Spoken Language Modeling from Raw Audio

Tu Anh Nguyen

Spoken Language Modeling from Raw Audio

Tu Anh Nguyen

PhD Dissertation
April 09, 2024

THÈSE DE DOCTORAT DE SORBONNE UNIVERSITÉ

Spécialité **Informatique**

École Doctorale Informatique, Télécommunications et Électronique de Paris (ED130)

Spoken Language Modeling from Raw Audio

Modèles de langue pour la parole appris à partir du signal audio

Présentée par

Tu Anh Nguyen

Pour obtenir le grade de

DOCTEUR de SORBONNE UNIVERSITÉ

Présentée et soutenue publiquement le 9 avril 2024

Devant le jury composé de :

Hung-yi LEE *Professor, National Taiwan University*	Rapporteur
David HARWATH *Assistant Professor, The University of Texas at Austin*	Rapporteur
Catherine PELACHAUD *Directrice de recherche, CNRS*	Examinatrice
Laurent BESACIER *Principal Scientist, Naver Labs Europe & Université Grenoble Alpes*	Examinateur
Tara SAINATH *Principal Research Scientist, Google*	Examinatrice
Benoît SAGOT *Directeur de Recherche, INRIA*	Directeur de thèse
Emmanuel DUPOUX *Directeur d'Études, ENS, PSL & EHESS & Meta*	Co-Encadrant de thèse

Tu Anh Nguyen
Spoken Language Modeling from Raw Audio
PhD Dissertation, April 09, 2024
Reviewers: Hung-yi Lee and David Harwath
Examiners: Catherine Pelachaud, Laurent Besacier and Tara Sainath
Supervisors: Benoît Sagot and Emmanuel Dupoux

Abstract

Speech has always been a dominant mode of social connection and communication. However, speech processing and modeling have been challenging due to its variability. Classic speech technologies rely on cascade modeling, i.e. transcribing speech to text with an Automatic Speech Recognition (ASR) system, processing transcribed text using Natural Language Processing (NLP) methods, and converting text back to speech with a Speech Synthesis model. This method eliminates speech variability but requires a lot of textual datasets, which are not always available for all languages. In addition, it removes all the expressivity contained in the speech itself.

Recent advancements in self-supervised speech learning (SpeechSSL) have enabled the learning of good discrete speech representations from raw audio, bridging the gap between speech and text technologies. This allows to train language models on discrete representations (discrete units, or pseudo-text) obtained from the speech and has given rise to a new domain called TextlessNLP, where the task is to learn the language directly on audio signals, bypassing the need for ASR systems. The so-called Spoken Language Models (Speech Language Models, or SpeechLMs) have been shown to be working and offer new possibilities for speech processing compared to cascade systems.

The objective of this thesis is thus to explore and improve this newly-formed domain. We are going to analyze why these discrete representations work, discover new applications of SpeechLMs to spoken dialogues, extend TextlessNLP to more expressive speech as well as improve the performance of SpeechLMs to reduce the gap between SpeechLMs and TextLMs.

Keywords: Spoken Language Model, Language Model, Unsupervised Speech Learning, TextlessNLP, Speech Processing

Résumé

La parole a toujours été un mode dominant de connexion sociale et de communication. Cependant, le traitement et la modélisation de la parole sont difficiles en raison de la variabilité le parole. Les technologies classiques de la parole reposent sur une modélisation en cascade, c'est-à-dire la transcription de la parole en texte avec un système de reconnaissance automatique de la parole (ASR), le traitement du texte transcrit à l'aide de méthodes de traitement du langage naturel (NLP) et la conversion du texte en parole avec un modèle de synthèse vocale. Cette méthode élimine la variabilité de la parole mais nécessite beaucoup de jeux de données textuelles, qui ne sont pas toujours disponibles pour toutes les langues. De plus, elle supprime toute l'expressivité contenue dans la parole elle-même.

De récentes avancées dans le domaine de l'apprentissage auto-supervisé de la parole (SpeechSSL) ont permis d'apprendre de bonnes représentations discrètes de la parole à partir du signal audio, comblant ainsi le fossé entre les technologies de la parole et du texte. Cela permet d'entraîner des modèles de langue sur des représentations discrètes (unités discrètes ou pseudo-texte) obtenues à partir de la parole et a donné naissance à un nouveau domaine appelé TextlessNLP, où la tâche consiste à apprendre la langue directement sur les signaux audio, sans avoir recours à des systèmes ASR. Les modèles de langue parlé (SpeechLMs) ont été montrés comme faisables et offrent de nouvelles possibilités pour le traitement de la parole par rapport aux systèmes en cascade.

L'objectif de cette thèse est donc d'explorer et d'améliorer ce domaine nouvellement formé. Nous allons analyser pourquoi ces représentations discrètes sont efficaces, découvrir de nouvelles applications des SpeechLMs aux dialogues parlés, étendre le TextlessNLP aux paroles plus expressives ainsi qu'améliorer les performances des SpeechLMs pour réduire l'écart entre les SpeechLMs et les TextLMs.

Mots-clés: Modèle de langue parlée, Modèle de langue, Apprentissage non supervisé de la parole, TextlessNLP, Traitement de la parole

Acknowledgement

First of all, I would like to thank all of the jury members – Hung-yi Lee, David Harwath, Catherine Pelachaud, Laurent Besacier, Tara Sainath, Benoît Sagot and Emmanuel Dupoux for spending their time reviewing and giving valuable comments for this dissertation. I would like to thank Hung-yi Lee and David Harwath for being the reporters of the thesis. I would like to thank Catherine Pelachaud and Laurent Besacier for being the members of my PhD committee.

I would like to give my sincere thanks to my supervisors – Emmanuel and Benoît for truly dedicating their time on my PhD. Emmanuel, this PhD subject couldn't have been realized without you. Thanks for giving me the opportunity to work on the internship on the ZeroSpeech 2021 challenge that finally came up with this PhD subject. Thank you for all the initiatives and connections that gave me the chance to work with amazing people at TextlessNLP, Seamless and FAIR teams. Benoît, thank you for accepting me to work on this PhD. Thank you for all the super ideas, the comments and the help for the paper writing and the presentations. Thank you for advertising our work at the conferences as well as at Collège de France. Finally, a big thank you to both of you for not really being my supervisors but rather being dear friends in life.

I would like to thank all members of the CoML team that I am lucky to work with since my internship in 2020. Thanks Xuan Nga and Catherine for the warm welcome to the team. Thanks Maureen, Patricia, Ewan, Mathieu Bernard, Nicolas for the great collaboration in the ZeroSpeech 2021 challenge. Thanks Manel, Marianne, Julien, Gwendal for helping me to overcome engineering problems. Thanks Mathieu, Robin, Marvin, Maureen, Maxime, Angelo, Jing for being my wonderful PhD peers at CoML. Thanks Rachid, Juliette, Paul Michel, Rahma, Alex for great advice during my PhD. Thanks Lin and Sabrina for all the kind support. Thanks Paul for doing a great job of being my first "stagiaire", it's been a pleasure mentoring you. Lastly, thank you for all the fun activities, seminars, raclettes, hangouts that made CoML like a family.

I would like to thank the members of the ALMAnaCH research team at Inria for hosting me during my PhD. Thanks to the "dinosaur" PhDs – Clementine, Pedro, Louis, Benjamin, Lionel for all the support and welcoming "baby" PhDs like us. Thanks Wissam, Menel, Alix, Anna, Francis for the joy you bring each time I come to

the C120 office. Thanks to my peers – Roman, Nathan, Mathieu, You, Arji, Lydia, Alafate, Biwesh for great discussions during my PhD. Thanks Tanti, Floriane, Hugo, Rua, Rian, Nicolas, Pierre for amusing conversations at Inria. Thanks Rachel, Djamé, Éric for the insightful exchanges. Thank you all for the seminars, team reading group, drinking and eating event and other activities (secret santa, halloween, galette des roi) that made my memorable memories.

I would like to thank my "metamates", without whom I couldn't accomplish the work mentioned in this thesis. Thanks to the early members of the Textless team – Eugene, Kushal, Morgane, Wei-Ning, Abdo, Yossi, Jade, Ann, Paden, Ali, Adam – that I had the chance to collaborate closely at the beginning of my PhD. Thanks to other members of the Audio/Textless EMEA team – Gabriel, Michael, Itai, Tal, Felix, Robin, Robin SR, Alex for the great conversations and collaboration on scaling SpeechLMs. Thanks Antony, Bowen, Maryam, Alexis for the delivery of the Expresso benchmark. Thanks Bokai, Benjamin, Paul-Ambroise, Juan, Sravya, Maha, Marta, Ruslan for the efforts of building the Spirit-LM model. I would also like to thank Alexei, Alexis, Michael Auli, Hugo, Thomas for the help and feedback on our work. I would like to thank all my CIFRE PhD fellows for the warm support and great moments during lunch at 6th floor, drinks and parties, babyfoot, social games that enrich my PhD life at FAIR. Thanks Pierre-Louis Xech for taking good care of all of us. Thanks to my FIFA friends at Meta for the fun moments and invaluable prizes. Thanks Jérémy for the support to FAIR new hires as well as organizing the FAIR Game Club.

I would like to thank my friends from the Vietnamese groups and communities who supported me and my wife during our life in France. Thanks to X-Việt family for always being with us, for the diverse nonsense and interesting daily conversations, and for the traditional events such as Tết X-Việt, Chào tân sinh viên, Trung thu, Chia tay chia chân that bring all the fun. Thanks to the FC Đồi group for fun football matches every weekend and for the joy of winning a match in a competition. Thanks to the AOE group for all the frustration but also the fun of raiding and climbing the ELO ladder. Thanks to the Orleans PhD group for the parties that make our life in Orleans less tedious. I would also like to thank my friends at the MVA Master for being there with me. Thanks to MaSSP for giving me the chance of meeting amazing mentors and students. Thanks to the Creteil Soleil family for considering us like brother and sister.

I would like to express my deep gratitude to the members of my family – my parents (mẹ Hoà, bố Hùng, mẹ Thuận, bố Nam), my French parents (Emmanuelle, Jean-Luc), my brother and sisters (anh Tuấn Anh, chị Linh, em Minh Anh) and cháu Bông, cháu

Thỏ, cháu Tiểu Hổ and other members of my family – who always care for me and support me unconditionally.

Last but absolutely not least, I would like to share my deepest gratitude to my beloved wife, Hải Yến, for being my greatest source of support both physically and mentally. Thank you for always being by my side, believing in me, and for your endless love. Thank you for all the support during my sleepless nights, for the delicious food everyday, for the tremendous fun that we share together. This thesis couldn't have been done without you. And also I don't forget to mention my gratitude for your untiring efforts to set up my wonderful PhD defense :p.

Publications

Included in the thesis

- Tu Anh Nguyen, Benoit Sagot, and Emmanuel Dupoux (2022a). "Are Discrete Units Necessary for Spoken Language Modeling?" In: *IEEE Journal of Selected Topics in Signal Processing* 16.6, pp. 1415–1423
- Tu Anh Nguyen, Eugene Kharitonov, Jade Copet, Yossi Adi, Wei-Ning Hsu, Ali Elkahky, Paden Tomasello, Robin Algayres, Benoît Sagot, Abdelrahman Mohamed, and Emmanuel Dupoux (Mar. 2023b). "Generative Spoken Dialogue Language Modeling". In: *Transactions of the Association for Computational Linguistics* 11, pp. 250–266
- Tu Anh Nguyen, Wei-Ning Hsu, Antony D'Avirro, Bowen Shi, Itai Gat, Maryam Fazel-Zarani, Tal Remez, Jade Copet, Gabriel Synnaeve, Michael Hassid, Felix Kreuk, Yossi Adi, and Emmanuel Dupoux (2023a). "Expresso: A Benchmark and Analysis of Discrete Expressive Speech Resynthesis". In: *Proc. INTERSPEECH 2023*, pp. 4823–4827
- Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R. Costa-jussa, Maha Elbayad, Sravya Popuri, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, Itai Gat, Gabriel Synnaeve, Juan Pino, Benoit Sagot, and Emmanuel Dupoux (2024). *SpiRit-LM: Interleaved Spoken and Written Language Model*. arXiv: `2402.05755 [cs.CL]`

Contents

0 Introduction

0.1 General Introduction

From the beginning of human society, speech has always been the dominant mode of human social bonding and information exchange, and has always been included in most communication technologies such as telephone, radio, television, and the Internet (Huang et al., 2001). However, modeling speech has not always been an easy task due to its variability (speaker's voice, accent, dialect), its ambiguity (homophones, contextual dependency) as well as the background noise contained in the speech itself. Modeling human spoken dialogues is, therefore, much more challenging and requires a great deal of effort, but it still remains not fully solved.

Classic cascaded speech systems (including popular systems like Siri, Alexa, or Google Assistant) rely on transcribing speech to text using an Automatic Speech Recognition (ASR) system and applying Natural Language Processing (NLP) systems on transcribed text. This approach removes the variability and ambiguity in the speech, but depends highly on the performance of the ASR system. In addition, the ASR system can potentially remove all the expressivity contained in the input speech and also in the output speech.

Recent years have witnessed the widespread of Large Language Models (LLMs), which heavily impacted not only the particular domain of NLP but also the whole field of Artificial Intelligence (AI). LLMs have been found to capture a general knowledge and understanding from a large amount of text corpora, and are able to perform various tasks without fine-tuning on single specific tasks in contrast to previous NLP systems (Brown et al., 2020). ChatGPT, one particular success of LLMs' applications, has gained gigantic attention from both the public and researchers since its release. It's among the first chatbot systems that can have human-like conversations and can generally answer questions in a human-desired manner and has been widely used as an AI assistant for daily tasks. However, LLMs have also been criticized for their trustworthiness, bias, and other legal issues (Weidinger et al., 2021).

Recent progress in Speech Processing, especially in Self-Supervised Speech Representation Learning (SpeechSSL) (Baevski et al., 2020c; Chung et al., 2021; Hsu et al., 2021a; Oord et al., 2018) has made it possible to learn from raw audio speech representations that are good for a variety of downstream tasks such as Automatic Speech Recognition (ASR), Speech Classification, or Speech Diarization (Yang et al., 2021). These methods allow learning discrete representations from speech, creating a bridge between speech and text technologies (Nguyen et al., 2020b). Language Models can thus be trained on the learned discrete speech representations, making it possible to model speech from raw audio without any text supervision (Lakhotia et al., 2021), this gives rise to a new research domain called "TextlessNLP". These so-called "SpeechLMs," however, still lag behind "TextLMs," but they allow for the processing of speech directly from audio with new potentials over cascaded systems.

The subject of this thesis is thus concentrated around this research domain. I will present our work that attempts to explore and improve the capability of SpeechLMs. Chapter 1 gives a background on the subject as well as on Spoken Language Models (SpeechLMs). I will then revisit the use of speech units in SpeechLMs (Chapter 2), then I will introduce an application of speech language modeling to spoken dialogue generation (Chapter 3), and continue to make resynthesized speech more expressive (Chapter 4), and, finally, I will present our attempt to combine SpeechLMs and TextLMs and close the gap between them (Chapter 5). In Chapter 6, I will talk about general contributions as well as the discussions/directions toward better spoken language systems.

0.2 Thesis Structure

This thesis is written with a publication-based structure. Each chapter from Chapters 2-5 represents one of my publications, generally followed by additional experiments attempting to explore or improve the work, and each one mainly covers one important subject in the field of spoken language modeling. However, they form a consistent body of work and can be seen as incremental efforts to build a spoken language modeling system. This will be discussed more in the last chapter (Chapter 6). The structure of the thesis is as follows:

Chapter 1: Background and Related Work

In this chapter, we will cover the background of the thesis as well as previous work related to the subject. We will start with a brief introduction of Language

Models, then we'll go through the basic knowledge of Speech Processing, followed by self-supervised speech models, which will play an important role in SpeechLM systems. Finally. we will give an introduction of the main subject of the thesis: *Spoken Language Modeling*. We'll start with initial work giving rise to the domain, as well as how spoken language modeling systems are evaluated and how they compare to text systems.

Chapter 2: Speech Tokenization Revisited

We study the question of whether tokenization is important for spoken language modeling, as presented in the paper: *Are discrete units necessary for Spoken Language Modeling?* We train language models on both tokenized and continuous speech features extracted from a Self-Supervised Speech Model and evaluate them on our spoken language modeling metrics. We find that tokenization is indeed important for spoken language modeling. We further analyze the influence of speech features as well as the number of speech tokens on the performance of different metrics. The chapter finishes with our supplementary experiments on analyzing larger-size speech units.

Chapter 3: Spoken Dialogue Language Modeling

This chapter covers the modelization of spoken dialogues, which is presented in the paper: *Generative Spoken Dialogue Language Modeling*, where we extend the generative spoken language modeling approach to spoken dialogues. We find that by representing dialogues as multi-channel audio, we can effectively modelize turn-taking in a conversation and are able to generate spoken dialogues with natural conversational cues (overlapping, laughter, back-channeling). However, we find a lack of semantics coherence in the generated speech, even for the cascaded model, suggesting a lack of training dataset. In the rest of the chapter, we present our attempts to improve the model by leveraging larger read speech datasets.

Chapter 4: Expressive Speech Resynthesis

Previous work on spoken language modeling only focused on read speech. We attempt to extend to more expressive speech by introducing a high-quality expressive dataset in the following work: *Expresso: A Benchmark and Analysis of Discrete Expressive Speech Resynthesis*. The dataset consists of 26 different expressive styles (e.g., angry, happy, sad, laughing, etc.). We also introduce a benchmark on expressive speech resynthesis and provide a detailed analysis of 2 different speech tokenization methods: HuBERT-based tokenization and EnCodec tokenization. We find that EnCodec is excellent in resynthesis tasks, but they are not controllable (i.e., cannot be used to resynthesize speech with other speakers or styles). Additionally, EnCodec

units don't capture phonetic information as well as HuBERT units. We'll give follow-up analyses of this work on comparing HuBERT tokens and EnCodec tokens on spoken language modeling tasks, and on improving HuBERT tokens with more expressive tokens.

Chapter 5: Text+Speech Language Modeling

We attempt to improve SpeechLMs by using TextLMs. We bridge the gap between Speech and Text LMs by combining them together into a single Speech+TextLM in: *SPIRIT: Interleaved Spoken and Written Language Model*. We found that LMs trained on interleaved speech and text can learn speech and text cross-modally and are able to generate language content in either modality. We evaluate the models with comprehension tasks in both speech and text, and extend few-shot prompting to speech-text tasks such as ASR, TTS or Speech Classification. We further proposed sentiment modeling metrics on speech generation and found that our model is able to preserve expressivity contained in the speech, in contrast to cascaded systems.

Chapter 6: Discussion and Perspectives

This chapter gives discussions on the general contributions of each chapter, which is followed by the perspectives and possible research directions that contribute to the domain of spoken language modeling.

1 Background and Related Work

In this chapter, we are going to introduce a new task called *Spoken Language Modeling (SLM)*, which aims to build spoken language systems directly on audio without any textual supervision[1]. We are going to present SLM in section 1.4, but before that we will learn more about language models (section 1.1), the basics of speech processing (section 1.2) as well as fundamental knowledge of self-supervised speech models (section 1.3).

1.1 Language Modeling

Language Models (LM) are models that are able to understand and/or generate natural language (texts). Formally, they are defined as probabilistic models that can assign probabilities to a sequence of words based on text corpora it was trained on (Jurafsky and Martin, 2009). For example, given two sentences:

```
The cat sat on the mat
The bat sat on the mat
```

a good language model should assign a higher probability to the first sentence rather than the second one. This is extremely beneficial for NLP tasks such as **speech recognition** or **machine translation**, where the model needs to choose the most probable output text sequence among a list of candidates (e.g. it's more probable to say *I love you* than *I law view*). It's worth noting that Language Models depend heavily on the training datasets. Return to the example above, a simple language model based on the occurences of single words (*1-gram*) trained on the plot of a Batman movie should assign higher probability to the second sentence than the first sentence.

[1]This comes from the following works: ZeroSpeech2021 Benchmark (Nguyen et al., 2020b) and GSLM (Lakhotia et al., 2021), which I contributed during my internship before this PhD.

Statistical Language Models approximate probability of word sequences using the chain rules:

$$P(\texttt{The cat sat}) = P(\texttt{The})P(\texttt{The} \rightarrow \texttt{cat})P(\texttt{The cat} \rightarrow \texttt{sat}) \tag{1.1}$$

where the probability of chaining the next word can be approximated using statistics over the training dataset. N-gram LMs (Jelinek and Mercer, 1980; Katz, 1987) are among the most popular Statistical LMs, they assume that the probability of predicting the next word only depends on previous n-1 words, and therefore only consider all *tuples of n consecutive words* (or *n-gram*). For example, in the previous example, a 2-gram model would approximate $P(\texttt{The cat} \rightarrow \texttt{sat}) \approx P(\texttt{cat} \rightarrow \texttt{sat})$, which is then computed by counting all the tuples starting with cat as follows

$$P(\texttt{cat} \rightarrow \texttt{sat}) = \frac{\text{number of } \texttt{cat sat}}{\text{number of } \texttt{cat *}}. \tag{1.2}$$

N-gram models are simple and are, therefore, widely used even in modern language systems. However, a drawback of n-gram is that the number of n-grams increases on a power scale (known as *curse of dimensionality*), making it difficult to compute n-gram for large n, and creating many zero-probability sequences. This later issue can be mitigated by *smoothing* over unseen sequences (Kneser and Ney, 1995).

Neural Language Models employ *neural networks* to approximate the probability of word sequences. Neural networks are a class of models that use a vast number of *parameters* to learn or estimate a given objective function. Similar to statistical language models, neural language models compute the probability of word sequences using the chain rules:

$$P(w_1, w_2, \ldots, w_t) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\ldots P(w_t|w_1, w_2, \ldots, w_{t-1}) \tag{1.3}$$

and the probability $P(w_t|w_1, w_2, \ldots, w_{t-1})$ is also approximated within a context of n words $P(w_t|w_1, w_2, \ldots, w_{t-1}) \approx P(w_t|w_{t-n}, \ldots, w_{t-1})$. Unlike n-gram models, neural networks permit to estimate efficiently $P(w_t|w_{t-n}, \ldots, w_{t-1})$ even for large n with its huge number of parameters. Bengio et al. (2000) was probably the earliest neural language model, which used a *shallow* network to estimate the probability of a word given n previous *context* words. At the beginning of the deep learning era (2010s), neural language models became much more popular, recurrent neural networks (RNN, Rumelhart and McClelland, 1987; LSTM, Hochreiter and Schmidhuber, 1997) have then become a norm for neural language models (Graves, 2014; Mikolov et al., 2010).

Since the apparition of Transformer (Vaswani et al., 2017), language models have proven to possess a good comprehension of texts given enough training data and computing resources. BERT (Devlin et al., 2019) and GPT (Radford et al., 2018) are two notable examples of Transformer-based language models. BERT utilizes a special Masked Language Modeling (MLM) task to predict the masked words in a sentence using a Transformer Encoder architecture (*Masked LM*), while GPT employ the classic Language Modeling task which consists of predicting the next word in the sentence using a Transformer Decoder architecture (*Autoregessive LM*). These models are widely used as pre-train models for fine-tuning specific domains or specific downstream tasks in NLP. Other popular Transformer-based LMs include RoBERTa (Liu et al., 2019), XLNET (Yang et al., 2019), Electra (Clark et al., 2020), BART (Lewis et al., 2020a), T5 (Raffel et al., 2020).

Large Language Models (LLM) have become a standard in NLP since the release of GPT-3 (Brown et al., 2020). They found that scaling autoregessive language models on billions of parameters (175B) with massive datasets helps to achieve general-purpose language understanding and generation, and that the models can solve new tasks by giving only a few examples in the *prompt*, or model input (*few-shot prompting*). This enables solving a number of NLP tasks just by appropriately setting the prompt without requiring to fine-tune the model as before (e.g. Chain-of-Thought prompting technique, Wei et al., 2022b). Since then, a number of LLMs have been developed (Chowdhery et al., 2022; Hoffmann et al., 2022; Touvron et al., 2023a; Zhang et al., 2022). Notably, LLaMA (Touvron et al., 2023a) showed that smaller LLMs (7B parameters) can achieve very good performance when training longer on more data using optimal-compute scaling laws (Kaplan et al., 2020), making LLMs more accessible for NLP research. It has been shown that larger LLMs tend to possess more abilities that smaller models don't have (*emergent abilities*). For example, GPT-3 model can perform 3-digit addition at 13B parameters but can only perform 4-digit addition at 175B parameters (Wei et al., 2022a).

ChatGPT (OpenAI, 2022) is probably the most successful system since the apparition of LLMs. They found that LLMs fine-tuned on Human Instructions using Reinforcement Learning with Human Feedback (RLHF, Ouyang et al., 2022) can become a powerful chatbot system and is able to generate human-like conversation and assist users on many tasks (Bowman, 2023). This has gained much attention from not only researchers but also from the public. Since then, a huge number of works have included RLHF in their LLMs, and many companies have integrated ChatGPT into their systems or developed their own chat-LLM systems. Despite

being popular for their success, chat-LLM systems also suffer from many critics (Deshpande et al., 2023). People can use them to help with harmful use cases. Hallucination is also a huge problem for LLMs. As they are probability-based, their generated texts are not truthful. Generation from LLMs could also be socially toxic. As their performance depends on the training dataset, these models can generate discriminative texts or be biased on some subjects. As part of Generative AI, LLMs could also be used to generate fake news or information, which is dangerous as soon as it becomes widespread. *Whether LLMs are able to understand what they said* is also a debatable question.[2] People were scared that one day, LLMs could achieve general understanding and could harm people.

Multimodal LM is another interesting direction since the LMs era. People are using LLM not only on text, but also extend on other modalities (image, audio, speech, video) to LLMs. Flamingo (Alayrac et al., 2022) is one of the first works to combine Text+Image in an LLM. Their method relies on integrating image features into LLM using an image encoder (e.g., CNN) and fine-tuning the model on image captions so that the model is able to understand the content of the images. Parti (Yu et al., 2022) tokenizes images into tokens with an image tokenizer and trains an LLM that is able to generate images from the text description. For speech modality, Lakhotia et al. (2021) introduced GSLM, a generative spoken language model that is able to generate speech from raw audio using a speech tokenizer obtained from self-supervised speech models. Since then, many speech and audio language models have also been introduced (Agostinelli et al., 2023; Borsos et al., 2023; Kreuk et al., 2023; Rubenstein et al., 2023).

1.2 Speech Processing

1.2.1 How Speech Signal is represented

Raw Audio Waveform Raw audio signal is represented as a sequence of numbers (or samples) indicated as the *amplitude of the audio* over time. The number of samples every second is called the *sample rate* and is measured in Hz (hertz). A higher sample rate means the speech is of higher resolution and quality. In speech processing, the general sample rate is 16Khz or 16,000 speech samples per second. This sample rate is enough for models to perceive the information in speech. For

[2] `https://twitter.com/geoffreyhinton/status/1728490334336770138`

more diverse audio (e.g. music), the sample rate is much higher, the standard sample rate for general audio is 44.1Khz and 48Khz.[3] Each audio sample can be stored in digital format using certain possible amplitude values. The number of possible amplitude values is called *audio bit depth*. For example, each sample of 4-bit audio can have $2^4 = 16$ different values. The most common bit depths are 16-bit, 24-bit or 32-bit. Most speech datasets are stored in 16-bit, which means 65 536 different amplitude values. Finally, an audio file can have only one channel (*mono*), where there is only one waveform representing the audio, or two channels (*stereo*), one waveform for the sound coming from the left and one for the right, which resembles how we hear in real worlds and creates a 3-D effect on the audio. This means that speech signals contain a lot of information and are costly to store, especially compared to text. For example, to store an audiobook of the first volume of Harry Potter, one needs 7 hours of speech,[4] or 400M speech samples at 16Khz, which takes 800 MB (megabytes)[5] in disk space. In contrast, the text version of the book contains about 76K words,[6] which takes up 400KB (kilobytes) in disk space, 2,000 times smaller than that of speech!

Spectrogram Raw audio signals contain only amplitude information of the speech but not other information (e.g. pitch, phonetic). The spectrogram is another representation of audio signals that contain the visual information of the *frequencies* (or *pitches*, measured in Hz) contained in the speech or audio. The spectrogram is a sequence of *spectrum*, or a decomposition of different frequencies contained in a short audio segment (or window), over time.[7] The spectrum is calculated using a special "frequency-analysis" operator, usually known as Fourier transform, and contains a density over a frequency bin (e.g. 0-10,000Hz). For example, a segment of a female voice will have a spectrum concentrated on around 200Hz, while a segment of a male voice will be around 100Hz. The spectrums are calculated over consecutive small segments in the speech (e.g., 0-100ms, 50-150ms, 100-150ms, ..), and are concatenated to form a single spectrogram of the speech, which can be visualized as a frequency-time image. In speech processing, frequencies in spectrogram are usually *scaled* (or converted) using a *mel scale* so that the scaled frequencies correspond with human-perceived frequencies, this is called the *mel-spectrogram*. *MFCC (Mel Frequency Cepstral Coefficients)* is another type of spectrogram representation where the mel frequencies are further converted using a *discrete cosine transform*,

[3] https://www.izotope.com/en/learn/digital-audio-basics-sample-rate-and-bit-depth.html
[4] https://www.youtube.com/watch?v=h72Mlk94wrQ
[5] https://www.colincrawley.com/audio-file-size-calculator/
[6] https://wordcounter.io/blog/how-many-words-are-in-harry-potter
[7] https://www.phon.ucl.ac.uk/courses/spsci/acoustics/week1-10.pdf

this representation can allow for better representation of speech[8], and is commonly used in speech and audio processing.

Speech Features While speech models can take the raw audio signal or spectrogram as input and directly optimize the given tasks (e.g., recognition or classification) in the output (called *end-to-end models*), many speech systems use learned features or representations, from speech to help to learn meaningful information from the speech. Classic speech recognition systems use speech features extracted from spectrogram to learn the acoustic or phonetic information from the speech (Alim and Rashid, 2018). Later, some classic statistical models (e.g., *Hidden Markov Model* or HMM) are used to learn speech features from the speech (Jelinek, 1976; Levinson et al., 1983). Recent speech systems use learned speech features or representations using neural networks. *x-vector* (Snyder et al., 2018) and *ECAPA-TDNN* (Desplanques et al., 2020) capture speaker representation from the speech using deep neural networks, and are commonly used to extract speaker features, or embeddings, from the speech. With the development of self-supervised speech models (*Wav2vec 2.0* Baevski et al., 2020c; *HuBERT* Hsu et al., 2021a; *wavLM* Chen et al., 2022), speech features obtained from these models can capture various meaningful speech representations, from paralinguistic (e.g. speaker, emotion), acoustic (e.g. phonetic) to higher linguistic levels (e.g. words, sentences). For example, in this thesis, we will introduce discrete representations (called *speech units*, or *pseudo-text*), which are quantized representations from self-supervised features that capture phonetic information from speech. *T-modules* (Duquenne et al., 2022) used wav2vec2.0 to embed the speech into a joint speech-text space to obtain sentence-level speech representations.

1.2.2 Speech Processing Tasks

In this section, we will introduce several significant speech processing tasks among the many that exist in the field.

Automatic Speech Recogntion (ASR) is probably the most well-known speech processing task and is the most popular research domain in speech processing. The task of ASR is to automatically transcribe the speech into text. ASR plays a vital role in most speech systems as it transforms user input into text so that the systems can apply NLP systems on transcribed texts. The history of ASR dates back to the year

[8] `https://en.wikipedia.org/wiki/Mel-frequency_cepstrum`

1950s (Juang and Rabiner, 2005) with early systems using *formant frequencies* to recognize spoken digits (Davis et al., 1952). Classic ASR systems require a complex combination of speech feature analysis, statistical acoustic models (e.g. HMM), and language models (e.g. LSTM). Since 2010s, neural network ASR systems have outperformed classic ASR systems and have become the standard in ASR for their simplicity. While classic ASR systems treat speech from spectrogram representations to extract useful information from speech, recent deep ASR systems show the possibility of training end-to-end and processing speech from the raw waveform by using convolution layers to extract low-level representations from the speech (Amodei et al., 2016; Sainath et al., 2015). Self-attention networks (Vaswani et al., 2017) also contribute significantly to current state-of-the-art ASR systems (Baevski et al., 2020c; Gulati et al., 2020; Radford et al., 2023). The most recent one, Whisper, has gained huge attention and success for its extreme performance in multilingual, noisy speech, achieving human performance in speech recognition.

Speech Synthesis, or Text to Speech (TTS), is another main research field in Speech Processing. As inverse to ASR, the task of TTS is to synthesize natural, human-like speech from a given text. From an answer from your home assistant, a train announcement to news you are listening to, synthetic speech (the output of a TTS system) can appear everywhere nowadays. One early known TTS system is Voder (Dudley, 1940), which is a complex machine that generates speech from hundreds of combinations of sounds and requires a trained person to perform the synthesis (similar to playing the piano, but more difficult!). Classic speech synthesis models were based on complex speech features such as formant frequencies or based on articulatory modeling. Similar to the ASR task, deep TTS models (neural network-based) have been well studied and surpassed the performances of classic TTS systems. Neural TTS models can either be composed of 2 stages: text-to-spectrogram (Ren et al., 2022; Shen et al., 2018) and spectrogram-to-waveform (also called *vocoder*, Kong et al., 2020; Prenger et al., 2019) or generate directly waveform from text (end-to-end) (Kim et al., 2021).

Speech Diarization is a special speech processing task where the goal is to distinguish each speaker from a speech recorded in a multi-speaker environment (e.g. a dialogue or a party). This task is important for robust ASR systems as it can help to differentiate the speaker's voice from a noisy speech input.

Speech Classification Tasks Speech Classification refers to a wide range of speech processing tasks where the goal is to predict the label of a speech input. Some examples of speech classification tasks include Speech Command Recognition, Speaker Recognition, Emotion Classification, or Language Identification. Recently, with the development of self-supervised speech systems, Speech Classification tasks could be done by fine-tuning pre-trained self-supervised speech models, which gives excellent results (Yang et al., 2021). Now we are going to discuss more about self-supervised speech models in the next section.

1.3 Self-supervised Speech Learning

Training a supervised speech system (e.g. ASR system) requires a lot of human-annotated labels, which can be very costly to obtain, especially for large-scale speech datasets. Self-supervised Speech Learning is a solution to this problem. Self-supervised speech models learn speech representations (or features) in an unsupervised manner, making use of large-scale unlabeled speech datasets to learn the underlying structure of the speech.

1.3.1 Self-supervised Learning

Self-supervised Learning (SSL) In Machine Learning, Self-supervised Learning methods refer to training methods where the training labels come from the data itself rather than from human-annotated labels. Therefore, it is an unsupervised learning method (learning without using labels, as contrasted with *supervised learning*). The trained model would be expected to learn a good representation of the data and can be used to extract data information or further fine-tune (continue to train) on specific downstream tasks (e.g. Image Classification for Images, Sentiment Analysis for Text, or Speech Recognition for Speech). SSL is often used to first leverage large amounts of unlabeled data and consequently applied on fewer sets of labeled data. It is therefore considered as the bridge between unsupervised learning (Rani et al., 2023).

Self-supervised Learning Objectives Without labels that act as targets for the prediction task, SSL needs a *pretext task*, or objective, to train the network. The SSL objectives can be classified into *contrastive* and *non-contrastive*. *Contrastive SSL methods* aim to learn data representations such that similar instances are close

together in the representation space, while dissimilar instances are far apart.[9] In the contrastive objective, there is often one target (or *positive sample*) sample and one (or multiple) *negative samples*, and the goal is to maximize the similarity of the output representation (or *anchor*), with the positive sample, while minimizing the similarity with the negative ones. In Computer Vision, the positive sample could be a crop of the input image, while negative ones are sampled across the dataset (Chen et al., 2020). *Non-constrastive SSL methods* often use a prediction task to either reconstruct the input or to predict other representations from the input. Auto Encoder models (models that *encode* input data to a hidden space and *decode* back to reconstruct the input), notable Variational Auto Encoder (VAE, Kingma and Welling, 2014), fall into non-constrative SSL methods. Language Models can also be considered as SSL methods as they also use input text as their training labels (e.g. BERT predicts the masked words in the input sentence, while GPT uses the next words as their prediction targets).

1.3.2 Self-supervised Speech Models

In Speech Processing, SSL models have been widely studied and introduced recently and are mostly employed for downstream speech recognition or speech classification tasks (Mohamed et al., 2022). We will introduce some popular Self-supervised Speech (SpeechSSL) models that will be discussed later in this thesis, please have a look at Mohamed et al. (2022) for a comprehensive review of SpeechSSL models.

CPC Contrastive Predictive Coding (Oord et al., 2018) (CPC) learns audio representations by predicting the *future* in hidden space using autoregressive models. The CPC model consists of a temporal convolutional encoder (CNN, LeCun et al., 1989) and an autoregressive predictor (LTSM, Hochreiter and Schmidhuber, 1997). The CNN Encoder takes the raw waveform $\mathbf{x}$ and produces hidden features $\mathbf{z} = (z_1, \ldots, z_T)$ of much lower rate (generally 100Hz) than the input waveform (16Khz). At each timestep t, the LSTM Predictor predicts the next k features $(z_{t+1}, \ldots, z_{t+k})$ given the past $(z_1, \ldots, z_t)$ with a contrastive objective. The negative samples are randomly sampled from the other hidden features (either the same or other sequences). The CPC model has been shown to capture speaker and acoustic information from the speech depending on how the negative samples are chosen (Oord et al., 2018; Rivière et al., 2020).

[9] `https://paperswithcode.com/task/contrastive-learning`

wav2vec 2.0 The wav2vec 2.0 (Baevski et al., 2020c) model learns speech representations by predicting the *masked quantized features* in the hidden space using a Transformer encoder (Vaswani et al., 2017). Similar to CPC, the wav2vec 2.0 model also consists of a CNN Encoder which extracts the hidden features $\mathbf{z} = (z_1, \ldots, z_T)$ from the speech. Inspired from BERT (Devlin et al., 2019), the hidden features $\mathbf{z}$ are partially masked (z_t is replaced with a masked feature vector $\hat{z}_{Mask}$ for all time steps t in $T_{Mask} \subset \{1, \ldots, T\}$) and then fed into a Transformer Encoder which captures the information from the whole sequence. The Transformer Encoder is trained with a contrastive objective to predict the quantized embeddings of the masked features $\{q_t = emb\,(g_q(z_t)) \mid t \in T_{Mask}\}$ where g_q is a quantization module (Gumbel-Softmax, Jang et al., 2017). This quantization step has been shown to benefit the learned representations in prior work (vq-wav2vec, Baevski et al., 2020b). The negative samples are uniformly sampled from the other quantized masked features of the same sequence. The wav2vec 2.0 model showed for the first time the powerful of Speech SSL model by achieving excellent preformances on ASR task by fine-tuning on only 10 minutes of labeled speech.

HuBERT The HuBERT (Hsu et al., 2021a) model has the same architecture as the wav2vec 2.0 model with a CNN Encoder followed by a Transformer Encoder. Unlike CPC and wav2vec 2.0 that use a contrastive loss, HuBERT is trained with a prediction objective to predict the target labels obtained from *teacher representations*. Similar to wav2vec 2.0, the speech is encoded to hidden features $\mathbf{z} = (z_1, \ldots, z_T)$ which are partially masked and fed to the Transformer Encoder. The teacher representations $\mathbf{y} = (y_1, \ldots, y_T)$ are clustered using k-means (MacQueen et al., 1967) and the cluster ids at the mask time steps are used as the target labels of the Transformer Encoder $\{l_t = cluster\_id(y_t) \mid t \in T_{Mask}\}$. The training of HuBERT model is done in multiple iterations, where the quality of the teacher $\mathbf{y}$ is improved in each iteration. The teacher of the first iteration are MFCC features. The next iterations use the hidden features of previous iterations as teacher. The hidden features are extracted from an intermediate layer of the Transformer Encoder, which was chosen to maximize phonetic mutual information metrics. HuBERT model therefore captures good semantic information from the speech and have excellent results, especially in speech recognition tasks.

1.3.3 Evaluation of Self-supervised Speech Models

There exist many metrics and benchmarks used to evaluate self-supervised speech models. The metrics can be evaluated by fine-tuning the models on downstream

tasks (e.g. SUPERB, Yang et al., 2021; LeBenchmark, Evain et al., 2021) or by extracting models' outputs (e.g. Zero Resource Speech, Dunbar et al., 2019, 2017, 2020; Versteegh et al., 2015)

SUPERB The SUPERB benchmark (Yang et al., 2021) is a collection of tasks used to evaluate the quality of speech reprentation of self-supervised speech models. It consists a wide range of downstream speech processing tasks ranging from content (PR, ASR, KS, QbE) and speaker identity (SID, ASV, SD) to semantics (IC, SF) and paralinguistic (ER). The speech SSL models are fine-tuned and evaluated on each task, and can be compared with other models through an active online leaderboard.[10]

Zero Resource Speech Challenge The Zero Resource Speech (ZeroSpeech) Challenge (Dunbar et al., 2019, 2017, 2020; Versteegh et al., 2015) is a series of benchmarks that evaluates the progress of speech systems that work without any textual supervisions. The series began since 2015 and the long-term objective was broken down into four incremental tasks: Acoustic Unit Discovery, Spoken Term Discovery, Discrete Resynthesis, and Spoken Language Modeling. The metrics can be evaluated in an unsupervised manner (without fine-tuning) by probing the systems' outputs directly. Similar to SUPERB, ZeroSpeech metrics and benchmarks are made available[11] which enable comparison and tracking progress of spoken language systems.

1.4 Spoken Language Modeling from Raw Audio

Classic approaches on speech generation rely on cascaded systems, i.e. transcribe speech to text with an ASR model, then generate text responses and convert it back to speech with a speech synthesis model. These approaches work well but they heavily depend on the performance of ASR systems, in addition normalize speech to text remove all acoustic information contained in speech (voice characteristics, expressivity) and can make the synthesized speech unnatural. In this section, we are going to introduce a new task called *spoken language modeling from raw audio* (Lakhotia et al., 2021; Nguyen et al., 2020b), where the goal is to perform language modeling on speech directly without passing through text. This can open up new

[10] `https://superbbenchmark.org/leaderboard`
[11] `https://www.zerospeech.com/`

possibilities of speech modeling, such as expressive speech modeling, multichannel speech modeling, or even general audio modeling.

1.4.1 Spoken Language Models (SpeechLMs)

Language Models work very well on text, but how about speech? As previously discussed, there are some main disavantages of speech compared with text for the task of language modeling:

- Speech waveforms are extremely long compared with text. For example, the sentence *one day the park plays* is presented as 5 text tokens, but is worth 64,000 audio frames. This long-term context dependency is a main issue for current language models, and therefore can severely affect the performance of LM on speech.

- Speech signals are continuous, making it hard for language modeling task. Text, on the other hand, is tokenized to discrete tokens, and can therefore be embedded to a vocabulary which is optimized both LM input and prediction stages.

- Speech contain various information compared with text: linguistic content, voice characteristics, background noise, etc. Training a language model on speech can be very difficult since LM doesn't know which information of speech to learn from.

To deal with the issues mentioned above, Nguyen et al. (2020b) and Lakhotia et al. (2021) proposed a simple yet effective approach to train language models on speech: Tokenize speech into discrete units of lower frame rate which contain linguistic information of the speech, and then train a language model on the discrete units (Figure 1.1).

1.4.1.1 ZeroSpeech 2021 Baseline System

In Nguyen et al. (2020b), we proposed a baseline system for the task of Spoken Language Modeling which consists of 3 composite systems: a self-supervised speech model (CPC), a clustering module (k-means) and a language model (BERT).

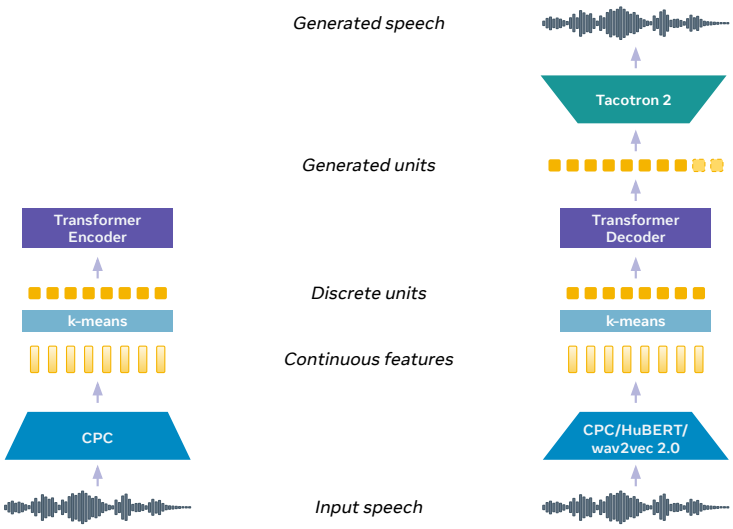


Fig. 1.1: Baseline Speech Language Models (SpeechLMs) ZeroSpeech 2021 Baseline System (left) and GSLM system (right). In both systems, the input speech is transformed into discrete units by clustering continuous speech representations obtained from self-supervised speech models. Then a language model is trained on discrete units. The ZeroSpeech 2021 Baseline system uses a Transformer Encoder with a masked prediction objective, while GSLM uses a Transformer Decoder with an autoregressive objective. GSLM further employs a Text-to-Speech model (Tacotron 2) to convert discrete units back to speech.

Acoustic Speech Features Extraction. Previous work (Niekerk et al., 2020; Rivière et al., 2020) show that representations from self-supervised speech models like CPC capture well phonetic information from the speech. Following this, we trained a CPC model to extract meaningful representations from the speech. The CPC model consists of a 5-layer 1D-CNN Encoder followed by a multi-layer LSTM Autoregressive network. We perform an in-depth analysis of phonetic quality over hidden representations of the CPC model using the ABX metrics [12] (Schatz et al., 2013) and extract the speech features from the hidden layer of the autoregressive network of the CPC model which has the best ABX. The final extracted speech features have a frame-rate of 100Hz (160 times smaller than the actual sample rate of 16KHz) and contain phonetic-like information in its representation.

