Interspeech 2024
1-5 September 2024, Kos, Greece

Word-level Text Markup for Prosody Control in Speech Synthesis

Yuliya Korotkova[1,2], *Ilya Kalinovskiy*[1,3], *Tatiana Vakhrusheva*[1,2]

[1]JustAI, Russia
[2]Higher School of Economics, Russia
[3]School of Computer Science & Robotics, Tomsk Polytechnic University, Russia
yuokorotkova@gmail.com, kua21@tpu.ru, t.vakhrusheva@just-ai.com

Abstract

Modern Text-to-Speech (TTS) technologies generate speech very close to the natural one, but synthesized voices still lack variation in intonation which, in addition, is hard to control. In this work, we address the problem of prosody control, aiming to capture information about intonation in a markup without hand-labeling and linguistic expertise. We propose a method of encoding prosodic knowledge from textual and acoustic modalities, which are obtained with the help of models pretrained on self-supervised tasks, into latent quantized space with interpretable features. Based on these features, the prosodic markup is constructed, and it is used as an additional input to the TTS model to solve the one-to-many problem and is predicted by text. Moreover, this method allows for prosody control during inference time and scalability to new data and other languages.
Index Terms: prosody control, prosody tagging, word-level prosody, speech synthesis, TTS

1. Introduction

Prosody plays an essential role in speech since it encodes a variety of information not inferred from the lexical content of an utterance. It is mainly reflected with the help of pitch, its level and contour, and duration. Although modern advances in Text-to-Speech (TTS) technologies have led to generating synthesized speech close to natural speech, there are still several drawbacks in modeling prosody. Firstly, TTS models tend to average the intonation of the database because of the one-to-many problem: the same textual input may have various audio references. Secondly, prosody is hard to control and model due to the difficulty of its interpretability and dependency on many factors: integral, including emotions and subjective attitude of the speaker, and local, such as semantic and syntactic relationships between the components of the phrase, accents, and topicality.

Methods aimed at resolving these problems can be divided into two main groups: unsupervised (data-based) and supervised (perception-based). Data-based approaches are cheap and consistent, but they lack interpretability and are mainly coarse-grained. Therefore, they do not represent phrasal segmentation and prominence. Examples of data-based methods are Global Style Token (GST) [1] and Style Reconstruction Loss [2], which can be added to the TTS training pipeline to improve and/or control intonation in synthesized speech.

Conversely, perception-based approaches are usually motivated by linguistic theories and, therefore, are interpretable, but time- and effort-consuming since they require manual labeling and often lack consistency between data and labels. Examples of explicit prosodic markups used in TTS are RFC [3], Tilt [4], and ToBI [5, 6]. To overcome the problem of hand-labeling, it is proposed to utilize tools for semi- and auto-labeling of prosody, for instance, PyToBI [7], AuToBI [8], and Wavelet [9]. However, these tools show insufficient results, do not reflect all crucial features of intonation, or require pretraining on labeled data.

A more balanced approach should be fine-grained, interpretable, and unsupervised at the same time as proposed in [10, 11, 12]. In all these methods, clustering is used to detect prosodic events at phoneme or word level based on prosodic features. Nonetheless, the method described in [12] does not allow control of intonation in synthesized speech. The method proposed in [10] mixes all factors influencing prosody. In addition, in this method, tags are different for each lexical group, which limits control. The authors of [11] cluster the whole space of prosodic features, which results in more neutral averaged intonational patterns rather than in expressive ones. Moreover, despite the authors' claims, the method is hardly applicable to the word level since clustering is performed separately per phoneme, and phonemes have a limited size of vocabulary, while the number of words is unlimited.

The main contributions of this work can be summarized as follows:

- We introduce an **unsupervised** method of **expressive** and **interpretable** word-level prosodic markup construction that reflects **fine-grained** prosody, while excluding coarse-grained prosody.
- We show that the integration of the markup into TTS makes synthesized speech richer with intonation and allows for **control** over it.
- We provide the code[1] and all the examples on our demo page[2].

