2751–2755.

[28] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, 1970.

[29] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[30] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 972–981.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *32nd International Conference on Machine Learning*, 2015, pp. 448–456.

[32] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *27th International Conference on Machine Learning*, 2010, pp. 807–814.

[33] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[34] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "TensorFlow: A system for large-scale machine learning." in *12th {USENIX} Symposium on Operating Systems Design and Implementation*, vol. 16, 2016, pp. 265–283.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[36] P. Żelasko, P. Szymański, J. Mizgajski, A. Szymczak, Y. Carmiel, and N. Dehak, "Punctuation prediction model for conversational speech," in *Interspeech*, 2018, pp. 2633–2637. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1096