Speech Feature Quantization. Quantizing speech features has been shown to be beneficial for acoustic unit discovery (Niekerk et al., 2020) and speech recognition (Baevski et al., 2020a). We also followed this and quantized the continuous speech features using the k-means clustering method. The clustering is done on the col-

[12]The ABX metrics will be presented in detail in section 1.4.2

lection of all the output features at every time step of all the audio files in a given training set. After training the k-means clustering, each feature is then assigned to a cluster, and each audio file can then be discretized to a sequence of discrete units corresponding to the index of the assigned clusters. We initially explored various numbers of clusters and evaluated the phonetic quality of the discrete units using the ABX metrics. We also included multiple-group clustering in our experiences as similar to Baevski et al. (2020b). We found that doing k-means with 50 clusters gives the best results in our case.

Language Modeling on Quantized Speech Units. With the speech feature quantization, we now have a discrete version of speech with low-level linguistic information (phonetic) and, therefore, can finally train a language model on the discrete units to capture high-level language properties from the speech. Following Baevski et al., 2020b, we trained a BERT (Devlin et al., 2019) model on the units with only the masked token prediction objective. We also followed Baevski et al. (2020b) by masking a span of tokens in the input sequence instead of a single token (otherwise, the prediction would be trivial to the model as discretized units tend to replicate).

This composite system partly solves the issues mentioned at the beginning of the section, as it discretizes speech into coarser tokens containing linguistic content and permits us to learn a language model on speech effectively. We will show in section 1.4.3 that the system is indeed able to learn useful information from the speech and that it is possible to perform language modeling from raw audio.

1.4.1.2 GSLM: Generative Spoken Language Modeling

The previous systems allow us to learn language models directly on raw speech, but they are not generative as they used a Transformer Encoder architecture. In addition, generating new speech requires a system that can generate speech back from the discrete units. In Lakhotia et al. (2021), we extend Nguyen et al. (2020b) to generative speech modeling. We propose GSLM (Generative Spoken Language Model), a system that can perform speech generation by training an autoregressive language model on the discrete units and synthesizing speech from the generated units using a TTS system.

The GSLM consists of 3 components: a Speech-to-unit (S2u) model that encodes speech to discrete units (or *pseudo-text*), a unit-Language Model (uLM) to perform generative modeling of the speech, and a unit-to-Speech (u2S) model that synthesizes speech from pseudo-text.

Speech-to-Unit (S2u) Similar to Nguyen et al. (2020b), the S2u model is composed of a speech feature extractor followed by a quantization module. We employed various self-supervised speech models as feature extractors, including CPC, wav2vec 2.0, and HuBERT. We also included a log Mel filter-bank baseline (with 80 frequency bands, computed every 10ms) to analyze the importance of the speech extractor quality. For the quantization, we used k-means to convert continuous speech features into discrete representations by training on the train-clean-100 subset of LibriSpeech. We experimented with different codebooks that have 50, 100, and 200 units.

unit-Language Model (uLM) We use a decoder Transformer architecture to train a language model on sequences of pseudo-text units. The Transformer model has 12 layers, 16 attention heads, an embedding size of 1024, an FFN size of 4096, and a dropout probability of 0.1. The uLM is trained on "clean" 6k hours sub-sample of LibriLight used in Rivière et al. (2020), transcribed with corresponding discrete units. Unlike Nguyen et al. (2020b), the discrete units are deduplicated[13] before being fed to the language model. In preliminary experiments, we found that removing sequential repetitions of units improves performance. We hypothesize that this simple modification allows us to use the Transformer's limited attention span more efficiently as in Hsu et al. (2021b).

Unit-to-Speech (u2S) We adopt the *Tacotron-2* model (Shen et al., 2018) such that it takes pseudo-text units as input and outputs a log Mel spectrogram. To enable the model to synthesize arbitrary unit sequences, including those representing incomplete sentences, we introduce two modifications. First, we append a special "end-of-input" (EOI) token to the input sequence, hinting the decoder to predict the "end-of-output" token when attending to this new token. However, this modification alone may not be sufficient, as the decoder could still learn to ignore the EOI token and correlate end-of-output prediction with the learned discrete token that represents silence as most of the speech contains trailing silence. To address this, we train the model using random chunks of aligned unit sequence and spectrogram, and append the EOI token to unit sequence chunks, such that the audio does not always end with silence. We implement chunking in the curriculum learning fashion, where the chunk size gradually grows (starting with 50 frames with an increment of 5 per epoch) to increase the difficulty of the task. For waveform generation, we use the pre-trained flow-based neural vocoder *WaveGlow* (Prenger et al., 2019). This model outputs the time-domain signal given the log Mel spectrogram as input. All u2S models were trained on LJ Speech (LJ) Ito and Johnson, 2017.

[13] For example, a pseudo-text `10 11 11 11 21 32 32 32 21` becomes `10 11 21 32 21`.

We will see in section 1.4.3 that it is possible to train a generative language model from quantized units and that the GSLM system can generate new speech in both prompted (given input speech) and unprompted (no input speech given) conditions. However, we first need to see some evaluation methods of spoken language models in the following section.

Note that in the following of this thesis, we will call these systems that perform language modeling on raw speech **Spoken Language Models**, or **Speech Language Models**, or **SpeechLMs** interchangeably. The same thing for classic **Language Models**, or **Text Language Models**, or **TextLMs**.

1.4.2 Spoken Language Models Evaluation Metrics

1.4.2.1 Zero-shot Comprehension Metrics

Inspired by textual evaluation metrics of language models, speech language models' comprehension can be evaluated using black-box tests. This involves providing speech inputs and computing model performances based on their outputs.

In Nguyen et al. (2020b), we proposed 4 metrics used to probe the understanding of speech language models at different linguistic level: Phonetics (Libri-light ABX Metrics), Lexicon (sWUGGY Spot-the-word Metrics), Syntax (sBLIMP Acceptability Metrics) and Lexical Semantics (sSIMI Similarity Metrics).

Phonetics: The Libri-light ABX Metrics. The ABX discriminability metric evaluates the speech representations in terms of phonetic quality, it can be simply seen as an unsupervised, contrastive version of phoneme classification accuracy and has been introduced in Schatz et al. (2013). Given a pair of similar triphones (e.g. 'aba'-'apa') and an intervening sound (either 'aba' or 'apa'), the model has to tell which sound has a closer representation to the intervening sound. The ABX metric is reported as the error rate in which the model fails to choose the correct triphone. Formally, given two speech categories A and B (e.g. triphones 'aba' and 'apa'), we compute the following asymmetric error score:

$$\hat{e}_{ABX}(A,B) := \frac{1}{n_A(n_A-1)n_B} \sum_{\substack{a,x\in A \\ x\neq a}} \sum_{b\in B} \left[\mathbb{1}_{d(b,x)<d(a,x)} + \frac{1}{2}\mathbb{1}_{d(b,x)=d(a,x)} \right] \quad (1.4)$$

Tab. 1.1: Summary description of the Spoken Zero-shot Comprehension Metrics. The metrics in light blue use model's representations to compute a pseudo-distance (distance d or similarity $\hat{s}_M$) between input embeddings, the metrics in light orange use a pseudo-probability P computed over the entire input sequence.

Linguistic Level	Metrics	Data	Task	Example
acoustic-phonetic	ABX	(a, b, x)	$d(a, x) < d(b, x)$?	(apa, aba, apa)
lexicon	sWUGGY	(w, nw)	$P(w) > P(nw)$?	(brick, blick) (squalled, squilled)
syntax	sBLIMP	(cor, inc)	$P(cor) > P(inc)$?	(dogs eat meat, dogs eats meat) (the boy can't help himself, the boy can't help herself)
lexical semantics	sSIMI	(w_1, w_2, s_H)	$\hat{s}_M(a, b) \propto s_H(a, b)$?	(abduct, kidnap, 8.63) (abduct, tap, 0.5)
contextual semantics	T-StoryCloze	$(cont, cor, inc)$	$P(cont, cor) > P(cont, inc)$?	(Shyanne had a spelling test. She wanted to pass it. She studied hard. She made a 100., Shyanne was overjoyed., Their mom enjoyed her new, broken vase.)
commonsense reasoning	S-StoryCloze	$(cont, cor, inc)$	$P(cont, cor) > P(cont, inc)$?	(Shyanne had a spelling test. She wanted to pass it. She studied hard. She made a 100., Shyanne was overjoyed., She was heartbroken.)

where n_A, n_B are the cardinalities of A, B respectively; a and x are two different sounds belonging to category A and b belonging to B (in this example a and x are the same triphones 'aba' and b is 'apa'); and $d(a, b)$ is a distance between the representations of a and b computed as the average cosine distance along the Dynamic Time Warping path of the two representations. The ABX error score is symmetrized and aggregated across all minimal pairs of triphones like ('aba', 'apa'), where the change only occurs in the middle phoneme. This score can be computed *within speaker* (in which case, all stimuli a, b and x are uttered by the same speaker) or *across speaker* (a and b are from the same speaker and x from a different speaker). The ABX metric is agnostic to the dimensionality of the embeddings, can work with discrete or continuous codes, and has been used to evaluate acoustic features in the Zero-Ressource Challenge Series (Dunbar et al., 2019, 2017, 2020; Versteegh et al., 2015). The Libri-light ABX metrics are computed on the pre-existing Libri-light dev and test sets, which have already been used to evaluate several self-supervised models (Kahn et al., 2020; Rivière et al., 2020).

Lexicon: The sWUGGY Spot-the-word Metrics. Unlike most text-based Language Models, where the text inputs are tokenized at word or sub-word levels, speech language models have much more fine-grained input features and therefore can struggle at recognizing words. Inspired by Godais et al. (2017), who used 'spot-the-word' task to evaluate character-level Language Models, we proposed the sWUGGY metric which evaluates model's capability to detect an existing word against a similar pseudo-word. In this task, the models are presented with a pair of sounds: an existing word and a matching nonword (e.g. 'brick' – 'blick'), and are evaluated on their capacity to attribute a higher probability to the existing word. The sWUGGY

spot-the-word metric corresponds to the accuracy of correctly classifying the words across all pairs:

$$score_{swuggy} := \frac{1}{n_D} \sum_{(w,nw)\in D} \left[\mathbb{1}_{P(w)>P(nw)} + \frac{1}{2} \mathbb{1}_{P(w)=P(nw)} \right] \tag{1.5}$$

where D is the set of all pairs of existing word and nonword (w, nw); n_D is the number of pairs; and $P(w)$ is a *pseudo-probability* given by the model for the input slimuli w. The nonwords are created with WUGGY (Keuleers and Brysbaert, 2010), which generates for a given word a list of candidate nonwords best matched in phonotactics and syllabic structure. The matching nonwords are carefully chosen from the candidates to ensure high-quality speech synthesis (i.e. all nonwords should sound plausible) and to match unigram and bigram phoneme frequencies. The word pairs are then synthesized to speech using a TTS system with 4 different speakers. The sWUGGY dataset consists of 2 subsets: an in-vocab-sWUGGY subset with the existing words being part of the LibriSpeech train vocabulary and an OOV-sWUGGY subset with existing words which do not appear in the LibriSpeech training set.

Syntax: The sBLIMP Acceptability Metrics. The sBLIMP metrics are adapted from BLIMP (Warstadt et al., 2020), a dataset of linguistic minimal sentence pairs of matched grammatical and ungrammatical sentences (e.g. 'He love eating pizza' - 'He loves eating pizza'). Similarly to the sWUGGY metrics, the task is to decide which of the two members of the pair is grammatical based on the probability of the sentence. The sBLIMP acceptability metric is thus computed as the accuracy that the model assigns a higher probability to the grammatical one. The sBLIMP dataset was created by adapting the code used for generating the BLIMP dataset (Warstadt et al., 2020), specifically tailored for speech purposes. In BLIMP, sentences are divided into twelve broad categories focusing on different linguistic paradigms in the fields of syntax, morphology, or semantics. These categories are themselves divided into 67 finer linguistic subcategories, containing 1000 sentence pairs each, automatically generated using expert hand-crafted grammar. To make the sBLIMP dataset 'speech-ready', 5 subcategories were discarded, and the grammar of 9 additional subcategories was slightly modified in order to avoid any difficulty in generating a prosodic contour for the ungrammatical sentences. Words not found in the LibriSpeech training set were excluded from the sBLIMP vocabulary. Compound words and homophones, which could cause further understanding issues during synthesis, were also removed. Finally, the generated texts are synthesized using a multi-speaker TTS system similar to sWUGGY.

Lexical Semantics: The sSIMI Similarity Metrics. The sSIMI metric assesses model comprehension of lexical semantics through its output representations. Here, the task is to compute the similarity of the representation of pairs of words (e.g. 'cat' – 'dog') and compare it to human similarity judgements. Formally, the sSIMI score is computed as the correlation of the similarity given by models compared with human similarity scores:

$$score_{ssimi} := \frac{1}{n_{\mathcal{D}}} \sum_{D_i \in \mathcal{D}} \rho \left(\{\hat{s}_M(w_1, w_2)\}_{(w_1,w_2)\in D_i} ; \{s_H(w_1, w_2)\}_{(w_1,w_2)\in D_i} \right) \quad (1.6)$$

where $\mathcal{D}$ contains all datasets D_i of word pairs (w_1, w_2); $n_{\mathcal{D}}$ is the number of datasets; $s_H(w_1, w_2)$ is the simlarity score given by human; $\hat{s}_M(w_1, w_2)$ is a similarity computed from output representations of w_1 and w_2; and ρ is the Spearman's rank correlation coefficient. Based on Chung and Glass (2018), a set of 13 existing semantic similarity and relatedness tests was used to construct the sSIMI benchmark. The similarity-based datasets include WordSim-353 (Yang and Powers, 2006), WordSim-353-SIM (Agirre et al., 2009), mc-30 (Miller and Charles, 1991), rg-65 Rubenstein and Goodenough, 1965, Rare-Word (or rw) (Luong et al., 2013), simLex999 (Hill et al., 2015), simverb-3500 (Gerz et al., 2016), verb-143 (Baker et al., 2014), YP-130 Yang and Powers, 2006 and the relatedness-based datasets include MEN (Bruni et al., 2012), Wordsim-353-REL (Agirre et al., 2009), mturk-287 (Radinsky et al., 2011), mturk-771 (Halawi et al., 2012). All scores were normalized on a 0-10 scale, and pairs within the same dataset containing the same words in a different order were averaged. Pairs containing a word absent from the LibriSpeech train set were discarded. Two versions of sSIMI were created, one synthetic and one natural. The synthetic subset was generated using a multi-speaker TTS system as similar to sWUGGY and sBLIMP. For the natural subset, audio segments corresponding to different words were obtained from the LibriSpeech dataset following the process presented in Chung and Glass (2018). The natural subset is comparatively smaller than the synthetic one due to the exclusion of pairs not present in the LibriSpeech test and dev sets. However, in this natural subset, each word can appear in multiple audio segments providing phonetic diversity; duplicated scores are averaged in the analysis step.

Previous metrics mainly focus on lower linguistic levels such as phonetic (ABX), lexical (sWUGGY, sSIMI), or single-sentence (sBLIMP) and therefore could hardly assess model capability to capture contextual semantics (at multi-sentence level). Later in Hassid et al. (2023), we introduced other zero-shot metrics used to evaluate the comprehension of speech language models at a broader level: Contextual Semantics and Commonsense Reasoning (T-StoryCloze and S-StoryCloze).

Commonsense Reasoning: The T-StoryCloze and S-StoryCloze Metrics. To better evaluate the capabilities of speech language models in capturing fine-grained textual nuances and continuation coherence, two spoken versions of the StoryCloze textual benchmark (Mostafazadeh et al., 2016), denoted by Spoken StoryCloze (S-StoryCloze) and Spoken Topic StoryClose (T-StoryCloze), were introduced. The StoryCloze dataset contains five-sentence commonsense stories, each story consists of a four-sentence 'context' and two alternative endings, called 'right ending' and 'wrong ending'. The goal of the model is to choose the right ending over the wrong ending given the context. To generate the spoken benchmarks, the stories from the textual dataset are generated using a single-speaker TTS system. The S-StoryCloze follows the original StoryCloze samples, and evaluates the models' capabilities to capture fine-grained causal and temporal commonsense relations. For the T-StoryCloze, or Spoken Topic StoryCloze, the wrong ending was sampled randomly from the dataset and could therefore be unrelated to the context. This version of StoryCloze aims to evaluate continuation coherence given a spoken prompt and is far easier. However, it has been shown that the T-StoryCloze task is still challenging for modern speech language models. Similar to sWUGGY and sBLIMP, the T-StoryCloze and S-StoryCloze metrics are computed as the percentage of examples where the model assigns a higher probability to the correct sample than the incorrect one.

1.4.2.2 Speech Resynthesis and Generation Metrics

Apart from the comprehension metrics, in Lakhotia et al. (2021) we proposed metrics used to assess the output of generative speech models in terms of resynthesis intelligibility, quality and diversity of the newly generated speech.

Speech Resynthesis Intelligibility: ASR-PER. Unlike text tokenizers in TextLMs, where the output texts are identical to the input texts (as text tokenizers are inversible), speech information can be loss during the tokenization-detokenization process (e.g. speech $\rightarrow$ units $\rightarrow$ resynthesized speech). The Speech Resynthesis Intelligibility metrics therefore aim to evaluate if the resynthesized output speech has the same content as the input speech. The ideal metric for intelligibility would be to use humans to transcribe the resynthesized speech and compare the text to the original input. However, an automatic proxy can be obtained by using a state-of-the-art ASR system pretrained on a large corpus of real speech. The resynthesized speech is thus transcribed to text with the ASR system, and then is compared with the original text of the input speech using any ASR metrics such as Phoneme Error Rate (PER), Character Error Rate (CER) or Word Error Rate (WER).

Speech Generation Quality and Diversity: AUC over Perplexity and VERT. In this scenario, the model is used to generate new speech using either a speech prompt (conditional generation) or no input (unconditional generation), and the generated speech is evaluated in terms of *meaningfulness*. Similar to previous metrics, the generated speech is transcribed to text using an ASR system and text-based generation metrics are then employed. Text generation evaluation typically involves two axes: the quality and diversity of the generated text (Hashimoto et al., 2019). Text quality can be automatically evaluated by computing the perplexity, or probability, of text using a reference language model trained on natural texts. A lower perplexity assumes a more probable generated sentence. However, a flaw of Natual Language Model-based perplexity is that a sentence contain of repeated words will likely have very low perplexity but is implausible in real life. The diversity metrics somewhat remedy this problem by computing how the generated texts are diverse in terms of vocabulary. In Lakhotia et al. (2021), we introduced VERT (diVERsiTy), a metrics used to compute the diversity of generated texts by calculating the geometric mean of self-BLEU and auto-BLEU. Self-BLEU (Zhu et al., 2018) evaluates how similar one sentence is compared to other generated sentences, while auto-BLEU measures within-sentence diversity. Low self-BLEU and auto-BLEU scores indicate higher diversity of the produced sentences. Typically, there is a trade-off between Perplexity and VERT based on the temperature hyperparameter used for sampling from the language model, whereby at low temperature, the system outputs good sentences (low Perplexity) but not varied (high VERT), and at high temperatures, it outputs varied sentences (low VERT), but not very good (high Perplexity). This results in model comparison being either based on 2D plots with lines representing the trade-off between quality and diversity, or estimation of how close these plots are to oracle (ground-truth text) Perplexity and VERT by computing the area under the curve (AUC).

Human Evaluation Metrics: MOS. Despite the practical of previously mentioned automatic metrics, they are still dependent on *off-the-shelf* ASR and scoring LM systems, and might not reflect the true quality of the generations as listened by humans. Human evaluations therefore still play an important role in the assessments of speech generation. Mean Opinion Scores (MOS) is a commonly used method for human evaluation, and is widely used to evaluate the quality of audiovisual data or systems (Streijl et al., 2016). In MOS, the humans are asked to give their opinions, generally scores from 1-5 with 1 being *Bad* and 5 being *Excellent*, over an audio file. The scores are then averaged over all humans and all files to obtain a mean opinion score. Depending on how the humans are instructed, the MOS score can be

used to evaluate one particular aspect of the speech such as *intelligibility (MOS)* or *meaningfulness (MMOS)*.

1.4.3 Performances of Spoken Language Models

The overall performances of the described spoken language models systems over Zero-shot Comprehension Metrics and Speech Generation Metrics are shown in Table 1.2 and Figure 1.2 respectively. We will discuss here some key insights obtained from the results.

Tab. 1.2: Spoken Zero-shot Comprehension Metrics Performances. Scores are taken from Nguyen et al. (2020b) and Lakhotia et al. (2021). The speech features are evaluated with the *ABX within* and *ABX across* metrics, while spoken language modeling performances are evaluate with *sWUGGY*, *sBLIMP*, and *sSIMI* metrics. The systems are described in section 1.4.1 and the metrics are described in section 1.4.2. ∅ denotes unobtainable scores, while – denotes scores not reported. The best scores for each speech and text systems are **bold**.

			ABX within ↓		ABX across ↓		sWUGGY ↑	sBLIMP ↑	sSIMI ↑	
Feature	nb units	LM	clean	other	clean	other	(invocab)		synth.	libri.
ZeroSpeech 2021 Systems (Nguyen et al., 2020b)										
Low-budget Baseline Systems										
CPC	50	BERT-small	6.38	10.22	8.26	14.86	65.81	52.91	3.88	5.56
	50	LSTM	6.38	10.22	8.26	14.86	66.13	53.32	4.42	7.56
High-budget Baseline Systems										
CPC-small	50	BERT	10.26	14.24	14.17	21.26	70.69	54.26	2.99	6.68
CPC	50	BERT	**6.38**	**10.22**	**8.26**	**14.86**	**75.56**	**56.14**	**6.25**	**8.72**
Text Topline Systems										
Aligned-phone	40	BERT	0.00	0.00	0.00	0.00	92.19	63.72	7.92	4.54
Phone	39	BERT	∅	∅	∅	∅	97.90	66.78	9.86	16.11
Sub-word	50K	RoBERTa large	∅	∅	∅	∅	**96.58**	**81.56**	**32.28**	**28.96**
GSLM Systems (Lakhotia et al., 2021)										
Baseline MFCC Features										
LogMel	50	GPT	23.95	–	35.86	–	51.48	53.22	–	–
	100	GPT	24.33	–	37.86	–	51.88	53.17	–	–
	200	GPT	25.71	–	39.65	–	50.38	52.24	–	–
Self-supervised Speech Features										
CPC	50	GPT	5.50	–	7.20	–	67.82	54.57	–	–
	100	GPT	**5.09**	–	**6.55**	–	68.28	55.65	–	–
	200	GPT	5.18	–	6.83	–	62.60	54.81	–	–
HuBERT-L6	50	GPT	7.37	–	8.61	–	67.12	55.94	–	–
	100	GPT	6.00	–	7.41	–	**68.70**	**57.06**	–	–
	200	GPT	5.99	–	7.31	–	63.48	52.97	–	–
wav2vec2-L14	50	GPT	22.30	–	24.56	–	48.08	54.25	–	–
	100	GPT	18.16	–	20.44	–	49.76	54.03	–	–
	200	GPT	16.59	–	18.69	–	55.32	54.30	–	–
Topline Text System										
ASR-text	–	GPT	∅	∅	∅	∅	**96.88**	**70.98**	–	–

1.4.3.1 Spoken Language Modeling is Feasible

For the zero-shot comprehension metrics (Table 1.2), we see that the best-performing speech models show better-than-chance performances in all tasks, although there is substantial variation between tasks. We see excellent performances at the acoustic-phonetic level (6-8% error rate for the best model on ABX-across), good performances at the lexical level (around 70% for best models), and poor results at larger linguistic levels (syntactic and lexical semantics). Overall, this still shows that the discrete units could capture acoustic information from speech, and training a language model on them can leverage language comprehension from raw speech.

Looking at Figure 1.2, we can see that the best GSLM systems perform well at the resynthesis task (speech $\rightarrow$ units $\rightarrow$ synthesized speech, with around 10% PER). On the speech generation tasks, we see that they obtain very good human MMOS scores (averaged on both prompted and unprompted generations), which correlate well with the automatic AUC metrics. This shows that it is possible to generate new speech without any text supervision using discrete units.

These results show the feasibility of both spoken language modeling and generative spoken language modeling tasks.

1.4.3.2 Quality of Speech Features is Important

We observe that speech features have critical impacts on the performances of spoken language modeling systems. LogMel features, which are not supposed to capture strong linguistic information of speech, have high ABX errors (around 20-40%), especially compared to CPC and hidden HuBERT features (under 10%). This is further reflected in spoken language modeling metrics like sWUGGY and sBLIMP, where features with very good ABX scores (CPC and HuBERT) tend to have better performances. This is also true for Speech Resynthesis and Generation Metrics.

The number of clusters also influences the performances of SpeechLMs. Interestingly, increasing the number of clusters helps improve Speech Resynthesis tasks (PER, MOS), but not for other comprehension tasks (ABX, sWUGGY, sBLIMP). There seems to be a sweet spot for CPC and HuBERT features at 100 units.

It's worth noting that autoregressive LMs (GPT, LSTM) are not as performant as encoder-based LMs like BERT, which is commonly observed in classic text LMs.

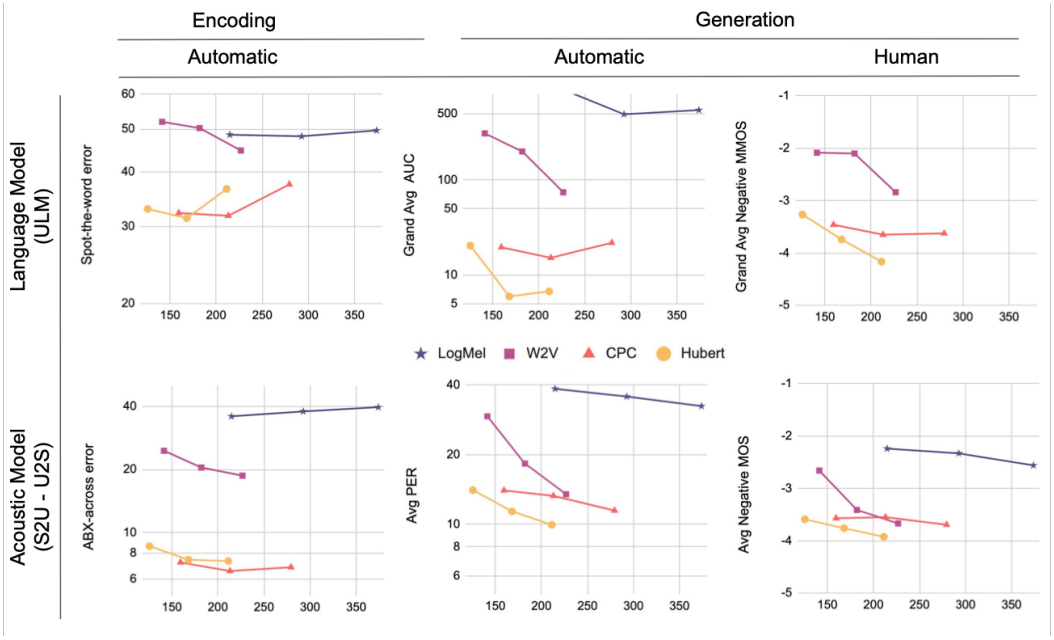


Fig. 1.2: Overall results of GSLM systems on comprehension and generation metrics (from Lakhotia et al., 2021). The results are presented with 4 speech encoders (LogMel, CPC, HuBERT and wav2vec 2.0) varying in number of k-means units (50, 100, 200). The metrics are described in section 1.4.2. Negative human opinion scores are shown for ease of comparison with automatic metrics (lower is better). The generation metrics have been averaged across LS and LJ (PER and MOS; resynthesis task) and across prompted and unprompted generations (AUC and MMOS; speech generation task). The LogMel-based systems were not evaluated by humans in the speech generation task.

1.4.3.3 There is a gap between SpeechLMs and TextLMs

In both encoder-based and decoder-based spoken language modeling systems, we see a clear gap in the performances of Text-based language models compared with their speech counterparts. This could be caused by many factors such as: i) the linguistic (phonetic) quality of the speech units; ii) the granularity of the speech units compared to text; and iii) the quantity of speech data compared to the vast amount of text data. To better disentangle these factors, we trained language models on two text levels: forced-aligned phonemes (which could be seen as perfect speech units with 0 ABX error), and normal phonemes (generally deduplicated phonemes, without SIL tokens) and compared them with a pre-trained text LM on large-scale dataset (RoBERTa). We see that each factor effectively contributes to the discrepancy between SpeechLMs and TextLMs. Going from speech to aligned-phone boosts a lot on lexical and syntactic tasks, but not on lexical semantics task. The sSIMI task is improved by using phonemes instead of aligned phones, and using large-scale data with sub-word tokens yields the best syntactic and lexical semantics scores.

There is also another factor of speech that can hinder their performance compared with text which is the absence of lexical information (e.g. word boundary). In Nguyen et al. (2020a), we analyzed this factor by comparing various LMs on character and phoneme levels with and without boundary information. We found that models without word boundaries underperform models that have boundaries which can rely on higher-order units like words or BPEs, and that part of this decrement can be compensated by using automatically generated word boundaries using unsupervised word segmentation.

The results show that there is still a lot of room for improvement for SpeechLMs, and more work could be done to bridge the gap between SpeechLMs and TextLMs.

1.4.3.4 Lacking Expressivity in Speech Generation

One huge advantage of SpeechLMs compared to cascaded systems is the possibility of learning and generating paralinguistic information (e.g. pitch, expressivity) in the speech. However, by listening to the speech examples generated by GSLM,[14] we can observe that although the generated speech contains good linguistic content, it still lacks natural intonation and expression.

This naturally comes from the fact that the models are trained on read speech datasets, which limits their ability to generate expressive speech. Having more expressive datasets could possibly improve the generation quality of SpeechLMs. In addition, the use of discrete speech units obtained from self-supervised speech features also contributes to this lack of expressivity. We have been optimizing the speech units using linguistically-based metrics (ABX, PNMI), which could inadvertently strip away paralinguistic cues from the speech. Using other speech features containing acoustic information could, therefore, provide SpeechLMs with expressive information and thus generate more natural speech.

1.4.4 Discussions

1.4.4.1 Applications of Spoken Language Models

TextlessNLP The previous works opened up the possibility of applying language models directly to audio inputs, sidestepping the need for textual resources or Automatic Speech Recognition (ASR), which leads to a new research domain called

[14] `https://speechbot.github.io/gslm/index.html`

TextlessNLP[15]. Polyak et al. (2021) further analyzed the speech resynthesis task by extracting disentangled discrete representations for speech content, prosodic information, speaker identity and managed to synthesize speech from discrete units in a controllable manner. Following this, Kharitonov et al. (2022b) introduced pGSLM, a multi-stream SpeechLM which jointly learns "pseudo-text" tokens together with quantized prosodic features (i.e. duration and F0). Kreuk et al. (2022) performed the speech emotion conversion task using discrete speech representations. Nguyen et al. (2023b) extended GSLM to multi-channel speech and performed a spoken dialogue language modeling task (dGSLM). Leveraging general audio neural codecs (SoundStream, Zeghidour et al., 2021), Borsos et al. (2023) proposed AudioLM, a speech language modeling system that can generate coherence speech with the "pseudo-text" speech units (called *semantic tokens*) while preserving paralinguistic information (e.g. speaker identity, prosody) with the audio codecs (called *acoustic tokens*).

Speech Translation Lee et al. (2022a) proposed using discrete speech units as target to perform direct speech-to-speech translation with an encoder-decoder architecture. Following this, Popuri et al. (2022) proposed pre-training a BART (Lewis et al., 2020a) model on the speech units and fine-tune on the speech translation task. Following AudioLM, AudioPaLM (Rubenstein et al., 2023) utilizes speech tokens to perform Speech Recognition and Translation tasks while keeping input speaker identity. Recently, Seamless (Communication et al., 2023a,b) proposed expressive, real-time Speech Translation systems that work for hundreds of languages.

Multimodal Speech Systems There is an intrinsic relation between text and speech, and therefore it is natural to bind SpeechLMs with TextLMs. Hassid et al. (2023) found that it is beneficial to continue training speech units on pre-trained TextLMs. Zhang et al. (2023a) proposed a spoken question-answering model, SpeechGPT, that utilizes speech units as a proxy to go from speech to text. Lastly, Nguyen et al. (2024) introduced Spirit-LM, a combined speech and text generative language model that can generate content cross-modally and allows speech-text few-shot in-context learning. In the audiovisual domain, Hsu et al. (2023b) proposed ReVISE which combines self-supervised audiovisual model (Shi et al., 2022) with speech units for the audiovisual generation task.

[15] `https://speechbot.github.io/`

Downstream Speech Tasks SpeechPrompt and SpeechPrompt-v2 (Chang et al., 2022, 2023b) explored the performance of GSLM and pGSLM systems on a variety of downstream speech classification tasks from Superb (Yang et al., 2021) using prompt tuning technique inspired from prefix-tuning (Li and Liang, 2021). Hsu et al. (2023a) found that GSLM systems do not have the in-context learning (ICL) capability, but are able to perform ICL on unseen tasks after warm-up training. SpeechGen (Wu et al., 2023) extended this prompting technique to Speech Translation, Speech Inpainting and Speech Genration tasks using the unit-mBART model in Popuri et al. (2022).

1.4.4.2 Concurrent Related Work

Audio Generation with Discrete Representations Oord et al. (2017a) applied their proposed VQ-VAE model to obtain discrete latent representations of audio. They show that the discrete representations can capture the speech content and can be used to reconstruct audio or to generate new audio using the WaveNet Decoder (van den Oord et al., 2016). Jukebox (Dhariwal et al., 2020) employed multi-scale VQ-VAE to compress audio to multi-level discrete codes and trained language models on the codes to perform music generation conditioned on artists, genres, and optionally lyrics. With the introduction of high-quality neural audio codecs (SoundStream, Zeghidour et al., 2021, Encodec, Défossez et al., 2022), more audio generation systems are introduced. VallE (Wang et al., 2023a) and SpearTTS (Kharitonov et al., 2023) performed the voice-conditioning (voice-cloning) TTS task by translating text to audio codec tokens. In the same spirit, Kreuk et al. (2023) proposed AudioGen, which employs a Transformer decoder to model text and Encodec units, permitting the generation of audio samples from text captions. Following AudioLM, Agostinelli et al. (2023) introduced MusicLM, a system that can generate music from text descriptions using a hierarchical sequence-to-sequence modeling task. MusicGen (Copet et al., 2023) allows the generation of high-quality music from both text descriptions and melody using a single language model operating on multiple streams of discrete tokens.

2 Speech Tokenization Revisited

We have seen in the first chapter that *discrete units* play an essential role in SpeechLMs systems by transforming the continuous speech representations into a discrete space, which made it possible to perform spoken language modeling from raw audio. However, we didn't really provide an analysis as to why such speech tokenization, or discretization, works, and what is contained in these discrete speech units. In this chapter, we are going to deal with these questions on the importance of discretization on our provided spoken language modeling metrics.

This chapter presents the following paper, which was published in IEEE Journal of Selected Topics in Signal Processing:

Tu Anh Nguyen, Benoit Sagot, and Emmanuel Dupoux (2022a). "Are Discrete Units Necessary for Spoken Language Modeling?" In: *IEEE Journal of Selected Topics in Signal Processing* 16.6, pp. 1415–1423

It is followed by Section 2.6, where I present my additional experiments on trying to improve SpeechLMs by using larger speech units.

Statement of contribution:

I implemented all the models as well as performed the experiments mentioned in this chapter, with the ideas and suggestions obtained from discussions with my supervisors as well as feedback from reviewers.

Publication: Are discrete units necessary for Spoken Language Modeling?

Tu Anh Nguyen[◇,†], **Benoît Sagot**[†], **Emmanuel Dupoux**[◇,‡]

[◇]Meta AI Research, [†]Inria, Paris, [‡]EHESS, ENS-PSL, CNRS, Paris

{nguyentuanh208, emmanuel.dupoux}@gmail.com, benoit.sagot@inria.fr

Abstract

Recent work in spoken language modeling shows the possibility of learning a language unsupervisedly from raw audio without any text labels. The approach relies first on transforming the audio into a sequence of discrete units (or pseudo-text) and then training a language model directly on such pseudo-text. Is such a discrete bottleneck necessary, potentially introducing irreversible errors in the encoding of the speech signal, or could we learn a language model without discrete units at all? In this work, we study the role of discrete versus continuous representations in spoken language modeling. We show that discretization is indeed essential for good results in spoken language modeling. We show that discretization removes linguistically irrelevant information from the continuous features, helping to improve language modeling performances. On the basis of this study, we train a language model on the discrete units of the HuBERT features, reaching new state-of-the-art results in the lexical, syntactic and semantic metrics of the Zero Resource Speech Challenge 2021 (Track 1 - Speech Only).

2.1 Introduction

Pre-training language models on large-scale text data have achieved tremendous success in natural language understanding and have become a standard in Natural Language Processing (NLP) (Brown et al., 2020; Devlin et al., 2019; Liu et al., 2019; Radford et al., 2018, 2019). Recently, Brown et al. (2020) showed that very large language models are actually few-shot learners, and manage to perform well even in zero-shot settings.

Large-scale self-supervised pre-training for speech data has also become more and more popular as a method to boost the performance of Automatic Speech Recognition (ASR) (Baevski et al., 2020c; Chung et al., 2021; Hsu et al., 2021a). However, these models mostly rely on fine-tuning, which requires more training and text labels, to either improve the model or evaluate the learned representations of the speech. Lately, Nguyen et al. (2020b) introduces a new unsupervised task: Spoken language modeling, the learning of a language unsupervisedly from raw audio without any text labels, along with a suite of 4 zero-shot metrics probing for the quality of the learned models at different linguistic levels: phonetic, lexical, syntactic, semantic. The metrics are evaluated using the representations extracted from the model (phonetic, semantic) or pseudo-probability scores given by the model (lexical, syntactic). Their proposed baseline approach relies on transforming the audio into a sequence of frame-by-frame discrete units (or pseudo-text) and training a language model on the pseudo-text. The trained models displayed better-than-chance performances on nearly all the evaluation metrics of the challenge (Dunbar et al., 2021; Nguyen et al., 2020b). However, this paradigm creates a discrete bottleneck between a speech encoder and a language model which could be a potential source of error, and in addition requires multiple training phases (learning an acoustic representation, clustering it, and learning a language model). Is such a discrete bottleneck necessary?

One way in which discrete units could help language modeling stems from the fact that in contrast to text, audio data contains a lot more details, some of which are linguistically relevant (intonation, rhythm, non verbal vocalization), others not so (background noise, reverberation, speaker identity, etc). To the extent that discretization effectively removes linguistically irrelevant information from the continuous features (Niekerk et al., 2021), it could indeed help language modeling. Of course, this potential gain could be counterbalanced by the fact that discretization could also make errors and remove useful information.

In this work, we analyse the importance of discretization in spoken language modeling. We employ a pre-trained acoustic model to obtain either continuous or discretized features from audio data. We then train BERT language models with a Masked Language Modeling (MLM) objective on both discrete and continuous features used either as inputs or as targets and evaluate the resulting systems on zero-shot spoken language modeling metrics. We also evaluate HuBERT (Hsu et al., 2021a), a single model trained from raw waveform with discrete targets, on these metrics and compare the results with our best models.

Our contributions can be listed as follows:

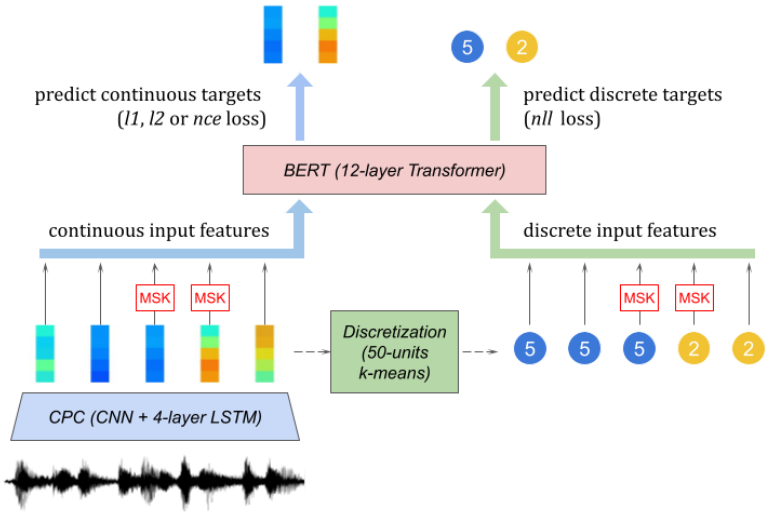


Fig. 2.1: Overview of the trained BERT models. The BERT model takes as input either the continuous features extracted from CPC or the sequences of frame-by-frame discretized units obtained from k-means, and tries to predict either continuous target features (with L1, L2 or NCE loss) or discrete target units (with NLL loss).

- We show experimentally that discretization is beneficial for spoken language modeling, but we can get rid of discrete bottlenecks by using low-level continuous inputs so long as we still use discrete targets.
- We show that discretization disentangles linguistic information from non-linguistic signals, forcing the transformer to focus on linguistic ones.
- We show that a self-supervised model trained with a MLM objective on discrete targets like HuBERT achieves very good results on spoken language modeling metrics, showing that it can learn not only acoustic but also high-level linguistic information.

2.2 Related Work

Discretization in Self-Supervised Approaches Self-supervised models for learning speech representation have become more and more popular as an effective pre-training method for downstream Automatic Speech Recognition (ASR) task, notably wav2vec2.0 (Baevski et al., 2020c) and HuBERT (Hsu et al., 2021a). Both models comprise a feature extractor (CNN Encoder) followed by a feature encoder (Transformer Encoder), and are trained with a MLM objective like BERT. However, wav2vec2.0 discretizes the latent features obtained by the CNN Encoder and uses

them as the target for the Transformer Encoder using a contrastive loss against negative samples in the sentence. On the other hand, HuBERT discretizes fixed features obtained from a teacher model and uses these fixed discrete units as the target for the Transformer Encoder using a cross-entropy loss. Finally, our work is mostly similar to Baevski et al. (2020a), where they compare BERT models training on discrete units obtained from vq-wav2vec (Baevski et al., 2020b) and continuous features obtained from wav2vec (Schneider et al., 2019) on the ASR task. They found that training BERT model on discrete vq-wav2vec units is more effective for ASR.

Spoken Language Modeling Following the huge success of language models on text data (Brown et al., 2020; Devlin et al., 2019; Radford et al., 2019), the Zero Resource Speech Challenge 2021 (Dunbar et al., 2021; Nguyen et al., 2020b) opens up new possibilities for learning high-level language properties from raw audio without any text labels. They introduced 4 zero-shot evaluation metrics at different linguistic levels (phonetic, lexical, syntactic, semantic), along with composite baseline systems consisting of an acoustic discretization module (Contrastive Predictive Coding, or CPC+k-means) followed by a language model (BERT or LSTM) on the discretized units. The CPC model takes the raw audio as input and produces phonetic representations at a lower frame rate of 100Hz, helping the language model to learn high-level information from the raw audio. In the same spirit, Lakhotia et al. (2021) introduced Generative Spoken Language Modeling (GSLM), the task of learning and generating spoken language from raw audio only. They provided baseline systems consisting of a discrete speech encoder (CPC, wav2vec 2.0, HuBERT), a generative language model (GPT-like model), and a speech decoder (Tacotron-2, Shen et al., 2018). The models are evaluated on spoken language modeling metrics (Nguyen et al., 2020b), ASR-based generation metrics (Lakhotia et al., 2021) as well as human evaluation metrics.

2.3 Experimental Setup

In this section, we first present the evaluation metrics as well as the dataset used to train and evaluate the models. We then explain our models and the inference methods for model evaluation.

2.3.1 Evaluation Metrics

We evaluate our models with the ZeroSpeech 2021 Benchmark Metrics (Nguyen et al., 2020b), consisting of 4 zero-shot tests probing for the quality of spoken language models at four linguistic levels: phonetic (Libri-light ABX metrics), lexical (sWUGGY spot-the-word metrics), syntactic (sBLIMP acceptability metrics) and semantic (sSIMI similarity metrics).

Libri-light ABX metrics Given a pair of similar triphones (e.g., 'aba'-'apa') spoken by a same speaker and an intervening sound (either 'aba' or 'apa'), the model has to tell which sound has a closer representation to the intervening sound. The ABX metrics is reported as the error rate that the model fails to choose the correct triphone.

sWUGGY spot-the-word metrics Given a pair of a word and a similar non-word (e.g., 'brick'-'blick'), the model has to tell which is the word based on their probability. The spot-the-word metrics is reported as the accuracy that the model assigns a higher probability to the word.

sBLIMP acceptability metrics Given a linguistic minimal sentence pair of matched grammatical and ungrammatical sentences (e.g., 'he loves it'-'he love it'), the model has to tell which is the grammatical sentence. The acceptability metrics is reported as the accuracy that the model assigns a higher probability to the grammatical sentence.

sSIMI similarity metrics Given a pair of words (e.g., 'happy'-'joyful'), the model has to compute a similarity score based on their representations. The similarity metrics is reported as the Pearson correlation coefficient (PCC) between model scores and human judgements. In this work, the sSIMI scores are weighted across different subsets according to their sizes and averaged across LibriSpeech and synthetic subsets to make it more accurate and consistent. We reported it as wSIMI.

2.3.2 Datasets

Training Dataset We train our models on LibriSpeech (Panayotov et al., 2015), an English corpus containing 1000 hours of read speech based on public domain audio

books. The models are validated on LibriSpeech dev-clean and dev-other subsets, comprising 10 hours of speech in total.

Metrics Datasets The metrics datasets are either extracted sounds from LibriSpeech (ABX, sSIMI) or synthesised using Google API[1] (sWUGGY, sBLIMP, sSIMI). The datasets containing words or sentences were filtered to only contain the LibriSpeech vocabulary[2] (except sWUGGY non-words), and are split into dev and test sets. The dev sets have been made publicly available at the ZeroSpeech 2021 Challenge website [3].

2.3.3 Models

ZeroSpeech 2021 Baseline The ZeroSpeech 2021 Baseline System (Nguyen et al., 2020b) is a composite of three components: an acoustic model (CPC, Oord et al., 2018; Rivière et al., 2020), a clustering module (k-means) and a language model (BERT, Devlin et al., 2019). The CPC model is first trained to obtain good phonetic representations of the speech, which are then discretized into sequences of units with the k-means model. The BERT model is finally trained on these discrete units to better learn linguistic information.

As we only focus on the language modeling system in this work, we shall use the best CPC model in the ZeroSpeech 2021 Baseline System, which comprises a 5-layer 1D-CNN Encoder followed by a 4-layer LSTM autoregressive model. The features are extracted from the 2nd layer (unless otherwise specified) of the LSTM model, with a rate of 100Hz, and are either discretized with a 50-unit k-means model (discrete) or left unchanged (continuous).