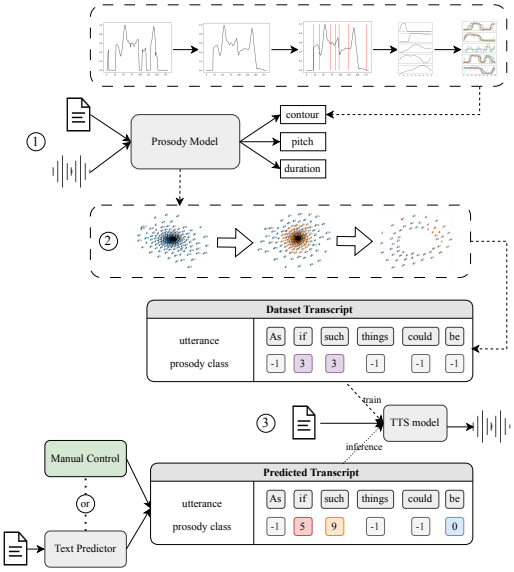
2. Methods

In this section, the whole procedure for obtaining a prosodic markup and integrating it into TTS is described (Figure 1a), which consists of the following steps:

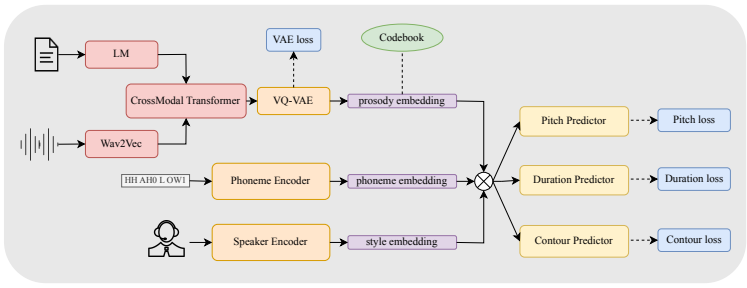
1. The Prosody Model learns to reconstruct prosodic features from textual and acoustic modalities encoded into a quantized latent space.
2. A prosodic markup is constructed based on this space that reflects a limited number of expressive prosodic events.
3. A TTS model is trained using the prosodic markup, while a language model is trained to predict the markup from text.

[1]https://github.com/just-ai/speechflow
[2]https://yuliya1324.github.io/prosody_control_TTS/

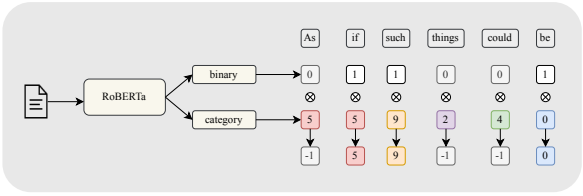
10.21437/Interspeech.2024-715



(a) Pipeline of prosodic markup construction and its integration into TTS



(b) Prosody Model



(c) Text Predictor

Figure 1: *Overview of the whole pipeline of prosodic markup construction and architectures of Prosody Model and Text Predictor. Class* -1 *in a markup denotes for neutral intonation.*

2.1. Prosody model's architecture

We propose a model aimed at constructing a latent space aligned at the word level that encodes fine-grained prosodic information not influenced by extralinguistic factors, as well as the pronunciation of lexical items. This latent space is encoded in a codebook learned by a Vector Quantised-Variational AutoEncoder (VQ-VAE) [13] from textual and acoustic modalities with the help of models pretrained with self-supervised learning (SSL). To learn this quantized space, the model learns to predict acoustic features that carry information about prosody: durations, pitch, and tonal contours. Additional information, such as speaker's style and pronunciation, is added to prosodic embeddings before predicting intonational features, to remove the influence of other factors that are not required in the markup. Figure 1b illustrates the architecture of the proposed model.