BERT with discrete and continuous features We modify the BERT model so that it is able to take as input either discrete units obtained from k-means or continuous features extracted from CPC, in which case the masking is done by replacing the features with a masked embedding vector. We also allow the model to predict either discrete target units or continuous target features, with multiple choices of an appropriate objective for each case. When predicting discrete targets, we use a cross-entropy objective (Negative Log-Likelihood, or NLL loss) but with two

[1] `https://cloud.google.com/text-to-speech`
[2] In this work, we only evaluated the sWUGGY metrics on the in-vocab subset, which contain the words from the LibriSpeech vocabulary.
[3] `https://zerospeech.com/tasks/task_4/benchmarks_datasets/`

	Training hyperparameters						Inference hyperparameters					
	n units (if discrete)		feat.	masking		num	prob.	est.	SIMI librispeech		SIMI synthetic	
id	input	target	stride	length	prob.	updates	M	Δt	layer	pooling	layer	pooling
	BERT Models on CPC-big Features											
1	50	50	10ms	10	0.5	250k	15	5	11	max	1	min
2	50	50	10ms	10	0.5	250k	15	5	5	min	7	mean
11	50	50	10ms	10	0.5	250k	15	5	11	mean	10	min
12	50	50	10ms	10	0.5	250k	15	5	9	mean	1	min
3	-	50	10ms	10	0.5	250k	35	5	6	min	4	max
4	-	50	10ms	10	0.5	250k	35	5	8	mean	11	min
13	-	50	10ms	10	0.5	250k	25	5	10	max	10	min
14	-	50	10ms	10	0.5	250k	35	5	4	max	6	mean
5	-	-	10ms	10	0.5	250k	45	5	8	max	1	mean
15	-	-	10ms	10	0.5	250k	35	5	11	max	11	mean
16	-	-	10ms	10	0.5	250k	45	5	12	mean	12	mean
6	-	-	10ms	10	0.5	250k	35	5	1	mean	12	mean
17	-	-	10ms	10	0.5	250k	25	5	12	min	12	max
18	-	-	10ms	10	0.5	250k	35	5	1	mean	12	mean
7	-	-	10ms	10	0.5	250k	35	5	3	mean	1	mean
8	50	-	10ms	10	0.5	250k	25	5	12	mean	5	min
19	50	-	10ms	10	0.5	250k	25	5	8	mean	4	min
20	50	-	10ms	10	0.5	250k	25	5	8	mean	11	mean
9	50	-	10ms	10	0.5	250k	15	5	12	max	12	min
21	50	-	10ms	10	0.5	250k	15	5	11	mean	7	min
22	50	-	10ms	10	0.5	250k	15	5	4	mean	12	max
10	50	-	10ms	10	0.5	250k	15	5	10	max	12	max
	HuBERT Base Models											
23	-	100	20ms	10	0.65	250k	15	5	2	max	5	min
24	-	500	20ms	10	0.65	400k	15	5	1	mean	9	max
25	-	500	20ms	10	0.65	400k	15	5	10	mean	6	max
	BERT Models on HuBERT Discrete Units											
26	-	500	20ms	10	0.5	250k	10	5	7	mean	8	mean

Tab. 2.1: Training and Inference hyperparameters of models trained in the paper. The id corresponds to the model index in other tables.

slightly different implementations. We could simply employ a linear classification head at the output of the BERT model as usual (which we denote by *linear NLL*, or NLL-l) or force the BERT output features to be similar to the embedding vectors of the target units as for HuBERT (cf. equation (3) from Hsu et al., 2021a, we denote this by *embedding NLL*, or NLL-e). In the case of continuous targets, it can be a reconstruction objective (L1 loss or L2 loss) or a contrastive objective (Noise Contrastive Estimation, or NCE loss). In the latter case, the predicted features are contrasted with 100 negative features sampled from the same phrase (similar to continuousBERT, Baevski et al., 2020a).

We use a BERT base model, which comprises a 12-layer Transformer Encoder. Our implementation is based on the wav2vec2.0 (Baevski et al., 2020c) Transformer Encoder [4] using fairseq (Ott et al., 2019). Each input sequence contains the features of a full audio file, and we consider at most 15.6 seconds of audio per file. We

[4] `https://github.com/pytorch/fairseq/tree/main/examples/wav2vec`

trained all models for 250k update steps on 32 GPUs, with a batch size of 175s per GPU. The learning rate was warmed up to a peak value of 1×10^{-5} after 32k steps. For the masking, we masked M consecutive tokens for each span, where $M \sim \mathcal{N}(10, 10)$, with a total masking coverage of roughly half of the input tokens (spans may overlap).

2.3.4 Model Inference for Evaluation

ABX Distance For the ABX metrics, we extract frame-by-frame representation features for each audio file. Then, the ABX distance between two files is computed as the average angular distance of the representations along the realigned Dynamic Time Wrapping path. Given two audio files x and y with two sequences of representation $\mathbf{r}^x = r_1^x, \ldots, r_T^x$ and $\mathbf{r}^y = r_1^y, \ldots, r_S^y$ respectively, the ABX distance between x and y is computed as follows:

$$d_{ABX}(x, y) = \frac{1}{|\text{path}_{\text{DTW}}(\mathbf{r}^x, \mathbf{r}^y)|} \sum_{(i,j) \in \text{path}_{\text{DTW}}(\mathbf{r}^x, \mathbf{r}^y)} sim(r_i^x, r_j^y), \tag{2.1}$$

where $sim(r_i^x, r_j^y)$ is the angular distance (in radian) between the embeddings r_i^x and r_j^y.

We note that in this paper the ABX metrics are mainly used to evaluate the input and target features of the BERT model, and therefore the ABX distances are mostly performed on the CPC features without using the BERT model.

Probability Estimation For sWUGGY and sBLIMP metrics, we compute for each audio file a model-based pseudo log-probability (m-PLP) of the trained BERT model. Given an audio file x with the input and target features for the BERT model $x_1 ... x_T$ and $\hat{x}_1 ... \hat{x}_T$ respectively, the m-PLP is computed as follows:

$$\text{m-PLP}(x) = \sum_{\substack{j=0 \\ i=j\Delta t}}^{\lfloor (T-M)/\Delta t \rfloor} \sum_{m=1}^{M} PLP(\hat{x}_{i+m} | \overline{x_{i+1} .. x_{i+M}}), \tag{2.2}$$

where M is a chosen size of a sliding window, Δt is a chosen step of the sliding window and $PLP(\hat{x}_{i+m} | \overline{x_{i+1} .. x_{i+M}})$ is a pseudo log-probability of the target $\hat{x}_{i+m}$ given by the BERT model with M-span masked inputs $x_1 .. x_i m .. m x_{i+M+1} .. x_T$ (m represents a masked feature).

For models with NLL or NCE loss, $PLP(\hat{x}_i|\overline{x_{i+1}..x_{i+M}})$ is computed as the log value of the probability given by the softmax layer of the BERT model (in the NLL case, the probability is computed over all tokens, while in the NCE case it is computed over all sampled negative examples). For models with L1 or L2 loss, we compute $PLP(\hat{x}_i|\overline{x_{i+1}..x_{i+M}})$ as the negative reconstruction loss of the predicted feature and the target feature $\hat{x}_i$. The negativity ensures that a correct target has a higher m-PLP.

The m-PLP extends the span-masked pseudo probability (span-PP) (Nguyen et al., 2020b) to BERT models with continuous targets. It is derived from the pseudo-loglikelihood score (PLL) for MLMs (Wang and Cho, 2019), which was shown to be an effective sentence scoring method for BERT models in many scenarios (Salazar et al., 2020).

The choice of M and Δt is determined for each model using the dev sets, and is given in Table 2.1. In our experiments, we always consider $\Delta t = 5$ and vary M in $\{15, 25, 35, 45, 55\}$. For models trained on HuBERT features (section 2.4.3), we vary M in $\{5, 10, 15, 20, 25\}$ as the frame rate is 50Hz instead of 100Hz as for CPC.

Similarity Score For the sSIMI metrics, we extract a fixed-length representation for each audio file by applying a pooling function (mean, max, min) over hidden features from one layer of the Transformer Encoder. The similarity score of two audio files is computed as the cosine similarity between the two corresponding representations. The choice of the hidden layer and the pooling function is determined for each model using the dev sets and is given in Table 2.1.

2.4 Results

2.4.1 Discrete bottleneck seems to be essential for spoken language modeling

Table 2.2 reports the performances of our BERT models, trained with either continuous or discrete CPC features of the LibriSpeech 960h dataset, on lexical (sWUGGY), syntactic (sBLIMP) and semantic (wSIMI) metrics.

We first examine how the continuity of the input and target features affects the quality of the BERT model on the evaluation metrics. By comparing the best scores in each case, we see that having discrete inputs helps the model learn better lexical

id	input	target	loss	sWUGGY↑ (invocab)	sBLIMP↑	wSIMI↑
	discrete input, discrete target					
1	disc.	disc.	NLL-l	79.28	59.71	6.32
2	disc.	disc.	NLL-e	**80.02**	**59.86**	**7.87**
	continuous input, discrete target					
3	*cont.*	disc.	NLL-l	**60.36**	**53.23**	8.39
4	*cont.*	disc.	NLL-e	60.20	52.78	**9.49**
	continuous input, continuous target					
5	*cont.*	*cont.*	NCE	56.84	52.62	**9.16**
6	*cont.*	*cont.*	L1	59.23	53.12	7.85
7	*cont.*	*cont.*	L2	**60.56**	**53.33**	6.55
	discrete input, continuous target					
8	disc.	*cont.*	NCE	65.69	**57.24**	9.33
9	disc.	*cont.*	L1	73.93	56.02	**10.69**
10	disc.	*cont.*	L2	**74.22**	55.75	5.97

Tab. 2.2: Discrete vs Continous Performances. Performances on the dev sets of sWUGGY, sBLIMP, wSIMI metrics of BERT models using either continuous ZeroSpeech CPC features (layer 2 of the LSTM module of CPC-big) or discretized features (with a 50-unit k-means model) as inputs and targets. Best scores in each category are in bold, best scores overall are underlined.

and syntactic information, whereas models with continuous inputs do have better than chance performance on the lexical task. We observe that the best models on the language model tasks are obtained with discrete inputs and discrete targets, which is the classic configuration of BERT. Predicting continuous targets from discrete inputs, where the model acts as an autoencoder decoder, is also beneficial and nearly catches up with the best models. It is interesting, still, to note that it is possible to acquire some language information without any discretization. The wSIMI scores are still quite low, but we see in general that having continuous information does help.

2.4.2 Is continuous input always bad?

We observe during our training experiments that the masked prediction objective is too easy for some models with continuous inputs and could quickly lead to overfitting. This could be explained by the fact that the input and target features are extracted from the same layer of the LSTM autoregressive module of CPC. As a consequence, we try using the input features from different layers of the LSTM module, while maintaining the same target layer. We keep using the NLL-e loss for

		ABX within↓		ABX across↓	
		clean	other	clean	other
layer 0	*cont.*	11.50	14.09	18.53	24.70
	disc.	21.46	24.21	30.77	34.91
layer 2	*cont.*	3.41	4.84	4.20	7.65
	disc.	6.38	10.22	8.22	14.86
layer 4	*cont.*	9.49	11.95	10.01	15.70
	disc.	19.81	21.64	24.39	28.04

Tab. 2.3: Layer-wise Analysis on Feature Quality. Within and Across Speaker ABX error (lower is better) on Libri-light dev-clean and -other for continuous and discretized features of different layers of the LSTM autoregressive module of CPC-big model. Layer 0 means the output of the CNN Encoder module.

discrete targets while using NCE and L1 loss for continuous targets. The results are reported in Table 2.4.

We observe that using continuous input features from a different layer does reduce overfitting during training, which significantly improves the performances of the models on LM metrics, especially for sWUGGY scores. Interestingly, we note that using continuous input features from a lower LSTM layer (layer 0, where the ABX errors are high, cf. Table 2.3) to predict target features from a higher LSTM layer (layer 2) is more beneficial to the model than using high quality continuous input features from the same or higher layer as the target features (layer 2, layer 4). This is not the case, however, for discrete input models, where the model benefits from good quality input units.

More analysis on the ABX errors of the trained BERT models' hidden features is reported in Table 2.5. We see that well-trained models with good language modeling scores (models 1,2,13,15,17,8,9,10) seem to have very good ABX errors compared to the others. By looking at the best hidden features of each model, we see that models with discrete targets (models 1,2,13) are able to reconstruct hidden features which are better than the targets, while this is not the case for most models with continuous targets (except model 10).

Overall, we observe that discrete-discrete model (with the same input and target units) yields the best performance when the quality of discrete units is good. Continuous-discrete is also a great choice when using low-level input features. When there are no discrete units at all, the LM performances are still limited, even if using a NCE loss could help a bit with syntactic and semantic metrics.

id	input	target	sWUGGY↑ (invocab)	sBLIMP↑	wSIMI↑
	discrete input, discrete target, NLL-e loss				
2	layer 2	layer 2	**<u>80.02</u>**	**<u>59.86</u>**	7.87
11	layer 0	layer 2	64.91	52.45	7.48
12	layer 4	layer 2	70.68	55.06	**8.61**
	continuous input, discrete target, NLL-e loss				
4	*layer 2*	layer 2	60.20	52.78	**9.49**
13	*layer 0*	layer 2	**77.19**	**55.30**	7.25
14	*layer 4*	layer 2	67.41	54.13	8.06
	continuous input, continuous target, NCE loss				
5	*layer 2*	*layer 2*	56.84	52.62	**9.16**
15	*layer 0*	*layer 2*	**65.53**	**55.20**	6.17
16	*layer 4*	*layer 2*	59.81	52.93	8.32
	continuous input, continuous target, L1 loss				
6	*layer 2*	*layer 2*	59.23	53.12	**7.85**
17	*layer 0*	*layer 2*	**67.83**	**53.59**	7.25
18	*layer 4*	*layer 2*	63.68	53.26	6.90
	discrete input, continuous target, NCE loss				
8	layer 2	*layer 2*	**65.69**	**57.24**	**9.33**
19	layer 0	*layer 2*	58.55	52.31	8.07
20	layer 4	*layer 2*	58.61	54.48	7.72
	discrete input, continuous target, L1 loss				
9	layer 2	*layer 2*	**73.93**	**56.02**	**<u>10.69</u>**
21	layer 0	*layer 2*	62.98	53.47	5.20
22	layer 4	*layer 2*	65.92	53.94	7.01

Tab. 2.4: Changing Model Input Features. Performances on the dev sets of sWUGGY, sBLIMP, wSIMI metrics of BERT models using the input features from different layers of the LSTM module of CPC-big model. Layer 0 means the output of the CNN Encoder module. Best scores in each category are in bold, best scores overall are underlined.

2.4.3 Varying the number of discrete units

Here, we address the question as to why discrete units are better than continuous ones. One hypothesis is that discrete units manage to remove linguistically irrelevant information and force the transformer to focus on linguistic ones. To test this, we run a speaker discrimination probe on the discrete units and continuous features. In addition, we run a new experiment varying the number of discrete units from 20 to 2000. Hypothetically, when the number of units is too small (eg, smaller than the number of phonemes), the resulting phonetic confusions should degrade the learning of higher linguistic representations. Conversely, when the number of units is

id	input feature	target feature	loss	input	layer 3	layer 6	layer 9	layer 12	target
	BERT Models on CPC-big Features								
1	CPC-l2+km50	CPC-l2+km50	NLL-l	7.3	5.90	6.13	5.64	3.87	7.3
2	CPC-l2+km50	CPC-l2+km50	NLL-e	7.3	5.36	6.97	5.40	3.95	7.3
11	CPC-l0+km50	CPC-l2+km50	NLL-e	26.11	15.08	12.16	10.63	7.01	7.3
12	CPC-l4+km50	CPC-l2+km50	NLL-e	22.1	13.57	9.40	10.05	6.06	7.3
3	*CPC-l2*	CPC-l2+km50	NLL-l	3.81	8.87	15.30	14.91	9.10	7.3
4	*CPC-l2*	CPC-l2+km50	NLL-e	3.81	8.47	16.82	21.47	12.84	7.3
13	*CPC-l0*	CPC-l2+km50	NLL-e	15.01	6.57	4.74	4.04	4.07	7.3
14	*CPC-l4*	CPC-l2+km50	NLL-e	9.75	7.03	8.78	10.49	4.98	7.3
5	*CPC-l2*	*CPC-l2*	NCE	3.81	6.47	6.83	6.09	5.98	3.81
15	*CPC-l0*	*CPC-l2*	NCE	15.01	7.45	5.63	4.73	5.93	3.81
16	*CPC-l4*	*CPC-l2*	NCE	9.75	6.82	7.19	5.46	5.50	3.81
6	*CPC-l2*	*CPC-l2*	L1	3.81	7.41	7.81	10.31	12.87	3.81
17	*CPC-l0*	*CPC-l2*	L1	15.01	6.64	4.73	4.71	5.48	3.81
18	*CPC-l4*	*CPC-l2*	L1	9.75	5.89	5.27	4.80	5.38	3.81
7	*CPC-l2*	*CPC-l2*	L2	3.81	6.35	6.58	7.65	9.20	3.81
8	CPC-l2+km50	*CPC-l2*	NCE	7.3	6.49	5.50	6.87	5.37	3.81
19	CPC-l0+km50	*CPC-l2*	NCE	26.11	15.93	12.85	10.55	9.09	3.81
20	CPC-l4+km50	*CPC-l2*	NCE	22.1	12.88	10.17	10.18	8.81	3.81
9	CPC-l2+km50	*CPC-l2*	L1	7.3	5.25	5.71	4.89	10.96	3.81
21	CPC-l0+km50	*CPC-l2*	L1	26.11	13.13	10.87	9.81	15.40	3.81
22	CPC-l4+km50	*CPC-l2*	L1	22.1	11.44	9.91	7.65	11.88	3.81
10	CPC-l2+km50	*CPC-l2*	L2	7.3	5.42	5.09	4.20	3.68	3.81
	HuBERT Base Models								
23	*waveform*	MFCC+km100	NLL-e	-	7.13	4.18	5.03	9.05	27.88
24	*waveform*	H1-l6+km500	NLL-e	-	6.83	4.71	4.42	3.53	6.97
25	*waveform*	H2-l12+km500	NLL-e	-	6.66	4.43	4.48	3.78	6.26
	BERT Models on HuBERT Discrete Units								
26	H2-l12+km500	H2-l12+km500	NLL-e	6.26	5.32	6.51	6.74	4.43	6.26

Tab. 2.5: ABX of Hidden Transformer Features. Average (within and across) dev-clean ABX error of input features, target features and features from different hidden layers of Transformer model. CPC-lx stands for layer x of CPC-big, Hj-lx stands for layer x of HuBERT j'th iteration.

too large, the quantization step would start to leak other-than-phonetic information into the representation, hence making it closer to the continuous representations.

To support our hypothesis, we run kmeans on the continuous features of both CPC and HuBERT models, and vary k to be 20, 50, 100, 200, 500, 1000, and 2000, after which we train a discrete-discrete BERT model. For the CPC features, we take the layer 2 features of the CPC-big model as usual. For the HuBERT features, we train our own HuBERT base model as described in Section 2.4.4, we then take the features from layer 12 of the Transformer Encoder after the 2nd iteration, which have the best ABX (cf. Table 2.5). Following Kharitonov et al. (2022a), we train a speaker classifier in the following way: We randomly split LibriSpeech dev-clean utterances into train/valid/test (80%/10%/10%) sets and train a two-layer Transformer classifier on the sequences of discrete units or continuous features of the utterances. The classification head is performed on the first token (`bos`, or

		unit quality		language modeling on units		
model	**n units**	spk prb↑	ABX↓	sWUGGY↑ (invocab)	sBLIMP↑	sSIMI↑
CPC	20	30.40	12.66	71.71	58.97	4.72
	50	34.00	9.89	80.02	**59.86**	**7.87**
	100	49.20	9.56	**80.47**	59.47	6.09
	200	56.00	9.72	79.90	58.90	4.45
	500	64.00	10.72	79.66	59.72	6.37
	1000	61.60	11.99	79.86	58.46	6.78
	2000	67.60	14.24	78.46	58.25	5.94
	cont.	*98.00*	*5.02*		-	
HuBERT	20	24.40	14.04	62.89	57.06	7.80
	50	38.00	9.19	76.79	61.12	8.61
	100	48.00	8.34	81.09	62.47	5.19
	200	61.60	7.57	81.54	62.78	7.03
	500	68.40	7.73	**83.06**	**62.89**	9.73
	1000	74.40	9.04	82.58	61.55	8.64
	2000	73.20	11.00	81.61	62.85	**10.66**
	cont.	*99.60*	*4.23*		-	
Forced Phones	*40*	*10.00*	*0.00*	*92.19*	*63.72*	*6.23*

Tab. 2.6: **Discrete Unit quality** (Speaker probing and ABX) and Performance of the BERT models trained on Discrete Units on the dev sets of LM scores (sWUGGY, sBLIMP, wSIMI) for different numbers of clusters on CPC and HuBERT features. The ABX is averaged on dev-clean and dev-other within and across subsets.

begin-of-sentence) of the transformer outputs. For this speaker probing task, there are 40 classes (speakers). The models are trained for 20 epochs and are validated on the valid set. We finally report the test accuracy. For reference, we also include the forced phonemes units (frame-by-frame phonemes). As the forced phonemes contain a silence, there are 40 units in total.

The results are reported in Table 2.6 and illustrated in Figure 2.2. As expected speaker classification accuracy increases with the number of clusters, and the continuous features yield the best classification. We can observe a U-shaped curve in performance across the different language metrics as a function of the number of units. Interestingly, the optimum number of units seems to be different across the model features (CPC, HuBERT) and linguistic levels. HuBERT features are better than CPC features in most cases, and seem to benefit from more clusters than CPC features. In general, we see that the language model scores seem to decrease slowly compared to ABX as the number of clusters becomes bigger. It is also interesting to note that the language model scores become steadily good as soon as the number of clusters is higher than the number of phonemes (40 units). This could be seen in Figure 2.3, where we analyze to what extent the discrete units obtained with different numbers of clusters correlate with the gold phonemes. Using the phoneme alignments of Librispeech available from Nguyen et al. (2020b), we collect all unit-phoneme pairs from the utterances of the dev-clean subset and compute the

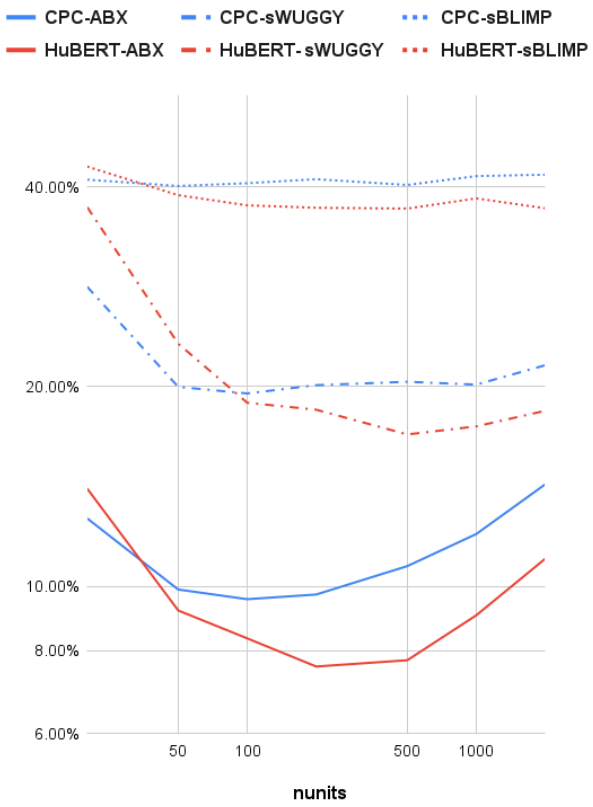


Fig. 2.2: Varying Number of Units. ABX of Discrete units and Error rate on the dev sets of LM scores (sWUGGY, sBLIMP) for different numbers of clusters for CPC and HuBERT features. The ABX is averaged on dev-clean and dev-other within and across subsets.

probability of each phoneme given a discrete unit. Figure 2.3 (top, middle, bottom) shows this unit-phone alignment for discrete units obtained from CPC features with 20, 50 and 500 clusters respectively. The phoneme order is obtained by clustering the rows of the 50-unit model with a hierarchical clustering method. We observe a limited unit-phoneme correspondence when having only 20 discrete units; but as soon as the number of clusters reaches 50, we see a clear correspondence between the units and the phonemes, although several "hard" phonemes are still dispersed and don't correspond to a single unit (e.g. ch, oy, th, uh); when there are 500 clusters, there are more units representing a single phoneme, and most "hard" phonemes are now assigned by certain units.

These results support the hypothesis that the superiority of the discrete units is due to the fact that they block the propagation and amplification of non-linguistic signals that may be present (even if attenuated) in continuous representations.

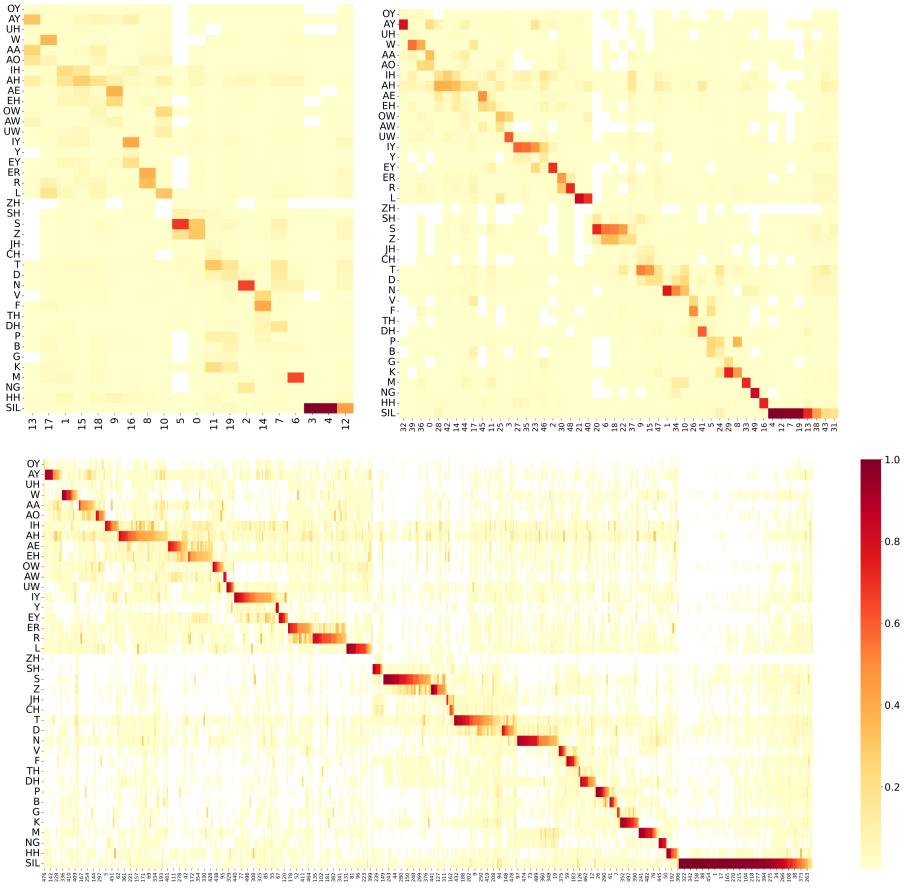


Fig. 2.3: **Unit-Phoneme Alignments.** Probability that each discrete unit belongs to possible phonemes $P\,(phoneme \mid unit)$ for discrete units obtained by clustering CPC features with different numbers of clusters: 20 (top-left), 50 (top-right) and 500 (bottom). Unit-Phoneme alignments are collected on Librispeech dev-clean subset. The phoneme order is obtained by clustering the rows of the 50-unit model with a hierarchical clustering method.

2.4.4 Comparison with state-of-the-art systems

We evaluate the HuBERT model on the zero-shots metrics and compare the results with our trained BERT models. The HuBERT model is trained iteratively, using clustering units from features of previous iteration as the teacher. We trained a HuBERT base model, which comprises a 7-layer CNN Encoder followed by a 12-layer Transformer Encoder, on the Librispeech 960h dataset for 3 iterations. The teachers for each iteration are MFCC features (100 units), Transformer's layer6 of 1st iteration (500 units) and Transformer's layer12 of 2nd iteration (500 units) respectively. Architecturally, the Transformer Encoder of the HuBERT model is very similar to our model 13 (*continuous input layer 0, discrete target layer 2, NLL-e loss*)

Systems id	input	target	loss	ABX (target features)↓ within clean	within other	across clean	across other	sWUGGY↑ (invocab)	sBLIMP↑	wSIMI↑
	ZeroSpeech 2021 Best Baseline System (Nguyen et al., 2020b)									
	CPC-layer2+km50	CPC-layer2+km50	NLL-l	6.71	10.62	8.41	15.06	75.51	56.16	2.05
	ZeroSpeech 2021 Text Topline Systems (Nguyen et al., 2020b)									
	Forced phones	Forced phones	NLL-l	0.00	0.00	0.00	0.00	91.88	63.16	4.44
	Phones	Phones	NLL-l	-	-	-	-	97.67	66.91	12.80
	BERT Models on CPC-big Features									
2	CPC-layer2+km50	CPC-layer2+km50	NLL-e	6.71	10.62	8.41	15.06	**80.29**	**59.93**	6.56
13	*CPC-layer0*	CPC-layer2+km50	NLL-e	6.71	10.62	8.41	15.06	77.22	55.62	**6.61**
17	*CPC-layer0*	*CPC-layer2*	L1	3.28	4.81	4.31	7.92	68.37	53.95	5.68
9	CPC-layer2+km50	*CPC-layer2*	L1	3.28	4.81	4.31	7.92	74.46	55.38	6.17
	HuBERT Base Models									
23	*waveform*	MFCC+km100	NLL-e	20.22	24.97	33.42	40.45	62.74	54.11	5.58
24	*waveform*	H-iter1-layer6+km500	NLL-e	6.29	7.51	8.76	12.82	79.13	58.89	5.45
25	*waveform*	H-iter2-layer12+km500	NLL-e	5.87	7.15	6.96	10.73	**80.19**	**59.29**	**5.87**
	BERT Models on HuBERT Discrete Units									
26	H-iter2-layer12+km500	H-iter2-layer12+km500	NLL-e	5.87	7.15	6.96	10.73	**83.29**	**61.93**	**9.73**

Tab. 2.7: Overall Results. Comparison on the test sets of the 4 ZeroSpeech 2021 metrics of our BERT models trained on continuous or discrete CPC features, BERT model trained on HuBERT discrete units and HuBERT Base models with ZeroSpeech 2021 Baseline and Topline Systems. For each continuous/discrete combination, we choose the best performing model on the dev set as reported in Table 2.4. We trained the HuBERT model for 3 iterations. The targets used to train the 3 iterations are discretized MFCC features (100 units), discretized features from Transformer's layer6 of 1st iteration (500 units) and discretized features from Transformer's layer12 of 2nd iteration (500 units) respectively. All models were trained on the LibriSpeech 960h dataset. For the ABX metrics, we report the scores on the target features used to train the model. Best scores in each category are in bold, best scores overall are underlined.

where they both take as input the continuous features of the CNN Encoder and predict discrete targets obtained from features of a higher level with a NLL-eloss.

Overall performances on the ZeroSpeech 2021 test sets are reported in Table 2.7. For each of discrete/continuous combinations, we choose the best performing model on the dev set as reported in Table 2.4. We also include the discrete-discrete model trained on HuBERT Discrete Units (500 units), which was reported to have the best LM scores in section 2.4.3. We first observe a huge improvement of model 2 compared with the baseline system, even if they both use the same units for the BERT model. This improvement greatly comes from the reimplementation of the BERT model, which uses the wav2vec2 Transformer Encoder model[5]. Changing the NLL-l loss to NLL-e loss also improves a little bit (cf. Table 2.2).

[5]One main difference between the two Transformer models is that wav2vec2 uses a Convolutional Positional Embedding instead of the standard Sinusoidal Positional Embedding. However, we did not study the effect of this difference in this paper.

It seems that using good quality discrete units as targets is very beneficial for the language models, achieving better scores than using continuous targets in all the metrics. The HuBERT model performs surprisingly well, approaching our best model on the language model tasks. This means that the Transformer Encoder of HuBERT acts as a language model as well. We see that as soon as the discrete targets have better quality, the HuBERT model manages to have better results on spoken language modeling metrics. We see that the discrete-discrete model on HuBERT Discrete Units (model 26) further improves the scores on all the metrics, confirming again our finding that it's better to train a discrete-discrete model when we have good quality units.

Comparing the results with the ZeroSpeech 2021 Systems, we observe that our models are closing the gap between spoken and text-based language models.

2.5 Conclusion

This work analyses the importance of discretization in spoken language modeling. We experimentally show that discretization is essential for spoken language modeling, although high-quality discrete units are required to obtain good performances. We also show the possibility of learning high-level language properties of a self-supervised speech representation learning model like HuBERT. Finally, we obtain state-of-the-art results on 3 out of 4 metrics of the Zero Resource Speech Challenge 2021 (Track 1 - Speech Only), bridging the gap between speech and text-based systems. Note though that because HuBERT requires a teacher that learns a discrete representation, the overall training of HuBERT is not end-to-end, because the training of the teacher is not (in fact, requires several iterations). Further work is needed to simplify this kind of training loop to learn language directly from speech inputs. Further work is also needed to assess whether the present results can generalize to other languages and datasets.

Additional Results

This part presents my supplementary experiments concerning the exploration of using larger speech units in SpeechLMs.

2.6 Exploration of Larger Speech Units

			units quality		language modeling on units		
Unit type	n units	unit/sec	1-hot ABX↓	Centroids ABX↓	LM data	sWUGGY↑ (invocab)	sBLIMP↑
Base units							
base (no dedup)	500	49.9	8.40	4.85	libri-light 6k	73.89	55.55
base+dedup	500	29.6	8.40	4.85	libri-light 6k	74.32	56.01
					libri-light 60k	**77.45**	**58.41**
BPE-based units							
base+dedup+BPE30k	30k	9.6	44.14	-	libri-light 6k	71.32	55.08
					libri-light 60k	**75.84**	**57.57**
base+dedup+BPE30k+BC2k	2k	9.6	36.77	-	libri-light 6k	69.00	54.44
					libri-light 60k	72.09	56.58
Subsampled units							
base+subsampled by 2	500	21.0	13.72	6.29	libri-light 6k	76.67	59.18
					libri-light 60k	78.78	**60.50**
base+subsampled by 3	500	15.5	22.87	10.00	libri-light 6k	**79.12**	59.51
base+subsampled by 4	500	12.1	32.27	17.32	libri-light 6k	76.79	60.37
Text controls							
asr character (w/ boundary)	28	14.9	-	-	libri-light 6k	**97.70**	**70.81**
asr BPE30k	30k	2.9	-	-	libri-light 6k	96.58	69.06

Tab. 2.8: BPE-based and Subsampled Speech Units Performances. The base units are inspired from Elkahky et al. (2023), which are the average features of layers 7-9 from the HuBERT Base followed by k-means 500. Following Elkahky et al. (2023), the deduplicated base units are processed with byte-pair encoding (BPE, Sennrich et al., 2016) with a vocabulary size of 30k, and are further applied Brown Clustering (Brown et al., 1992) to reduce the number of possible units to 2k. The subsampled units are sampled (i.e., take one unit for every n unit) to reduce the unit rate (unit/sec, the average number of units per second calculated on the LibriSpeech dev-clean set). The 1-hot and Centroids ABX are computed on 1-hot and centroids vectors of units, respectively, and are averaged over the within- and across- tasks on LibriSpeech dev-clean and dev-other subsets. We then further train a *12-layer transformer decoder LM* on the units with either the clean-6k subset of libri-light as in Lakhotia et al. (2021) or the full libri-light dataset. The reported sWUGGY and sBLIMP scores are calculated with *unnormalized log-likelihood* of speech stimulus (not divided by number of tokens).

One issue of speech units is their small granularity, possibly making it hard for LM to learn long context. Although SpeechSSL models downsample speech features from 16Khz to 50hz, and deduplication also reduces substantially the number of units, the rate of speech units is still high compared to text tokens. Following Elkahky et al. (2023), we tried to reduce the frame rate of speech units by applying BPE

(Sennrich et al., 2016), and possibly further Brown Clustering (Brown et al., 1992), on the units and evaluate their quality in terms of spoken language model metrics (sWUGGY, sBLIMP). The results are reported in Table 2.8.

We note that BPE-based units, although have better frame rate than the character control, don't have a good ABX as well as LM metrics. We see that BPE30k units have worse performance than the deduplicated base units, and applying Brown Clustering is not helpful. It is however interesting to note that scaling the LM dataset from 6k to 60k hours of speech is beneficial, and help to boost the performances of BPE units by a great deal. This could suggest that BPE units could benefit scaling much more data. The results are a little bit contradictory to a recent work (Chou et al., 2023), where they found doing BPE on speech units is beneficial for spoken language modeling. However, they trained their speech language model jointly with text and mainly evaluated the models on speech generation tasks. At the time of the experiements, we didn't have the chance to scale the models with more data and probe them with more semantic tasks such as StoryCloze. We leave this to possible further work.

Additionally, we tried to reduce the unit rate naively by subsampling the speech units (i.e. take one unit for every n unit) and then train language models on the subsampled units (cf. Table 2.8) and interestingly found that this helps to improve the performance of language models. This show that larger units could be beneficial for SpeechLMs and 50Hz seems not to be an optimal value. However, doing subsampling could probably loss information from the speech and we actually observed a loss in the resynthesis quality. This leads to our next experiments where we designed HuBERT models specifically with smaller frame rates.

Inspired from the previous results, we trained new HuBERT model with smaller speech rates to obtain a better speech tokenizer for our SpeechLMs. We compare the 50hz speech units obtained from HuBERT Base with HuBERT Mix, trained on a mix of Vox Populi, Common Voice, MLS, People, Spotify, Fisher, with varying number of features rate (50hz, 25hz, 16.6hz, 12.5hz) in the last iteration. We then tried different number of clusters for each rate and compared the speech unit quality (with ABX on LibriSpeech and Fisher) and spoken language modeling metrics (sWUGGY and sBLIMP). The scores are reported in Table 2.9.

We first observe that speech units obtained from HuBERT Mix have better phonetic quality than HuBERT Base, especially on natural datasets like Fisher. This results in better performances on spoken language modeling metrics (HuBERT Base 50hz vs HuBERT Mix 50hz). Concerning different frame rates of HuBERT Mix, we see a similar trend from the previous experiment, where reducing the frame rate

Tokenizer	n units	unit/sec (dedup)	units quality				LM on units	
			1-hot ABX↓		Centroids ABX↓		sWUGGY↑	sBLIMP↑
			LS	Fisher	LS	Fisher	(invocab)	
HuBERT Base 50hz								
HuBERT Base 50hz+km100	100	26.1	8.74	16.62	6.84	13.85	71.61	**56.82**
HuBERT Base 50hz+km200	200	28.3	**8.53**	**16.57**	5.94	12.76	72.60	56.32
HuBERT Base 50hz+km500	500	31.9	8.66	17.22	**5.04**	**11.67**	**72.76**	56.06
HuBERT Mix 50hz								
HuBERT Mix 50hz+km100	100	27.4	7.23	11.57	6.42	10.47	72.42	57.61
HuBERT Mix 50hz+km200	200	28.2	**6.70**	10.43	5.29	8.88	74.78	**58.36**
HuBERT Mix 50hz+km500	500	29.6	7.24	**10.19**	4.67	7.63	**75.26**	56.88
HuBERT Mix 50hz+km1024	1024	33.1	8.18	10.65	**4.38**	**7.09**	74.42	57.03
HuBERT Mix 25hz								
HuBERT Mix 25hz+km200	200	18.4	**10.17**	**14.85**	6.87	11.28	79.13	**59.63**
HuBERT Mix 25hz+km500	500	19.7	11.10	15.26	5.80	9.81	**81.22**	59.26
HuBERT Mix 25hz+km1024	1024	20.8	12.81	16.26	**5.17**	**8.85**	78.98	59.14
HuBERT Mix 25hz+km500+robust	501	19.0	10.46	**14.75**	-	-	**82.23**	**60.62**
HuBERT Mix 16.6hz								
HuBERT Mix 16.6hz+km200	200	13.9	**16.50**	**21.86**	10.73	16.26	73.91	58.92
HuBERT Mix 16.6hz+km500	500	14.5	17.16	22.42	8.34	14.03	78.79	**59.94**
HuBERT Mix 16.6hz+km1024	1024	15.0	19.98	23.88	**7.28**	**12.38**	**79.11**	59.24
HuBERT Mix 12.5hz								
HuBERT Mix 12.5hz+km200	200	10.9	**22.44**	**26.71**	15.87	20.67	75.70	58.09
HuBERT Mix 12.5hz+km500	500	11.3	24.78	27.67	13.13	17.68	**78.33**	59.00
HuBERT Mix 12.5hz+km1024	1024	11.5	26.89	29.12	**11.40**	**15.92**	77.67	**59.10**

Tab. 2.9: HuBERT Tokenizers with Different Framerates. We compare speech units obtained from HuBERT Base (trained on LibriSpeech, from (from Hsu et al., 2021a) and HuBERT Mix (trained on a mix of Vox Populi, Common Voice, MLS, People, Spotify, Fisher, from Hassid et al., 2023). The HuBERT Mix models are trained for 4 iterations, with the 4th iteration varying in downsample sizes (50hz, 25hz, 16.6hz, 12.5hz). The HuBERT features are clustered with different number of clusters. The *HuBERT Mix 25hz+km500+robust* line corresponds to applying augmentation invariant method in Gat et al. (2023) to *HuBERT Mix 25hz+km500* units, resulting in 501 units. We evaluate speech units quality in terms of 1-hot ABX and Centroids ABX, calculated on LibriSpeech (dev-clean and dev-other) and Fisher (valid) datasets. We then train a *12-layer Transformer Decoder LM* on the clean-6k subset of libri-light, and evaluated on zerospeech metrics. The reported sWUGGY and sBLIMP scores are calculated with *unnormalized log-likelihood* of speech stimulus (not divided by number of tokens).

substantially improves the sWUGGY and sBLIMP metrics. We observe that going beyond 25hz doesn't help much in this case, and using 500 units for 25hz gives the best performance in general. We then decided to further improve this tokenizer by applying the augmentation method in Gat et al. (2023) to make the speech units more robust. This speech tokenizer is later used in Hassid et al. (2023) and in the work of Chapter 5.

3 Spoken Dialogue Language Modeling

Spoken dialogue has always been an essential part of human conversation. Modeling such dialogues is not only important but also very challenging due to the richness of human speech. Following the "textless" approach, we could extend to spoken dialogue modeling, where we consider dialogues as parallel streams of audio. This enables us to not only modelize spoken dialogues but also generate them. We'll see in this chapter how we could deal with such spoken dialogues.

This chapter presents the following paper that was published in the Transactions of the Association for Computational Linguistics (TACL):

Tu Anh Nguyen, Eugene Kharitonov, Jade Copet, Yossi Adi, Wei-Ning Hsu, Ali Elkahky, Paden Tomasello, Robin Algayres, Benoît Sagot, Abdelrahman Mohamed, and Emmanuel Dupoux (Mar. 2023b). "Generative Spoken Dialogue Language Modeling". In: *Transactions of the Association for Computational Linguistics* 11, pp. 250–266

It is followed by Section 3.7, which consists of my preliminary experiments in order to improve the dGSLM model.

Statement of contribution:

I implemented all the models as well as performed the experiments mentioned in this chapter, with the ideas and suggestions obtained from discussions with my colleagues as well as feedback from reviewers.

Publication: Generative Spoken Dialogue Language Modeling

Tu Anh Nguyen[◇,†], **Eugene Kharitonov**[◇1], **Jade Copet**[◇], **Yossi Adi**[◇], **Wei-Ning Hsu**[◇], **Ali Elkahky**[◇], **Paden Tomasello**[◇], **Robin Algayres**[◇1], **Benoît Sagot**[†], **Abdelrahman Mohamed**[◇], **Emmanuel Dupoux**[◇,‡]

[◇]Meta AI Research, [†]Inria, Paris, [‡]EHESS, ENS-PSL, CNRS, Paris

{ntuanh, abdo, dpx}@meta.com

Abstract

We introduce dGSLM, the first "textless" model able to generate audio samples of naturalistic spoken dialogues. It uses recent work on unsupervised spoken unit discovery coupled with a dual-tower transformer architecture with cross-attention trained on 2,000 hours of two-channel raw conversational audio (Fisher dataset) without any text or labels. We show that our model is able to generate speech, laughter and other paralinguistic signals in the two channels simultaneously and reproduces more naturalistic and fluid turn taking compared to a text-based cascaded model.[2,3]

3.1 Introduction

In natural conversations, speakers spontaneously coordinate who is currently speaking and when the other person will speak next. As a result, conversations end up being a fluent succession of *turns* without much overlapping speech or long stretches of silence. Of course, silences and overlaps also occur naturally and they carry significant information which is interpreted within the conversation setting. For instance, when overlapping speech occurs it often contains content-neutral verbal information (e.g. "hmm", "yeah") or non-verbal vocalization (e.g. laughter), used to convey a listening attitude (back-chanelling) (Schegloff, 1982; Yngve, 1970). Short

[1]Work done while at Meta.
[2]Generation samples can be found at `https://speechbot.github.io/dgslm`
[3]Code and pre-trained models are made available at `https://github.com/facebookresearch/fairseq/tree/main/examples/textless_nlp/dgslm`

silences between turns do occur and show both cross-cultural variations and universal dependence on dialogue related variables, for instance, straight and positive answers to questions are typically faster than non-responses or negative responses (Stivers et al., 2009).

All of this *turn-taking* coordination is natural to humans, and starts to be learned at an early age by infants (Nguyen et al., 2022b). In contrast, it remains a challenging area of research in human/machine interactions (Skantze, 2021). One of the reason is that much of the research into natural dialogue modeling is taking place with text-based interfaces. Here, the coordination problem is primarily focused on semantic coherence and appropriateness of the artificial agent in interaction with a human (see Ni et al., 2021 for a review). The turn-taking problem itself is being taken care of by an artificially imposed walkie talkie arrangement; each agent is writing in turn and signalling the end of it's turn by pressing carriage return.

Within speech-based systems, it is very similar, as current spoken assistants like Siri or Alexa are triggered by a predetermined wake word, and wait for the end of an utterance followed by sufficient silence to segment the turns of the human interlocutor. This may give rise to slow and unnatural conversations. In fact, in human-human conversation, *pauses* within speaker turns tend to be on average longer than *gaps* between speaker turns (Brady, 1968; Heldner and Edlund, 2010; Ten Bosch et al., 2005), indicating that silence may not be the main cue for humans to switch turns. Because most speech-based systems are based on Automatic Speech Recognition (ASR), and that many significant aspects of speech like prosody and nonverbals are typically not annotated in naturalistic speech dialogues, current dialogue systems have been struggling with generating naturalistic dialogue.

Here we capitalize on recent progress in self-supervised learning and textless speech processing (Borgholt et al., 2022; Borsos et al., 2023; Lakhotia et al., 2021) to investigate the possibility to directly train a spoken dialogue model from raw audio, bypassing the need for text or ASR. Briefly, we build on self-supervised discrete speech representations models, which we train on spontaneous conversations with each speaker having his or her own audio channel. After training, the speech units come to represent not only verbal but also nonverbal materials. We can now encode a conversation between two interlocutors as two parallel streams of discrete tokens. We then introduce a novel dual-tower transformer architecture, where each channel is processed by one "tower" of the model which learn via an autoregressive loss, but the two towers also communicate via cross-attention in their hidden units. This cross-attention is critical for the correct synchronization of the two channels and result in a naturalistic distribution of turns, overlap and pauses. While this system

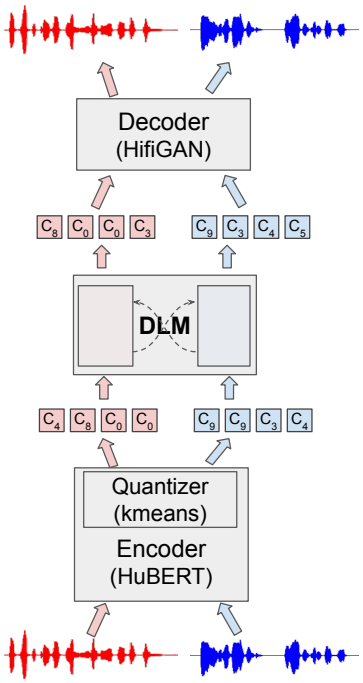


Fig. 3.1: **General Schema for dGSLM**: A discrete encoder (HuBERT+kmeans) turns each channel of a dialogue into a string of discrete units $(c_1, ..c_N)$. A Dialogue Language Model (DLM) is trained to autoregressively produce units that are turned into waveforms using a decoder (HifiGAN).

is not trained on enough data to capture deep syntactic and semantic aspects of dialogue, and indeed scores below a text-based cascaded ASR+LM+TTS model on semantic content, it does capture better surface characteristics of chitchat in mimicking accurately turn-taking and backchanneling. This can be seen as a proof of principle that previously difficult to capture aspects of spontaneous conversations can be captured with minimally modified language modeling techniques. Finally, our model opens up new possibilities to create more natural naturalistic human-machine dialogue systems in the future.

3.2 Related work

Unsupervised Spoken Language Modeling. Recently great advances have been achieved in the area of representation learning from raw audio. Models trained with either autoencoder objectives (Ondel et al., 2016; Oord et al., 2017a) or masked objectives (CPC: Oord et al., 2018; APC: Chung and Glass, 2020; wav2vec 2.0: Baevski et al., 2020c; HuBERT: Hsu et al., 2021a; MockingJay: Liu et al., 2020)

from raw speech can learn audio representation that can be used for a variety of downstream tasks (Yang et al., 2021), see Borgholt et al. (2022) for a review.

Most of these models build a codebook of discrete units, either as latent representation or as targets. The discrete representation can in turn be fed to a standard autoregressive language model, which can then be sampled to generate new speech sequences (Dieleman et al., 2021; Lakhotia et al., 2021). An interesting aspect of this procedure is that it can capture aspects of speech that are typically not available in written transcriptions and can therefore model prosody and intonation (Kharitonov et al., 2022b), or non verbal vocalizations typical of emotional speech (Kreuk et al., 2022). Up to now, however, no such model has been applied to multi-party conversational speech.

Dialogue Generation. Since the early work on end-to-end neural dialogue generation (Li et al., 2015; Serban et al., 2016; Vinyals and Le, 2015), empowered by scalable methods for language representation (Lewis et al., 2020a; Radford et al., 2018), there has been enormous progress in the area of dialogue generation (Adiwardana et al., 2020; Roller et al., 2020; Zhang et al., 2019). More recent research focused on utilizing retrieval augmented generation methods (Lewis et al., 2020b) for long-context, multi-session conversations (Xu et al., 2021a), and grounding responses on fresh information from the internet (Komeili et al., 2021; Shuster et al., 2022). However, all the progress in this research work centered around text dialogues leaving out non-lexical information (Ang et al., 2002; Schuller et al., 2013) in human-human dialogues, e.g., emotion, pauses, laughter, hesitation, and interruption. Our work builds on end-to-end techniques while taking a speech-first approach to address this shortcoming, where prompts and generated sequences are represented as self-supervised discrete speech representations (Lakhotia et al., 2021). As a result, the capacity of our models is constrained by the amount of publicly available speech dialogues; for example, the LDC English Fisher dialogues corpus (Cieri et al., 2004) contains roughly 12M words compared to tens of billions of words in the case of text-based dialogue systems. There have been recent calls for large-scale end-to-end benchmarks and datasets with spoken input to fill this gap (Faruqui and Hakkani-Tür, 2021).

Turn-taking Modeling. Decades-long research on conversation analysis (Duncan, 1972; Gravano and Hirschberg, 2011; Levinson and Torreira, 2015; Sacks et al., 1974; Schegloff, 2000; Ward, 2019) has shown that human turn-taking relies on a variety of complex signals, or cues, including prosodic cues, linguistic cues and even

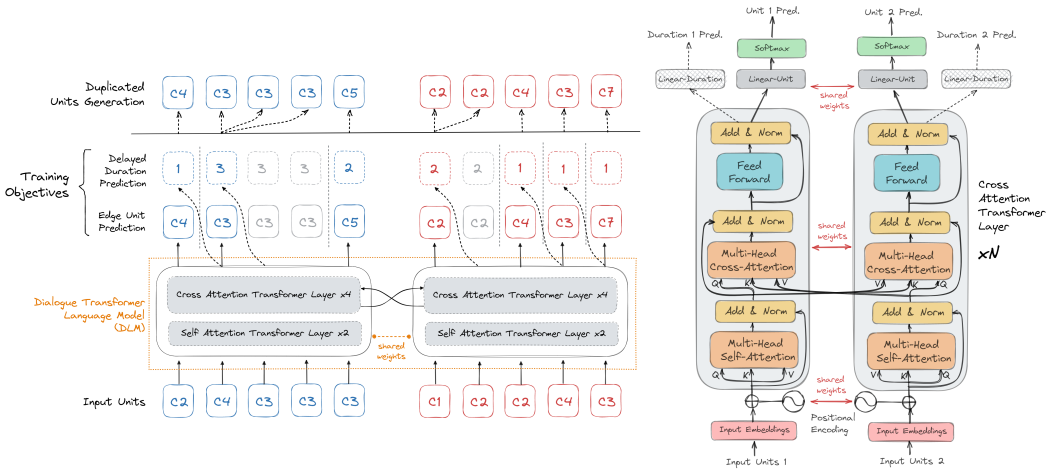


Fig. 3.2: Illustration of the Dialogue Transformer Language Model (DLM). Left: DLM Training Objectives. During training, the loss is applied only to edge units and their durations. During generation, the model duplicates the units with the corresponding predicted durations. Right: The Cross-Attention Transformer Layer Architecture.

non-verbal cues such as gaze or gestures, making turn-taking modeling a challenging problem. Simple turn-taking models using finite-state machines have been proposed to predict the distribution and durations of turn-taking events (Cassell et al., 2001; Raux and Eskenazi, 2009; Thórisson, 2002). More recently, more sophisticated machine learning-based models of turn-taking have been introduced (Masumura et al., 2018; Meena et al., 2014; Roddy et al., 2018; Skantze, 2017). These models used multi-modal features including simple linguistic features and prosodic features extracted from the speech to predict turn shifts. Most recently, Ekstedt and Skantze (2020) has shown the possibility of turn-taking prediction in spoken dialogue using only linguistics features (text input). We use these definitions of turn-taking events to analyse the output of our models.

3.3 Approach

Our approach is based on the availability of a dataset constructed along the Fisher Telephone conversation collection protocol (Cieri et al., 2004) where each conversation involves two speakers, and each speaker is recorded in a separate audio channel while having a very casual conversation. We follow the textless generative spoken language modeling pipeline of Lakhotia et al. (2021), which decomposes the problem of speech generation into three components: a Speech-to-Units encoder, a Units-to-Units language model and a Units-to-Speech decoder. For the encoder

we adopt HuBERT, (Hsu et al., 2021a) followed by k-means clustering; for the decoder network we use a modified Hifi-GAN neural vocoder (Kong et al., 2020), similarly to Polyak et al. (2021). These models are trained on single channel data from the Fisher dataset and applied to each channel separately, which do not model cross-channel interactions. For the language model, we introduce our new Dialogue Transformer Language Model, or DLM. Figure 3.1 presents an overview of our system. The following sections (Sections 3.3.1–3.3.3) will present at a high level each component of our model and review the turn-taking terminology in this study (Section 3.3.4).

3.3.1 Discrete Phonetic Representation

Conversational speech contains casual expressions (filler words like 'hmm') and a variety of non verbal sounds (e.g., laughter) that do not appear in formal or read speech. We therefore train a HuBERT model (Hsu et al., 2021a) directly on our conversation dataset in order to obtain domain-appropriate phonetic representation. Specifically, it is trained on the collection of voice segments extracted of all speakers in the dataset. The discrete units are then obtained by clustering the representation of the HuBERT model using the k-means algorithm. At inference time, the two-channel speech waveform is encoded channel-wise into two time-aligned streams of discrete units.

In Table 3.1, we compare the HuBERT Base model (Hsu et al., 2021a) trained on 2000h of Fisher dataset versus 1000h of Librispeech dataset on the machine-ABX phonetic test. We used Libri-light ABX (Kahn et al., 2020) for the Lirispeech test. For the Fisher, we generated a Fisher ABX dataset using the phonetic alignments obtained from Fisher development set. The results clearly show a domain effect, whereby the Fisher dataset is a better training set than the Librispeech dataset for ABX discriminations in Fisher.

Tab. 3.1: **Within and Across-Speaker ABX error** on Fisher dev and LibriSpeech dev-clean datasets for HuBERT Base and HuBERT Fisher models.

	Fisher		LibriSpeech	
	within↓	across↓	within↓	across↓
HuBERT Base	7.77	12.57	**3.95**	**4.69**
HuBERT Fisher	**5.50**	**8.35**	11.17	14.70

3.3.2 Waveform Generation

For the waveform generation, we used the discrete unit-based HiFi-GAN vocoder from Polyak et al. (2021) trained on a small subset of high quality single-channel voice segments of our conversation dataset, using discrete units obtained from the HuBERT model and 1-hot speaker information from the dataset. During generation, we generate each channel of discrete units with one different speaker, and combine the audio generated from the two channels. Voices for the waveform generation are chosen from the speakers in the HifiGAN training set.

3.3.3 Dialogue Transformer Language Model

We introduce our Dialogue Transformer Language Model (DLM), which is a two-tower transformer network with *Cross-Attention* and shared weights trained with *Edge Unit Prediction* and *Delayed Duration Prediction* objectives. The model is illustrated in Figure 3.2 and its components will be detailed below, and we will perform ablations to test for the effects of each of these components.

We will also compare the two-tower model with a simpler single-tower model with dual inputs. This last model is inspired by previous work in multi-stream language model (Kharitonov et al., 2022b). It consists of a single transformer, with two embedding heads in the input and two softmax heads in the output. This model combines very early the two speaker channels at the embedding layer and models them jointly, only to separate them again in the last layer. We call this model MS-TLM (Multi-Stream Transformer Language Model)

Cross-Attention Transformer Layer. When modeling separate channels of dialogue, we would like the LM to not only get information from the history of each channel itself, but also have information from other channels as well. As a result, we add an additional Muti-Head Cross-Attention block after the Multi-Head Self-Attention block to share information between different channels (cf. Figure 3.2, right). We train a single Transformer model which we clone into the two towers with shared weights, which allows the model to be speaker-independent without having to do permutation invariant training.

Edge Unit Prediction. Previous work (Kharitonov et al., 2022b) disentangles the content modeling problem from the duration modeling problem by training the language model on deduplicated discrete units and the corresponding unit durations with different objectives. However, in our setting, units from different channels are time-aligned and there would be no easy way to keep the alignment if we were to deduplicate each input stream. On the other hand, training a language model on duplicated units is more difficult as content and duration information are entangled and learnt simultaneously, resulting in a poor modeling performance. From this point of view, we introduce an edge unit prediction objective, which forces the model to predict the next unit only if it is different from the current one (i.e. edge unit). We use cross-entropy loss for this objective, and the edge unit prediction loss is then defined as:

$$\mathcal{L}_{EU} = \sum_{c=1}^{2} \sum_{\substack{t \\ u_t^{(c)} \neq u_{t-1}^{(c)}}} \log p(u_t^{(c)} \mid u_{1:t-1}^{(1,2)}; \theta),$$

where $u_t^{(c)}$ represents the discrete unit from channel c at time t and θ denotes the model parameters.

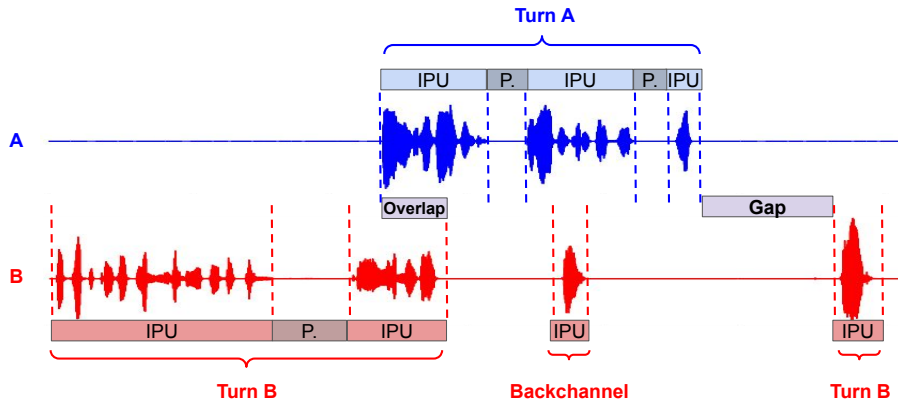


Fig. 3.3: Illustration of turn-taking events: IPU (Interpausal Unit), Turn (for speaker A and Speaker B, resp), P. (within-speaker Pause), Gap, Overlap and Backchannel.

Delayed Duration Prediction. Besides the unit prediction objective, DLM models the duration of the edge units with a duration prediction objective. As unit durations are highly varied, we output a continuous duration prediction and employ an L1 loss. Due to the high correlation between the duration and the unit itself, we follow Kharitonov et al. (2022b) and perform a delayed unit duration prediction, which predicts the duration of an edge unit at time t given the first $t - 1 + \Delta$ units, where

Δ is a delay factor ($\Delta \geq 0$). The delayed duration prediction loss is then defined as:

$$\mathcal{L}_{ED} = \sum_{c=1}^{2} \sum_{\substack{t \\ u_t^{(c)} \neq u_{t-1}^{(c)}}} \left| d_t^{(c)} - \hat{d}_t^{(c)} \left(u_{1:t-1+\Delta}^{(1,2)}; \theta \right) \right|,$$

where $d_t^{(c)}$ represents the target duration (number of repetitions) of the edge unit $u_t^{(c)}$ and $\hat{d}_t^{(c)}$ is the continuous duration prediction of the DLM model.

Training objective. The training loss of DLM is the sum of the edge unit prediction loss and the delayed duration prediction loss:

$$\mathcal{L}_{DLM} = \mathcal{L}_{EU} + \mathcal{L}_{ED}. \tag{3.1}$$

Model Inference for Generation. For generation, we autoregressively generate edge units and the corresponding durations in both channels. Even though the loss is applied only at the edge units, the model may generate spurious and inconsistent data at other non-edge time steps. We give precedence to the predicted duration associated with the first edge unit predicted in each channel and overwrite the network output with this edge units for the corresponding number of steps. It is this overwritten content which is used as input to the network till the next edge unit. For example, if we predict a unit $u_t^{(c)}$ at time t and the corresponding duration $d_t^{(c)}$ at time $t+1$, we replace the next $d_t^{(c)}$ units of channel c by $u_t^{(c)}$ and only alter the unit at time $t + d_t^{(c)}$. The duration prediction is rounded during generation.

3.3.4 Definitions of turn-taking metrics

Because our model generates two audio channels in parallel, it is possible to use simple Voice Activity Detection (VAD) tools on the output to derive turn-taking metrics. Following Figure 3.3, we define an *Inter-Pausal Unit* (IPU) as continuous stretch of speech in one speaker's channel, delimited by a VAD silence of more than 200ms on both side. We define *silence* as sections of the recording with no voice signals on either channel and *overlap* as sections where there are voice signals on both channels. Silences can be subdivided into *gaps* (when it occurs between two IPUs by distinct speakers) and *pauses* (when they occur for the same speaker). Successive IPUs by the same speaker separated by a pause are regrouped into a *turn*. Overlap could also theoretically be subdivided into *backchannel* (when it is rather short IPU contained within an IPU of the other speaker) and *interruption* (when it

starts within an IPU of the other channel and continues after its end), but the exact definition is dependant on high-level linguistic features, which we will not attempt to extract here. In our analysis, we will therefore tally the distribution of duration of IPUs, gaps, pauses and overlaps in the training corpus and in generated dialogues of our various models.

3.3.5 Cascaded Dialogue Baseline System

We compare our textless-based dialogue models with a traditional cascaded dialogue system which consists of an ASR model, followed by a text-based language model and a Text-To-Speech (TTS) module. We first transcribe each channel of the dialogue with the ASR model, we then combine the transcribed text into a turn-based conversation,[4] we ignore any turns that are completely contained inside an other turn. We train a Transformer Language Model on these conversations and we finally employ a TTS module to synthesize the generated text into a turn-based conversation.

3.4 Experimental Setup

3.4.1 Training Set

We use in this work the Fisher Dataset (Cieri et al., 2004), a conversation corpus consisting of more than 16,000 English telephone conversations averaging ten-minutes in duration and focusing on various topics. The audio was recorded separately in two channels resulting in 2000 hours of transcribed speech.[5]

For the training of HuBERT and HifiGAN models, we follow the preprocessing steps of Kuchaiev et al. (2019)[6] to obtain a collection of single-channel voice segments of the Fisher dataset. The segments vary mostly from 10–15 seconds, with a total duration of about 1,800 hours. We divide the Fisher dataset into train/valid/test sets with a 98/1/1 split (different speakers in each split).

[4]Example: `<A> hi <B> hi how you doing <A> great <B> good good my name is marine.`

[5]The transcription was done using the Quick Transcription specification (Cieri et al., 2004), resulting in some inaccuracies and untranscribed portions. Here, we only used the transcriptions to obtain speech segments containing vocal activity to train the HifiGan and HuBERT model. The DLM was trained on the unsegmented raw data.

[6]https://gitlab.nrp-nautilus.io/ar-noc/nemo/-/blob/master/scripts/process_fisher_data.py

Tab. 3.2: **Unit Prediction loss (NLL) & Accuracy metrics** of DLM models as a function of number of Cross-Attention layers. When the number of cross-attention layers is less than 6, they are put on top of self-attention layers. The models are trained with the Next-step Unit Prediction Objective on the parallel unit streams of the Fisher stereo audio dataset.

n cross layers	NLL↓	Acc↑
0/6	1.387	71.77
2/6	1.341	72.06
4/6	**1.338**	**72.10**
6/6	**1.337**	**72.11**

3.4.2 Model Training

We train a HuBERT Base model (Hsu et al., 2021a) from raw audio. The encoder contains seven 512-channel CNN layers with strides [5,2,2,2,2,2,2] and kernel widths [10,3,3,3,3,2,2], converting the signal rate from 16,000 samples/sec down to 50 frames/sec. It is followed by 12 Transformer blocks. The model is trained with a masked objective for 3 iterations following the same recipe as in (Hsu et al., 2021a). The model alternates between feature extraction/quantization and masked-prediction training in each iteration. We used the k-means algorithm with codebook sizes of 100, 500, and 500 to quantize the MFCC features, the 6th transformer layer features, and the 9th transformer layer features for the three HuBERT training iterations. After training, we quantize the final transformer layer features into 500 units for the DLM training. We choose a large codebook size of 500 to model various kinds of vocalizations beyond broad phonetic classes. Following Hsu et al. (2021a), we use 250k training updates in the first iteration and 400k model updates in subsequent training iterations using 32 V100 32GB GPUs. As the transformer does not change the input frame rate, the encoded discrete units have a frame rate of 50 units per second (one every 20ms). We show in Table 3.1 that our HuBERT model trained on the Fisher dataset learns better phonetic information suitable for conversations than the publicly available HuBERT model trained on audiobooks (Hsu et al., 2021a).

We train the HifiGAN model on a small subset of the Fisher dataset segments consisting of 120 speakers with 10 minutes each. These speakers were selected to be of high intelligibility using the average perplexity of a phone recognizer trained on the clean Librispeech 100h training subset (Rivière and Dupoux, 2021). The model is trained to generate the audio waveform given HuBERT units of a segment and a speaker embedding vector.

For the DLM models, we use a transformer model consisting of 6 layers, with 8 attention heads per layer, and an embedding size of 512. When cross-attention is used, it is added to the top 4 transformer layers. We show in Table 3.2 the effect of the number of cross-attention layers on language modeling metrics. We find that more layers give better scores, but that 4 layers of cross-attention give almost the same performance as 6 for less complexity. We train the DLM model on the parallel unit streams encoded from 2000 hours of stereo audio, each sample contains up to 6144 unit pairs, an equivalent of 123 seconds. The models are trained on a total of 32 V100 32GB GPUs, with a batch size of 370 seconds of audio per GPU for a total number of 250k steps. We used an Adam optimizer (Kingma and Ba, 2015) with a max learning rate of 5×10^{-4}. The implementation of the DLM model is done using the fairseq (Ott et al., 2019) toolkit. It took us 66 hours on average to train 100k steps of DLM models without edge unit prediction, and 95 hours with additional edge unit prediction objective.

We also train a Multi-Stream Transformer Language Model (MS-TLM, Kharitonov et al., 2022b), a single transformer model taking two streams of units as input and autoregressively predict the next units in both streams. It is a standard Transformer Language Model, with 6 layers, 8 attention heads per layer and an embedding size of 512, with the difference that the embedding layer concatenates the two embeddings of the two parallel units, and the output layer produces two softmax layers to predict the next units in both streams. We train the MS-TLM model similarly to the DLM models as previously mentioned. Training 100k steps of MS-TLM model took us 40 hours.

For the cascaded system, we use a pre-trained ASR model[7] to decode the Fisher dataset. We then train a standard 6-layer Transformer Language Model on the turn-based conversations obtained from the ASR. We pre-process the text using a byte pair encoding (BPE, Sennrich et al., 2016) with 20k iterations and limit each sample to have 512 tokens. We trained the language model for 100k steps on 32 V100 32GB GPUs with a batch size of 2048 tokens per GPU. We use the same optimizer as for other models. Finally, we use the Google TTS API to synthesize generated conversations, with two different voices indicating two different speakers.

[7]We use the robust wav2vec2-large model fine-tuned on Switchboard dataset (Hsu et al., 2021c). For decoding, we use the 4-gram KenLM language model trained on Switchboard dataset.

3.4.3 Evaluation Metrics

This section presents the evaluation metrics used to assess our dialogue models on two dimensions: Training and Generation.

Tab. 3.3: **Training Metrics** across the DLM models that differ in Cross-Attention Layer (CA), Edge Unit Prediction (EP), Duration Prediction (DP) and Duration Delayed Factor (Δ). The MS-TLM model used a single transformer with two input and output streams.

					Edge Unit		**Duration**	
Id	CA	EP	DP	Δ	NLL↓	Acc↑	MAE↓	Acc↑
	MS-TLM							
0	-	-	-	-	3.05	34.14	-	-
	DLM							
1	×	×	×	-	3.07	34.13	-	-
2	✓	×	×	-	2.95	35.68	-	-
3	✓	✓	×	-	2.49	48.36	-	-
4	✓	✓	✓	0	2.26	54.09	1.47	51.90
5	✓	✓	✓	1	**2.25**	**54.27**	**1.23**	**58.18**

3.4.3.1 Training Metrics

These metrics evaluate the dialogue modeling performance in each channel separately using metrics close to the training loss. They are computed by encoding files from the development set and extracting statistics on the predicted outputs at each time steps. They are used to compare the different versions of the DLMs and therefore not applied to the cascaded model.

Edge Unit Prediction. We report the Negative Log-Likelihood (NLL), or Cross Entropy loss when predicting edge units. We also compute the Prediction Accuracy.

Edge Duration Prediction. We use the Mean Absolute Error (MAE), or L1 Loss when evaluating edge duration prediction (a MAE of 1 corresponds to 20ms of error). The Duration Accuracy is also reported.

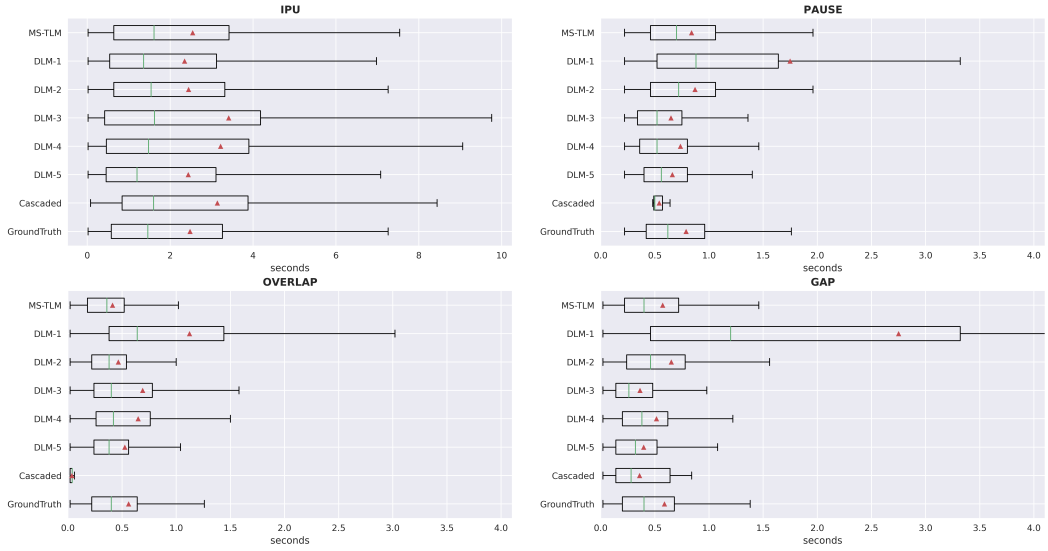


Fig. 3.4: Distributions of durations of turn-taking events in prompted continuations across models, compared to the prompts' continuation ground truth segments (see models ids in Table 3.3). The green line and the red triangle represent the mean and the median of the events respectively.

3.4.3.2 Dialogue Generation Metrics

We evaluate the generation properties of our models using descriptive statistics, automatic metrics and human-based judgements. Unless otherwise written, we perform conditional generation and generate 90-second long continuations using 117 30-second long prompts extracted from the development set and use the default generation temperature of 1.0. We generate the units by sampling among the top 20 possible units.

Turn-taking Event Statistics. We compute the turn-taking events as defined in section 3.3.4 using the samples generated by the models with a Voice Activity Detection (VAD) using pyannote library[8] (Bredin et al., 2020). We then analyse the statistics of these turn-taking events (number of events and their durations) across different models.

Turn-taking Event Consistency. We evaluate the model's capacity to generate consistent conversations in terms of turn-taking events. We measure the Pearson correlation between the total duration of events in each prompt and in the corresponding continuation.

[8]https://github.com/pyannote/pyannote-audio

Tab. 3.4: Number of turn-taking events and cumulated durations per minute across models for prompted continuations, compared to ground truth continuations, and to the same statistics in the training set.

Id	Model	Number of occurrences / min				Cumulated duration /min			
		IPU	Pause	Gap	Overlap	IPU	Pause	Gap	Overlap
0	**MS-TLM**	19.4	10.6	5.1	3.3	49.4s	8.9s	2.9s	1.3s
1	**DLM-1**	17.7	7.9	3.9	5.5	41.4s	13.8s	10.7s	6.1s
2	**DLM-2**	20.0	10.4	5.5	3.6	48.9s	9.1s	3.6s	1.7s
3	**DLM-3**	19.0	1.8	4.9	11.7	65.0s	1.1s	1.8s	8.1s
4	**DLM-4**	18.9	3.2	5.6	9.4	60.7s	2.4s	2.9s	6.1s
5	**DLM-5**	24.2	5.4	7.2	10.9	59.1s	3.6s	2.9s	5.8s
6	**Cascaded**	17.5	0.0	14.9	0.0	54.8s	0.0s	5.3s	0.0s
	Ground Truth	21.6	7.0	7.5	6.5	53.5s	5.5s	4.4s	3.6s
	Training Set	25.9	7.2	8.6	10.0	54.5s	5.6s	4.6s	4.7s

Natural Dialogue Event Statistics. We evaluate the naturalness of the generated speech by focusing on the Speaking Rate (WPM, words per minute), Laughter Frequency (LPM, laughs per minute), Filler Word Rate (FWR, filler words per 100 words) and Floor Transfer Offset (duration between two consecutive turns of the two speakers, a positive FTO represents a gap while a negative FTO represents an overlap). For this evaluation, we use the same ASR model used to decode the Fisher dataset[7] to transcribe the generated speech. To detect laughs in the speech, we use an open-source model described in Gillick et al. (2021).[9] To compute the FWR, we use the following filler words set: {'uh', 'um', 'like', 'i mean', 'you know'}.

Semantic Evaluation. We use two evaluation metrics proposed in Lakhotia et al. (2021), perplexity (PPL) and VERT, to assess the generation quality and diversity of the models. We first transcribe the generated speech using the ASR system. As these metrics are calculated on text sequences, we combine the text from two channels into a single turn-based text sequence[4], ignoring any turns that are completely contained inside an other turn. We employ the open-source DialoGPT model[10] (Zhang et al., 2019) to compute the perplexity on the turn-based sequences. We simply replace the speaker tokens (`<A>`, `<B>`) with the `<|endoftext|>` token, indicating a turn switch. For the VERT metrics, we also compute the self-BLEU and auto-BLEU on the turn-based text sequences. As the conversation texts contain a lot of repetitions, we report the VERT-4 score instead of VERT-2 score as in Lakhotia et al. (2021).

Since the PPL and VERT scores highly depend on the generation temperature, we perform generation on different temperatures ranging from 0.3–2.0. We then

[9] https://github.com/jrgillick/laughter-detection
[10] https://huggingface.co/microsoft/DialoGPT-medium

Tab. 3.5: **Natural Dialogue Event Statistics.** Speaking Rate (WPM, words per minute), Laughter Frequency (LPM, laughs per minute) and Filler Word Rate (FWR, filler words per 100 words) of the prompted continuation speech across models, compared to ground truth continuations.

Id	Model	WPM	LPM	FWR
0	**MS-TLM**	139.17	1.88	9.36
1	**DLM-1**	123.60	1.98	9.39
2	**DLM-2**	141.09	2.06	10.36
3	**DLM-3**	281.41	7.08	3.40
4	**DLM-4**	244.13	6.05	3.38
5	**DLM-5**	211.98	3.62	5.50
6	**Cascaded**	216.73	0.00	7.08
	Ground Truth	181.46	3.60	7.25

compute the PPL and VERT for each temperature and fit the points corresponding to different temperatures with an exponential line and report the PPL@GT (PPL with respect to the ground truth VERT) score (cf. Figure 3.7). For the conditional generation case, we compute instead the conditional perplexity (cond. PPL), which is the perplexity of the generated sequence given the concatenation of the prompt sequence and generated sequence as input to the DialoGPT model.

Human Opinion Score. We perform a human evaluation on the generated examples. The opinions are based on two dimensions: *N-MOS (naturalness Mean Opinion Score)* representing naturalness and turn-taking conversationality, and *M-MOS (meaningfulness Mean Opinion Score)* for meaningfulness and content quality. For N-MOS, we asked the participants to concentrate on the fluidity and naturality of the interaction as well as the expressiveness of the speakers regardless of meaning. For M-MOS, they should focus on what is being said and if it is semantically coherent. For these two evaluations, we used a scale of 1-5 (1: worst, 5: best). The CrowdMOS package (Ribeiro et al., 2011) was used for all subjective evaluations using the recommended recipes for detecting and discarding inaccurate scores. Indeed, we remove all workers whose correlation with the mean scores is lower than 0.25, and then filter out outlier workers whose correlation with the mean scores is lower than 0.6. We enforced at least six raters for each of the generated samples. Participants were recruited using a crowd-sourcing platform.

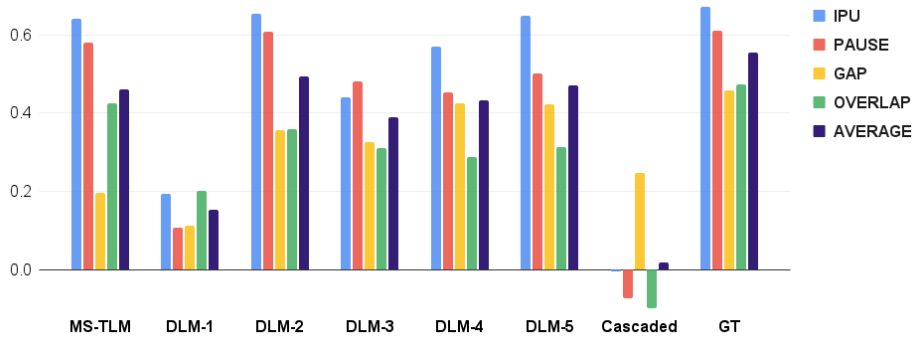


Fig. 3.5: **Correlation between the duration of events in the prompts and in the continuations** across models, compared to ground truth (GT), where the correlation is computed between the first 30 seconds and the following 90 seconds of the samples.

3.5 Results

3.5.1 Content and Duration Modeling

Table 3.3 reports the modeling evaluation metrics on our development subset of the Fisher dataset. In rows Id 1-5, we compare different DLM models, while row Id 0 represents the MS-TLM model, which takes as input multiple unit streams from different channels, and predicts the next-step units only. We note that for models Id 1-3, the next-step unit prediction objective is also included in the training process, but when the duration prediction objective is employed (models Id 4-5), the next-step unit prediction objective is omitted.

We observe that by using the self cross-attention layers, the edge unit prediction metrics slightly improve (u NLL: 3.07 vs 2.95). On considering models Id 2 & 3, we observe a huge improvement in edge unit NLL & Accuracy when introducing the edge unit prediction objective (u NLL: 2.95 vs 2.49). By introducing the duration prediction objective and removing the next-step unit prediction objective, we see that the model performs even better on the edge unit prediction metrics (u NLL: 2.26), and finally the duration metrics greatly improves when we apply a delayed duration prediction (d MAE: 1.47 vs 1.23).

On comparing with the MS-TLM model, we see that our best DLM model perform much better on content modeling. The reason, we believe, is related to the entangled modeling of content and duration in the MS-TLM model.

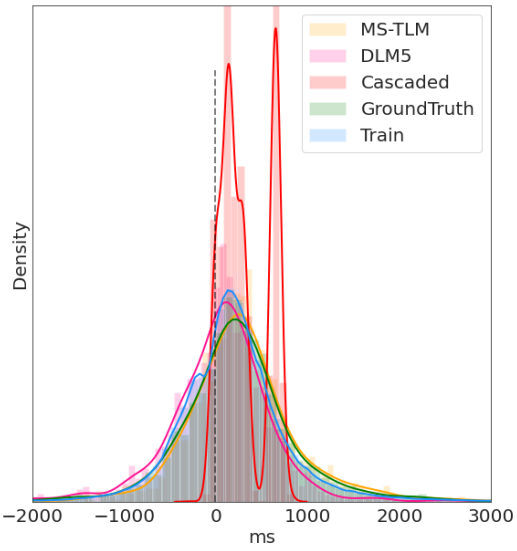


Fig. 3.6: **Histogram of Floor Transfer Offset (FTO)** in the generated speech across models, compared to ground truth continuations and the training set.

3.5.2 Turn-taking Event Statistics

In this section, we analyse the distribution of the turn-taking events (as described in section 3.3.4) in the dialogue continuations generated by our models. The statistics are computed over 3 hours of generated speech per model.

Figure 3.4 shows the distribution of each of the 4 turn-taking events: IPU, pause, gap and overlap. In this figure, the Ground truth corresponds to the true continuation of the prompts in the original corpus. Despite having a reasonably good modeling score (cf. Table 3.3), DLM-1, which has no cross-attention layers between the two transformer towers, has poor performance on turn-takings events, except for the IPU event. The lack of communication between the two channels during generation creates huge gaps and overlaps in the generated samples. The MS-TLM and DLM-2 models have similar distributions of shorter overlaps and longer pauses and gaps. They were trained using the next-step prediction loss on duplicated unit sequences, which could lead to repeated unit generation, causing a slow pace and more extended silences in the generated audio. The opposite effect happens when we introduce the edge unit prediction (DLM-3-5). These models manage to generate more overlaps, with pauses and gaps of shorter duration. These observations are further reinforced in Table 3.4, which details the number of events and their total durations per minute. It is interesting to note that all models, except DLM-1, manage to capture the empirical fact that *intra-turn* pauses tend to be longer than *between-turn* gaps (Brady, 1968; Heldner and Edlund, 2010; Ten Bosch et al., 2005).

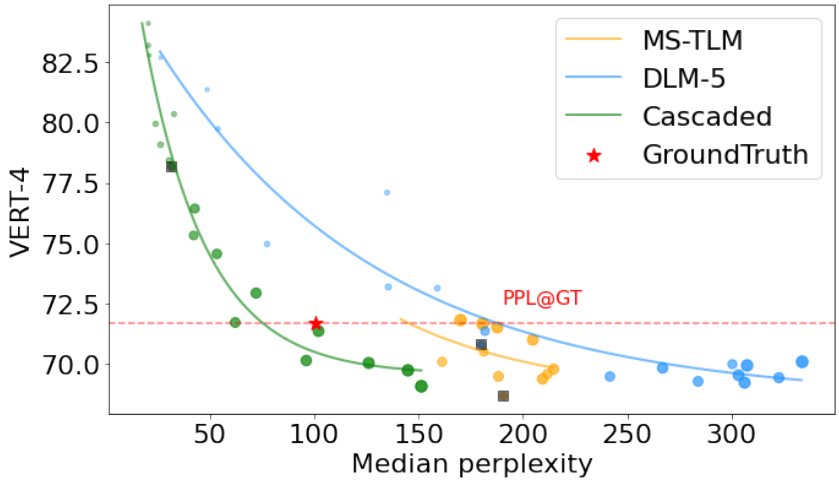


Fig. 3.7: **PPL vs VERT scores with unconditioned generation** for MS-TLM, DLM-5 and Cascaded models compared to ground truth transcriptions. The sizes of the points correspond to the temperature used for generation (0.3–2.0), squares mean default temperature 1.0. The turn-based sequences are limited to 50 words.

The cascaded model only produces alternating speech turns and therefore has almost no overlap and pause. This also results in low variance in the gap distribution, making the geneation sounds like machine conversation.

3.5.3 Turn-taking Event Consistency

Figure 3.5 shows the correlation between the total duration of turn-taking events in the prompts and in the generated continuations. For the ground truth, we compute the correlation of the events' duration between the first 30 seconds and the folowing 90 seconds in each sample. We observe that in general all models except DLM-1 and cascaded have good correlations, showing their ability to maintain the dialogue consistency. Unsurprisingly, the cascaded model has no correlation with the prompt events, except for the gaps, which are proportional to the number of turn changes.

3.5.4 Natural Dialogue Event Statistics

Table 3.5 reports the naturalness statistics on the generated samples of our models. We first notice that, compared to ground truth, models that don't have edge unit prediction (MS-TLM, DLM-1–2) tend to produce speech with less information and more hesitations (lower rate, less laughter, more filler words) than those with edge unit prediction (DLM-3–5). Adding duration prediction can effectively help to

Tab. 3.6: Semantic Evaluation. Perplexity of ASR-transcribed generated speech at default temperature (@t1) and at ground truth VERT (@GT) in both unconditional and conditional generation across models compared to ground truth transcriptions. We limit the transcribed turn-based sequences to 50 words.

		unconditional		**conditional**	
		PPL↓		cond. PPL↓	
Id	Model	@t1	@GT	@t1	@GT
0	**MS-TLM**	190.59	144.82	741.86	-
1	**DLM-1**	145.85	-	195.89	-
2	**DLM-2**	218.30	-	453.73	-
3	**DLM-3**	155.17	161.58	463.27	329.74
4	**DLM-4**	290.07	231.00	693.48	314.49
5	**DLM-5**	179.65	187.16	605.84	365.08
6	**Cascaded**	32.23	80.80	45.93	117.06
	Ground Truth	100.85	100.85	65.00	65.00

produce more natural speech, but it still produces more words than ground truth. The cascaded model is unable to produce laughter as the ASR and TTS modules are not able to capture these information, it also generate nearly "non-stop" speech at a faster rate than natural speech. Looking at Figure 3.6, we see indeed that the cascaded model has no negative FTO (overlap), and the positive FTOs (gaps) fall mostly in the range of one second. In general, other models seem to have good FTO distribution compared to the reference ground truth and training set.

3.5.5 Semantic Evaluation

For semantic metrics, we perform both conditional and unconditional generations. For conditional generation, we select 50 10-second long prompts in the validation set. For each model and temperature, we generate 50 samples and limit the transcribed turn-based text sequences to 50 words.

We found that certain models is not possible to obtain PLL@GT as they tend to generate repeated units at low temperatures, creating complete noise in the synthesis. We therefore report the PPL scores for the default temperature 1.0 (@t1). As shown in Table 3.6, we see that the dialogue models fail to generate semantically coherent speech, resulting in high perplexity, especially in prompted generation. The cascaded model has a very good perplexity as the language model was trained on word and sub-word levels, it even has a higher PPL@GT than the ground truth in the

Tab. 3.7: Human Evaluations. Conversation Naturalness (N-MOS) and Conversation Meaningfulness (M-MOS) on a 5 point scale (5 is best) with 95% CI.

Id	Model	N-MOS↑	M-MOS↑
0	**MS-TLM**	3.31 ± 0.43	2.29 ± 0.49
1	**DLM-1**	2.25 ± 0.60	1.70 ± 0.44
2	**DLM-2**	2.95 ± 0.37	2.24 ± 0.47
3	**DLM-3**	3.29 ± 0.43	2.20 ± 0.44
4	**DLM-4**	3.36 ± 0.44	2.18 ± 0.46
5	**DLM-5**	3.70 ± 0.46	2.48 ± 0.49
6	**Cascaded**	2.38 ± 0.63	2.70 ± 0.38
	Ground Truth	4.23 ± 0.26	4.21 ± 0.25

unconditional case. When it comes to conditional generation, the cascaded model has a good PPL, but is still way below the ground truth.

3.5.6 Human evaluation

For this evaluation, we filter the prompts to contain genuine alternations between the two interlocutors and balanced gender. We retained 50 10-second long prompts and generated 10 20-second long continuations for each prompt. Human evaluation results are reported in Table 3.7. The naturalness and meaningfulness MOS scores correlate well with results in previous sections. The DLM-5 model has the best performance among dialogue models, while the DLM-1 performs significantly worse on both scores. Interestingly, whereas there is a large gap between our best model and ground truth on meaningfulness (1.73 points on the 5-point scale) this gap is much reduced on turn-taking (.53 points). The cascaded model shows a lack of naturalness, while having better scores on meaningfulness than all dialogue models. However, it is still far below the ground truth despite having a very good semantic scores. Overall, our models can generate dialogues mimicking natural turn-taking, while fail maintaining cross-sentence meaningfulness. We believe the lack of semantic coherence in generated dialogues results from the fine-grained acoustic units used for modeling and the small training corpus size.

3.6 Conclusion and Future Work

We have presented dGSLM, the first model for spoken dialogue generation trained from raw audio. This model has been shown to reproduce naturalistic intelligible speech, while trained on only 2k hours of audio from telephone conversations. Informal inspection of the generated samples[2] shows that it is able to reproduce non-verbal vocalizations (laughter, backchannels). Detailed analysis of the turn-taking events show that the model is able to reproduce accurate synchronization including distribution and duration of turn-taking events like IPU, gaps, pauses and overlaps. In particular, it is able to reproduce the rather puzzling observation that inter-turn pauses tend to be on average longer than between turn gaps, suggesting the pauses alone are not a sufficient signal to indicate a change of turn.

Although the model lacks the ability to produce semantically coherent speech, it paves the way for the construction of more naturalistic human-machine dialogue systems. The logic and timing of turn-taking which has been up to now very difficult to model artificially emerges naturally from our system, while it is clearly not yet able to process speech at a deep semantic level. This indicates that a model that correctly predicts synchronization between turns can be learned from relatively a small amount of data. This is surprising given that one major paralinguistic information, intonation, was not explicitely encoded in the input (or the output) of the system. Further work incorporating pitch (Kharitonov et al., 2022b) could potentially improve the current results. Results from the cascaded system also suggest that either using larger linguistic units (like BPE) from raw audio (Borsos et al., 2023) or combining our model with text-based models would create systems which could generate more natural and meaningful conversations.

Additional Results

This part presents my supplementary experiments concerning the exploration of improving dGSLM models.

3.7 Improve dGSLM with large-scale single-channel speech dataset

As discussed, we found that even the cascaded system struggled to generate a meaningful conversation, suggesting the insufficient of 2000 hours of Fisher dataset. Therefore, a natural direction of improvement is to make use of large-scale datasets (e.g., 60K hours of Libri-light). However, most available large-scale speech datasets are single-channel, meaning that they can't produce parallel streams of units as the Fisher dataset.