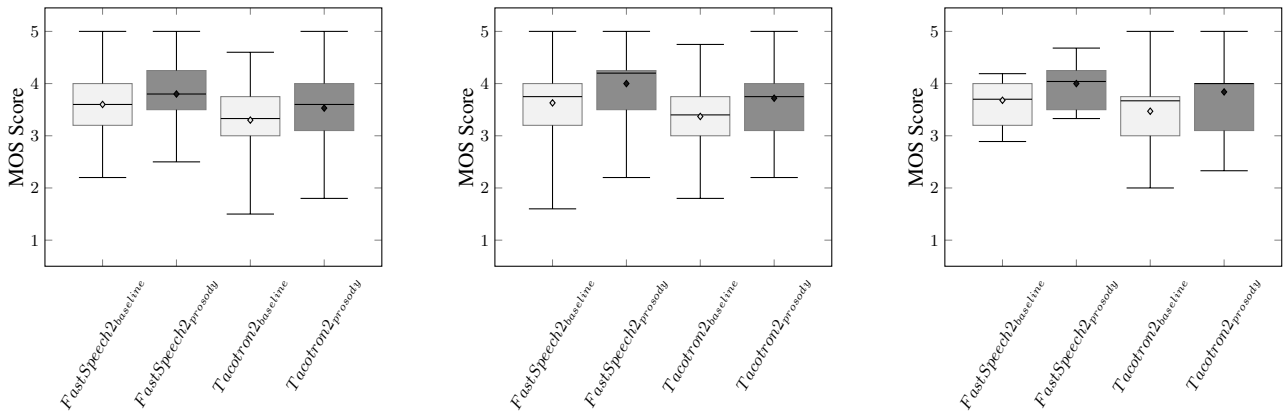
At first, acoustic and textual representations of speech are obtained with the help of Wav2Vec [14] and RoBERTa [15], respectively, since these models provide good representations of the data generalizing common patterns and utilizing context. An acoustic model encodes lower-level acoustic features, while a language model encodes higher-level linguistic information, both of which contribute to prosody. The embeddings from SSL models are averaged by words and then fused together by a Multimodal Transformer with Crossmodal Attention [16] to use both modalities simultaneously. Within this transformer, features from the language model $e_{lm}^{w\times1024}$ (w denotes the number of words) are added to features from the acoustic model $e_{am}^{w\times1024}$ with learnable attention weights resulting in $e_{cmodal}^{w\times1024}$. Then, $e_{cmodal}^{w\times1024}$ is fed to the VQ-VAE. During training, Wav2Vec and RoBERTa are frozen, while the Multimodal Transformer with Crossmodal Attention is trained.

The core of the model is Vector Quantization, which serves as an information bottleneck and compresses the space of all possible features into a limited number. This enforces the model to learn only essential information from speech, through which prosodic features can be predicted. Moreover, to ensure disentanglement of speech factors, two additional encoders are used: the Phoneme Encoder encodes lexical content, while the Speaker Encoder encodes speaker characteristics and the style of speaking. The VQ-VAE takes $e_{cmodal}^{w\times1024}$ and outputs $e_{prosody}^{w\times128}$, which are mapped to embeddings of the codebook $e_{vq}^{1024\times128}$ inside the VQ. The Phoneme Encoder encodes the phoneme sequence and outputs $e_{phon}^{p\times256}$ (p denotes the number of phonemes). The Speaker Encoder is based on Gaussian Mixture Variational AutoEncoder (GMVAE) [17]. It predicts the style embedding $e_{style}^{1\times32}$ based on biometric features obtained by ECAPA-TDNN [18]. Then, embeddings from all encoders are extrapolated to the length of the phoneme sequence and concatenated resulting in $e_{out}^{p\times416}$. Finally, $e_{out}^{p\times416}$ is fed to three different predictors of prosodic parameters: the Duration Predictor, the Pitch Predictor and the Tonal Contour Predictor, all based on predictors from FastSpeech2 [19].

The target prosodic features are durations, pitch mean, pitch range, the mean of pitch derivatives and tonal contours. Durations and pitch are normalized by the minimum and maximum values of an utterance. In addition, pitch is interpolated on unvoiced phonemes and smoothed to reduce the influence of phoneme pronunciation and pitch fluctuations. Tonal contours represent classes, each reflecting the trend of pitch movement realized on a single word. The procedure for tonal contour modeling is as follows (upper dashed box in Figure 1a). Firstly, the pitch, normalized by the length and level, as well as smoothed, is cut by word boundaries, resulting in tonal contours per word. Then, they are normalized by words to reduce the effect of the level and length. Finally, tonal contours are clustered into 500 classes using Agglomerative clustering with a precomputed Euclidean distance matrix. Durations, pitch mean, pitch range, and the mean of pitch derivatives are calculated for each phoneme. Tonal contours are modeled for each word and then extrapolated to all phonemes in the word.