Actually, the shared-weight dual tower architecture of DLM allows pre-training single tower on single-channel audio, with the cross-attention now becomes self-attention, and using the learned weights to initialize the multi-channel training on Fisher dataset. We follow this and perform the pre-training of the DLM on the Libri-light 60k dataset. Training metrics and learning curves of the models are shown in Table 3.8 and Figure 3.8.

	Duration Loss	Dataset	Unit Rate	**Fisher Valid** Edge Unit PPL↓	Edge Unit Acc↑	Duration MAE↓	Duration Acc↑	**LibriSpeech Dev** Edge Unit PPL↓	Edge Unit Acc↑	Duration MAE↓	Duration Acc↑
DLM model trained on HuBERT Fisher units											
dGSLM (DLM-5)	MAE	Fisher	50hz	4.75	54.27	1.23	58.18	22.36	31.29	0.53	61.89
DLM model trained on HuBERT Mix units											
dGSLM	MAE	Fisher	25hz	6.95	48.01	1.43	69.95	11.04	39.45	**0.28**	78.08
+Cross-Entropy Duration Loss	CE	Fisher	25hz	6.58	**49.22**	**0.83**	**71.76**	11.17	39.98	0.54	79.15
Train on LL60K	CE	LL60k	25hz	11.44	39.80	0.94	70.08	**5.76**	51.17	0.46	**81.47**
+Fine-tune on Fisher	CE	Fisher	25hz	**6.56**	49.17	**0.83**	71.74	11.67	39.10	0.54	79.37
Mixed Train on LL60K+Fisher	CE	Fisher+LL60k	25hz	7.91	45.60	0.86	71.09	**5.76**	**51.18**	0.46	81.43

Tab. 3.8: Modeling Metrics on Fisher Valid and Librispeech Dev sets for DLM models trained on HuBERT Mix units compared with the DLM-5 model which was trained on HuBERT Fisher units. The HuBERT Mix units are 25hz+km500+robust in Table 2.9. We replace the MAE loss for Duration Prediciton in dGSLM by a Cross-Entropy loss over discrete durations. We also pre-train DLM on the libri-light 60k hours dataset with a single tower architecture and self-attention, and then fine-tune on the Fisher dataset with cross-attention. We also train DLM model on a mix of mono and stereo datasets (LL60k+Fisher). All models are trained on a 100k steps, except DLM-5 where it was trained on 250k steps.

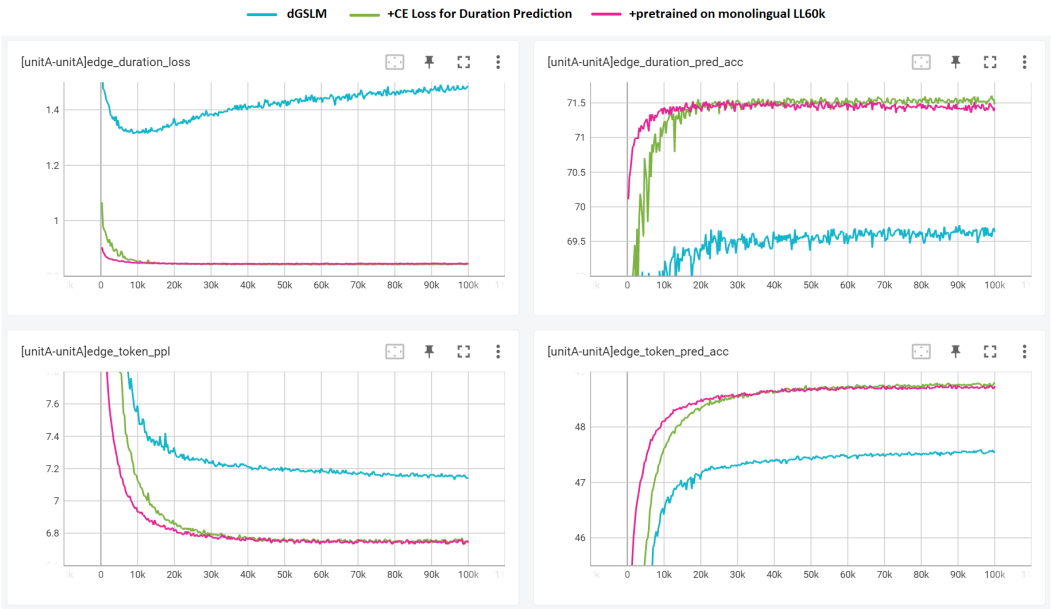


Fig. 3.8: Training Curves of DLM models on modeling metrics. All the models are trained using HuBERT Mix 25hz units on the Fisher dataset for 100K iterations. The reported metrics are evaluated on the Fisher valid dataset. The blue curve corresponds to the same dGSLM model, the green curve corresponds to dGSLM model with Cross-Entropy Duration Prediction loss instead of MAE loss, the pink curve corresponds to the model initialized from a pretrained model on the Libir-light 60K dataset.

In preliminary experiments, we found that HuBERT Fisher units, although have good resynthesis quality for Fisher dataset, perform badly on resynthesizing audio from Libri-light (this is further confirmed by the bad ABX of HuBERT Fisher on LibriSpeech, cf. Table 3.1). We then decided to use the HuBERT Mix 25hz robust units as mentioned in Section 2.6, which have been shown to have good ABX in both LibriSpeech and Fisher. We then first re-train the dGSLM model using the 25hz units. We observe that the model trained on this new units has higher training perplexity on the Fisher valid set, which comes from the difference in the frame rate of speech units as well as training steps, but has much better perplexity on the LibriSpeech dev set, even if the model doesn't see any Libri-light data during training.

Another small improvement of dGSLM model is the use of the Cross-Entropy loss over the duration prediction objective. In our experiments, we found that the duration prediction loss of dGSLM overfits very quick due to the use of the MAE loss (Figure 3.8 top-left blue curve), we then replaced the MAE loss with the CE loss, where we consider duration as discrete targets. This substantially improves not only the duration modeling of DLM, but also the content modeling with a gain in both perplexity and edge unit prediction accuracy.

Now comes to the main part, where we first pre-train the DLM model with a single tower architecture on the Libri-light 60k dataset, and then fine-tune the model on the Fisher dataset. Looking at Table 3.8, we see that the model initialized from pre-trained model doesn't perform better than the model trained from scratch, even on the LibriSpeech metrics. However, when looking at the training curves in Figure 3.8, we see indeed that the fine-tuned model learns much faster at initial iterations and eventually converges to the same points as the model from scratch, indicating that there is still a transfer between the pre-trained model on the large-scale single-channel dataset to multi-channel ones.

Finally, the DLM model also allows training on a mix of single-channel and multi-channel datasets. We make use of this and train a model on a combination of Libri-light 60k and Fisher. We see that this helps to achieve good modeling metrics on both Fisher and LibriSpeech datasets. This shows the potential of training a large-scale model on the mix of different datasets.

We did not really evaluate the models in terms of speech generation metrics because the training metrics are not much improved. The obtained results suggest that even 60k hours of Libri-light is not enough to train a good pre-trained SpeechLM for fine-tuning on Fisher, which led us to the priority of scaling and improving SpeechLM systems so that they can be later fine-tuned on dialogue datasets. This led to the efforts in Chapter 5.

4 Expressive Speech Resynthesis

Previous work focus on using speech units obtained from self-supervised speech models and resynthesizing the speech with a vocoder model trained on read or casual speech datasets, which can potentially remove all expressivity contained in the input speech. A critical problem for expressive speech generation is the lack of high-quality datasets used for training speech synthesis models. In this chapter, we are going to introduce a new open-source expressive dataset that can be used to train discrete unit-based speech synthesis models along with a benchmark on discrete expressive speech resynthesis.

This chapter presents the following paper that was published in the Proceedings of Interspeech 2023 :

Tu Anh Nguyen, Wei-Ning Hsu, Antony D'Avirro, Bowen Shi, Itai Gat, Maryam Fazel-Zarani, Tal Remez, Jade Copet, Gabriel Synnaeve, Michael Hassid, Felix Kreuk, Yossi Adi, and Emmanuel Dupoux (2023a). "Expresso: A Benchmark and Analysis of Discrete Expressive Speech Resynthesis". In: *Proc. INTERSPEECH 2023*, pp. 4823–4827

It is followed by Section 4.7 and Section 4.8, where I present my experiments considering the disentanglement of expressive speech units and the comparison of language modeling on HuBERT and Encodec units, respectively.

Statement of contribution:
In this work, I pre-processed the collected Expresso dataset, trained all the vocoders mentioned in the paper, evaluated the quality of the speech units, performed the speech resynthesis, and created the emotion classification benchmarking. The additional experiments of the chapter are performed by myself, with the help of Bokai, Maha and Sravya mostly for the expressive tokenizer evaluation.

Publication: Expresso: A Benchmark and Analysis of Discrete Expressive Speech Resynthesis

Tu Anh Nguyen[*,◇,†], **Wei-Ning Hsu**[*,◇], **Antony D'Avirro**[*,◇], **Bowen Shi**[*,◇], **Itai Gat**[◇], **Maryam Fazel-Zarani**[◇], **Tal Remez**[◇], **Jade Copet**[◇], **Gabriel Synnaeve**[◇], **Michael Hassid**[◇,♠], **Felix Kreuk**[◇], **Yossi Adi**[+,◇,♠], **Emmanuel Dupoux**[+,◇,‡]

[◇]Meta AI Research, [†]Inria, Paris,
[♠] The Hebrew University of Jerusalem, [‡]EHESS, ENS-PSL, CNRS, Paris

{ntuanh, adiyoss, dpx}@meta.com

Abstract

Recent work has shown that it is possible to resynthesize high-quality speech based, not on text, but on low bitrate discrete units that have been learned in a self-supervised fashion and can therefore capture expressive aspects of speech that are hard to transcribe (prosody, voice styles, non-verbal vocalization). The adoption of these methods is still limited by the fact that most speech synthesis datasets are read, severely limiting spontaneity and expressivity. Here, we introduce EXPRESSO, a high-quality expressive speech dataset for textless speech synthesis that includes both read speech and improvised dialogues rendered in 26 spontaneous expressive styles. We illustrate the challenges and potentials of this dataset with an *expressive resynthesis benchmark* where the task is to encode the input in low-bitrate units and resynthesize it in a target voice while preserving content and style. We evaluate resynthesis quality with automatic metrics for different self-supervised discrete encoders, and explore tradeoffs between quality, bitrate and invariance to speaker and style. All the dataset, evaluation metrics and baseline models are open source[1].

[*,+] Core contribution as first and last authors
[1] https://speechbot.github.io/expresso/

4.1 Introduction and related work

Speech synthesis has been traditionally approached as a mapping between text and speech. This has a series of limiting consequences: text is an impoverished representation of language, that does not specify many expressive dimensions: rhythm, intonation, emotion, emphasis, and often fails to encode non-verbals like laughter, cries, lip smacks, etc. As a result, speech synthesizers typically resort to standard read speech as the main target, which severely limits the expressivity of AI systems.

Everything changed with the advent of Self Supervised Learning (SSL) speech models (Chung et al., 2021; Hsu et al., 2021a), which enabled to build discrete representations for speech without needing any textual annotation (Borsos et al., 2023; Gat et al., 2023; Lakhotia et al., 2021). Because such representations can be learned from much more diverse audio than read speech (conversational, casual speech), this opens up the possibility to build more expressive systems based on SSL units instead of text. Recently, Kharitonov et al. (2023) and Wang et al. (2023a) utilize neural audio codecs (Défossez et al., 2022; Zeghidour et al., 2021) to encode speech features into codes and generate natural speech from textual input. However, these models still rely on large-scale read speech that lacks expressivity. One of the roadblocks in building such expressive systems is the lack of datasets that are sufficiently expressive and of high audio quality for learning a synthesizer. Most existing expressive datasets (e.g., EmoV, Adigwe et al., 2018) have used expressive reading, where voice actors read a (neutral) sentence in different expressions (happy, sad, etc.). This method puts the voice actors in an artificial situation, resulting in not very plausible rendering of these expressions. In this work, we also reproduced this protocol to create an expressive speech dataset but added a section based on conversational improvisation. The two actors are prompted with a situation and a character (e.g., two drivers involved in a car accident, a parent and a child, etc.) and improvise a dialogue impersonating their characters. This yields much more realistic and casual speech, with spontaneous hesitations, laughter, etc., that would be extremely hard to transcribe accurately, but, which can be in principle captured by SSL units.

Next, we illustrate the potential of this dataset by setting up the task of discrete expressive resynthesis. As in discrete resynthesis (Maimon and Adi, 2022; Polyak et al., 2021), it consists in taking audio as input, encoding it in low bitrate discrete units, and synthesizing the same content with a different target voice. In this work, we additionally include the task of preserving expressive style. We introduce

automatic metrics for content and style preservation and evaluate a HuBERT (Hsu et al., 2021a) encoder trained on a masked prediction objective followed by k-means clustering, which we compare to Encodec (Défossez et al., 2022), a compression model which acts as a high bitrate baseline. We compare different pretraining sources for the HuBERT units based on public datasets of read speech and/or spontaneous speech. Synthesis is done with a units-based HiFi-GAN vocoder (Polyak et al., 2021) conditioned on speaker or speaker and style.

4.2 The Expresso dataset

The Expresso dataset consists of 47 hours of expressive speech from 4 speakers of North American English. The dataset is divided into two main sections: an *expressive reading* section (37% of the corpus) where actors read short prompts in a parallel fashion in 7 different styles, with additional long-form and emphasis material (see 4.2.1), and an *improvised dialog* section (72% of the corpus) where pairs of actors are prompted to improvise a conversation in a fictive setting that illustrates one of 25 specified styles (see 4.2.2). In a small additional singing section, actors sing a few of their favorite songs.

The 26 different styles were chosen for their universality/recognizability (i.e. common emotions like happiness, sadness), utility for current/anticipated speech applications (i.e. whispered, enunciated speech), and to elicit the large range of possible vocalizations of the human voice (including addressing or imitating a child or an animal, and non-verbals like grunting, coughing, whistling, etc., see Table 4.1 for a full list).

Data was recorded in a professional recording studio with minimal background noise at 48kHz/24bit. The files for read speech and singing are in a mono wav format; and for the dialog section in stereo (one channel per actor), where the original flow of turn-taking is preserved.

4.2.1 Expressive reading

Seven of the styles (confused, default, enunciated, happy, laughing, sad, whisper) were applied in a parallel fashion to the same set of prompts, so content did not necessarily reflect the emotion being conveyed, and we relied on the actor's expertise to convey the desired style. Written instructions were delivered for each style

Tab. 4.1: EXPRESSO's expressive styles. * singing is the only improvised style that is not in dialogue format.

Style	Read (min)	Improvised (min)	total hours
angry	-	82	1.4
animal	-	27	0.4
animal_directed	-	32	0.5
awe	-	92	1.5
bored	-	92	1.5
calm	-	93	1.6
child	-	28	0.4
child_directed	-	38	0.6
confused	94	66	2.7
default	133	158	4.9
desire	-	92	1.5
disgusted	-	118	2.0
enunciated	116	62	3.0
fast	-	98	1.6
fearful	-	98	1.6
happy	74	92	2.8
laughing	94	103	3.3
narration	21	76	1.6
non_verbal	-	32	0.5
projected	-	94	1.6
sad	81	101	3.0
sarcastic	-	106	1.8
singing*	-	4	.07
sleepy	-	93	1.5
sympathetic	-	100	1.7
whisper	79	86	2.8

describing the upper and lower performative bounds of the style, some including video examples.

Aside from a small corpus of shared essential lines (greetings, common phrases, numbers, letters), each speaker had a unique script in order to maximize linguistic diversity across the entire dataset. In total, the written corpus contains roughly 21,000 words over 2,400 unique lines. Material was scraped from open-source datasets like Wikipedia and commissioned datasets containing voice-assistant style utterances and then proofread and scrubbed for PII. Although not specifically balanced for phonetic coverage, the corpus was tuned both overall and per-speaker for a desired ratio of statements, questions, exclamations, jokes, etc.

Contrastive emphasis. We enclose certain words/spans in asterisks to denote emphasis, designed to convey contrastive focus in the reading of an utterance. Actors were trained to read this syntax with the desired prosodic effect. These occur in isolation throughout the read-speech corpus, but also in a subset of each

speaker's lines labeled "emphasis" where the same line is repeated 2 to 4 times with contrastive emphasis placed on different words/spans.

Long-form material. To capture longer-range prosodic dependencies, each speaker's script contains one news article (read in the "default" style) and one long-form narrative piece (read in the "narration" style), roughly totalling 100 lines per speaker.

4.2.2 Improvised dialogs section

Dialogs were elicited via a set of situational prompts designed to evoke the desired styles or emotions. Some prompts resemble voice-application domains such as reporting the weather, navigation, information retrieval, while others are more open-ended scenarios; some of them were proposed by the actors.

Tradeoffs were made to capture usable data while preserving the feel of natural conversation, i.e. actors were recorded in separate booths but watching each other over video conference. A small number of dialogs ($<$10) were interrupted by the studio director to provide notes. These dialogs were edited to remove the pause and maintain a more natural flow.

4.2.3 Singing section

Each speaker recorded several versions of popular nursery rhymes and public domain songs. The original recording had more data (93 minutes total) but could not be shared because the songs turned out not to be in the public domain.

4.2.4 Data preparation

The raw dataset contains mostly pre-cut segments of 3-4 seconds for the read section, except for long-form ones, and long waveforms ranging from 2 to 10 minutes for dialogs section. For the purpose of speech synthesis, we cut long files into segments of 15 seconds. We split the dataset into train/dev/test subsets such that each speaker-style contains roughly 60s in dev/test splits, resulting in 1.5 hours of speech in each subset.

4.3 Method

Additional datasets. We refer to the read section of EXPRESSO as Exp-R and the improvised section as Exp-I. To train the units and vocoder, and to evaluate the results, we use additional open source datasets: LJspeech (Ito and Johnson, 2017) (LJ), VCTK (Veaux et al., 2017), Librispeech dev-other (Panayotov et al., 2015) (LS), Fisher (Cieri et al., 2004) and EmoV-DB (EmoV) (Adigwe et al., 2018).

Evaluation metrics For evaluating the discrete units, we compute their bitrate, ABX discrimination (the probability that the DTW distance between minimally different triphones like /bit/ versus /bet/ are more distant to one another than two instances of the same triphone, Schatz et al., 2013) both on the 1-hot representations (as in Nguyen et al., 2020b), and using the dense embedding corresponding to the centroid of the units, and PNMI (phone-normalized mutual information between units z and ground truth phonemes y: $I(y;z)/H(y)$ as in Hsu et al., 2021a).

For evaluating the quality of resynthesized speech, we build automatic metrics for the preservation of content, pitch, and expressivity. As in Polyak et al. (2021), content preservation is evaluated by running a publicly available Automatic Speech Recognition (ASR) model (Xu et al., 2021b) on the resynthesized sentence and computing the Word Error Rate (WER) relative to the transcription of the input sentence[2]. We run this on in-domain inputs (LJ, VCTK and Exp-R) and out-of-domain inputs (LS and Fisher). Pitch preservation is evaluated by computing F0 Frame Error (FFE), which measures the percentage of frames with a deviation of more than 20% in pitch value between the input and resynthesized output. Expressivity preservation is computed by training an expressive style classifier[3] on the train set of EXPRESSO and applying it to resynthesized versions of its dev set. These classifiers are also run on the original data for comparison.

4.4 Models

4.4.1 Unit encoding

For unit encoding, we compared three models: two HuBERT-based, one Encodec-based. The HuBERT models use the same architecture (HuBERT base with 12

[2] `https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_vox_960h_pl.pt`

[3] We fine-tune the wav2vec2 base model (Baevski et al., 2020c) on a 26 style classes audio classification task (as in the SUPERB benchmark, Yang et al., 2021) using Huggingface transformers library (Wolf et al., 2020).

Tab. 4.2: Encoder and Tokenizer used for our discrete units. Bitrate is $\log_2$(codebook size) $\times$ n units per sec in BPS. Mean within- and across-speaker ABX discrimination scores, resp., on 1-hot vectors and units' centroids. PNMI is mutual information between units and phonemes. For Encodec RVQ8, we concatenate multiple codebooks for ABX. PNMI is not available for this tokenizer.

Model	Tokenizer	BPS	ABX 1-hot (↓)		ABX-centr. (↓)		PNMI (↑)	
			LS	Fisher	LS	Fisher	LS	Fisher
HuBERT (LS960)	KM500 (LS960)	450	**8.98**	15.45	5.28	10.84	67.57	48.42
	KM2000 (Expr)	550	11.00	17.81	**4.32**	9.07	73.23	56.33
HuBERT (Mix1)	KM2000 (Mix1)	550	10.27	**14.96**	4.92	8.93	72.36	55.77
	KM2000 (Expr)	550	10.50	15.27	4.39	**8.04**	**74.37**	**59.14**
Encodec	RVQ1	500	45.47	44.29	26.10	29.57	21.42	13.17
	RVQ8	4000	41.38	40.50	20.20	25.60	-	-

Tranformer layers), but are trained on different corpora. HuBERT-LS960 was trained on LibriSpeech 960 as in Lakhotia et al. (2021). We used the available model in textless-lib (Kharitonov et al., 2022a). We use HuBERT-Mix1 from Hsu et al. (2023b), which was trained with a more varied mixture of datasets: an 8 language subset of VoxPopuli (Wang et al., 2021) (167K h), Common Voice (Ardila et al., 2020) (4K h) and Multilingual LibriSpeech (MLS) (Pratap et al., 2020) (50K h), totalling 221K hours. For quantization, we trained k-means models on HuBERT features either on a subset of HuBERT pre-training dataset or on EXPRESSO, with k=500 on LS960 or k=2000 on other datasets. The Encodec model is from Défossez et al. (2022), we used two models with 1 and 8 codebooks of cardinality 1024, which were trained on VoxPopuli400k English, People's Speech (Galvez et al., 2021), LibriSpeech 960, LibriLight (Kahn et al., 2020) and Spotify (Clifton et al., 2020). Training hyperparameters were identical to Défossez et al. (2022) except for having no audio normalization and using zero padding instead of reflect.

4.4.2 Vocoder

For HuBERT units, we produce the waveform using HiFiGAN, which we train on the units presented above. For each set of units, we train either on LJ and VCTK, or on LJ, VCTK and EXPRESSO. The vocoder is either conditioned on speaker ID (using a look-up table), or on speaker ID and expression ID (also using a look-up table). We distinguish the read and improvised versions of the expressions, yielding a total of 34 expressions. For Encodec units, we used the Encodec decoder to directly produce the waveform, and compared systems with 1 and 8 codebooks.

Tab. 4.3: Content preservation evaluation: WERs (%) of speech resynthesized by our models. We bold the best HuBERT or best Encodec model within each column. We denote speaker conditioning as S, and speaker with expression conditioning as S_E. Results are reported for Expresso (E), VCTK (V) and Fisher (Fish).

				In Domain Source					Out of Domain Source							
				Same Speaker			Swapped Speaker									
			Src.	LJ	V	E	V	E	LS	Fish	LS	Fish	LS	Fish	LS	Fish
			Tgt.	LJ	V	E	V	E	LJ	LJ	V	V	E	E	Orig Voices	
Model	Tokenizer	Data	Cond.	WER												
Original audio	_	_	_	*2.04*	*1.74*	*14.76*	_	_	_	_	_	_	_	_	*3.55*	*30.26*
HuBERT (LS960)	KM500 (LS960)	LJ+V	S	3.12	6.85	_	7.20	_	11.56	50.13	10.83	48.26	_	_	_	_
		E+LJ+V	S	3.22	7.19	24.21	6.80	24.34	11.61	49.24	10.42	46.63	10.93	47.18	_	_
		E+LJ+V	S_E	3.65	6.76	23.65	7.79	24.82	12.00	50.49	10.75	47.72	10.57	46.57	_	_
	KM2000 (E)	LJ+V	S	2.98	7.15	_	7.09	_	10.95	47.44	9.98	47.06	_	_	_	_
		E+LJ+V	S	3.41	7.09	21.64	7.01	22.80	10.80	46.90	10.20	45.79	10.61	46.77	_	_
		E+LJ+V	S_E	2.83	6.48	22.35	6.56	21.92	10.34	45.93	9.72	45.08	9.52	43.13	_	_
HuBERT (Mix1)	KM2000 (Mix1)	LJ+V	S	**2.60**	6.98	_	7.60	_	9.60	41.78	8.34	40.53	_	_	_	_
		E+LJ+V	S	2.80	6.84	21.25	7.20	22.52	9.17	40.61	8.38	38.91	9.76	42.87	_	_
		E+LJ+V	S_E	2.85	7.17	20.36	7.33	20.81	9.50	41.09	8.92	40.82	8.39	38.47	_	_
	KM2000 (E)	LJ+VCTK	S	2.77	5.60	_	5.89	_	9.48	41.42	8.39	40.81	_	_	_	_
		E+LJ+V	S	2.95	**4.85**	20.64	**5.07**	21.01	**9.04**	39.62	7.91	38.45	8.46	39.84	_	_
		E+LJ+V	S_E	3.05	5.48	**19.52**	5.59	**20.27**	9.20	**38.79**	**7.75**	**37.48**	**8.00**	**36.67**	_	_
Encodec (RVQ-1)		_	None	5.52	17.46	34.36	_	_	_	_	_	_	_	_	18.88	60.68
Encodec (RVQ-8)		_	None	**2.20**	**2.52**	**16.85**	_	_	_	_	_	_	_	_	**4.62**	**35.64**

4.5 Results

Table 4.2 shows the phonetic quality metrics across different SSL units. The HuBERT encoders trained on the larger and noisier corpus (Mix1) tend to have overall better results than when trained on LS960 only, especially when tested on Fisher. The ABX-centroid and PNMI metrics gave better results when k-means clustering was run on EXPRESSO (a small high quality, high diversity dataset) than on the large dataset used to train HuBERT itself. This was not the case, however with the ABX 1-hot metric, so further study is necessary to confirm this result. The Encodec units gave poor results, which is not surprising given that Encodec units are generic representations trained for audio compression that also encodes non-phonetic variations whereas HuBERT units are trained with a masked language modeling objective.

Table 4.3 shows the result of content preservation for the resynthesis task (WER), distinguishing the case where the input sentences are drawn from the same distribution as the vocoder training sets (In-domain Source: LJ, VCTK and Exp-R) and when the input sentences are from different datasets (Out-of-domain Source: LS and Fisher). For in-domain, we further distinguish the cases where the input voice is the same as the target voice (Same Speaker) and when the target voice is randomly sampled from the same training set (Swapped Speaker; not applicable to LJ speech). We find that swapping speakers costs on average only a small decrement in WER (3% relative), suggesting that the units are well (although not totally) disentangled from speaker information. On average, the HuBERT units trained on Mix1 give better performances (about 10% relative) than units trained on LS960, a result consistent with the phonetic quality metrics. As for the vocoder, the training voices and conditioning (either speaker alone or speaker+style) did not give systematic results.

Tab. 4.4: **Expressive style classification accuracy** using a pre-trained emotion classification model. We denote speaker conditioning as S, and speaker with expression conditioning as S_E. Results are reported for Expresso (E) and VCTK (V).

				Same		Swapped		Zero-shot				Out of dom.
			Src.	E_R	E_I	E_R	E_I	E_R	E_I	E_R	E_I	EmoV
			Tgt.	E	E	E	E	LJ	LJ	V	V	E
Model	Tokenizer	Data	Cond.					Accuracy				
Original Audio	_	_	_	*92.47*	*75.69*	_	_	_	_	_	_	*27.46*
HuBERT (LS960)	KM500 (LS960)	LJ+V	S	-	-	_	_	13.36	9.14	26.42	12.43	
		E+LJ+V	S	33.18	14.99	29.80	13.53	8.45	10.79	28.26	10.97	11.56
		E+LJ+V	S_E	**81.57**	58.68	81.26	56.12	23.96	36.93	**56.07**	28.15	43.35
	KM2000 (E)	LJ+V	S	-	-	_	_	5.22	10.42	27.04	11.70	
		E+LJ+V	S	39.17	23.95	33.95	19.38	11.67	12.07	27.34	14.63	9.25
		E+LJ+V	S_E	78.34	**62.16**	76.96	54.11	22.12	39.31	55.76	**32.54**	46.24
HuBERT (Mix1)	KM2000 (Mix1)	LJ+V	S	-	-	_	_	7.99	9.32	27.50	11.52	
		E+LJ+V	S	25.81	17.73	28.57	15.72	5.53	8.96	27.65	11.33	11.27
		E+LJ+V	S_E	78.80	61.06	**81.41**	**58.14**	31.64	**40.77**	40.86	31.63	48.27
	KM2000 (E)	LJ+V	S	-	-	_	_	7.68	10.24	27.80	12.07	
		E+LJ+V	S	37.02	16.82	35.33	16.09	5.99	9.69	26.73	11.52	14.45
		E+LJ+V	S_E	72.81	**62.16**	73.12	55.76	**31.18**	39.85	55.61	28.52	**49.71**
Encodec (RVQ-1)		_	None	57.76	44.42	_	_	_	_	_	_	22.25
Encodec (RVQ-8)		_	None	**78.65**	**64.53**	_	_	_	_	_	_	**26.88**

Overall, the best HuBERT results on the same speaker showed a drop in performance compared to the original audio files (between 30% relative to more than double the error), and compared to the Encodec-8 units. Regarding out-of-domain resynthesis, the HuBERT-Mix1 units again generally outperform the HuBERT-LS960 units (19% relative). In addition, the tokenizer trained on EXPRESSO tend to be better by a small margin (3% relative). The best resynthesis models suffer from a large drop in WER compared to original audio and Encodec-8 for LS (twice the errors) but a much smaller drop for Fisher. Encodec-1 consistently underperform HuBERT-based resynthesis, indicating that one Encodec codebook of size 1024 is not enough to fully capture linguistic content.

Tables 4.4 and 4.5 show the results on style and pitch preservation, respectively; these experiments are exclusively ran on EXPRESSO inputs (in-domain) or EmoV inputs (out-of-domain). We group the in-domain results in 3 conditions. The first two are similar to the same-speaker and swapped speaker conditions discussed above. Unsurprisingly, for both style and pitch, the results are uniformly better when the vocoder was conditioned on speaker+expression than in speaker alone. Note, though, that even without expression conditioning, the style classification score is much higher than the chance level (3.8%), suggesting that style is partly transmitted through the units. The results on swapped speakers are slightly worse than on same speaker (cost around 10% relative accuracy on style and 20% rel. on pitch error). Globally, the style scores of Encodec-8 are on par or better than the best HuBERT resynthesis models, but much better for pitch preservation (6-fold). Next, we explored whether expressivity could be transferred to voices that were only trained in the default read speech style (VCTK and LJ). This change cuts the style score in half, still remaining way above chance, and better than the EXPRESSO voices

Tab. 4.5: F0 Frame Error (FFE). Bold best absolute scores. We denote speaker conditioning as S, and speaker with expression conditioning as S_E. Results are reported for Expresso (E), Expresso Read (E_R), Expresso Improvised (E_I), LJ, VCTK (V), and EMOV.

				Same		Swapped		Zero-shot				OOD
			Src.	E_R	E_I	E_R	E_I	E_R	E_I	E_R	E_I	EMOV
			Tgt.	E	E	E	E	LJ	LJ	V	V	E
Model	Tokenizer	Data	Cond.	FFE								
HuBERT (LS960)	KM500 (LS960)	E+LJ+V	S	0.31± 0.10	0.33± 0.12	0.37± 0.12	0.38± 0.13	0.34± 0.10	0.35± 0.12	0.38± 0.11	0.38± 0.14	0.27± 0.08
		E+LJ+V	S_E	0.27± 0.13	0.30± 0.13	0.34± 0.16	0.36± 0.15	0.33± 0.15	0.35± 0.14	0.34± 0.16	0.36± 0.15	**0.25± 0.09**
	KM2000 (E)	E+LJ+V	S	0.31± 0.12	0.33± 0.13	0.36± 0.13	0.36± 0.13	0.35± 0.09	0.35± 0.11	0.38± 0.11	0.38± 0.14	0.26± 0.09
		E+LJ+V	S_E	**0.26± 0.13**	**0.29± 0.13**	0.34± 0.16	0.36± 0.15	**0.31± 0.14**	**0.33± 0.13**	**0.33± 0.16**	**0.35± 0.15**	0.26± 0.09
HuBERT (Mix1)	KM2000(Mix1)	E+LJ+V	S	0.32± 0.11	0.33± 0.13	0.38± 0.12	0.37± 0.13	0.36± 0.09	0.35± 0.11	0.38± 0.12	0.38± 0.14	0.26± 0.09
		E+LJ+V	S_E	0.28± 0.14	0.29± 0.13	0.34± 0.16	0.36± 0.15	0.32± 0.15	0.34± 0.14	0.34± 0.16	0.36± 0.15	0.27± 0.09
	KM2000 (E)	E+LJ+V	S	0.31± 0.10	0.32± 0.12	0.37± 0.12	0.37± 0.13	0.35± 0.09	0.35± 0.11	0.38± 0.11	0.37± 0.14	0.26± 0.08
		E+LJ+V	S_E	0.27± 0.13	0.30± 0.13	**0.34± 0.16**	**0.36± 0.14**	0.32± 0.14	0.34± 0.14	0.34± 0.16	0.36± 0.15	0.25± 0.09
Encodec (RVQ-1)		_	None	0.08± 0.04	0.11± 0.07	_	_	_	_	_	_	0.09± 0.06
Encodec (RVQ-8)		_	None	**0.04± 0.02**	**0.05± 0.03**	_	_	_	_	_	_	**0.04± 0.02**

not conditioned by expression, showing that expressive styles can, to a certain extent, generalize to untrained voices in a zero-shot fashion. Finally, we tested expressive resynthesis could be applied out-of-domain, using input data from EmoV. This dataset uses 5 expressions (neutral, amused, angry, sleepy, disgust) which we mapped based on the description to EXPRESSO's neutral, laughing, angry, sleepy and disgusted. The performance of the classifier was much lower than in the in-domain case, and unexpectedly higher for resynthesis than the original file. Inspection of the errors pattern showed some systematic style confusions across datasets. For instance, original EmoV voices in angry style were classified as "projected" by our classifier, but as "angry" once resynthetized with the "angry" conditioning. This suggests discrepancies in style rendering across the two datasets for identical labels (e.g., anger rendered as shouting in one versus cold rage in the other). This also suggests that the style-conditioned vocoder can to a certain extent remap input styles to different ones (confirmed by a style swapping experiment not reported in the table). More research and expressive datasets are needed to develop a dataset-independent and speaker-independent expressive style classifiers. Despite these limitations, the results were congruent with in the in-domain case, with better than chance performance and improved performance with expression-conditioned vocoders.

4.6 Conclusion

We presented a new dataset for expressive discrete resynthesis, and analysed content and style preservation for several baseline discrete SSL models. We showed that Encodec systems which are designed for general audio compression are generally better for resynthesis, although they lack the controllability in output voices and style made possible by the fact that HuBERT units are disentangled from speaker identity

and (partially) from expressivity. Further work is needed to improve HuBERT-based expressive resynthesis, and reach the quality of Encodec units, while retaining controllability. In particular, improved performance on pitch preservation could be obtained by conditioning the vocoder on discrete pitch units, as in Polyak et al. (2021).

Additional Results

This part presents my supplementary experiments concerning the exploration of better expressive speech units as well as comparing language modeling performances of HuBERT and Encodec units.

4.7 Disentangled Expressive Speech Units

We see that HuBERT units only don't capture pitch information from the speech, and we don't always have the ground truth style of the audio to condition the HifiGAN model. Therefore, it is natural to think of generating pitch and style units capturing the intonation and expressivity in the speech and then condition the HifiGAN model on these pitch and style units in addition to the HuBERT units.

Following Polyak et al. (2021), we trained a VQ-VAE (Oord et al., 2017b) model on the F0 of the speech to obtain pitch units capturing the intonation in the speech. For the style units, we extract Speechprop's features (Duquenne et al., 2023), which is supposed to contain the expressivity style of the speech. We further fine-tune the features to predict Expresso styles in order to reinforce the expressivity captured in the features as well as remove unnecessary speaker information. We finally performed a k-means clustering on the features to obtain the style units. Based on the assumption that pitch and style change at a lower rate than the phonetic content of speech, we take the pitch units and style units with a lower frame rate (12.5hz for pitch and 1hz for style), with the hope that more compressed units will potentially be beneficial for language models. We then trained a HifiGAN model conditioned on the 3 types of units: HuBERT, pitch and style and evaluated the resynthesized speech using the Expresso benchmark. The results are shown in Table 4.6.

We can see effectively that using additional pitch and style units helps to improve the expressivity and pitch preservation metrics. On content preservation, the model shows good results, largely surpassing Encodec with 1 codebook. We see that using style units indeed allows capturing the expressive style from the input speech without relying on the ground truth style, reaching the performances of Encodec with 1 codebook. Finally, using the pitch units effectively reduce substantially the F0 Frame Error compared with using only HuBERT units. We see, in general, that we have succeeded in extracting disentangled speech units with good expressive resynthesis quality while reducing the bitrate of previous models by a large margin. This later

Model	Metrics	Bitrate BPS	Content Word Error Rate (WER)↓			Expressive Style Classification Accuracy↑			Pitch F0 Frame Error (FFE)↓		
			E. Read	LS	Fisher	E. Read	E. Imp.	EmoV	E. Read	E. Imp.	EmoV
Original Audio		-	14.76	3.55	30.26	92.47	75.69	27.46	-	-	-
Expresso models											
HuBERT + HifiGAN		550	20.64	8.46	39.84	37.02	16.62	14.45	0.31	0.32	0.26
HuBERT + HifiGAN cond. on GT Style		550	19.52	8.00	36.67	72.81	62.16	**49.71**	0.27	0.30	0.25
Encodec (RVQ=1)		500	34.36	18.88	60.68	57.76	44.42	22.25	0.08	0.11	0.09
Encodec (RVQ=8)		4000	**16.85**	**4.62**	**35.64**	**78.65**	**64.53**	26.88	**0.04**	**0.05**	**0.04**
HuBERT Mix 25hz Tokenizers											
HuBERT + HifiGAN		225	22.90	11.66	**35.64**	28.25	19.78	13.29	0.41	0.43	0.36
(HuBERT, Pitch, Style) + HifiGAN		307	**22.35**	**10.60**	36.58	**56.02**	**47.66**	**20.52**	**0.16**	**0.17**	**0.16**

Tab. 4.6: Expressive Speech Resynthesis Evaluation with Disentangled Speech Units. Performances of resynthesized speech using HifiGAN model trained on Expresso dataset using either HuBERT (robust 25hz with codebook size of 501) units only or HuBERT, Pitch and Style units compared with HuBERT (Mix1+km2000E) + HifiGAN (with and without conditioning on the Ground Truth Style) and Encodec (with 1 and 8 codebooks) systems of Expresso. The resynthesis is done with the same input speaker for Expresso subsets and with random Expresso speaker for other datasets. The bitrate is bit-per-second (BPS) computed as $log_2(\textit{codebook size}) \times \textit{n tokens per second}$. The Pitch units are obtained by training a VQ-VAE model on the F0 extracted from the speech and have a vocab of 64 and a frame rate of 12.5hz. The Style units are obtained by clustering the features extracted from Speechprop (Duquenne et al., 2023) and have a vocab of 100 and a frame rate of 1hz.

became the speech tokenizer for the SPIRIT-LM-EXPRESSIVE model which will be introduced in Chapter 5.

4.8 Comparison of HuBERT and Encodec on Spoken Language Modeling

Encodec units show excellent results in audio compression and have recently been widely used for Speech- and Audio-LMs (Borsos et al., 2023; Kreuk et al., 2023; Wang et al., 2023a). However, we have seen in this work that Encodec units don't have good phonetic quality. This suggests that the language models trained on Encodec will have poor results compared with HuBERT units.

To confirm this hypothesis, we trained language models on Encodec and HuBERT units and evaluated the model performances on spoken language modeling metrics. Following Hassid et al. (2023), we initialized the SpeechLMs with the LLAMA 1.5B model to quicken the training and trained the models on 140k hours of speech datasets, including Librilight, Spotify, and People's Speech, for 30k steps. The HuBERT units are deduplicated HuBERT Mix 25hz mentioned in Section 2.6, and

the Encodec units are the Encodec with 1 codebook used in Expresso. The results are illustrated in Figure 4.1.

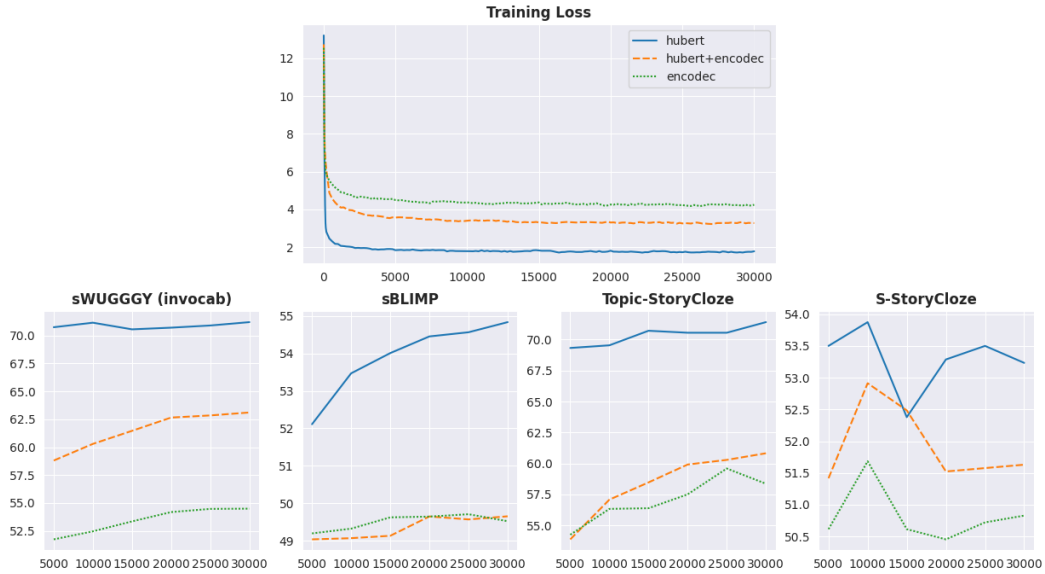


Fig. 4.1: Comparison of SpeechLMs trained on HuBERT vs Encodec units. The language models are initialized from the TextLM LLAMA 1.5B, and are trained for 30k iterations on Librilight, Spotify and People's Speech datasets, totaling 140k hours of speech. For HuBERT, we used the deduplicated HuBERT Mix 25hz units. For Encodec we used the Encodec with 1 codebook from Expresso. We additionally trained a model on the mix of HuBERT and Encodec units, where the units are interleaved and sorted with time. We plot the training loss (top) as well as evaluation metrics (bottom) of the models during training. The reported evaluation scores are calculated with *normalized log-likelihood* of speech stimulus (divided by the number of tokens).

We see that the model trained on HuBERT units has better loss than the Encodec one, which may be due to the frame rate and the codebook size of the units. However, when looking at the evaluation metrics, we see a clear gap between the two model. The Encodec LM seems to perform near chance at almost all metrics, lagging behind HuBERT LM by a large margin. We further tried to train a language model on a mix of HuBERT and Encodec units (the two kinds of units are interleaved and sorted by time stamps) and interestingly found that this model performs in-between HuBERT and Encodec on most tasks (except sBLIMP), where the performances are almost at chances. This indicates that the semantic HuBERT units acutually *guide* the Encodec units to perform better in spoken language modeling tasks, and that the frame rate is not the issue for Encodec in this case.

These results further confirmed that Encodec units, while being excellent at capturing acoustic information from the speech, perform poorly for SpeechLMs, and that HuBERT units are still desirable for SpeechLMs to learn semantic information in

the speech. It is also the reason why AudioLM (Borsos et al., 2023) employed a cascaded multi-stage language modeling scheme, where they first train a language model on semantic units (w2v-BERT), then train other language models to translate semantic units to fine-grain acoustic units (SoundStream). However, doing this will possibly remove all acoustic information in the semantic modeling stage. We will see in the next chapter that by using disentangled speech units presented in the previous section, the language model can also capture expressive information while generating semantic content appropriately.

5 Text+Speech Language Modeling

In a prior work (Hassid et al., 2023), we found that SpeechLMs benefit from TextLMs. However, we didn't focus on continuing training on text, and therefore, the trained model loses its capacity in text, which is undesirable. In this chapter, we are going to combine "textless" with text. We will analyze the best way to combine text and speech modalities in one single language model so that it can benefit from both textual semantics and speech expressivity.

This chapter presents the following paper that was published in arxiv, it is a preprint version of a work in progress :

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R. Costa-jussa, Maha Elbayad, Sravya Popuri, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, Itai Gat, Gabriel Synnaeve, Juan Pino, Benoit Sagot, and Emmanuel Dupoux (2024). *SpiRit-LM: Interleaved Spoken and Written Language Model*. arXiv: 2402.05755 [cs.CL]

Statement of contribution:

I initiated the Speech-Text LLM project, pre-processed training speech and speech-text datasets as well as "zero-shot" evaluation datasets, implemented the interleaving scheme, and trained most ablation models. I then experimented with pitch and style units and also trained the expressive models.

Publication: SPIRIT-LM: Interleaved Spoken and Written Language Model

Tu Anh Nguyen[a,+,†], **Benjamin Muller**[a,+], **Bokai Yu**[a,+], **Marta R. Costa-jussa**[b,+], **Maha Elbayad**[b,+], **Sravya Popuri**[b,+], **Paul-Ambroise Duquenne**[b,+,†], **Robin Algayres**[b,‡], **Ruslan Mavlyutov**[b,+], **Itai Gat**[b,+], **Gabriel Synnaeve**[c,+], **Juan Pino**[c,+], **Benoît Sagot**[c,†], **Emmanuel Dupoux**[c,+,‡]

[+] Meta AI, [†] Inria, Paris, [‡] EHESS, ENS-PSL, CNRS, Paris

{ntuanh, benjaminmuller, bokai, dpx}@meta.com

Abstract

We introduce SPIRIT-LM, a foundation multimodal language model that freely mixes text and speech. Our model is based on a pretrained text language model that we extend to the speech modality by continuously training it on text and speech units. Speech and text sequences are concatenated as a single set of tokens, and trained with a word-level *interleaving* method using a small automatically-curated speech-text parallel corpus. SPIRIT-LM comes in two versions: a BASE version that uses speech semantic units and an EXPRESSIVE version that models expressivity using pitch and style units in addition to the semantic units. For both versions, the text is encoded with subword BPE tokens. The resulting model displays both the semantic abilities of text models and the expressive abilities of speech models. Additionally, we demonstrate that SPIRIT-LM is able to learn new tasks in a few-shot fashion across modalities (i.e. ASR, TTS, Speech Classification)[1].

5.1 Introduction

Prompting Large Language Models (LLMs) has become a standard in Natural Language Processing (NLP) since the release of GPT-3 (Brown et al., 2020). Scaling language models to billions of parameters with massive datasets helps to achieve general-purpose language understanding and generation. Additionally, large-scale language models can solve new tasks by providing the model with a few examples

[a,b,c] Equally contributed as co-first, co-second and co-last authors, resp.

[1] Generation samples can be found at: https://speechbot.github.io/spiritlm

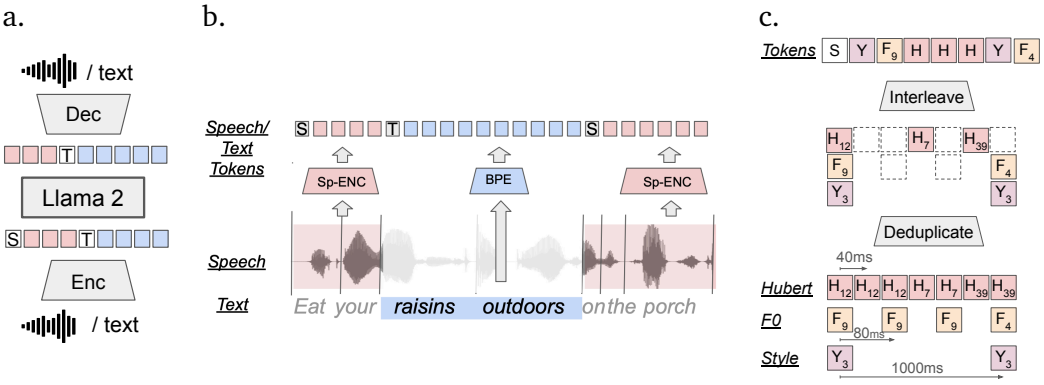


Fig. 5.1: **a. The SPIRIT-LM architecture.** A language model trained with next token prediction; tokens are derived from speech or text with an encoder, and rendered back in their original modality with a decoder. SPIRIT-LM models are trained on a mix of text-only sequences, speech-only sequences, and *interleaved* speech-text sequences. **b. Speech-text interleaving scheme.** Speech is encoded into tokens (pink) using clusterized speech units (Hubert, Pitch, or Style tokens), and text (blue) using BPE. We use special tokens [TEXT] to prefix text and [SPEECH] for speech tokens. During training, a change of modality is randomly triggered at word boundaries in aligned speech-text corpora. Speech tokens are deduplicated and interleaved with text tokens at the modality change boundary. **c. Expressive Speech tokens.** For SPIRIT-LM-EXPRESSIVE, pitch tokens and style tokens are interleaved after deduplication.

through in-context few-shot learning. Since then, a number of LLMs have been developed (Chowdhery et al., 2022; Hoffmann et al., 2022; Touvron et al., 2023a; Zhang et al., 2022). Notably, LLaMA (Touvron et al., 2023a) showed that smaller LLMs can achieve very good performance when training longer on more data using optimal-compute scaling laws (Kaplan et al., 2020), making LLMs more accessible for NLP research.

Speech Language Models (SpeechLMs), i.e. language models trained directly on speech, have been introduced (Algayres et al., 2023; Borsos et al., 2023; Lakhotia et al., 2021) and have recently become an active field of research (Hassid et al., 2023; Nguyen et al., 2023b; Rubenstein et al., 2023; Wang et al., 2023a). These models are either trained on speech-only datasets or datasets of specific tasks, e.g. Text-To-Speech (TTS), Automatic Speech Recognition (ASR), or Speech Translation, making the LMs focus on certain modality or tasks potentially loosing their generalization capabilities.

Given the increasing quality of text-only LLMs (Brown et al., 2020; Touvron et al., 2023b), one successful approach to generate speech has been to build pipelines that first transcribe input speech with ASR, then generate text using a text-only LLM and finally synthesize the generated text into speech with TTS. However, with

Inference	SPIRIT-LM generations	
	Prompt	**Generation**
	SPIRIT-LM-BASE	
S→S	[SPEECH][Hu34][Hu301][Hu280]..[Hu34] 🔊 *a b c d e*	[Hu28][Hu41][Hu123]..[Hu254] 🔊 *f g h i j k l m n o p q r c s t u v*
T→S	[TEXT]The largest country in the world is	[SPEECH][Hu34][Hu20][Hu340]..[Hu489] 🔊 *Russia. It has about 150 millions inhabitants..*
T→S	[TEXT]Here's a story about a llama that can speak:	[SPEECH][Hu12][Hu41][Hu123]..[Hu254] 🔊 *This little llama had a friend named dobby...*
S→T	[SPEECH][Hu34][Hu71][Hu405]..[Hu34] 🔊 *Yellowstone national park is an american national park located in*	[TEXT] the northwest corner of Wyoming. It is located in the Greater Yellowstone area...
S→T	[SPEECH][Hu34][Hu301][Hu280]..[Hu34] 🔊 *one two three four five*	[TEXT] 6 7 8 9 10...
	SPIRIT-LM-EXPRESSIVE	
S →T	[SPEECH][St3][Pi0][Hu34][Hu103]..[Hu22] 🔊 *Are you really going to do that <Angry>*	[TEXT] he said in a voice that was almost a scream i'm afraid
S →T	[SPEECH][St5][Pi5][Hu34][Hu409]..[Hu24] 🔊 *Are you really going to do that <Disbelief>*	[TEXT] she said turning her head quickly and putting out her hand for the glasses
T→S	[TEXT]I am so deeply saddened	[SPEECH][Hu34][St2][Pi9][Hu371][Hu20]..[Hu89] 🔊 *...this moment is very very hard to me... <Sad>*
T→S	[TEXT]Your actions have made me incredibly angry	[SPEECH][Hu37][St1][Pi3][Hu38][Hu111]..[Hu98] 🔊 *So what you think you could talk about it to me <Angry>*

Tab. 5.1: SPIRIT-LM generations with text (T) or speech (S) prompt and elicited to generate text (marked with special token [TEXT]) or speech (marked with special token [SPEECH]). We report the transcripted speech examples under the speech sequence indicated with 🔊 and < > (e.g., <Angry>) is appended when the speech is presented with the associated emotion. SPIRIT-LM models are Llama-2 7B models (Touvron et al., 2023a) fine-tuned with text (BPE) and speech tokens where Hubert token (cf.§ 5.3.1) is denoted as [Hu], while [Pi] and [St], used exclusively in SPIRIT-LM-EXPRESSIVE (cf.§ 5.3.2), represent the Pitch token and the Style token, respectively. SPIRIT-LM models enable semantically consistent multimodal generations, few-shot learning for text and speech tasks, cross-modal inference (text to speech and speech to text) and expressive generations. The samples can be found at our demo webpage[1].

such pipelines, modeling and generating expressive speech is constrained out of the language model, leading to poor generation from an expressive point of view.

In this work, we aim to combine the generative abilities and pretrained knowledge of text LLMs with the expressive capacities of speech-language models. We show that LLMs trained on interleaved speech and text can learn speech and text cross-modally and are able to generate language content in either modality. We evaluate the models with comprehension tasks in both speech and text, and extend few-shot prompting to speech-text tasks such as ASR, TTS or Speech Classification. We further extend the semantic speech tokens with expressive tokens that capture the pitch and style of the speech, and evaluate the models with newly introduced sentiment modeling tasks. Our contributions are the following:

- We introduce SPIRIT-LM, a single language model that can generate both speech and text. SPIRIT-LM is based on continuously pretraining LLAMA 2 with *interleaving* speech and text data.
- Similarly to text LLMs, we find that SPIRIT-LM can learn new tasks in the few-shot setting in text, speech and in the cross-modal setting (i.e. speech to text and text to speech).
- To evaluate the expressive abilities of generative models, we introduce the SPEECH-TEXT SENTIMENT PRESERVATION BENCHMARK (noted STSP) that measures how well generative models preserve the sentiment of the prompt within and across modalities for both spoken and written utterances.
- Finally, we propose an expressive version of SPIRIT-LM (SPIRIT-LM-EXPRESSIVE). Using STSP, we show that SPIRIT-LM is the first language model that can preserve the sentiment of text and speech prompts both within and across modalities.

The rest of the paper is structured as follows: We describe relevant related work (Section 5.2), our methods for model training and evaluation (Section 5.3), text and speech understanding evaluation results (Section 5.4), sentiment modeling evaluation (Section 5.5), an in-depth responsible AI evaluation of SPIRIT-LM with a focus on spoken and written toxicity detection (Section 5.6), and finally the broader impact of this work (Section 5.7).

5.2 Related Work

Textless NLP Recent progress in Self-Supervised Speech Representation Learning (SSL) (Baevski et al., 2020c; Chen et al., 2022; Chung et al., 2021; Hsu et al., 2021a) has made it possible to learn from raw audio speech representations that are good for a variety of downstream tasks (Yang et al., 2021). In addition, these methods can be used to derive discrete tokens that operate as a kind of pseudo-text and can be used to learn a language model from raw audio (Lakhotia et al., 2021) which is able to capture both the linguistic content and the prosody (Kharitonov et al., 2022b), giving rise to a host of applications: emotion conversion (Kreuk et al., 2022), dialogue generation (Nguyen et al., 2023b), speech classification (Chang et al., 2023b). Even though these models are good at capturing expressivity, they trail text models in capturing semantics when trained with comparable amounts of data (see Nguyen et al., 2023b, 2020b). In this work, we use semantic speech

	Hours	N Tokens		P Samp.	Epochs
		Speech	Text		
Speech-only	458K	28.2B		33.3%	1.24
Speech+Text	111K	7.0B	1.4B	33.3%	3.81
Text-only			307B	33.3%	0.11

Tab. 5.2: Statistics of training data. P Samp. is the Sampling Proportion of each subset for a training batch. Epochs is the number of epochs seen for each subset after 100K training steps or equivalently 100B tokens. For Speech+Text datasets, Epochs can be varied for different training tasks as speech & text tokens can be dropped.

tokens extracted from HuBERT (Hsu et al., 2021a), possibly combined with pitch and style tokens (as in Kharitonov et al., 2022b), and supplement the model training with textual bpe-units.

Speech and Speech+Text LMs There has been an increasing number of SpeechLMs since GSLM (Lakhotia et al., 2021). AudioLM (Borsos et al., 2023) utilizes two types of discrete speech tokens: semantic tokens (derived from w2v-BERT, Chung et al., 2021), and acoustic tokens (derived from SoundStream, Zeghidour et al., 2021) to capture semantic and acoustic information from speech respectively. They model speech in a multi-stage fashion (semantic $\rightarrow$ coarse acoustic $\rightarrow$ fine-grained acoustic) in order to generate speech in the same acoustic style as the prompt while being semantically coherent. Vall-E (Wang et al., 2023a) models speech with acoustic tokens (Encodec, Défossez et al., 2022) and perform TTS task by translating phonemes to tokens using an autoregressive LM. Hassid et al. (2023) found that fine-tuning pre-trained TextLMs helps boost the performance of SpeechLMs. SpeechGPT (Zhang et al., 2023a) further fine-tune speechLMs on cross-modal tasks (ASR, TTS) and chain-of-modality Question-Answering (QA) task (Q-speech $\rightarrow$ Q-text $\rightarrow$ A-text $\rightarrow$ A-speech) to perform spoken QA tasks. Similar to SpeechGPT, Spectron (Nachmani et al., 2023) utilizes text as a proxy for spoken QA and speech continuation tasks (speech-prompt $\rightarrow$ text-prompt $\rightarrow$ text-continuation $\rightarrow$ speech-continuation). Unlike previous work, they represent speech using a spectrogram and employ a pre-trained speech encoder (USM, Zhang et al., 2023b) to extract speech features. In the same spirit, Fathullah et al. (2023) propose replacing the text questions with their speech versions during the fine-tuning of a chat LLAMA 2 model to obtain an end-to-end model able to perform speech question answering, speech translation, and audio summarization tasks. AudioPALM (Rubenstein et al., 2023) and VioLA (Wang et al., 2023b) both train autoregressive language models on text and speech in a multi-task fashion and focus on Speech Recognition (ASR), Speech Synthesis (TTS) and Speech Translation (AST, S2ST) tasks. Most recently, VoxtLM (Maiti et al., 2023) and SUTLM (Chou et al., 2023) jointly trained speech and text LMs on ASR, TTS, and speech/text

Model	#shots	Accuracy ↑			
		T→T	T→S	S→S	S→T
SPIRIT-LM-BASE	0	0.69	0.33	0.33	0.32
SPIRIT-LM-EXPRESSIVE	0	0.68	**0.43**	**0.48**	0.33
Few-Shot Prompting					
	3	0.67	0.34	0.42	0.34
SPIRIT-LM-EXPRESSIVE	6	**0.70**	0.36	0.45	**0.37**
	9	0.63	0.36	0.46	0.34
Random Predictor		0.33	0.33	0.33	0.33
Cascade Topline					
(ASR) + LLAMA 2 + (TTS)	0	0.64	0.34	0.32	0.36
Prompt Performance	0	0.86		0.96	

Tab. 5.3: Zero-Shot and Few-Shot Performance on the SPEECH-TEXT SENTIMENT PRESERVATION BENCHMARK. SPIRIT-LM models (trained for 100k steps) are presented with prompts expressing a positive, negative or neutral sentiment. In the speech modality the sentiment is in the audio quality (laughter, cries, etc), and in text it is in the semantic content. The continuation is then elicited across modalities or, as a control, in the same modality, and tested with pretrained classifiers. The last row (Prompt Performance) presents the performance when we apply the classifier directly on the text or speech prompt.

continuation tasks. Our work is mainly similar to Chou et al. (2023) in the training tasks but with the additional capacity of performing cross-modal generation and expressive speech and text generation. We also study larger models and evaluate their zero-shot and in-context learning capabilities.

5.3 Methods

SPIRIT-LM models are based on continuously pretraining a text-pretrained language model on a combination of text and speech (Figure 5.1.a). Following Hassid et al., 2023, we continuously pretrain LLAMA 2 (Touvron et al., 2023b) using a collection of text-only datasets, speech-only datasets and aligned speech+text datasets fed to the model with *interleaving*. We evaluate all our models on speech and text comprehension metrics (sWUGGY, sBLIMP, Nguyen et al., 2020b; sStoryCloze, tStoryCloze, Hassid et al., 2023; MMLU Hendrycks et al., 2021) and downstream tasks such as ASR, TTS and speech classification.

SPIRIT-LM comes in two versions: SPIRIT-LM-BASE and SPIRIT-LM-EXPRESSIVE. SPIRIT-LM-BASE models speech using HuBERT tokens (Hsu et al., 2021a) while SPIRIT-LM-EXPRESSIVE uses the concatenation of HuBERT, pitch and style tokens.

Model	Task	WUGGY↑		BLIMP↑		Topic-StoryCloze↑				StoryCloze↑				MMLU↑
		T	S	T	S	T	S	T→S	S→T	T	S	T→S	S→T	T
Previous Work														
GSLM (Lakhotia et al., 2021)		∅	64.8	∅	54.2	∅	66.6	∅	∅	∅	53.3	∅	∅	∅
AudioLM (Borsos et al., 2023)		∅	71.5	∅	**64.7**	∅	–	∅	∅	∅	–	∅	∅	∅
Voxtlm (Maiti et al., 2023)		**80.3**	66.1	**74.2**	57.1	–	–	–	–	–	–	–	–	–
TWIST (Hassid et al., 2023)		∅	**74.5**	∅	59.2	∅	76.4	∅	∅	∅	55.4	∅	∅	∅
Ours														
SPIRIT-LM-BASE		**80.3**	69.0	73.3	58.3	**98.0**	**82.9**	**72.7**	**88.6**	**79.4**	**61.0**	**59.5**	**64.6**	**36.9**
SPIRIT-LM-EXPRESSIVE		75.8	65.0	73.6	54.2	97.9	75.4	61.6	73.2	78.9	56.9	54.6	58.8	33.3
Cascade Topline														
(ASR +) LLAMA 2		84.1	79.2	72.8	71.6	98.5	94.76	94.76	94.76	81.9	75.7	75.7	75.7	46.2

Tab. 5.4: Zero- and few-shot comprehension evaluation. Reporting accuracy based on negative-log-likelihood – normalized by the number of tokens – minimization prediction. MMLU is evaluated in the 5-shots prompting setting. The other tasks are evaluated in the zero-shot setting. T refers to the text modality and S to the Speech modality. We fill with ∅ the task and modality that are not supported by the reported system, and with _ the scores that are not publicly available.

5.3.1 SPIRIT-LM-BASE

The SPIRIT-LM-BASE model is based on the 7B version of LLAMA 2 trained on Text-only, Speech-only, and aligned Speech+Text datasets.

Speech Encoder We use the same HuBERT model as in TWIST (Hassid et al., 2023), which is trained on a mixture of datasets: Multilingual LibriSpeech (Pratap et al., 2020), Vox Populi (Wang et al., 2021), Common Voice (Ardila et al., 2020), Spotify (Clifton et al., 2020), and Fisher (Cieri et al., 2004). The HuBERT model was trained for 4 iterations, with a downsampling factor of 640, resulting in a sample rate of 25hz. For the quantization, we utilized k-means 500 units from TWIST as base units and trained a feed-forward quantizer using data-invariant augmentation technique from Gat et al. (2023). We finally obtained a vocabulary of 501 semantic speech tokens.

Speech and Text Tokenization We tokenize text with the default LLaMA's tokenizer and speech with the HuBERT tokenizer described above. Following previous work, HuBERT tokens are deduplicated for betting modeling quality. For uni-modal datasets (Text-only and Speech-only), we tokenize the data and prepend them with the corresponding modality token, i.e. "[TEXT]this is a text sentence" or "[SPEECH][Hu262][Hu208][Hu499][Hu105]".

Interleaving Speech and Text For the aligned Speech+Text datasets, we mix text and speech by interleaving speech and text at the word level (Figure 5.1.b), making the input look like this "[TEXT]the cat [SPEECH][Hu3][Hu7]..[Hu200][TEXT]the mat"[2]. Our hypothesis is that interleaving training will help the model learn an alignment between speech and text tokens, unlocking better text to speech transfer. The speech and text spans within the sentences are sampled randomly at each training step.

Speech Decoder As for speech synthesis from speech tokens, we train a HifiGAN (Kong et al., 2020; Polyak et al., 2021) vocoder on the Expresso dataset. The HifiGAN model is conditioned on HuBERT speech tokens and 1-hot speaker embedding from one of 4 Expresso's voices.

5.3.2 SPIRIT-LM-EXPRESSIVE

Previous work shows that HuBERT tokens can capture good semantic information from speech but perform badly at expressivity (Nguyen et al., 2023a). Our goal is to have a model that can understand and preserve the emotion in the input speech while being biometric-free. We therefore supplement semantic speech tokens from HuBERT with additional *pitch tokens* and *style tokens* and include them in language model training so that our trained SPIRIT-LM-EXPRESSIVE model can capture and generate more expressive speech.

Pitch Tokens Following Polyak et al. (2021) and Kharitonov et al. (2022b), we produce pitch tokens using a VQ-VAE (Oord et al., 2017b) model trained on the F0 of the input speech. Following the implementation of Polyak et al. (2021)[3], we trained a VQ-VAE model on the Expresso (Nguyen et al., 2023a) dataset with a codebook size of 64 and a downsampling rate of 128, resulting in 12 pitch tokens per second. For training the pitch quantizer, the F0 is extracted using pyannote[4]. However, for the language model training, we extract F0 using FCPE[5], a fast pitch estimator using Transformer, for inference speed.

[2]with "[Hu3][Hu7]..[Hu200]" being the tokenization of the spoken utterance "sat on"
[3]https://github.com/facebookresearch/speech-resynthesis
[4]https://github.com/pyannote/pyannote-audio
[5]https://github.com/CNChTu/FCPE

Style Tokens We extract speechprop features from Duquenne et al. (2023), which capture speech input's expressive style. The features were pooled with average pooling over input segments of 1 second, making one feature every one second. In order to keep style tokens biometric-free, we further remove speaker information from speechprop features by fine-tuning the features to predict the expressive style on the Expresso dataset which serves as a normalization step to obtain the style features. We finally train a k-means clustering on the normalized features of Expresso dataset with 100 units.

Expressive Speech Tokenization We mix the 3 types of tokens (HuBERT tokens at 25hz, pitch tokens at 12.5hz, style tokens at 1hz) into a single sequence of tokens by sorting the tokens with their corresponding timestamps (Figure 5.1.c). Similar to SPIRIT-LM-BASE, we deduplicate HuBERT tokens as well as pitch tokens, making the input sequence look like this: "[SPEECH][St10][Pi0][Hu28][Hu22][Pi14][Hu15][Pi32][Hu78][Hu234][Hu468]"

Apart from the speech tokenization, the training details of SPIRIT-LM-EXPRESSIVE are the same as for SPIRIT-LM-BASE.

Expressive Speech Decoder We train a HifiGAN model conditioned on HuBERT tokens, pitch tokens, style tokens and 1-hot speaker embedding from Expresso's voices.

5.3.3 Training Details

Our SPIRIT-LM models are trained on a combination of speech, text and aligned speech+text sequences. We report in Table 5.2 the amount and sampling proportion of each type of data and list the datasets we use here:

Text-only datasets We include a subset of LLaMA (Touvron et al., 2023a) training datasets, where we exclude datasets that are unrelated to speech, like code, totaling 300B text tokens.

Speech-only datasets We employ open-sourced large-scale speech datasets, totaling 460K hours of speech or 30B speech tokens.

Model	Task	WUGGY↑		BLIMP↑		Topic-StoryCloze↑				StoryCloze↑				MMLU↑
		T	S	T	S	T	S	T→S	S→T	T	S	T→S	S→T	T
SPIRIT-LM variants														
SPIRIT-LM-BASE		**80.3**	69.0	73.3	58.3	98.0	**82.9**	**72.7**	**88.6**	**79.4**	**61.0**	59.5	64.6	36.9
- No Interleaving		74.7	67.1	72.6	57.2	97.7	74.0	57.5	71.9	78.2	60.1	54.2	56.4	32.1
- Randomly-initialize		78.1	**69.9**	72.9	**58.8**	97.6	81.8	70.2	88.1	73.7	58.0	58.2	62.5	25.8
- Rope θ default		78.2	69.5	73.3	57.7	**98.2**	82.0	72.0	88.3	78.9	60.9	**59.8**	**65.5**	34.3
- +ASR+TTS		76.8	68.7	71.7	57.2	97.7	81.6	71.6	86.1	77.4	59.9	58.8	63.5	31.4
Parallel Data Training														
Word-level transcription		74.7	67.1	72.6	57.2	98.0	80.3	57.5	71.9	78.2	60.1	54.2	56.4	32.1
ASR+TTS-only		76.5	69.8	73.3	57.6	97.3	74.9	63.5	71.8	76.3	54.6	53.9	54.0	34.4
Unimodal Models														
Speech Only		67.1	69.5	53.7	58.0	54.8	72.9	52.2	49.4	53.7	54.8	52.6	49.3	27.2
Text Only		72.6	46.8	**73.9**	52.6	98.2	51.7	47.5	51.7	79.0	50.2	47.3	52.1	**40.1**

Tab. 5.5: Ablation experiments in Zero- and few-shot comprehension evaluation. All the models reported are initialized from LLAMA 2 7B (except Randomly-initialize one) and are trained for 100k steps. Reporting accuracy based on negative-log-likelihood – normalized by the number of tokens – minimization prediction. MMLU is evaluated in the 5-shots prompting setting. The other tasks are evaluated in the zero-shot setting. T refers to the text modality and S to the Speech modality. For a full comparison of unnormalized and normalized scoring accuracy, refer to Table 5.10.

Aligned Speech+Text datasets We use a small subset of speech datasets that came along with text transcriptions. We then collect speech-text alignments at word-level either through the provided dataset or by performing an alignment at the word level using aligner tool from Pratap et al. (2023)[6]. All the alignments are automatically curated, and thus, possible errors in the alignments are admitted. The speech+text datasets comprise of 110K hours of speech or 7B speech tokens (HuBERT) and 1.5B text tokens.

In total, we have 570K hours of speech. As the number of tokens differs a lot in different modalities, we tuned the sampling weights of the datasets so that the model sees each modality (speech, text, speech+text) roughly equal number of times during training.

Optimization Following Rubenstein et al. (2023), we extend the embeddings of LLaMa vocabulary with new speech tokens and modality tokens. The new tokens' embeddings are initialized randomly. We then continue to pre-train the 7B LLAMA 2 model with the constant final learning rate of $3.0e^{-5}$, a sequence length of 4k (equivalent to 200 seconds of speech only), and a batch size of 4 per GPU. We trained the model on 64 A100 GPUs, making an efficient batch size of 1M tokens, for 200K steps. Following Xiong et al. (2023) and Rozière et al. (2024), we make a small

[6] https://pytorch.org/audio/main/tutorials/ctc_forced_alignment_api_tutorial.html

Model	Task	LS clean (10 shots)		LS other (10 shots)		IC (30 shots)
		ASR↓	TTS↓	ASR↓	TTS↓	↑
SPIRIT-LM *variants*						
SPIRIT-LM-BASE		21.9	45.5	29.2	43.8	71.9
+ASR+TTS		**6.0**	**6.7**	**11.0**	**7.9**	**75.8**
SPIRIT-LM-EXPRESSIVE		37.9	52.0	50.0	53.6	66.2
Parallel Data Training						
Word-level transcription		113.2	85.2	111.6	75. 2	22.6
ASR+TTS only		7.7	8.1	11.9	9.4	7.4
Cascade Topline						
(WHISPER +) LLAMA 2 (+MMS TTS)		3.7	4.0	7.2	4.9	89.6

Tab. 5.6: Few-shot tasks. We evaluate SPIRIT-LM models for Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) Evaluation on LibriSpeech (LS) and Intent Classification (IC). ASR scores correspond to Word-Error-Rate (% WER) evaluated in the 10-shots setting with a max context length of 1024. TTS scores correspond to the Character-Error-Rate (% CER) in the 10-shots setting with a max context length of 2048. IC scores correspond to accuracy in the 30 shots setting.

modification to the RoPE positional encoding by increasing the "base frequency" θ of ROPE from 10,000 to 100,000, which has been shown to benefit long-context modeling. Finally, for the speech-text interleaving sampling strategy, we randomly select the word spans so that each text sequence contains 10-30 words and each speech sequence contains 5-15 words, we do this in order to balance the portion of speech tokens and text tokens in the input sequences[7].

5.3.4 Evaluation

We evaluate SPIRIT-LM checkpoints in a large number of scenarios and use cases. First, to showcase the semantic abilities of our models in speech, we report the transcript of speech generations collected by prompting the model with text or speech sequences. As illustrated in Table 5.1, SPIRIT-LM is able to generate semantically and expressively consistent speech when prompted with speech tokens or text tokens.

Second, we evaluate our models quantitatively with an extensive collection of benchmarks that require generating text or speech tokens:

Speech- and Text- only Tasks We use sWUGGY, sBLIMP, StoryCloze, and speech classification tasks. All these tasks take as input a sequence of speech tokens and measure if the model is able to find the correct sequence among two choices.

[7] In our initial experiments, we found that changing the length of word spans has little impact on our evaluation metrics, but we do expect a more detailed analysis of this on longer context metrics in further work.

sWUGGY and sBLIMP are described in detail in Nguyen et al. (2020b). Briefly, sWUGGY measures if the model can discriminate between existing spoken words and non-words (e.g., "brick" vs. "blick"). sBLIMP measures if the model can distinguish between a spoken grammatically correct sentence and an ungrammatical spoken variant of the same sentence (e.g., "cats are lazy" vs. "cats is lazy"). Given the beginning of a short spoken story, StoryCloze measures if the model can find the plausible ending among two sentences, which typically requires some high-level semantic understanding and common sense (Mostafazadeh et al., 2017). We use the spoken version of the original storycloze (S-StoryCloze) and the topic-Storycloze (T-StoryCloze) assembled by Hassid et al. (2023) based on simpler negative samples. All of these tasks have a random baseline performance of 50%. All these tasks are evaluated in the 0-shot prompting setting. We predict the sample with the highest likelihood of the two choices. In addition to speech, these benchmarks are also available in the text modality. We, therefore, measure the text-modeling abilities of SPIRIT-LM on these. In addition, we evaluate SPIRIT-LM on MMLU (Hendrycks et al., 2021), a popular evaluation benchmark for LLMs in the text modality. Finally, we evaluate SPIRIT-LM on the Intent-Classification task from Chang et al. (2023b).

Speech-to-Text and Text-to-Speech Tasks SPIRIT-LM is trained in both speech and text. For this reason, it has the ability to model tasks that require both text and speech modeling. We evaluate SPIRIT-LM for ASR. We report the Word-Error-Rate (WER) between the generated and the gold transcriptions. For text-to-speech (TTS), we consider our system's ability to generate the audio corresponding to the inputted text. We measure the performance by transcribing the generated audio with Whisper (Radford et al., 2023), a state-of-the-art ASR model, and we compare it with the original text with Character-Error-Rate. Both these tasks are evaluated in English with Librispeech clean and other test sets.

5.3.5 Baselines

We compare our results with previously published generative speech systems. All these methods use one or several Transformer (Vaswani et al., 2017) decoder-only models trained on speech units. They differ in how they are trained (pretrained from scratch or fine-tuned), the types of speech units they model, and their amount of training data. GSLM (Lakhotia et al., 2021) is based on speech units (e.g. Hubert) and trained from scratch on speech-unit modeling. TWIST (Hassid et al., 2023) is a textually pretrained speech model based on Llama-13B (Touvron et al.,

2023a). AudioLM (Borsos et al., 2023) is a cascade system made of a semantic sequence model (using w2v-BERT, Chung et al., 2021) combined with coarse-acoustic and fine-acoustic models (using SoundStream units, Zeghidour et al., 2021). In contrast with SPIRIT-LM, the approach mentioned above only relies on speech units during training, making them speech-only models (i.e. they do not support text understanding nor generation).

We also compare our models to VoxtLM (Maiti et al., 2023), a concurrent work on speech and text language modeling. We report the best scores from the original published papers for all the mentioned methods.

As a top-line comparison, we compare our models with cascade models that use LLAMA 2 as a text generative model. For text-to-text (T→T), we only rely on LLAMA 2-7B. For speech-to-speech (S→S), we utilize the cascade model, ASR from WHISPER-MEDIUM (Radford et al., 2023), followed by LLAMA 2, synthesized by MMS-TTS (Pratap et al., 2023).

5.4 Speech and Text Understanding

5.4.1 Lexical, Grammatical and Semantic Knowledge in Text and Speech

We find that SPIRIT-LM-BASE competes with the baselines for WUGGY, BLIMP, and Storycloze in the speech modality while preserving competitive text performance (cf. Table 5.4). More specifically, SPIRIT-LM-BASE outperforms the baselines by a large margin on StoryCloze, which requires the most advanced speech semantic abilities compared to the other reported benchmarks.

Interleaving is critical We run ablation experiments (cf. Table 5.5) to understand what leads to this performance by controlling for the training budget and ablating a large number of training parameters. We set the training budget at 100k training steps or 100B tokens.

We compare SPIRIT-LM-BASE to a LLAMA 2 model continuously pretrained with two parallel data training settings. First, the ASR+TTS-only model consists of training with pairs of semantically equivalent sequences of speech and text (e.g. "[TEXT] the cat jumped by the window [TTS][Hu12]..[Hu54]" or "[SPEECH][Hu12]..[Hu54][ASR]

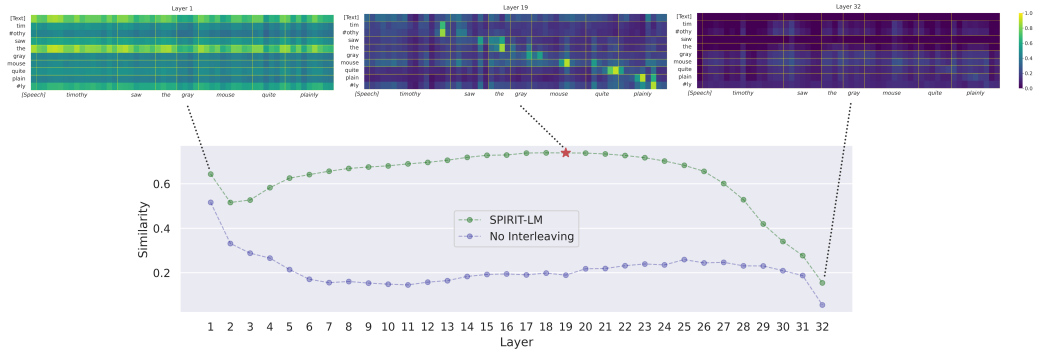


Fig. 5.2: Alignments of features obtained from Text and Speech Inputs. Bottom: Similarity of speech and text features extracted from different layers of SPIRIT-LM compared with the model training without speech-text interleaving. The similarity is computed as the maximum similarity over speech and text features of the same words and is averaged over a test set. **Top:** Pairwise cosine similarity between text features and speech features of the same sentence extracted from different layers of SPIRIT-LM.

the cat jumped by the window"[8]). Second, the Word-level Transcription model consists of training on sequences of pairs of textual and spoken words (e.g. "[TEXT] the [SPEECH][Hu12]..[Hu34] [TEXT] cat [SPEECH][Hu454]..[Hu90]...[TEXT] window [SPEECH][Hu15]..[Hu54]"). Additionally, we compare SPIRIT-LM-BASE to models trained on a single modality (speech or text) and with speech+text but without any interleaving data (cf. No Interleaving in Table 5.5).

Based on these experiments, we conclude that interleaving training is the primary factor leading to good-quality speech generation. Fine-tuning LLAMA 2 on parallel data leads to lower performance on tasks such as StoryCloze and BLIMP. Notably, fine-tuning the model on speech-only tokens leads to a much lower performance (e.g. more than 6 points difference with SPIRIT-LM on spoken Storycloze). This shows that interleaving training not only helps preserve the text generation abilities of the model but also leads to better speech understanding and generation performance. We measure the importance of the amount of aligned data used for interleaving training in Figure 5.3. We find that the model's performance in speech (T-StoryCloze) steadily increases with the amount of aligned data.

As shown in Table 5.4, SPIRIT-LM-EXPRESSIVE performs lower than SPIRIT-LM-BASE on these tasks, indicating that the expressive speech units lead to moderate lexical, grammatical, and semantic understanding degradation. We explain this with the following intuition. Modeling a given raw speech for SPIRIT-LM-EXPRESSIVE is more

[8]with "[Hu12]..[Hu54]" being the tokenization of the spoken utterance "the cat jumped by the window"

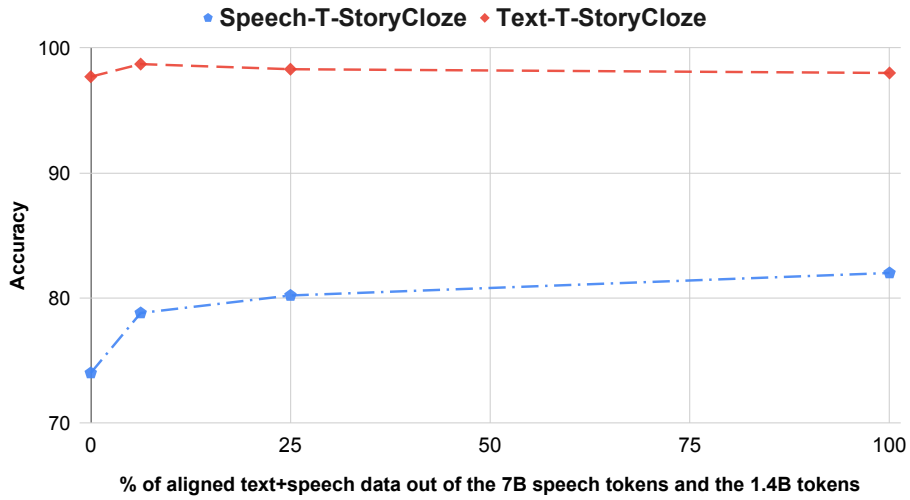


Fig. 5.3: Performance of SPIRIT-LM-BASE on Topic-StoryCloze in speech and text with regard to the sampled amount of aligned speech+text data from 0% to 100% out of the 8.4B tokens aligned tokens. (1.4B text tokens and 7B tokens speech tokens.)

costly than for SPIRIT-LM-BASE. Indeed, in contrast with SPIRIT-LM-BASE, SPIRIT-LM-EXPRESSIVE is based on integrating expressive speech units in the sequence during training, in addition to Hubert-tokens. This leads to extending the sequence length in the number of tokens for a fixed raw input speech. This added complexity leads to a degradation of speech modeling performance.

In the text modality, despite being fine-tuned on billions of speech tokens, SPIRIT-LM still performs decently on MMLU (above 33%) and degrades by less than 2 points on WUGGY, BLIMP, and StoryCloze compared to LLAMA 2.

Finally, on these tasks, the cascade approach (ASR with WHSIPER followed by LLAMA 2) is above SPIRIT-LM by a large margin.

5.4.2 Cross-Modal Evaluation

SPIRIT-LM can also model sequences that are made of both speech and text tokens.

Cross-Modal StoryCloze Based on the text and speech versions of StoryCloze, we build a speech to text (S→T) and text to speech (T→S) Storycloze for which the context is in one modality (e.g. speech) and the hypothesis is in the other modality (e.g. text). As seen in Table 5.5, we find the performance of SPIRIT-LM-BASE in the text to speech direction (T→S) on par with the speech only performance (S). In

contrast, the (S→T) direction is about 5 points above the speech performance (S). This suggests that the model performs better at text generation compared to speech generation even when it is conditioned on a speech sequence.

ASR & TTS Similarly to text language models, SPIRIT-LM can be prompted with few-shot examples to perform specific tasks. We illustrate this with ASR and TTS. We show in Table 5.6 that SPIRIT-LM models reach non-trivial performance in ASR and TTS. We find that few-shot prompting leads to the best performance with 10 shot prompting (cf. Figure 5.4).[9] Our best SPIRIT-LM-BASE model is at 21.9 Word-Error-Rate in Librispeech clean and 45.5 in Character-Error-Rate in TTS. We observe that when we add parallel ASR and TTS examples during training (cf. +ASR+TTS in Table 5.6), we can improve the performance from a very large margin. We note that adding ASR and TTS data has a very moderate impact on the rest of the tasks. We report the detailed prompting used for ASR and TTS in Section 5.9.1.

Cross-Modal Alignment To understand better the hidden mechanism that enables SPIRIT-LM to deliver good cross-modal performance while only being trained on *interleaved* data and raw speech and text, we look at the token-level similarity of the model's features from input sequences of HuBERT tokens and the corresponding BPE tokens. We illustrate this in Figure 5.2 (bottom), where we compute the maximum similarity over the same words of speech and text features extracted from different layers of SPIRIT-LM. We find that the similarity between spoken and written sequences inside the model increases from layer 2 and layer 20. In comparison, this alignment does not occur when the model is trained without interleaving (cf. Figure 5.2 bottom). This suggests that interleaving enables the model to map speech sequences with corresponding text sequences. Figure 5.2 (top) shows the alignments of BPE tokens and HuBERT tokens of the sentence *Timothy saw the gray mouse quite plainly* on layers 1, 19, 32. We see that the middle layers of SPIRIT-LM capture the same semantics information from both input modalities, with high alignments towards the end of each word (last BPE tokens, late HuBERT tokens).

5.4.3 Downstream Speech Classification

Finally, we report in Table 5.6 the abilities of SPIRIT-LM to perform speech classification task. We experiment with Intent-Classificaton (IC). We find that the accuracy

[9] We note that above 20 shots, we reach the maximum number of tokens that fit in the context for ASR and TTS.

improves with the number of shots (cf. Figure 5.4). Our best SPIRIT-LM model reaches up to 79% accuracy (compared to 89% of the topline performance). The detailed prompting used for IC is given in Section 5.9.1.

Pretrained Knowledge is Essential for Few-Shot Learning We report in Figure 5.6 the task-specific performance of SPIRIT-LM-BASE with regard to the number of training steps compared to a randomly initialized model trained in the same setting. After only 25k training steps, SPIRIT-LM-BASE reaches more than 75% accuracy on Intent Classification while the randomly initialized model is below 20%. This means that starting from a pretrained LLAMA 2 model is essential for few-shot in-context learning and that our method successfully transfers the pretrained few-shot learning abilities of the model to the speech modality.

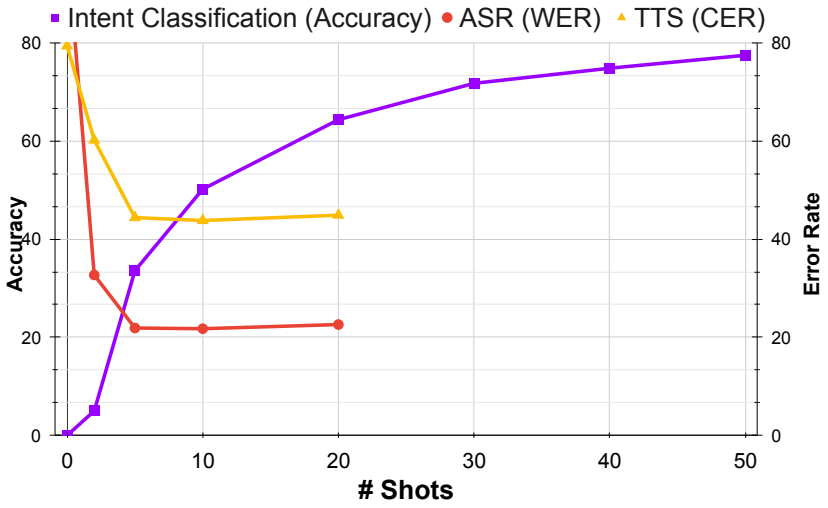


Fig. 5.4: SPIRIT-LM-BASE performance with regard to the number of shots presented to the model context for Intent Classification, ASR and TTS.

5.5 Expressivity Evaluation

One of the core contributions of this work is the expressivity modeling. To measure the expressivity of our model we first evaluate the quality of the introduced pitch and style tokens (§ 5.5.1). Second, we evaluate our SPIRIT-LM models on the newly introduced SPEECH-TEXT SENTIMENT PRESERVATION BENCHMARK (§ 5.5.2).

5.5.1 Style and Pitch Tokens Evaluation

We model expressive speech by complementing semantic speech tokens (HuBERT) with Pitch and Style tokens. To evaluate the quality of our tokenization, we use the speech resynthesis task from Nguyen et al. (2023a). It measures how well the resynthesized speech is compared with the original audio in terms of preserved content, expressive style, and pitch.

Table 5.7 shows the performance of SPIRIT-LM-BASE and SPIRIT-LM-EXPRESSIVE tokenizers compared to Encodec and Hubert-only baselines. We see the SPIRIT-LM-EXPRESSIVE tokenizer can capture good expressive style and pitch from the input speech. Additionally, we observe a very large improvement in Style and Pitch resynthesis when we compare SPIRIT-LM-BASE tokenizer with SPIRIT-LM-EXPRESSIVE.

		Bitrate	**Content**	**Style**	**Pitch**
Model	**Metrics**	BPS↓	WER↓	EMO↑	FFE↓
Original Audio		-	16.2	65.2	-
Expresso models (Nguyen et al., 2023a)					
Hubert + HifiGAN		550	23.0	22.7	0.30
Hubert + HifiGAN w/ GT Style		550	21.4	61.6	0.27
Encodec (RVQ=1)		500	38.0	41.5	0.09
Encodec (RVQ=8)		4000	19.0	56.7	0.04
SPIRIT-LM *Tokenizers*					
SPIRIT-LM-BASE		225	23.4	20.4	0.40
SPIRIT-LM-EXPRESSIVE		307	23.2	41.4	0.16

Tab. 5.7: **Expressive Speech Resynthesis Evaluation**. Performances of SPIRIT-LM Tokenikers on the Expresso Benchmark (Nguyen et al., 2023a) compared with their systems. The scores are averaged across datasets. For the detailed scores, refer to Table 4.6.

5.5.2 The SPEECH-TEXT SENTIMENT PRESERVATION BENCHMARK (STSP)

To evaluate how well our SPIRIT-LM models can understand and generate expressive speech and text, we introduce the SPEECH-TEXT SENTIMENT PRESERVATION BENCHMARK. It is made of a collection of speech and text prompts in the positive, negative or neutral sentiment. Given a spoken or written prompt , the task consists

The SPEECH-TEXT SENTIMENT PRESERVATION BENCHMARK			
Prompt origin	EXPRESSO-READ	EXPRESSO-ASR	EMOV
Prompt Type	Speech	Text	Speech
#Samples	1020/60/54	1373/479/462	1053/351/351
#Speakers	4	-	3
Classes	Positive(33%) / Negative(33%) / Neutral(33%)		

Tab. 5.8: Statistics of the SPEECH-TEXT SENTIMENT PRESERVATION BENCHMARK. (#Samples indicates the number of samples in each train/dev/test split.)

in generating a text or speech sequence of tokens that preserves the sentiment of the prompt.

For instance, in the text-to-X direction (T→T and T→S), given a written sentence bearing sadness, we check if the spoken generated text/utterance is also sad. On the other hand, the direction speech-to-X (S→S and S→T), given a spoken happy-sounding utterance, we check whether the model generates a positively written text or positive utterance.

5.5.2.1 Sentiment-Rich Spoken and Written Prompts

Speech Prompt In order to have the read speech of different expressive styles (e.g. *he's done it again* in happy/sad style). We utilize two datasets: 1) *Expressive reading* from EXPRESSO(Nguyen et al., 2023a) consisting of 47 hours of expressive North American English speech where 7 different styles are applied on the same content that does not reflect the emotion being conveyed. We use only the speech from 3 emotions: "happy", "sad" and "default". (we will refer to this dataset as EXPRESSO-READ) 2) EMOV (Adigwe et al., 2018), composed of emotional speech from 5 different speakers and 2 languages (North American English and Belgian French). We select only the English speech from 3 speakers when the same content is recorded in three different emotions: "Amused", "Angry" and "Neutral".