The model is optimized by the sum of four losses (1): Duration loss, Pitch loss, Contour loss, and VAE loss.

$$L_{Total} = L_{Dur} + L_{Pitch} + L_{Cont} + L_{VAE} \qquad (1)$$



(a) *English* (b) *Russian* (c) *Portuguese*

Figure 2: *Results of MOS tests (expressiveness of intonation) comparing two variants of each model: baseline (the markup is excluded from the input) and prosody (the markup is included in the input). Tests were performed for three languages: English (a), Russian (b) and Portuguese (c). All the results are significant according to the Mann-Whitney U-test for difference in means between two groups (p-value* < 0.001 *for all English and Russian models, p-value* < 0.005 *for all Portuguese models).*

For duration and pitch predictions, the mean-squared error (MSE) between the target and predicted features is calculated, while the Contour Predictor is optimized by the Cross-Entropy loss since it predicts the class of tonal contour. The VAE loss (2) is the sum of two MSE losses required for the codebook learning, which pushes the latent vectors and the codewords towards each other.

$$L_{VAE} = MSE(sg[e_{prosody}], e_{vq}) + MSE(e_{prosody}, sg[e_{vq}]) \quad (2)$$

where sg denotes the stopgradient operator.

2.2. Construction of the prosodic markup

In our many experiments, we found that the quantized space $e_{vq}^{1024\times128}$, learned by the Prosody Model, represents one big cluster where the neutral intonation lies in the center and the expressive one lies in the periphery (Figure 1a-2). Therefore, to construct a markup capturing **expressive** prosodic events present in speech and enabling the **control** of intonation in synthesized speech, additional clusterization of $e_{vq}^{1024\times128}$ is required. This clustarization is performed in two steps. Firstly, outliers are extracted by Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to detect expressive intonational patterns and exclude neutral ones. DBSCAN is a clustering technique well known for its ability to detect outliers by iteratively merging neighbors within a given distance and leaving out those that lie further. These outliers are then clustered by Gaussian Mixture Models (GMM) into 10 clusters to unite similar patterns and to distinguish them from one another. The resulting prosodic markup consists of these 10 classes with an additional neutral prosodic class.

2.3. Integration of the prosodic markup into TTS

The markup is fed to the TTS model as an additional input on par with text to resolve the one-to-many problem. During the training stage, the prosodic markup is used from the database, and during inference, it is predicted by the Text Predictor (Figure 1c).

The Text Predictor solves a sequence labeling task to annotate text with a prosodic markup. It is based on RoBERTa followed by two classification heads, with one head predicting the placement of a prosodic label (binary head) and the other predicting a particular prosodic class (category head). This helps to control prosody more precisely: to make prosodic events more or less frequent or to make them more or less varied. Specifically, the prediction of the binary head after softmax is validated by checking whether the probability of the first class (presence of a prosodic event) is higher than the given $thres_{bin}$ or not. Lowering this threshold makes prosodic labels more frequent and, therefore, the synthesized speech is more expressive. The classes of the prosodic markup can be sampled by argmax, which makes the markup deterministic – it is identical for the same textual input. Alternatively, to make the markup and the speech more varied, classes can be sampled randomly or from the distribution of probabilities of the category head.

During training, the last 15 layers and both classification heads are fine-tuned, while the rest of the model is frozen. The model is optimized by the sum of two Cross Entropy losses (3).

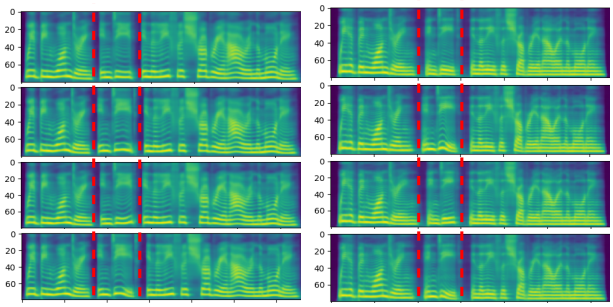
$$L_{text\_predictor} = L_{binary} + L_{category} \quad (3)$$

The first one is calculated between predictions of the binary head and binary target labels, where 0 denotes the neutral prosodic class of the markup, while 1 denotes all other classes of the markup. The second one is calculated between the predictions of the category head and the classes of the markup excluding the neutral prosodic class.