Text Prompt In order to have expressive text (e.g. *he's such an amazing player* for positive) as prompt, we transcribe[10] *improvised dialog* from EXPRESSOfor 4 emotions: "happy", "angry", "sad" and "default" to obtain an aligned Speech-Text dataset. Then we filter the samples if the transcription has less than 10 words (separated by space) or it has one word appearing more than 10 times. We refer to this aligned dataset by EXPRESSO-ASR.

[10]The transcription is done by WHISPER-MEDIUM (Radford et al., 2023).

Sentiment Mapping To unify different sets of emotional classes, we associate the emotions "happy"/"Amused", "sad"/"Angry" and "default"/"Neutral" to the "positive", "negative" and "neutral" sentiments.

Data Splits We split the datasets into train/dev/test subsets for later usage. Table 5.8 presents a comprehensive statistical overview of the datasets used. For EXPRESSO-READ, we use the original train/dev/test splits; while for the EMOV, we split it randomly into train/dev/test subsets with the ratios of 60/20/20. The EXPRESSO-ASR dataset is also divided into train/dev/set with the ratios of 60/20/20[11]. We use the train and dev subsets to train the sentiment classifiers and the test subset to prompt the SPIRIT-LM models.

5.5.2.2 Evaluation Metrics

For both tasks, we check if the generated utterance has a sentiment that is consistent with the sentiment of the prompt. We assess the sentiment of the produced utterance using sentiment classifiers and report its accuracy. The accuracy for speech-to-X directions is averaged over EXPRESSO-READ and EMOV.

We obtain text and speech sentiment classifiers by fine-tuning pre-trained text and speech models respectively. For the speech classifier, similar to Nguyen et al. (2023a), we fine-tune the wav2vec2 model[12] on the training sets of EXPRESSO-READ, EXPRESSO-ASR[13] and EMOV. For the text classifier, we fine-tune the 3-classes sentiment classifier from Hartmann et al. (2021) on the transcriptions of the EXPRESSO-ASR training set.

5.5.2.3 Evaluation Settings

We tune the generation parameters on the dev sets. In terms of the maximal number of generated tokens, we use 50 for T→T and S→T, 200 for T→S, and 300 for S→S. We use a temperature of 0.8 and nucleus sampling (Holtzman et al., 2020) with a top_p of 0.95 for all the directions. All the SPIRIT-LM models reported have been trained for 100k steps.

[11] We don't use the original data splits because the amount of data in the dev and test subsets is not enough.
[12] https://huggingface.co/facebook/wav2vec2-base
[13] We use only the speech data

Zero-Shot We prompt SPIRIT-LM using positive, negative or neutral text/speech input from the test sets of the datasets described in section 5.5.2.1. Then 1) for S→S and T→S, we classify the generated speech with the speech classifier. 2) for T→T and S→T, we assess the text continuation with the text classifier.

In-context Few-Shot We also evaluate SPIRIT-LM in a few-shot setting by constructing a set of few-shot examples (see section 5.9.2) and feed them as the in-context prompt.

5.5.2.4 Results

We report the results evaluated on the test sets in Table 5.3. For zero-shot performance, SPIRIT-LM-EXPRESSIVE surpasses SPIRIT-LM-BASE in all directions, with the exception of T→T where they perform comparably. Compared to the cascade baseline, SPIRIT-LM-EXPRESSIVE outperforms it over all the directions except S→T.

In the case of few-shot results, we observe that few-shot is only beneficial when the output modality is text.

For both zero-shot and few-shot, the sentiment continuation is better preserved within the same modality than across different modalities. Among all the directions, S→T scores the lowest.

The final row of Table 5.3 also includes an evaluation of performance directly on the input prompt. All prompts receive high scores, suggesting a significant potential for improvement in the preservation of expressivity.

5.6 Responsible Evaluation in Speech and Text

Responsible AI is at the core of our research. This section discusses and evaluates responsibility aspects from SPIRIT-LM. Speech language models (SpeechLMs) have the potential to bring the same benefits as text-based language models (textLMs), such as access to various downstream applications like search, language generation, summarization, translation, chatbots, etc. in the audio modality. This can increase their reach to more use cases and languages, including those that are not written or sparsely written.

Quantifying and working on user safety is a key aspect from generative model development. These models can inadvertently generate content that is harmful, offensive, or inappropriate is essential for generative language models (Deshpande et al., 2023; Touvron et al., 2023a). While safety is a broad concept, we focus on the specific problem of added toxicity in the generation of the SPIRIT-LM. Inspired by conditional language models' previous studies (Communication et al., 2023b), we define added toxicity as a toxicity increase in the generation compared to the initial source utterance.

5.6.1 Data

We use the HOLISTICBIAS dataset (Smith et al., 2022) and its synthesized speech extension (Communication et al., 2023b). This dataset has been shown to trigger toxicity for conditional language models (Costa-jussà et al., 2023). The dataset consists of over 472k English sentences. Sentences are created by combing a sentence template, such as "What do you think about *PLURAL NOUN PHRASE]*?", a noun (e.g. parents) and a descriptor (e.g., disabled), covering a list of 26 templates and 600 descriptors across 13 demographic axes (e.g., ability, race or gender). We utilize the dataset as the prompt for generating text (T→T) and speech (S→S), respectively.

Task	T→T		S→S	
	ETOX ↓	MUTOX ↓	ASR-ETOX ↓	MUTOX ↓
SPIRIT-LM-BASE	1.19	2.69	1.06	3.75
(ASR) + LLAMA 2 + (TTS)	1.22	2.63	1.17	2.70

Tab. 5.9: **Added Toxicity Detection**. The proportion of sentences with added toxicity divided by the total number of sentences. For the LLAMA 2 baseline, we use a cascaded pipeline made of WHISPER for ASR and MMS for TTS; for SPIRIT-LM-BASE, we use the model trained for 200k steps.

5.6.2 Evaluation Metrics

Similar to Seamless M4T V2 (Communication et al., 2023a), we use MUTOX and ETOX[14] (Costa-jussà et al., 2023) as our toxicity classifiers. For speech, we simply run ASR and evaluate toxicity with ETOX (we refer to this as ASR-ETOX). MUTOX can be directly applied on both text and speech generations, without the need for an ASR system.

[14]Freely available at `https://github.com/facebookresearch/seamless_communication`

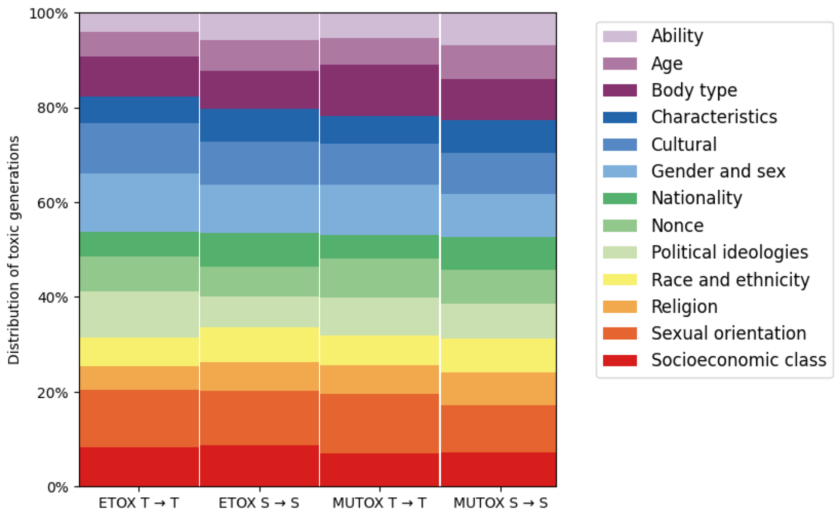


Fig. 5.5: Toxicity Distribution Relative Distribution of added toxicity over the 13 demographic axes for T→T and S→S generations. The number of added toxicities are normalized by the number of occurrences in each demographic axis.

To compute the added toxicity, we evaluate toxicity at the sentence level, both in the input utterance/prompt and in the generated output. We report the proportion of sentences with added toxicity divided by the total number of sentences. For ETOX and ASR-ETOX, a sentence has added toxicity when there are more toxic words found in the generated content than in the prompt. For MuTox, a sentence has added toxicity when the MuTox scores are more than 0.7 higher in the generated content than in the prompt.

5.6.3 Results

We report results in Table 5.9. In terms of ETOX, both SpiRit-LM and (Whisper) + Llama 2 + (MMS-TTS) have comparable results. When evaluated with MuTox, however, SpiRit-LM shows higher added toxicity especially in S→S. This might come from the fact that there exists more toxic contents in our speech training dataset. We leave the mitigation to future work.

Figure 5.5 shows the distribution of added toxicity in SpiRit-LM in terms of the 13 demographic axes represented in HolisticBias and how they vary in modality. We observe that *Gender and sex* and *Sexual orientation* tend to generate more added toxicity than the rest of demographic axes, while *ability* and *nationality* tend to be among the ones that generate the least. There is no big difference in distribution across modalities or metrics.

5.7 Limitations and Broader Impacts

Harmful applications SPIRIT-LM also shares the same risks as its generative model predecessors (Touvron et al., 2023a), such as intentionally harmful applications like fake news and spamming as well as unintentionally harmful ones like unfair or biased results, toxic or untrustworthy generations. These risks can be assessed and mitigated using watermarking e.g Kirchenbauer et al. (2023) or existing reinforcement learning from human feedback (RLHF) e.g. Bai et al. (2022). In addition to these traditional text risks, SPIRIT-LM, being a speech model, also extends risks associated with this modality with intentionally harmful applications like impersonating a specific speaker by continuing short speech segments while maintaining speaker identity and prosody. Mitigation measures for this risk include similar ones as with text (speech watermarking Communication et al., 2023a and RLHF). Similarly to text models, unintentionally harm may arise such as the lack of speaker robustness where the model can generate speech continuations inconsistent with the prompt in terms of accent and dialect only for underrepresented groups in the training data. Among the mitigation strategies, we can include: increasing the variety of the dataset, compensating for bias in representation of different demographics.

Future Work In this paper, we showed how combining style and pitch tokens with semantics tokens and continuously pretraining a text language model delivers very promising multimodal semantic abilities while enabling expressive speech generations. However, several architectural and training improvements could further progress in speech generation.

First, training multimodal models remains a challenge. In this work, we observed that despite training on both speech and text, our SPIRIT-LM models do not perform as well as the initial LLAMA 2 model in text generation. Refining the training procedure could potentially reduce this gap. Second, we restricted our evaluation to English. SPIRIT-LM models were trained on a large amount of non-English data. More investigation is needed to assess the quality and safety of the model in non-English languages. Third, we only experimented with 7B models. Scaling our experiments beyond 7B could lead to much better performance. Finally, the introduced SPIRIT-LM models are foundational models. This means that more work is needed to make them safe and aligned with user expectations. As it is now commonly done with text (Ouyang et al., 2022; Touvron et al., 2023b), fine-tuning a model with instructions and preference data in speech could potentially unlock new experiences such as fully expressive dialog systems.

5.8 Conclusion

We introduced SPIRIT-LM, a speech + text generative language model based on LLAMA 2 that can generate both speech and text in a cross-modal manner. We showed that by alternating speech and text in the input sequence during training, the model is able to generate the content fluidly by changing from one modality to another. We evaluated our models on a collection of speech and text metrics. We plan to make future improvements both in the area of model capability and in transparency and safety.

5.9 Additional Material

5.9.1 Few-Shot Prompts

Speech Recognition (ASR)

For ASR, we prompt the model and add special start and end flags. Indeed, we find that without these flags the model tends to hallucinate after transcripting the input sequence.

For SPIRIT-LM, we use the following prompting. We find that 10 examples leads to the best performance. We illustrate the prompting of SPIRIT-LM for ASR with a single few-shot example:

[SPEECH] *Speech token sequence*
[TEXT] <START Transcript> *Text transcript* <END>
[SPEECH] *Speech token sequence*
[TEXT]

For the models trained with parrallel ASR data (e.g. SPIRIT-LM-BASE +ASR+TTS), [SPEECH] is replaced with the [ASR] special token to trigger the transcription prediction as seen during training.

Text-to-Speech (TTS)

We find that prompting SPIRIT-LM with 10-shots leads to the best performance in TTS. We illustrate the prompting with a single example for few-shot learning:

[TEXT] *Input Text* 'stop'

[SPEECH] *Speech token sequence* <speech:STOP>

[TEXT] *Input Text* 'stop'

[SPEECH]

With <speech:STOP>, the spoken utterance "stop" tokenized into speech tokens[15]. For models trained with parallel TTS data (e.g. SPIRIT-LM-BASE +ASR+TTS), the token [SPEECH] is replaced with [TTS].

Intent Classification

For Intent Classification, we illustrate the prompting used in SPIRIT-LM-BASE with single example for few-shot:

[SPEECH] *Speech token sequence* [TEXT]

A:activate lights bedroom

[SPEECH] *Speech token sequence* [TEXT]

A:

For both ASR, TTS and Intent Classification, we postprocess the output of the model using the special tokens and beginning/end of sequence flags in order to extract the predicted text or speech sequence.

5.9.2 Construction of Few-Shot examples for Sentiment Continuation

We use S→T as an illustration, the identical process is applied to the remaining modality directions.

1. From the EXPRESSO-READ training set, we select only the speech samples where the waveform length exceeds 200,000, dividing each into two equal parts. The speech in the second segment is then transcribed.[16]

[15]For SPIRIT-LM-BASE, the spoken word "stop" is tokenized as [Hu481][Hu149][Hu40][Hu48][Hu315][Hu242][Hu428][Hu494][Hu75][Hu497][Hu188][Hu388][Hu109][Hu23][Hu338][Hu23][Hu481]

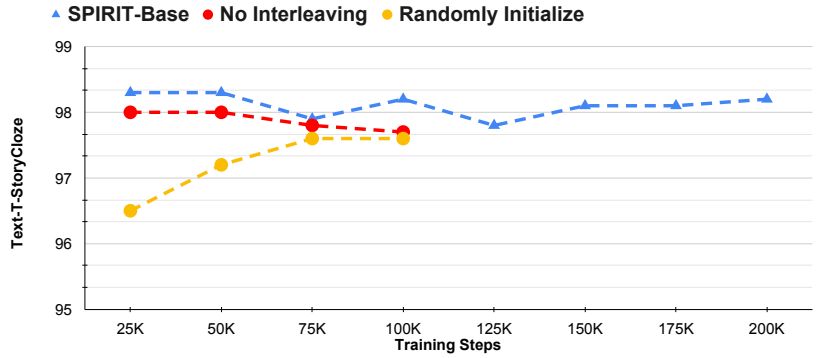
[16]The transcription is done by WHISPER-MEDIUM (Radford et al., 2023).

2. We apply the fine-tuned speech classifier and text classifier mentioned in 5.5.2.2 to the speech of the first segment and the transcription of the second segment, respectively. We retain only those pairs where the sentiment of the transcription in the second segment matches that of the speech in the first segment.

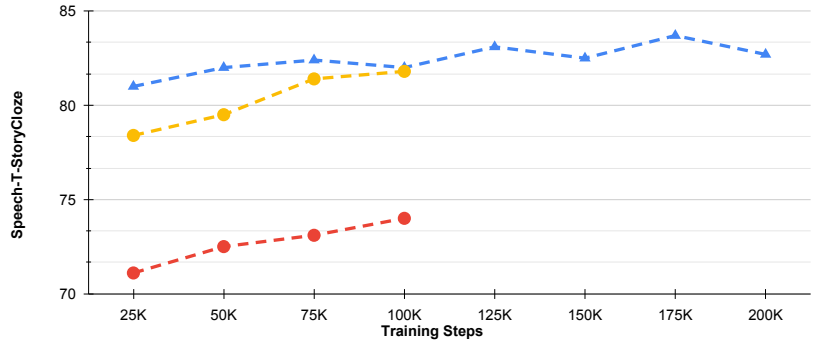
3. At the start of each run, we randomly select 3/6/9 samples from the above subset, ensuring a balanced distribution of samples for each sentiment. These samples are then combined to form the in-context prompt, which is reused for all subsequent iterations.

Model	Task	WUGGY↑		BLIMP↑		Topic-StoryCloze↑				StoryCloze↑			
		T	S	T	S	T	S	T→S	S→T	T	S	T→S	S→T
Previous Work													
GSLM (Lakhotia et al., 2021)		∅	65.4/64.8	∅	57.2/54.2	∅	56.3/66.6	∅	∅	∅	51.0/53.3	∅	∅
AudioLM (Borsos et al., 2023)		∅	– / 71.5	∅	**– / 64.7**	–	–	∅	∅	∅	–	∅	∅
Voxtlm (Maiti et al., 2023)		– / **80.3**	– / 66.1	– / **74.2**	– / 57.1	–	–	–	–	–	–	∅	∅
TWIST (Hassid et al., 2023)		∅	– / **74.5**	∅	– / 59.2	–	– / 76.4	∅	∅	∅	– / 55.4	∅	∅
SPIRIT-LM *variants*													
SPIRIT-LM-BASE		**95.1/80.3**	71.4/69.0	75.7/73.3	**63.2/58.3**	94.5/98.0	**69.2/82.9**	**66.6/72.7**	**83.8/88.6**	**76.6/79.4**	**56.2/61.0**	56.2/59.5	64.3/64.6
+ASR+TTS		94.5/76.8	71.8/68.7	74.3/71.7	62.4/57.2	93.1/97.7	69.1/81.6	66.0/71.6	81.6/86.1	75.3/77.4	55.5/59.9	55.5/58.8	63.5/63.5
Rope θ default		95.2/78.2	71.7/69.5	75.8/73.3	62.9/57.7	**94.5/98.2**	69.5/82.0	66.1/72.0	83.5/88.3	76.6/78.9	56.3/60.9	**56.4/59.8**	**64.1/65.5**
SPIRIT-LM-EXPRESSIVE		95.2/75.8	66.2/65.0	76.6/73.6	58.7/54.2	94.3/97.9	58.2/75.4	57.7/61.6	81.3/73.2	75.7/78.9	51.8/56.9	52.5/54.6	61.4/58.8
Parallel Data Training													
Word-level transcription		94.7/74.7	71.2/67.1	75.9/72.6	62.8/57.2	94.3/98.0	68.1/80.3	53.9/57.5	67.0/71.9	75.8/78.2	55.0/60.1	51.0/54.2	55.1/56.4
ASR+TTS		94.0/76.5	**72.6/69.8**	75.7/73.3	62.2/57.6	92.7/97.3	62.7/74.9	56.9/63.5	67.8/71.8	73.6/76.3	50.7/54.6	49.9/53.9	53.5/54.0
Unimodal Ablations													
Speech Only		67.4/67.1	71.8/69.5	54.1/53.7	63.0/58.0	49.7/54.8	62.2/72.9	48.3/52.2	49.0/49.4	48.2/53.7	51.0/54.8	48.1/52.6	49.2/49.3
Text Only		94.5/72.6	53.1/46.8	**77.3/73.9**	54.6/52.6	**94.5/98.2**	48.0/51.7	47.3/47.5	51.5/51.7	76.1/79.0	47.0/50.2	47.1/47.3	50.3/52.1
Cascade Topline													
(WHISPER) + LLAMA 2		– / 84.1	– / 79.2	– / 72.8	– / 71.6	– / 98.5	– / 94.76	– / 94.76	– / 94.76	– / 81.9	– / 75.7	– / 75.7	– / 75.7

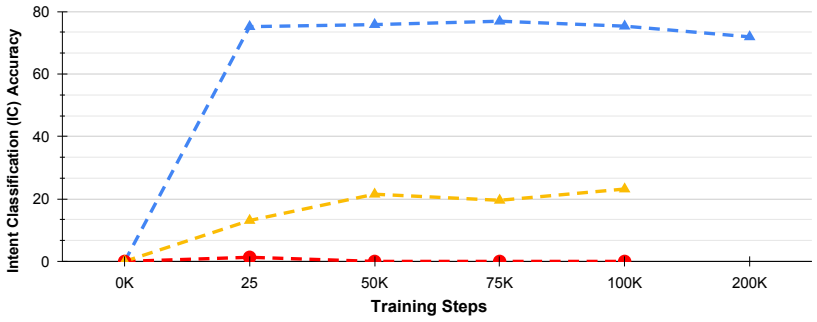
Tab. 5.10: Zero-shot Comprehension Evaluation in Speech (S) and Text (T). We report Accuracy / Accuracy-token for all the SPIRIT-LM models. Both metrics are based on selecting the hypothesis (among two choices) with the highest log-likelihood according to the model. The log-likelihood is based on the sum of each token likelihood in the sequence. The Accuracy is computed based on the prediction that maximizes the log-likelihood of the hypothesis. Accuracy-token adds a normalizing step of the log-likelihood by the number of tokens in the hypothesis. The related work performance (except GSLM) comes from the original published papers of each reported system. We recomputed the scores of GSLM on our metrics.



(a) Accuracy on Text T-StoryCloze (0-Shot)



(b) Accuracy on Speech T-StoryCloze (0-Shot)



(c) Accuracy on Intent Classification (30-Shot)

Fig. 5.6: Comparing SPIRIT-LM-BASE to a randomly initialized model trained in the same way and to a model trained with no Interleaving data. (i.e. the model is only trained on sequences of raw speech or raw text data without any interleaved aligned data.)

Discussion and Perspectives

6

6.1 General Contributions

Chapter 1 introduced SpeechLMs and discussed their new capabilities compared to more traditional cascaded systems: multichannel speech modeling (discussed in Chapter 3) and expressive speech modeling (discussed in Chapter 4). We also discussed different factors that could hinder SpeechLMs compared to TextLMs including the quality of speech units (discussed in Chapter 2) and the small scale of Speech Datasets and Language Models compared with Textual counterparts (discussed in Chapter 5).

In Chapter 2, we analyzed the importance of discretization in spoken language models. We compared discrete speech units and continuous speech features which were used to train SpeechLMs, and evaluted the LMs on zero-shot spoken language modeling metrics. We found that while discrete units yielded systematically superior results in lexical and syntactic zero-shot metrics, the gap between discrete and continuous representations was not large, suggesting that it is possible to learn lexical and syntactic information with continuous units only. We further analyzed the discrete speech units, and found that discretization indeed removes non-linguistic information (e.g. speaker information) from continuous speech features, which potentially helped the LM to focus on learning high-level semantic information from the language rather than focus on the acoustic levels. We also found that the phonetic quality of the units as mesured by the ABX metrics is a reliable predictor of the performance of LMs trained on these units. In addition, we also explored larger speech units using BPE-based methods as well as downsampling methods. We found that bitrate also has a huge impact on the quality of SpeechLMs.

In Chapter 3, we introduced dGSLM, an application of SpeechLMs to spoken dialogues. We found that by modeling the spoken dialogue as multi-channel audio, we could generate conversations with natural turn-takings as well as paralinguistic cues such as laughter or back-channeling. However, the model was not able to generate semantically coherent speech, which possibly came from the quality and granularity of speech units as well as the limited amount of the speech dataset. We further tried to improve the dGSLM model by fine-tuning it on a SpeechLM

pre-trained on a larger single-channel dataset and found that this helped to accelerate the training of dialogue LM, suggesting a transferability from LMs trained on large-scale speech datasets to conversational dialogues. This led to the need to have large-scale SpeechLMs trained on more datasets, which was studied in Chapter 5 (SPIRIT-LM).

In Chapter 4, we introduced EXPRESSO, a high-quality expressive speech dataset, along with an expressive resynthesis benchmark. We compared different discrete units for expressive speech resynthesis task: Encodec units (audio compression-based unit) and HuBERT units (SpeeechSSL-based unit that was studied in previous SpeechLMs). We found that Encodec units, despite having excellent resynthesis results, are not good at capturing phonetic information from speech (as measured by ABX and PNMI metrics), which is undesirable for SpeechLMs. We further confirmed this by training language models on Encodec and HuBERT units and evaluated their performances. We found that LM trained on HuBERT units achieved better results than LM trained on Encodec units in all zero-shot speech metrics. In addition, Encodec units encode speech information in an entangled fashion of phonetic content, speaker characteristics, pitch and expression, meaning that they cannot be used for speaker- and style-conditional speech generation. HuBERT units, on the other hand, only capture phonetic content but not pitch or expression, resulting in poor quality in pitch preservation and emotion preservation metrics. We further extended this with additional expressivity units, one for pitch, one for style to complement HuBERT units and enable jointly represent these dimensions in a disentangled fashion, which was later used in Chapter 5 (SPIRIT-LM-EXPRESSIVE).

In Chapter 5, we introduced SPIRIT-LM, a large language model that combines both speech and text. We find that training a LM on speech and text using an interleaved task helps the model to learn text and speech cross-modally. SPIRIT-LM had comparable results in zero-shot speech metrics compared with previous speech-only LMs, and achieved state-of-the-art results in spoken StoryCloze metrics while being almost on par with LLAMA 2 on text reasoning metrics like textual StoryCloze or MMLU. In addition, SPIRIT-LM is able to perform cross-modal few-shot learning on speech-text tasks such as ASR, TTS or Speech Intent Classification, and is able to generate speech and text cross-modally while preserving the expressivity contained in the speech. We additionally introduced a benchmark on speech-text sentiment preservation, which probes the capability of SpeechLMs to generate consistent expressivity both within and across modalities. We found that while the cascaded model can produce consistent expressivity in text-text generation, it is not capable of processing and producing the correlates of expressivity in the speech modality.

We found that our SPIRIT-LM-EXPRESSIVE model, however, can produce better than chance consistent expressivity both within and across modalities.

6.2 Towards a Unified Spoken Language System

Each piece of work in this thesis covers one important aspect of spoken language modeling with the ultimate aim to develop a unified spoken language system. SPIRIT-LM is the first attempt to integrate Speech LLMs, Text LLMs and Expressivity but more work is required to reach the final goal. In this section, I'll open up some research questions and a few possible directions that can be considered in order to achieve better spoken language systems.

6.2.1 On the Improvement of SpeechLMs

A first direction is to focus on **improving the quality of speech units**. In Chapter 2, we have found that speech unit quality is critical for the performance of SpeechLMs, and we have explored the improvements of speech units by training different speech encoders (CPC, HuBERT) and adjusting frame rates and number of units. It would be interesting to try other SpeechSSL models (wavLM, Chen et al., 2022; w2v-BERT, Chung et al., 2021; dinoSR, Liu et al., 2023), which have been claimed to be good at capturing phonetic information. We also explored methods to make the units *robust* to noise or disruptions with augmentation methods as in Gat et al. (2023), but would be worth trying other speaker-invariant methods for self-supervised speech features (ContentVec, Qian et al., 2022; Spin, Chang et al., 2023a) or normalizing speech units as in Lee et al. (2022b). It is worth noting that there is currently a trade-off between resynthesis quality and language modeling quality. Fine-grained units tend to have better resynthesis but perform poorly on language modeling (cf. poor performances of LM trained on Encodec units on zero-shot speech metrics). It is therefore natural to explore units that both work well for language modeling and resynthesis tasks. Finally, another possible direction for improving discrete units could be using *soft-discrete units*, i.e. using probability-based speech vectors (over a discrete codebook) instead of one-hot discrete vectors, which have been shown to benefit discrete-based TTS models (Niekerk et al., 2022).

In the same direction, **improving the TTS module** could also help to improve the quality of speech generation. We mainly focused on using HifiGAN in this thesis, which is only a speech vocoder, but other TTS models could be considered

like FastSpeech 2 (Ren et al., 2022), VITS (Kim et al., 2021) or diffusion-based models such as Voicebox (Le et al., 2023). Speech codec-based methods could also be considered, where the speech units are first converted to speech codec units (SoundStream, Zeghidour et al., 2021; Encodec, Défossez et al., 2022) before being translated to audio waveform (Borsos et al., 2023; Kharitonov et al., 2023; Wang et al., 2023a).

Another topic worth exploring is the **disentanglement of speech units**. In Chapters 4 and 5, we found that speech can be decomposed into disentangled units representing phonetic content, prosodic, and expressive style, respectively, which can be fed to SpeechLMs achieving better results than units like Encoder which contain the same information in an entangled fashion. One could speculate that the success of disentanglement comes into enabling a low bitrate representation, where the total bitrate is the sum of the bitrates of the individual channels, instead of being the product in the case of an entangled representation. Yet, the units we proposed are not perfect. It is possible that speaker or pitch information is still present in SSL representations (e.g. Seyssel et al., 2022). It thus would be useful to improve further speech disentanglement to achieve even better compression quality and language modeling. First, each component of the disentangled units (content, pitch, style) could be improved by making them more invariant to the other sources of information (as well as from speaker identity). Second, a single multitask speech tokenizer model could be trained via a reconstruction loss to directly decompose speech into different streams of units, each corresponding to one type of information. This is similar to Zhang et al. (2024), but they did not try to disentangle further non-semantic units to other information contained in the speech (e.g. pitch, style, speaker).

On the language modeling part, one question is **what is the best way to integrate Speech Units into SpeechLMs?** When speech is represented as one single sequence of units, we can simply treat them as text tokens and train language models on speech units (Lakhotia et al., 2021). However, it's more complicated when speech units are composed of multiple unit types (e.g. HuBERT, pitch, duration units in pGSLM, Kharitonov et al., 2022b) with possibly different sampling rates (e.g. HuBERT, pitch, style units in SpiRit-LM). pGSLM dealt with this by training a multi-stream transformer LM with multiple streams of input and multiple output heads predicting each stream. In SpiRit-LM, we simply interleaved different unit types and considered them as one stream of speech units. However, we found that doing this can be harmful to the SpeechLM as the unit rate increases with additional units. Further work is therefore needed to determine the best way to integrate speech units into SpeechLMs.

It is also worth re-visiting the question: **Is using continuous speech features always bad for SpeechLMs**? Along this thesis, we have been using discrete speech representations as input for SpeechLMs for their simplicity and efficiency. In Chapter 2, we found that training language models on continuous speech features is not as good as discrete speech units on zero-shot speech metrics. However, the study was mainly done with CPC features and encoder-based LMs (e.g. BERT). Recently, Algayres et al. (2023) found that autoregressive LMs trained on word-size continuous speech features could achieve good performances in zero-shot speech metrics. However, one specific problem with training autoregressive LMs on continuous speech features is how to perform generation. Algayres et al. (2023) employed a pre-defined lexical vocab and looked for the nearest neighbors of the generated features in the lexical space. Nachmani et al. (2023) proposed a Spoken Question Answering system (Spectron) that uses spectrograms to represent speech in both input and output. They used text as an intermediate proxy to help the language model learn speech content (speech prompt $\rightarrow$ text prompt $\rightarrow$ text continuation $\rightarrow$ speech continuation) and employed both a cross-entropy loss to predict the next words and a regression loss to predict the speech continuation output.

Scaling the language model is a viable direction to improve the performances of SpeechLMs. In SPIRIT-LM, we have just experimented with 7B models. Going for bigger models (30B, 70B, 150B) could improve the spoken as well as cross-modal aspects even more, as they have shown to have excellent results and possess many emergent abilities in TextLLMs (Wei et al., 2022a). However, scaling speech and multimodal LMs could be challenging as they require a lot of engineering effort (e.g., stabilizing training, introducing smoothly new modalities into the models, etc.). This also comes with larger speech (and speech-text) datasets that would not sacrifice quality, which will be discussed shortly.

Integration of an end-to-end SpeechLM is one of our ultimate goals. Most SpeechLMs at the moment rely on independent components (Speech Encoder, Language Model, Speech Decoder) that involve multiple training stages. Having an end-to-end model not only facilitates the training efforts but also helps each subcomponent to be optimized toward the final objective (e.g. speech tokenizer focused on producing speech units with excellent linguistic information). In our initial experiments, we found that this goal is not easy to achieve as the model tends to collapse (i.e., producing only zero tokens). One possible solution could be first to train each component with independent objectives and then start to train the whole model. We can also think of ways to add multiple objectives in order to avoid mode collapse during training.

6.2.2 Rethinking Evaluation Metrics and Datasets

Is ABX always a good metric to evaluate the quality of speech units? We have been relying on ABX to evaluate the quality of speech units, and we have seen in Chapter 2 that it reflects the linguistic quality of units and, eventually, the language modeling performances. However, we also learned that ABX is not an absolute indicator of how well the units perform on language modeling tasks. For example, HuBERT 50Hz units have much better ABX than HuBERT 25Hz units, but perform less well on spoken language modeling tasks. It is, however, safe to say that a low ABX means that the units are more phonetic-like, and it is most likely comparable if the units are in the same condition (e.g. comparing units that have same rate, vocab size). There are also other metrics used to evaluate speech unit quality at phonetic levels such as PNMI (as in Hsu et al., 2021a and Nguyen et al., 2023a) or without-context ABX (Hallap et al., 2023). It is also worth noting that there are more metrics that focus on higher-level linguistic information that can be considered such as ABX_{sem} and ABX_{POS} (Algayres et al., 2022) which have been used in Algayres et al. (2023) to evaluate speech embeddings in terms of lexical semantic and Part-Of-Speech (POS) tagging.

It is also important to think of having **more and better metrics for SpeechLMs**. The current evaluation metrics for SpeechLMs cover a wide range of linguistic levels (from phonetic to commonsense reasoning). However, they are still very limited compared with the evaluation metrics for TextLLMs (Liang et al., 2023; Srivastava et al., 2023). An interesting direction could be to *speechify* appropriate text benchmarks and create a benchmark for SpeechLMs using TTS systems. It would also be interesting to evaluate the performances of SpechLMs on self-supervised speech benchmarks like SUPERB (Yang et al., 2021). Of course, if most speech evaluation datasets are generated with TTS systems, this could bias the SpeechLMs towards synthesized speech, and not reflect their performance on real speech input. It is, therefore, important to build evaluation datasets based on real speech. Finally, as more and more SpeechLMs are produced, it could be beneficial to have a unified evaluation pipeline for SpeechLMs, which is currently missing.

Data quality is an important factor to consider when scaling speech LMs on increasingly larger corpora. We currently do not know what kinds of tradeoffs exist between data quality and quantity when it comes to speech LMs. In Chapter 4, we found that it is critical to have good-quality datasets for the speech synthesis as well as the k-means modules. However, we have found (in Section 2.6 and later in Hassid et al., 2023) that having more speech datasets is beneficial for the language models. Later, in Chapter 5, we found that aligned speech-text data, although of low quality,

is crucial not only for multimodal speech-text language modeling but also for speech language modeling. More research on the scaling laws regarding data quantity and data quality are needed for SpeechLMs.

Multilinguality. SPIRIT-LM was trained on limited multilingual datasets and showed some promising results with other languages. However, we did not really probe the models on multilingual tasks. This comes from several factors: First, English is still dominant in the training datasets compared to other languages. In addition, our SLM metrics (such as sWUGGY, sBLIMP, StoryCloze) only exist in English, and would need to be translated to evaluation speechLMs in other languages. Finally, we would need to find or build open-source expressive multilingual TTS datasets to train the vocoder. It is necessary to address all of these roadblocks to unlock the possibility of developing a high-quality multilingual speech LM.

6.2.3 Towards Better Dialogue Systems

In Chapter 3, we discussed the need for a good SpeechLM to fine-tune the dialogue datasets and partly solved this problem in Chapter 5. Therefore, a straightforward direction for future work would be to **fine-tune SPIRIT-LM on dialogue datasets**. The fine-tuning could be done either on instruction-tuning datasets (Ouyang et al., 2022) to create a reliable and helpful spoken chatbot system or on Fisher-like datasets to create a conversational agent or on a mix of both. The model should be able to transfer knowledge to dialogue-based datasets and perform well on these tasks. The advantage of SPIRIT-LM is it's cross-modal abilities, allowing the fine-tuning to be done not only using speech-only instruction datasets (as in Zhang et al., 2023a) but also text datasets and speech-text datasets.

A question that arises from this is **how to deal with multichannel dialogues**. The DLM model in Chapter 3 uses a dual-tower transformer architecture with cross-attention, and has been designed to be adaptable to single-channel speech. Inversely, adapting the transformer decoder architecture of SPIRIT-LM to multichannel speech is not straightforward, and may require some modifications in architecture. In addition, SPIRIT-LM utilizes deduplicated speech units, while the DLM model expects duplicated speech units in inputs, which may cause inconsistency in model fine-tuning. Another way to deal with multichannel dialogues is to mix them to single-channel speech. This can be done in two fashions: The first approach consists of mixing the multichannel audio into single-channel audio. The second approach considers dialogues as consecutive turns of speech and concatenates the segments of each channel into one single stream of audio. The advantage of the first approach

is that it allows the modelization of natural turn-taking with overlapping and back-channeling. However, the synthesis model will struggle to disentangle the speakers in overlapping segments. The second approach permits to efficiently modelize the content of the dialogue and synthesize multiple people in the conversation. It will, however, remove the naturalness of the conversation. This later approach has been shown to work well in recent work (SpeechGPT, Zhang et al., 2023a; USDM, Kim et al., 2024)

Safe Speech System. Responsible AI is a vital aspect of text-LLM systems. Spoken models can contain even more risks as i) speech contains biometric information from the speakers ii) spoken language is much more colloquial compared with written language. Developing a safe spoken system is therefore crucial but also very challenging. Among the possible directions could be fine-tuning the models with human instructions and preference speech datasets (Ouyang et al., 2022) so that the model answer could be more safe and instructive. In addition, adding watermarking to generated speech so that they can be easily detected is an interesting approach (San Roman et al., 2024). Finally, biometric-free model can be obtained by using disentangled speech units (as in SPIRIT-LM-EXPRESSIVE) that remove any speaker information before feeding to the SpeechLMs.

Going towards **an Embedded Dialogue System** is also an ultimate goal of spoken language modeling. It requires however solving many questions in order to achieve this goal. We can first think of an interactive dialogue agent that can interact with humans in real time. Following Communication et al. (2023a), this can be solved using streaming transformer architecture (Ma et al., 2023). Another interesting direction of an embedded dialogue system is the integration of visual features. This could be done by using features obtained from self-supervised audiovisual models (Hsu et al., 2023b; Shi et al., 2022).

6.3 It is only the Start of a Journey

This thesis was such a beautiful and memorable journey for me, and I am really grateful to be a part of it! However, I know that it is just the beginning of the development of SpeechLMs, and many things still need to be done in order to have a powerful spoken language system.

I hope that this thesis will be somewhat useful for the community, and I really look forward to excellent speech systems in the future!!

Bibliography

Adigwe, Adaeze, Noé Tits, Kevin El Haddad, Sarah Ostadabbas, and Thierry Dutoit (2018). "The emotional voices database: Towards controlling the emotion dimension in voice generation systems". In: *arXiv preprint arXiv:1806.09514* (cit. on pp. 85, 89, 118).

Adiwardana, Daniel, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le (2020). "Towards a Human-like Open-Domain Chatbot". In: *CoRR* abs/2001.09977. arXiv: 2001.09977 (cit. on p. 61).

Agirre, Eneko, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa (2009). "A study on similarity and relatedness using distributional and wordnet-based approaches". In: (cit. on p. 25).

Agostinelli, Andrea, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank (2023). *MusicLM: Generating Music From Text*. arXiv: 2301.11325 [cs.SD] (cit. on pp. 10, 33).

Alayrac, Jean-Baptiste, Jeff Donahue, Pauline Luc, et al. (2022). "Flamingo: a Visual Language Model for Few-Shot Learning". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (cit. on p. 10).

Algayres, Robin, Yossi Adi, Tu Anh Nguyen, Jade Copet, Gabriel Synnaeve, Benoit Sagot, and Emmanuel Dupoux (2023). *Generative Spoken Language Model based on continuous word-sized audio tokens*. arXiv: 2310.05224 [cs.CL] (cit. on pp. 101, 133, 134).

Algayres, Robin, Tristan Ricoul, Julien Karadayi, Hugo Laurençon, Salah Zaiem, Abdelrahman Mohamed, Benoît Sagot, and Emmanuel Dupoux (2022). "DP-Parse: Finding Word Boundaries from Raw Speech with an Instance Lexicon". In: *Transactions of the Association for Computational Linguistics* 10. Ed. by Brian Roark and Ani Nenkova, pp. 1051–1065 (cit. on p. 134).

Alim, Sabur Ajibola and Nahrul Khair Alang Rashid (2018). "Some Commonly Used Speech Feature Extraction Algorithms". In: *From Natural to Artificial Intelligence*. Ed. by Ricardo Lopez-Ruiz. Rijeka: IntechOpen. Chap. 1 (cit. on p. 12).

Amodei, Dario, Sundaram Ananthanarayanan, Rishita Anubhai, et al. (2016). "Deep speech 2: end-to-end speech recognition in English and mandarin". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, 173–182 (cit. on p. 13).

Ang, Jeremy, Rajdip Dhillon, Ashley Krupski, Elizabeth Shriberg, and Andreas Stolcke (2002). "Prosody-based automatic detection of annoyance and frustration in human-computer dialog". In: *INTERSPEECH* (cit. on p. 61).

Ardila, Rosana, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber (May 2020). "Common Voice: A Massively-Multilingual Speech Corpus". English. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 4218–4222 (cit. on pp. 90, 106).

Baevski, Alexei, Michael Auli, and Abdelrahman Mohamed (2020a). "Effectiveness of Self-Supervised Pre-Training for ASR". In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. Barcelona, Spain + Online, pp. 7694–7698 (cit. on pp. 19, 39, 42).

Baevski, Alexei, Steffen Schneider, and Michael Auli (2020b). "vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations". In: *Proc. Int. Conf. Learn. Represent. (ICLR)*. Addis Ababa, Ethiopia (cit. on pp. 16, 20, 39).

Baevski, Alexei, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli (2020c). "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations". In: *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*. Vol. 33. Online, pp. 12449–12460 (cit. on pp. 4, 12, 13, 16, 37, 38, 42, 60, 89, 103).

Bai, Yuntao, Andy Jones, Kamal Ndousse, et al. (2022). *Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback*. arXiv: 2204.05862 [cs.CL] (cit. on p. 123).

Baker, Simon, Roi Reichart, and Anna Korhonen (2014). "An unsupervised model for instance level subcategorization acquisition". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 278–289 (cit. on p. 25).

Bengio, Yoshua, Réjean Ducharme, and Pascal Vincent (2000). "A Neural Probabilistic Language Model". In: *Advances in Neural Information Processing Systems*. Ed. by T. Leen, T. Dietterich, and V. Tresp. Vol. 13. MIT Press (cit. on p. 8).

Borgholt, Lasse, Jakob Drachmann Havtorn, Joakim Edin, Lars Maaløe, and Christian Igel (2022). "A Brief Overview of Unsupervised Neural Speech Representation Learning". In: *arXiv preprint arXiv:2203.01829* (cit. on pp. 59, 61).

Borsos, Zalán, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. (2023). "Audiolm: a language modeling approach to audio generation". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (cit. on pp. 10, 32, 59, 79, 85, 96, 98, 101, 104, 106, 112, 126, 132).

Bowman, Samuel R. (2023). *Eight Things to Know about Large Language Models*. arXiv: 2304.00612 [cs.CL] (cit. on p. 9).

Brady, Paul T (1968). "A statistical analysis of on-off patterns in 16 conversations". In: *Bell System Technical Journal* 47.1, pp. 73–91 (cit. on pp. 59, 75).

Bredin, Hervé, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill (2020). "pyannote.audio: neural building blocks for speaker diarization". In: *ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing*. Barcelona, Spain (cit. on p. 71).

Brown, Peter F., Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer (1992). "Class-Based *n*-gram Models of Natural Language". In: *Computational Linguistics* 18.4, pp. 467–480 (cit. on pp. 54, 55, 165).

Brown, Tom, Benjamin Mann, Nick Ryder, et al. (2020). "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 1877–1901 (cit. on pp. 3, 9, 36, 39, 100, 101).

Bruni, Elia, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran (2012). "Distributional semantics in technicolor". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 136–145 (cit. on p. 25).

Cassell, Justine, Tim Bickmore, Lee Campbell, Hannes Vilhjálmsson, and Hao Yan (2001). "Human Conversation as a System Framework: Designing Embodied Conversational Agents". In: *Embodied Conversational Agents*. Cambridge, MA, USA: MIT Press, 29–63 (cit. on p. 62).

Chang, Heng-Jui, Alexander H. Liu, and James Glass (2023a). "Self-supervised Fine-tuning for Improved Content Representations by Speaker-invariant Clustering". In: *Proc. Interspeech* (cit. on p. 131).

Chang, Kai-Wei, Wei-Cheng Tseng, Shang-Wen Li, and Hung yi Lee (2022). "An Exploration of Prompt Tuning on Generative Spoken Language Model for Speech Processing Tasks". In: *Proc. Interspeech 2022*, pp. 5005–5009 (cit. on p. 33).

Chang, Kai-Wei, Yu-Kai Wang, Hua Shen, Iu thing Kang, Wei-Cheng Tseng, Shang-Wen Li, and Hung yi Lee (2023b). *SpeechPrompt v2: Prompt Tuning for Speech Classification Tasks*. arXiv: 2303.00733 [eess.AS] (cit. on pp. 33, 103, 111).

Chen, Sanyuan, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei (Oct. 2022). "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing". In: *IEEE Journal of Selected Topics in Signal Processing* 16.6, 1505–1518 (cit. on pp. 12, 103, 131).

Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton (2020). "A Simple Framework for Contrastive Learning of Visual Representations". In: *CoRR* abs/2002.05709. arXiv: 2002.05709 (cit. on p. 15).

Chou, Ju-Chieh, Chung-Ming Chien, Wei-Ning Hsu, Karen Livescu, Arun Babu, Alexis Conneau, Alexei Baevski, and Michael Auli (2023). "Toward Joint Language Modeling for Speech Units and Text". In: *arXiv preprint arXiv:2310.08715* (cit. on pp. 55, 104, 105).

Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, et al. (2022). *PaLM: Scaling Language Modeling with Pathways*. arXiv: 2204.02311 [cs.CL] (cit. on pp. 9, 101).

Chung, Yu-An and James Glass (2020). "Generative pre-training for speech with autoregressive predictive coding". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 3497–3501 (cit. on p. 60).

– (2018). "Speech2Vec: A Sequence-to-Sequence Framework for Learning Word Embeddings from Speech". In: *Proc. Interspeech 2018*, pp. 811–815 (cit. on p. 25).

Chung, Yu-An, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu (2021). “W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training”. In: *IEEE Autom. Speech Recognit. Understanding Workshop (ASRU)*. Cartagena, Colombia, pp. 244–250 (cit. on pp. 4, 37, 85, 103, 104, 112, 131).