3. Experiments and results

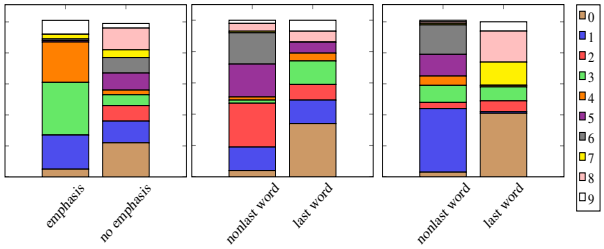
3.1. Data

The proposed method is applicable to nearly any language and dataset with sufficient intonational pattern diversity, as the prosodic features considered here are universal for most languages, except tonal languages like Chinese. To show the scalability of the method, experiments were performed for three different languages: English, Russian and Portuguese. The datasets consist of the internal data recorded by professional speakers with a sample rate of 22.05 kHz, open-source datasets,



(a) *FastSpeech2* (b) *Tacotron2*

Figure 3: *Example of prosody control illustrated by changes in a spectrogram chunk between red dashed lines. Sentence used in this example: "We had been wandering,* ***indeed****, in the leafless shrubbery an hour in the morning", the prosodic tag was modified on a word indeed.*



(a) *emphasis* (b) *questions* (c) *declaratives*

Figure 4: *Percentage of each prosodic class spread through (a) accented vs nonaccented words, (b) last vs nonlast words in questions, (c) last vs nonlast words in declarative sentences.*

such as the LJ Speech Dataset [20], and audio-books [21]. Our internal data includes sets with different sentence types, such as alternative, special and general questions, exclamation, and declarative sentences, as well as a set with emphasised words, which is useful to show interpretability of the markup (Section 3.3). Overall, there are recordings of 2699 (422 hours), 1344 (405 hours) and 10 (28 hours) speakers in English, Russian and Portuguese datasets, respectively. The prosodic markup covers about 20% of the English dataset, about 45% of the Russian one, and about 14% of the Portuguese one, while all the rest of the words are marked with neutral intonation. The datasets were divided into train and test parts in proportions of 99:1. Transcripts and audio files were aligned on the word and phoneme level by [22].

3.2. Models

To show the capabilities of the proposed markup, experiments were performed using two TTS models: Tacotron2 [23], an autoregressive acoustic model consisting of an encoder and a decoder with an attention mechanism, and FastSpeech2 [19], a parallel acoustic model with predictors of local prosody attributes used as conditional inputs to predict a mel spectrogram. Each model takes the same input consisting of phonemes, PoS tags, syntactic tags and prosodic tags, all of which are extrapolated to the phoneme-level and concatenated together. We compared two variants of each model: the *baseline* where the markup is excluded from the input to the TTS model, and the *prosody*, where the markup is included in the input to the TTS model.

3.3. Results

We conducted crowdsourced MOS tests for each language and model, comparing *baseline* and *prosody* variants. A total of 256, 190 and 170 native speakers participated in the English, Russian and Portuguese tests, respectively. They were asked to rate models on a 5-point Likert scale with respect to the expressiveness of intonation. The results of the tests (Figure 2) reveal that TTS models with the prosodic markup outperform their counterparts in terms of intonation.

To illustrate the controllability of intonation with the help of the prosodic markup, we visualized and synthesized spectrograms of sentences where a prosodic tag on a single word was changed. Figure 3 shows an example in which the word *indeed* between the red dashed lines is manually controlled with prosodic tag modifications. It can be seen that the prosody of the word with different prosodic tags changes, demonstrating the controllability of the markup.

To illustrate the interpretability of the markup, we visualized the percentage of all classes highlighting different prosodic patterns in English. We assume that the last word in general question is usually associated with rising intonation, while the last word in a declarative sentence is associated with falling intonation; and emphasised words inside phrase usually have prolonged stressed vowels and a peak of pitch on them. For this aim, we utilized our internal data, which has separate sets with general questions, declarative sentences, and a labeled set with emphasized words. Our hypothesis is that there are specific prosodic classes that mark emphasis, questions, and the end of a sentence. Figure 4a shows that emphasized words are mostly marked with classes $1, 3, 4$. Conversely, class 0 mostly appears at the end of the phrase, while classes 7 and 8 perhaps reflect falling intonation at the end of declarative sentences (Figures 4b-c).

4. Conclusion

Overall, in this work, we propose a method for obtaining prosodic information from speech in an unsupervised way, which is encoded in the form of a markup aligned with the word level. The results show that a TTS model conditioned on the proposed markup produces speech with more varied and natural intonation. Besides, it is possible to control intonation in synthesized speech by modifying the markup. Finally, the proposed method is cost-effective in terms of time and effort, does not require expertise in linguistics, and facilitates easy scaling to new data and languages.

One of the obvious limitations of the proposed method is that it requires the data to contain expressive prosody; otherwise, the Prosody Model will not be able to learn anything from the monotonic data. However, even a small proportion of expressive intonation (as seen in the Portuguese dataset in our experiments) is sufficient for the Prosody Model to learn significant intonational patterns. In addition, we cannot be sure about the compatibility of the markup with any TTS model. For example, we doubt that this markup concatenated with sequence of phonemes will correctly work in models such as VALL-E [24], since this model utilizes in-context learning for prosody modeling.

5. References

[1] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.

[2] R. Liu, B. Sisman, G. Gao, and H. Li, "Expressive tts training with frame and style reconstruction loss," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1806–1818, 2021.

[3] P. Taylor, "The rise/fall/connection model of intonation," *Speech Communication*, vol. 15, no. 1-2, pp. 169–186, 1994.

[4] P. A. Taylor and A. W. Black, "Synthesizing conversational intonation from a linguistically rich input," 1994.

[5] K. E. Silverman, M. E. Beckman, J. F. Pitrelli, M. Ostendorf, C. W. Wightman, P. Price, J. B. Pierrehumbert, and J. Hirschberg, "Tobi: A standard for labeling english prosody." in *ICSLP*, vol. 2, 1992, pp. 867–870.

[6] Y. Zou, S. Liu, X. Yin, H. Lin, C. Wang, H. Zhang, and Z. Ma, "Fine-grained prosody modeling in neural speech synthesis using tobi representation." in *Interspeech*, 2021, pp. 3146–3150.

[7] M. Domínguez Bajo, P. L. Rohrer *et al.*, "Pytobi: a toolkit for tobi labeling under python," *Interspeech 2019; 2019 Sept 15-19; Graz, Austria. Baixas: ISCA; 2019. p. 3675-6.*, 2019.

[8] A. Rosenberg, "Autobi-a tool for automatic tobi annotation," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[9] A. Suni, J. Šimko, D. Aalto, and M. Vainio, "Hierarchical representation and estimation of prosody using continuous wavelet transform," *Computer Speech & Language*, vol. 45, pp. 123–136, 2017.

[10] Y. Guo, C. Du, and K. Yu, "Unsupervised word-level prosody tagging for controllable speech synthesis," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7597–7601.

[11] A. Vioni, M. Christidou, N. Ellinas, G. Vamvoukakis, P. Kakoulidis, T. Kim, J. S. Sung, H. Park, A. Chalamandaris, and P. Tsiakoulis, "Prosodic clustering for phoneme-level prosody control in end-to-end speech synthesis," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5719–5723.

[12] R. Huang, C. Zhang, Y. Ren, Z. Zhao, and D. Yu, "Prosody-TTS: Self-supervised prosody pretraining with latent diffusion for text-to-speech," 2023. [Online]. Available: https://openreview.net/forum?id=y6EnaJlhcWZ

[13] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[14] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[16] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov, "Multimodal transformer for unaligned multimodal language sequences," in *Proceedings of the conference. Association for Computational Linguistics. Meeting*, vol. 2019. NIH Public Access, 2019, p. 6558.

[17] A. C. Aguilera, P. M. Olmos, A. Artés-Rodríguez, and F. Pérez-Cruz, "Regularizing transformers with deep probabilistic layers," *Neural Networks*, vol. 161, pp. 565–574, 2023.

[18] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[19] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," *arXiv preprint arXiv:2006.04558*, 2020.

[20] K. Ito and L. Johnson, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[21] S. King and V. Karaiskos, "The blizzard challenge 2013," 2014.

[22] R. Badlani, A. Łańcucki, K. J. Shih, R. Valle, W. Ping, and B. Catanzaro, "One tts alignment to rule them all," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6092–6096.

[23] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[24] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.