Cieri, Christopher, David Miller, and Kevin Walker (2004). “The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text”. In: *LREC* (cit. on pp. 61, 62, 67, 89, 106).

Clark, Kevin, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning (2020). “ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators”. In: *ICLR* (cit. on p. 9).

Clifton, Ann, Sravana Reddy, Yongze Yu, Aasish Pappu, Rezvaneh Rezapour, Hamed Bonab, Maria Eskevich, Gareth Jones, Jussi Karlgren, Ben Carterette, and Rosie Jones (Dec. 2020). “100,000 Podcasts: A Spoken English Document Corpus”. In: *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 5903–5917 (cit. on pp. 90, 106).

Communication, Seamless, Loïc Barrault, Yu-An Chung, et al. (2023a). *Seamless: Multilingual Expressive and Streaming Speech Translation* (cit. on pp. 32, 121, 123, 136).

– (2023b). *SeamlessM4T: Massively Multilingual & Multimodal Machine Translation*. arXiv: 2308.11596 [cs.CL] (cit. on pp. 32, 121).

Copet, Jade, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez (2023). “Simple and Controllable Music Generation”. In: *Thirty-seventh Conference on Neural Information Processing Systems* (cit. on p. 33).

Costa-jussà, Marta R, Eric Smith, Christophe Ropers, Daniel Licht, Jean Maillard, Javier Ferrando, and Carlos Escolano (2023). “Toxicity in multilingual machine translation at scale”. In: *EMNLP* (cit. on p. 121).

Davis, K. H., R. Biddulph, and S. Balashek (Nov. 1952). “Automatic Recognition of Spoken Digits”. In: *The Journal of the Acoustical Society of America* 24.6, pp. 637–642. eprint: https://pubs.aip.org/asa/jasa/article-pdf/24/6/637/18730880/637_1_online.pdf (cit. on p. 13).

Deshpande, Ameet, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan (2023). *Toxicity in ChatGPT: Analyzing Persona-assigned Language Models*. arXiv: 2304.05335 [cs.CL] (cit. on pp. 10, 121).

Desplanques, Brecht, Jenthe Thienpondt, and Kris Demuynck (2020). “ECAPA-TDNN: Emphasized Channel Attention, propagation and aggregation in TDNN based speaker verification”. In: *Interspeech 2020*, pp. 3830–3834 (cit. on p. 12).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proc. North Am. Ch. Assoc. Comput. Linguist. (NAACL)*. Vol. 1. Minneapolis, Minnesota, USA, pp. 4171–4186 (cit. on pp. 9, 16, 20, 36, 39, 41).

Dhariwal, Prafulla, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever (2020). "Jukebox: A Generative Model for Music". In: *arXiv preprint arXiv:2005.00341* (cit. on p. 33).

Dieleman, Sander, Charlie Nash, Jesse Engel, and Karen Simonyan (2021). "Variable-rate discrete representation learning". In: *arXiv preprint arXiv:2103.06089* (cit. on p. 61).

Dudley, Homer (1940). "The Carrier Nature of Speech". In: *Bell System Technical Journal* 19.4, pp. 495–515. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1940.tb00843.x (cit. on p. 13).

Dunbar, Ewan, Robin Algayres, Julien Karadayi, Mathieu Bernard, Juan Benjumea, Xuan-Nga Cao, Lucie Miskic, Charlotte Dugrain, Lucas Ondel, Alan W. Black, Laurent Besacier, Sakriani Sakti, and Emmanuel Dupoux (2019). *The Zero Resource Speech Challenge 2019: TTS without T*. arXiv: 1904.11469 [cs.CL] (cit. on pp. 17, 23).

Dunbar, Ewan, Mathieu Bernard, Nicolas Hamilakis, Tu Anh Nguyen, Maureen de Seyssel, Patricia Rozé, Morgane Rivière, Eugene Kharitonov, and Emmanuel Dupoux (2021). "The Zero Resource Speech Challenge 2021: Spoken Language Modelling". In: *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*. Brno, Czech Republic, pp. 1574–1578 (cit. on pp. 37, 39).

Dunbar, Ewan, Xuan Nga Cao, Juan Benjumea, Julien Karadayi, Mathieu Bernard, Laurent Besacier, Xavier Anguera, and Emmanuel Dupoux (2017). *The Zero Resource Speech Challenge 2017*. arXiv: 1712.04313 [cs.CL] (cit. on pp. 17, 23).

Dunbar, Ewan, Julien Karadayi, Mathieu Bernard, Xuan-Nga Cao, Robin Algayres, Lucas Ondel, Laurent Besacier, Sakriani Sakti, and Emmanuel Dupoux (2020). "The Zero Resource Speech Challenge 2020: Discovering discrete subword and word units". In: *CoRR* abs/2010.05967. arXiv: 2010.05967 (cit. on pp. 17, 23).

Duncan, Starkey (1972). "Some signals and rules for taking speaking turns in conversations." In: *Journal of personality and social psychology* 23.2, p. 283 (cit. on p. 61).

Duquenne, Paul-Ambroise, Hongyu Gong, Benoît Sagot, and Holger Schwenk (Dec. 2022). "T-Modules: Translation Modules for Zero-Shot Cross-Modal Machine Translation". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 5794–5806 (cit. on p. 12).

Duquenne, Paul-Ambroise, Kevin Heffernan, Alexandre Mourachko, Benoît Sagot, and Holger Schwenk (2023). *SONAR EXPRESSIVE: Zero-shot Expressive Speech-to-Speech Translation* (cit. on pp. 95, 96, 108, 167).

Défossez, Alexandre, Jade Copet, Gabriel Synnaeve, and Yossi Adi (2022). "High Fidelity Neural Audio Compression". In: *arXiv preprint arXiv:2210.13438* (cit. on pp. 33, 85, 86, 90, 104, 132).

Ekstedt, Erik and Gabriel Skantze (2020). "TurnGPT: a transformer-based language model for predicting turn-taking in spoken dialog". In: *arXiv preprint arXiv:2010.10874* (cit. on p. 62).

Elkahky, Ali, Wei-Ning Hsu, Paden Tomasello, Tu-Anh Nguyen, Robin Algayres, Yossi Adi, Jade Copet, Emmanuel Dupoux, and Abdelrahman Mohamed (2023). "Do Coarser Units Benefit Cluster Prediction-Based Speech Pre-Training?" In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5 (cit. on pp. 54, 165).

Evain, Solène, Ha Nguyen, Hang Le, Marcely Zanon Boito, Salima Mdhaffar, Sina Alisamir, Ziyi Tong, Natalia Tomashenko, Marco Dinarelli, Titouan Parcollet, Alexandre Allauzen, Yannick Estève, Benjamin Lecouteux, François Portet, Solange Rossato, Fabien Ringeval, Didier Schwab, and Laurent Besacier (2021). "LeBenchmark: A Reproducible Framework for Assessing Self-Supervised Representation Learning from Speech". In: *Proc. Interspeech 2021*, pp. 1439–1443 (cit. on p. 17).

Faruqui, Manaal and Dilek Hakkani-Tür (2021). "Revisiting the Boundary between ASR and NLU in the Age of Conversational Dialog Systems". In: *CoRR* abs/2112.05842. arXiv: 2112.05842 (cit. on p. 61).

Fathullah, Yassir, Chunyang Wu, Egor Lakomkin, Junteng Jia, Yuan Shangguan, Jay Mahadeokar, Ozlem Kalinli, Christian Fuegen, and Mike Seltzer (2023). *Towards General-Purpose Speech Abilities for Large Language Models Using Unpaired Data*. arXiv: 2311.06753 [cs.CL] (cit. on p. 104).

Galvez, Daniel, Greg Diamos, Juan Torres, Keith Achorn, Juan Cerón, Anjali Gopi, David Kanter, Max Lam, Mark Mazumder, and Vijay Janapa Reddi (2021). "The People's Speech: A Large-Scale Diverse English Speech Recognition Dataset for Commercial Usage". In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. Ed. by J. Vanschoren and S. Yeung. Vol. 1 (cit. on p. 90).

Gat, Itai, Felix Kreuk, Tu Anh Nguyen, Ann Lee, Jade Copet, Gabriel Synnaeve, Emmanuel Dupoux, and Yossi Adi (July 2023). "Augmentation Invariant Discrete Representation for Generative Spoken Language Modeling". In: *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*. Ed. by Elizabeth Salesky, Marcello Federico, and Marine Carpuat. Toronto, Canada (in-person and online): Association for Computational Linguistics, pp. 465–477 (cit. on pp. 56, 85, 106, 131, 165).

Gerz, Daniela, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen (2016). "Simverb-3500: A large-scale evaluation set of verb similarity". In: *arXiv preprint arXiv:1608.00869* (cit. on p. 25).

Gillick, Jon, Wesley Deng, Kimiko Ryokai, and David Bamman (2021). "Robust Laughter Detection in Noisy Environments". In: *Proc. Interspeech 2021*, pp. 2481–2485 (cit. on p. 72).

Godais, Gaël, Tal Linzen, and Emmanuel Dupoux (Jan. 2017). "Comparing Character-level Neural Language Models Using a Lexical Decision Task". In: pp. 125–130 (cit. on p. 23).

Gravano, Agustín and Julia Hirschberg (2011). "Turn-taking cues in task-oriented dialogue". In: *Computer Speech & Language* 25.3, pp. 601–634 (cit. on p. 61).

Graves, Alex (2014). *Generating Sequences With Recurrent Neural Networks*. arXiv: 1308.0850 [cs.NE] (cit. on p. 8).

Gulati, Anmol, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang (2020). *Conformer: Convolution-augmented Transformer for Speech Recognition*. arXiv: 2005.08100 [eess.AS] (cit. on p. 13).

Halawi, Guy, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren (2012). "Large-scale learning of word relatedness with constraints". In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1406–1414 (cit. on p. 25).

Hallap, Mark, Emmanuel Dupoux, and Ewan Dunbar (2023). "Evaluating context-invariance in unsupervised speech representations". In: *Proc. INTERSPEECH 2023*, pp. 2973–2977 (cit. on p. 134).

Hartmann, Jochen, Mark Heitmann, Christina Schamp, and Oded Netzer (2021). "The Power of Brand Selfies". In: *Journal of Marketing Research* (cit. on p. 119).

Hashimoto, Tatsunori B., Hugh Zhang, and Percy Liang (June 2019). "Unifying Human and Statistical Evaluation for Natural Language Generation". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1689–1701 (cit. on p. 27).

Hassid, Michael, Tal Remez, Tu Anh Nguyen, Itai Gat, Alexis Conneau, Felix Kreuk, Jade Copet, Alexandre Defossez, Gabriel Synnaeve, Emmanuel Dupoux, et al. (2023). "Textually Pretrained Speech Language Models". In: *arXiv preprint arXiv:2305.13009* (cit. on pp. 25, 32, 56, 96, 99, 101, 104–106, 111, 126, 134, 165).

Heldner, Mattias and Jens Edlund (2010). "Pauses, gaps and overlaps in conversations". In: *Journal of Phonetics* 38.4, pp. 555–568 (cit. on pp. 59, 75).

Hendrycks, Dan, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt (2021). "Measuring Massive Multitask Language Understanding". In: *International Conference on Learning Representations* (cit. on pp. 105, 111).

Hill, Felix, Roi Reichart, and Anna Korhonen (2015). "Simlex-999: Evaluating semantic models with (genuine) similarity estimation". In: *Computational Linguistics* 41.4, pp. 665–695 (cit. on p. 25).

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-Term Memory". In: *Neural Comput.* 9.8, 1735–1780 (cit. on pp. 8, 15).

Hoffmann, Jordan, Sebastian Borgeaud, Arthur Mensch, et al. (2022). *Training Compute-Optimal Large Language Models*. arXiv: 2203.15556 [cs.CL] (cit. on pp. 9, 101).

Holtzman, Ari, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi (2020). "The Curious Case of Neural Text Degeneration". In: *International Conference on Learning Representations* (cit. on p. 119).

Hsu, Ming-Hao, Kai-Wei Chang, Shang-Wen Li, and Hung yi Lee (2023a). *An Exploration of In-Context Learning for Speech Language Model*. arXiv: 2310.12477 [eess.AS] (cit. on p. 33).

Hsu, Wei-Ning, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed (2021a). "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units". In: *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 29, 3451–3460 (cit. on pp. 4, 12, 16, 37, 38, 42, 56, 60, 63, 68, 85, 86, 89, 103–105, 134, 165).

Hsu, Wei-Ning, David Harwath, Tyler Miller, Christopher Song, and James Glass (Aug. 2021b). "Text-Free Image-to-Speech Synthesis Using Learned Segmental Units". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Online: Association for Computational Linguistics, pp. 5284–5300 (cit. on p. 21).

Hsu, Wei-Ning, Tal Remez, Bowen Shi, Jacob Donley, and Yossi Adi (2023b). "ReVISE: Self-Supervised Speech Resynthesis With Visual Input for Universal and Generalized Speech Regeneration". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18795–18805 (cit. on pp. 32, 90, 136).

Hsu, Wei-Ning, Anuroop Sriram, Alexei Baevski, Tatiana Likhomanenko, Qiantong Xu, Vineel Pratap, Jacob Kahn, Ann Lee, Ronan Collobert, Gabriel Synnaeve, and Michael Auli (2021c). "Robust wav2vec 2.0: Analyzing Domain Shift in Self-Supervised Pre-Training". In: *Proc. Interspeech 2021*, pp. 721–725 (cit. on p. 69).

Huang, Xuedong, Alex Acero, Hsiao-Wuen Hon, and Raj Reddy (2001). *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. 1st. USA: Prentice Hall PTR (cit. on p. 3).

Ito, Keith and Linda Johnson (2017). *The LJ Speech Dataset*. `https://keithito.com/LJ-Speech-Dataset/` (cit. on pp. 21, 89).

Jang, Eric, Shixiang Gu, and Ben Poole (2017). "Categorical Reparameterization with Gumbel-Softmax". In: *International Conference on Learning Representations* (cit. on p. 16).

Jelinek, F. (1976). "Continuous speech recognition by statistical methods". In: *Proceedings of the IEEE* 64.4, pp. 532–556 (cit. on p. 12).

Jelinek, Frederick and Robert L. Mercer (1980). "Interpolated estimation of Markov source parameters from sparse data". In: *Proceedings of the Workshop on Pattern Recognition in Practice* (cit. on p. 8).

Juang, B. and Lawrence Rabiner (Jan. 2005). "Automatic Speech Recognition - A Brief History of the Technology Development". In: (cit. on p. 13).

Jurafsky, Daniel and James H. Martin (2009). *Speech and Language Processing (2nd Edition)*. USA: Prentice-Hall, Inc. (cit. on p. 7).

Kahn, J., M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux (2020). "Libri-Light: A Benchmark for ASR with Limited or No Supervision". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7669–7673 (cit. on pp. 23, 63, 90).

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei (2020). *Scaling Laws for Neural Language Models*. arXiv: 2001.08361 [cs.LG] (cit. on pp. 9, 101).

Katz, S. (1987). "Estimation of probabilities from sparse data for the language model component of a speech recognizer". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 35.3, pp. 400–401 (cit. on p. 8).

Keuleers, Emmanuel and Marc Brysbaert (2010). "Wuggy: A multilingual pseudoword generator". In: *Behavior research methods* 42.3, pp. 627–633 (cit. on p. 24).

Kharitonov, Eugene, Jade Copet, Kushal Lakhotia, Tu Anh Nguyen, Paden Tomasello, Ann Lee, Ali Elkahky, Wei-Ning Hsu, Abdelrahman Mohamed, Emmanuel Dupoux, and Yossi Adi (2022a). "textless-lib: a Library for Textless Spoken Language Processing". In: *Proc. North Am. Ch. Assoc. Comput. Linguist. (NAACL)*. Hybrid: Seattle, Washington, USA + Online, pp. 1–9 (cit. on pp. 48, 90).

Kharitonov, Eugene, Ann Lee, Adam Polyak, Yossi Adi, Jade Copet, Kushal Lakhotia, Tu Anh Nguyen, Morgane Riviere, Abdelrahman Mohamed, Emmanuel Dupoux, and Wei-Ning Hsu (May 2022b). "Text-Free Prosody-Aware Generative Spoken Language Modeling". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, pp. 8666–8681 (cit. on pp. 32, 61, 64, 65, 69, 79, 103, 104, 107, 132).

Kharitonov, Eugene, Damien Vincent, Zalán Borsos, Raphaël Marinier, Sertan Girgin, Olivier Pietquin, Matt Sharifi, Marco Tagliasacchi, and Neil Zeghidour (2023). *Speak, Read and Prompt: High-Fidelity Text-to-Speech with Minimal Supervision*. arXiv: 2302.03540 [cs.SD] (cit. on pp. 33, 85, 132).

Kim, Heeseung, Soonshin Seo, Kyeongseok Jeong, Ohsung Kwon, Jungwhan Kim, Jaehong Lee, Eunwoo Song, Myungwoo Oh, Sungroh Yoon, and Kang Min Yoo (2024). *Unified Speech-Text Pretraining for Spoken Dialog Modeling*. arXiv: 2402.05706 [cs.CL] (cit. on p. 136).

Kim, Jaehyeon, Jungil Kong, and Juhee Son (2021). *Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech*. arXiv: 2106.06103 [cs.SD] (cit. on pp. 13, 132).

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun (cit. on p. 69).

Kingma, Diederik P. and Max Welling (2014). "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. arXiv: http://arxiv.org/abs/1312.6114v10 [stat.ML] (cit. on p. 15).

Kirchenbauer, John, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein (2023). *A Watermark for Large Language Models*. arXiv: 2301.10226 [cs.LG] (cit. on p. 123).

Kneser, R. and H. Ney (1995). “Improved backing-off for M-gram language modeling”. In: *1995 International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1, 181–184 vol.1 (cit. on p. 8).

Komeili, Mojtaba, Kurt Shuster, and Jason Weston (2021). “Internet-Augmented Dialogue Generation”. In: *CoRR* abs/2107.07566. arXiv: 2107.07566 (cit. on p. 61).

Kong, Jungil, Jaehyeon Kim, and Jaekyoung Bae (2020). “HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis”. In: *Advances in Neural Information Processing Systems*. Vol. 33, pp. 17022–17033 (cit. on pp. 13, 63, 107).

Kreuk, Felix, Adam Polyak, Jade Copet, Eugene Kharitonov, Tu Anh Nguyen, Morgan Rivière, Wei-Ning Hsu, Abdelrahman Mohamed, Emmanuel Dupoux, and Yossi Adi (Dec. 2022). “Textless Speech Emotion Conversion using Discrete & Decomposed Representations”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 11200–11214 (cit. on pp. 32, 61, 103).

Kreuk, Felix, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi (2023). “AudioGen: Textually Guided Audio Generation”. In: *The Eleventh International Conference on Learning Representations* (cit. on pp. 10, 33, 96).

Kuchaiev, Oleksii, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, Patrice Castonguay, Mariya Popova, Jocelyn Huang, and Jonathan M. Cohen (2019). *NeMo: a toolkit for building AI applications using Neural Modules*. arXiv: 1909.09577 [cs.LG] (cit. on p. 67).

Lakhotia, Kushal, Eugene Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, and Emmanuel Dupoux (Dec. 2021). “On Generative Spoken Language Modeling from Raw Audio”. In: *Transactions of the Association for Computational Linguistics* 9, pp. 1336–1354. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00430/1976784/tacl_a_00430.pdf (cit. on pp. 4, 7, 10, 17, 18, 20, 26–28, 30, 39, 54, 59, 61, 62, 72, 85, 90, 101, 103, 104, 106, 111, 126, 132, 159, 163, 165).

Le, Matthew, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and Wei-Ning Hsu (2023). *Voicebox: Text-Guided Multilingual Universal Speech Generation at Scale*. arXiv: 2306.15687 [eess.AS] (cit. on p. 132).

LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel (1989). “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1.4, pp. 541–551 (cit. on p. 15).

Lee, Ann, Peng-Jen Chen, Changhan Wang, Jiatao Gu, Sravya Popuri, Xutai Ma, Adam Polyak, Yossi Adi, Qing He, Yun Tang, Juan Pino, and Wei-Ning Hsu (May 2022a). “Direct Speech-to-Speech Translation With Discrete Units”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, pp. 3327–3339 (cit. on p. 32).

Lee, Ann, Hongyu Gong, Paul-Ambroise Duquenne, Holger Schwenk, Peng-Jen Chen, Changhan Wang, Sravya Popuri, Yossi Adi, Juan Pino, Jiatao Gu, and Wei-Ning Hsu (July 2022b). "Textless Speech-to-Speech Translation on Real Data". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz. Seattle, United States: Association for Computational Linguistics, pp. 860–872 (cit. on p. 131).

Levinson, S. E., L. R. Rabiner, and M. M. Sondhi (1983). "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition". In: *The Bell System Technical Journal* 62.4, pp. 1035–1074 (cit. on p. 12).

Levinson, Stephen C and Francisco Torreira (2015). "Timing in turn-taking and its implications for processing models of language". In: *Frontiers in psychology* 6, p. 731 (cit. on p. 61).

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer (July 2020a). "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault. Online: Association for Computational Linguistics, pp. 7871–7880 (cit. on pp. 9, 32, 61).

Lewis, Patrick S. H., Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela (2020b). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *CoRR* abs/2005.11401. arXiv: 2005.11401 (cit. on p. 61).

Li, Jiwei, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan (2015). "A Diversity-Promoting Objective Function for Neural Conversation Models". In: *CoRR* abs/1510.03055. arXiv: 1510.03055 (cit. on p. 61).

Li, Xiang Lisa and Percy Liang (Aug. 2021). "Prefix-Tuning: Optimizing Continuous Prompts for Generation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Online: Association for Computational Linguistics, pp. 4582–4597 (cit. on p. 33).

Liang, Percy, Rishi Bommasani, Tony Lee, et al. (2023). "Holistic Evaluation of Language Models". In: *Transactions on Machine Learning Research*. Featured Certification, Expert Certification (cit. on p. 134).

Liu, Alexander H., Heng-Jui Chang, Michael Auli, Wei-Ning Hsu, and James R. Glass (2023). "DinoSR: Self-Distillation and Online Clustering for Self-supervised Speech Representation Learning". In: *Thirty-seventh Conference on Neural Information Processing Systems* (cit. on p. 131).

Liu, Andy T, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee (2020). "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 6419–6423 (cit. on p. 60).

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *ArXiv* abs/1907.11692 (cit. on pp. 9, 36).

Luong, Minh-Thang, Richard Socher, and Christopher D Manning (2013). "Better word representations with recursive neural networks for morphology". In: *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pp. 104–113 (cit. on p. 25).

Ma, Xutai, Anna Sun, Siqi Ouyang, Hirofumi Inaguma, and Paden Tomasello (2023). *Efficient Monotonic Multihead Attention*. arXiv: `2312.04515 [cs.CL]` (cit. on p. 136).

MacQueen, James et al. (1967). "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA, pp. 281–297 (cit. on p. 16).

Maimon, Gallil and Yossi Adi (2022). "Speaking style conversion with discrete self-supervised units". In: *arXiv preprint arXiv:2212.09730* (cit. on p. 85).

Maiti, Soumi, Yifan Peng, Shukjae Choi, Jee-weon Jung, Xuankai Chang, and Shinji Watanabe (2023). "Voxtlm: unified decoder-only models for consolidating speech recognition/synthesis and speech/text continuation tasks". In: *arXiv preprint arXiv:2309.07937* (cit. on pp. 104, 106, 112, 126).

Masumura, Ryo, Tomohiro Tanaka, Atsushi Ando, Ryo Ishii, Ryuichiro Higashinaka, and Yushi Aono (2018). "Neural dialogue context online end-of-turn detection". In: *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 224–228 (cit. on p. 62).

Meena, Raveesh, Gabriel Skantze, and Joakim Gustafson (2014). "Data-driven models for timing feedback responses in a Map Task dialogue system". In: *Computer Speech & Language* 28.4, pp. 903–922 (cit. on p. 62).

Mikolov, Tomas, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur (Sept. 2010). "Recurrent neural network based language model". In: vol. 2, pp. 1045–1048 (cit. on p. 8).

Miller, George A and Walter G Charles (1991). "Contextual correlates of semantic similarity". In: *Language and cognitive processes* 6.1, pp. 1–28 (cit. on p. 25).

Mohamed, Abdel-Rahman, Hung yi Lee, Lasse Borgholt, Jakob D. Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, Tara N. Sainath, and Shinji Watanabe (2022). "Self-supervised speech representation learning: A review". In: *IEEE JSTSP Special Issue on Self-Supervised Learning for Speech and Audio Processing* (cit. on p. 15).

Mostafazadeh, Nasrin, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen (June 2016). "A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kevin Knight, Ani Nenkova, and Owen Rambow. San Diego, California: Association for Computational Linguistics, pp. 839–849 (cit. on p. 26).

Mostafazadeh, Nasrin, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen (Apr. 2017). "LSDSem 2017 Shared Task: The Story Cloze Test". In: *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*. Ed. by Michael Roth, Nasrin Mostafazadeh, Nathanael Chambers, and Annie Louis. Valencia, Spain: Association for Computational Linguistics, pp. 46–51 (cit. on p. 111).

Nachmani, Eliya, Alon Levkovitch, Roy Hirsch, Julian Salazar, Chulayuth Asawaroengchai, Soroosh Mariooryad, Ehud Rivlin, RJ Skerry-Ryan, and Michelle Tadmor Ramanovich (2023). *Spoken Question Answering and Speech Continuation Using Spectrogram-Powered LLM*. arXiv: 2305.15255 [cs.CL] (cit. on pp. 104, 133).

Nguyen, Tu Anh, Wei-Ning Hsu, Antony D'Avirro, Bowen Shi, Itai Gat, Maryam Fazel-Zarani, Tal Remez, Jade Copet, Gabriel Synnaeve, Michael Hassid, Felix Kreuk, Yossi Adi, and Emmanuel Dupoux (2023a). "Expresso: A Benchmark and Analysis of Discrete Expressive Speech Resynthesis". In: *Proc. INTERSPEECH 2023*, pp. 4823–4827 (cit. on pp. xi, 83, 107, 117–119, 134, 169).

Nguyen, Tu Anh, Eugene Kharitonov, Jade Copet, Yossi Adi, Wei-Ning Hsu, Ali Elkahky, Paden Tomasello, Robin Algayres, Benoît Sagot, Abdelrahman Mohamed, and Emmanuel Dupoux (Mar. 2023b). "Generative Spoken Dialogue Language Modeling". In: *Transactions of the Association for Computational Linguistics* 11, pp. 250–266 (cit. on pp. xi, 32, 57, 101, 103).

Nguyen, Tu Anh, Benjamin Muller, Bokai Yu, Marta R. Costa-jussa, Maha Elbayad, Sravya Popuri, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, Itai Gat, Gabriel Synnaeve, Juan Pino, Benoit Sagot, and Emmanuel Dupoux (2024). *SpiRit-LM: Interleaved Spoken and Written Language Model*. arXiv: 2402.05755 [cs.CL] (cit. on pp. xi, 32, 99).

Nguyen, Tu Anh, Benoit Sagot, and Emmanuel Dupoux (2022a). "Are Discrete Units Necessary for Spoken Language Modeling?" In: *IEEE Journal of Selected Topics in Signal Processing* 16.6, pp. 1415–1423 (cit. on pp. xi, 35).

Nguyen, Tu Anh, Maureen de Seyssel, Robin Algayres, Patricia Roze, Ewan Dunbar, and Emmanuel Dupoux (Sept. 2020a). *Are word boundaries useful for unsupervised language learning?* arXiv: 2210.02956 [cs.CL] (cit. on p. 31).

Nguyen, Tu Anh, Maureen de Seyssel, Patricia Rozé, Morgane Rivière, Evgeny Kharitonov, Alexei Baevski, Ewan Dunbar, and Emmanuel Dupoux (2020b). "The Zero Resource Speech Benchmark 2021: Metrics and baselines for unsupervised spoken language modeling". In: *NeurIPS Workshop Self-Superv. Learn. Speech Audio Process.* Online (cit. on pp. 4, 7, 17, 18, 20–22, 28, 37, 39–41, 44, 49, 52, 89, 103, 105, 111, 163).

Nguyen, Vivian, Otto Versyp, Christopher Cox, and Riccardo Fusaroli (2022b). *A systematic review and Bayesian meta-analysis of the development of turn taking in adult-child vocal interactions*. psyarXiv: vn62t (cit. on p. 59).

Ni, Jinjie, Tom Young, Vlad Pandelea, Fuzhao Xue, Vinay Adiga, and Erik Cambria (2021). "Recent advances in deep learning based dialogue systems: A systematic survey". In: *arXiv preprint arXiv:2105.04387* (cit. on p. 59).

Niekerk, Benjamin van, Marc-André Carbonneau, Julian Zaidi, Mathew Baas, Hugo Seuté, and Herman Kamper (2022). "A Comparison of Discrete and Soft Speech Units for Improved Voice Conversion". In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. Singapore, pp. 6562–6566 (cit. on p. 131).

Niekerk, Benjamin van, Leanne Nortje, Matthew Baas, and Herman Kamper (2021). "Analyzing Speaker Information in Self-Supervised Models to Improve Zero-Resource Speech Processing". In: *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*. Brno, Czech Republic, pp. 1554–1558 (cit. on p. 37).

Niekerk, Benjamin van, Leanne Nortje, and Herman Kamper (2020). "Vector-Quantized Neural Networks for Acoustic Unit Discovery in the ZeroSpeech 2020 Challenge". In: *Proc. Interspeech 2020*, pp. 4836–4840 (cit. on p. 19).

Ondel, Lucas, Lukás Burget, and Jan Cernocký (2016). "Variational Inference for Acoustic Unit Discovery". In: *SLTU*. Vol. 81. Procedia Computer Science. Elsevier, pp. 80–86 (cit. on p. 60).

Oord, Aäron van den, Yazhe Li, and Oriol Vinyals (2018). "Representation Learning with Contrastive Predictive Coding". In: *CoRR* abs/1807.03748. arXiv: 1807.03748 (cit. on pp. 4, 15, 41, 60).

Oord, Aaron van den, Oriol Vinyals, and koray kavukcuoglu (2017b). "Neural Discrete Representation Learning". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc. (cit. on pp. 95, 107).

Oord, Aäron van den, Oriol Vinyals, and Koray Kavukcuoglu (2017a). "Neural Discrete Representation Learning". In: *CoRR* abs/1711.00937. arXiv: 1711.00937 (cit. on pp. 33, 60).

OpenAI (2022). *Introducing ChatGPT*. https://openai.com/blog/chatgpt (cit. on p. 9).

Ott, Myle, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli (June 2019). "fairseq: A Fast, Extensible Toolkit for Sequence Modeling". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 48–53 (cit. on pp. 42, 69).

Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe (2022). "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., pp. 27730–27744 (cit. on pp. 9, 123, 135, 136).

Panayotov, Vassil, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur (2015). "Librispeech: An ASR corpus based on public domain audio books". In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 5206–5210 (cit. on pp. 40, 89).

Polyak, Adam, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux (2021). "Speech Resynthesis from Discrete Disentangled Self-Supervised Representations". In: *Proc. INTERSPEECH*. arXiv: 2104.00355 [cs.SD] (cit. on pp. 32, 63, 64, 85, 86, 89, 94, 95, 107).

Popuri, Sravya, Peng-Jen Chen, Changhan Wang, Juan Pino, Yossi Adi, Jiatao Gu, Wei-Ning Hsu, and Ann Lee (2022). "Enhanced Direct Speech-to-Speech Translation Using Self-supervised Pre-training and Data Augmentation". In: *Proc. Interspeech 2022*, pp. 5195–5199 (cit. on pp. 32, 33).

Pratap, Vineel, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli (2023). *Scaling Speech Technology to 1,000+ Languages*. arXiv: 2305.13516 [cs.CL] (cit. on pp. 109, 112).

Pratap, Vineel, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert (2020). "MLS: A Large-Scale Multilingual Dataset for Speech Research". In: *Proc. Interspeech 2020*, pp. 2757–2761 (cit. on pp. 90, 106).

Prenger, R., R. Valle, and B. Catanzaro (2019). "Waveglow: A Flow-based Generative Network for Speech Synthesis". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3617–3621 (cit. on pp. 13, 21).

Qian, Kaizhi, Yang Zhang, Heting Gao, Junrui Ni, Cheng-I Lai, David Cox, Mark Hasegawa-Johnson, and Shiyu Chang (2022). "ContentVec: An Improved Self-Supervised Speech Representation by Disentangling Speakers". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 18003–18017 (cit. on p. 131).

Radford, Alec, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever (2023). "Robust speech recognition via large-scale weak supervision". In: *International Conference on Machine Learning*. PMLR, pp. 28492–28518 (cit. on pp. 13, 111, 112, 118, 125).

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018). "Improving Language Understanding by Generative Pre-Training". In: *OpenAI blog* (cit. on pp. 9, 36, 61).

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog* (cit. on pp. 36, 39).

Radinsky, Kira, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch (2011). "A word at a time: computing word relatedness using temporal semantic analysis". In: *Proceedings of the 20th international conference on World wide web*, pp. 337–346 (cit. on p. 25).

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020). "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21.140, pp. 1–67 (cit. on p. 9).

Rani, Veenu, Syed Tufael Nabi, Munish Kumar, Ajay Mittal, and Krishan Kumar (2023). "Self-supervised Learning: A Succinct Review". In: *Archives of Computational Methods in Engineering* 30.4, pp. 2761–2775 (cit. on p. 14).

Raux, Antoine and Maxine Eskenazi (2009). "A finite-state turn-taking model for spoken dialog systems". In: *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pp. 629–637 (cit. on p. 62).

Ren, Yi, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu (2022). *FastSpeech 2: Fast and High-Quality End-to-End Text to Speech*. arXiv: 2006.04558 [eess.AS] (cit. on pp. 13, 132).

Ribeiro, Flávio, Dinei Florêncio, Cha Zhang, and Michael Seltzer (2011). "Crowdmos: An approach for crowdsourcing mean opinion score studies". In: *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, pp. 2416–2419 (cit. on p. 73).

Rivière, Morgane and Emmanuel Dupoux (2021). "Towards unsupervised learning of speech features in the wild". In: *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, pp. 156–163 (cit. on p. 68).

Rivière, Morgane, Armand Joulin, Pierre-Emmanuel Mazaré, and Emmanuel Dupoux (May 2020). "Unsupervised pretraining transfers well across languages". In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. Barcelona, Spain + Online, pp. 7414–7418 (cit. on pp. 15, 19, 21, 23, 41).

Roddy, Matthew, Gabriel Skantze, and Naomi Harte (2018). "Investigating speech features for continuous turn-taking prediction using lstms". In: *arXiv preprint arXiv:1806.11461* (cit. on p. 62).

Roller, Stephen, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric Michael Smith, Y-Lan Boureau, and Jason Weston (2020). "Recipes for building an open-domain chatbot". In: *CoRR* abs/2004.13637. arXiv: 2004.13637 (cit. on p. 61).

Rozière, Baptiste, Jonas Gehring, Fabian Gloeckle, et al. (2024). *Code Llama: Open Foundation Models for Code*. arXiv: 2308.12950 [cs.CL] (cit. on p. 109).

Rubenstein, Herbert and John B Goodenough (1965). "Contextual correlates of synonymy". In: *Communications of the ACM* 8.10, pp. 627–633 (cit. on p. 25).

Rubenstein, Paul K., Chulayuth Asawaroengchai, Duc Dung Nguyen, et al. (2023). *AudioPaLM: A Large Language Model That Can Speak and Listen*. arXiv: 2306.12925 [cs.CL] (cit. on pp. 10, 32, 101, 104, 109).

Rumelhart, David E. and James L. McClelland (1987). "Learning Internal Representations by Error Propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, pp. 318–362 (cit. on p. 8).

Sacks, Harvey, Emanuel A. Schegloff, and Gail Jefferson (1974). "A Simplest Systematics for the Organization of Turn-Taking for Conversation". In: *Language* 50.4, pp. 696–735 (cit. on p. 61).

Sainath, Tara N., Oriol Vinyals, Andrew Senior, and Haşim Sak (2015). “Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks”. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4580–4584 (cit. on p. 13).

Salazar, Julian, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff (2020). “Masked Language Model Scoring”. In: *Proc. 58th Annu. Meeting Assoc. Comput. Linguist. (ACL)*. Online, pp. 2699–2712 (cit. on p. 44).

San Roman, Robin, Pierre Fernandez, Hady Elsahar, Alexandre D´efossez, Teddy Furon, and Tuan Tran (2024). “Proactive Detection of Voice Cloning with Localized Watermarking”. In: *arXiv preprint* (cit. on p. 136).

Schatz, T., V. Peddinti, F. Bach, A. Jansen, H. Hermansky, and E. Dupoux (2013). “Evaluating speech features with the Minimal-Pair ABX task: Analysis of the classical MFC/PLP pipeline”. In: *INTERSPEECH* (cit. on pp. 19, 22, 89).

Schegloff, Emanuel A (1982). “Discourse as an interactional achievement: Some uses of ‘uh huh’and other things that come between sentences”. In: *Analyzing discourse: Text and talk* 71, pp. 71–93 (cit. on p. 58).

– (2000). “Overlapping talk and the organization of turn-taking for conversation”. In: *Language in society* 29.1, pp. 1–63 (cit. on p. 61).

Schneider, Steffen, Alexei Baevski, Ronan Collobert, and Michael Auli (2019). “wav2vec: Unsupervised Pre-Training for Speech Recognition”. In: *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*. Ed. by Gernot Kubin and Zdravko Kacic. Graz, Austria, pp. 3465–3469 (cit. on p. 39).

Schuller, Björn, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian Müller, and Shrikanth Narayanan (2013). “Paralinguistics in speech and language—State-of-the-art and the challenge”. In: *Computer Speech & Language* 27.1. Special issue on Paralinguistics in Naturalistic Speech and Language, pp. 4–39 (cit. on p. 61).

Sennrich, Rico, Barry Haddow, and Alexandra Birch (Aug. 2016). “Neural Machine Translation of Rare Words with Subword Units”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725 (cit. on pp. 54, 55, 69, 165).

Serban, Iulian Vlad, Ryan Lowe, Laurent Charlin, and Joelle Pineau (2016). “Generative Deep Neural Networks for Dialogue: A Short Review”. In: *CoRR* abs/1611.06216. arXiv: 1611.06216 (cit. on p. 61).

Seyssel, Maureen de, Marvin Lavechin, Yossi Adi, Emmanuel Dupoux, and Guillaume Wisniewski (2022). “Probing phoneme, language and speaker information in unsupervised speech representations”. In: *Interspeech 2022-23rd INTERSPEECH Conference* (cit. on p. 132).

Shen, Jonathan, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, Rif A. Saurous, Yannis Agiomvrgiannakis, and Yonghui Wu (2018). “Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions”. In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. Calgary, Alberta, Canada, pp. 4779–4783 (cit. on pp. 13, 21, 39).

Shi, Bowen, Wei-Ning Hsu, Kushal Lakhotia, and Abdelrahman Mohamed (2022). "Learning Audio-Visual Speech Representation by Masked Multimodal Cluster Prediction". In: *International Conference on Learning Representations* (cit. on pp. 32, 136).

Shuster, Kurt, Mojtaba Komeili, Leonard Adolphs, Stephen Roller, Arthur Szlam, and Jason Weston (2022). *Language Models that Seek for Knowledge: Modular Search & Generation for Dialogue and Prompt Completion* (cit. on p. 61).

Skantze, Gabriel (2017). "Towards a general, continuous model of turn-taking in spoken dialogue using LSTM recurrent neural networks". In: *SIGdial* (cit. on p. 62).

– (2021). "Turn-taking in conversational systems and human-robot interaction: a review". In: *Computer Speech & Language* 67, p. 101178 (cit. on p. 59).

Smith, Eric Michael, Melissa Hall, Melanie Kambadur, Eleonora Presani, and Adina Williams (Dec. 2022). ""I'm sorry to hear that": Finding New Biases in Language Models with a Holistic Descriptor Dataset". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 9180–9211 (cit. on p. 121).

Snyder, David, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur (2018). "X-Vectors: Robust DNN Embeddings for Speaker Recognition". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333 (cit. on p. 12).

Srivastava, Aarohi, Abhinav Rastogi, Abhishek Rao, et al. (2023). "Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models". In: *Transactions on Machine Learning Research* (cit. on p. 134).

Stivers, Tanya, Nicholas J Enfield, Penelope Brown, Christina Englert, Makoto Hayashi, Trine Heinemann, Gertie Hoymann, Federico Rossano, Jan Peter De Ruiter, Kyung-Eun Yoon, et al. (2009). "Universals and cultural variation in turn-taking in conversation". In: *Proceedings of the National Academy of Sciences* 106.26, pp. 10587–10592 (cit. on p. 59).

Streijl, Robert C., Stefan Winkler, and David S. Hands (2016). "Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives". In: *Multimedia Syst.* 22.2, 213–227 (cit. on p. 27).

Ten Bosch, Louis, Nelleke Oostdijk, and Lou Boves (2005). "On temporal aspects of turn taking in conversational dialogues". In: *Speech Communication* 47.1-2, pp. 80–86 (cit. on pp. 59, 75).

Thórisson, Kristinn R. (2002). *Natural Turn-Taking Needs No Manual: Computational Theory And Model, From Perception To Action* (cit. on p. 62).

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. (2023a). "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971*. arXiv: 2302.13971 [cs.CL] (cit. on pp. 9, 101, 102, 108, 111, 121, 123, 168).

Touvron, Hugo, Louis Martin, Kevin Stone, et al. (2023b). *Llama 2: Open Foundation and Fine-Tuned Chat Models*. arXiv: 2307.09288 [cs.CL] (cit. on pp. 101, 105, 123).

van den Oord, Aäron, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu (2016). "WaveNet: A Generative Model for Raw Audio". In: *Proc. 9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*, p. 125 (cit. on p. 33).

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc. (cit. on pp. 9, 13, 16, 111).

Veaux, Christophe, Junichi Yamagishi, and Kirsten MacDonald (2017). "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit". In: (cit. on p. 89).

Versteegh, Maarten, Roland Thiollière, Thomas Schatz, Xuan Nga Cao, Xavier Anguera, Aren Jansen, and Emmanuel Dupoux (2015). "The zero resource speech challenge 2015". In: *Proc. Interspeech 2015*, pp. 3169–3173 (cit. on pp. 17, 23).

Vinyals, Oriol and Quoc V. Le (2015). "A Neural Conversational Model". In: *CoRR* abs/1506.05869. arXiv: 1506.05869 (cit. on p. 61).

Wang, Alex and Kyunghyun Cho (2019). "BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model". In: *Proc. Workshop Methods Optim. Eval. Neural Lang. Gener.* Minneapolis, Minnesota, USA, pp. 30–36 (cit. on p. 44).

Wang, Changhan, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux (Aug. 2021). "VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation". In: *Proceedings of Association for Computational Linguistics*, pp. 993–1003 (cit. on pp. 90, 106).

Wang, Chengyi, Sanyuan Chen, Yu Wu, Zi-Hua Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei (2023a). "Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers". In: *ArXiv* abs/2301.02111. arXiv: 2301.02111 [cs.CL] (cit. on pp. 33, 85, 96, 101, 104, 132).

Wang, Tianrui, Long Zhou, Ziqiang Zhang, Yu Wu, Shujie Liu, Yashesh Gaur, Zhuo Chen, Jinyu Li, and Furu Wei (2023b). *VioLA: Unified Codec Language Models for Speech Recognition, Synthesis, and Translation*. arXiv: 2305.16107 [cs.CL] (cit. on p. 104).

Ward, Nigel G (2019). *Prosodic patterns in English conversation*. Cambridge University Press (cit. on p. 61).

Warstadt, Alex, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman (2020). "BLiMP: The Benchmark of Linguistic Minimal Pairs for English". In: *Transactions of the Association for Computational Linguistics* 8. Ed. by Mark Johnson, Brian Roark, and Ani Nenkova, pp. 377–392 (cit. on p. 24).

Wei, Jason, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus (2022a). *Emergent Abilities of Large Language Models*. arXiv: 2206.07682 [cs.CL] (cit. on pp. 9, 133).

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou (2022b). "Chain of Thought Prompting Elicits Reasoning in Large Language Models". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (cit. on p. 9).

Weidinger, Laura, John Mellor, Maribeth Rauh, et al. (2021). *Ethical and social risks of harm from Language Models*. arXiv: 2112.04359 [cs.CL] (cit. on p. 3).

Wolf, Thomas, Lysandre Debut, Victor Sanh, et al. (2020). "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45 (cit. on p. 89).

Wu, Haibin, Kai-Wei Chang, Yuan-Kuei Wu, and Hung-yi Lee (2023). "SpeechGen: Unlocking the Generative Power of Speech Language Models with Prompts". In: *arXiv preprint arXiv:2306.02207* (cit. on p. 33).

Xiong, Wenhan, Jingyu Liu, Igor Molybog, et al. (2023). *Effective Long-Context Scaling of Foundation Models*. arXiv: 2309.16039 [cs.CL] (cit. on p. 109).

Xu, Jing, Arthur Szlam, and Jason Weston (2021a). "Beyond Goldfish Memory: Long-Term Open-Domain Conversation". In: *CoRR* abs/2107.07567. arXiv: 2107.07567 (cit. on p. 61).

Xu, Qiantong, Alexei Baevski, Tatiana Likhomanenko, Paden Tomasello, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli (2021b). "Self-training and pre-training are complementary for speech recognition". In: *ICASSP*. IEEE, pp. 3030–3034 (cit. on p. 89).

Yang, Dongqiang and David Martin Powers (2006). *Verb similarity on the taxonomy of WordNet*. Masaryk University (cit. on p. 25).

Yang, Shu wen, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung yi Lee (2021). *SUPERB: Speech processing Universal PERformance Benchmark*. arXiv: 2105.01051 [cs.CL] (cit. on pp. 4, 14, 17, 33, 61, 89, 103, 134).

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le (2019). "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc. (cit. on p. 9).

Yngve, Victor H (1970). "On getting a word in edgewise". In: *Chicago Linguistics Society, 6th Meeting, 1970*, pp. 567–578 (cit. on p. 58).

Yu, Jiahui, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu (2022). "Scaling Autoregressive Models for Content-Rich Text-to-Image Generation". In: *Transactions on Machine Learning Research*. Featured Certification (cit. on p. 10).

Zeghidour, Neil, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi (2021). "Soundstream: An end-to-end neural audio codec". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30, pp. 495–507 (cit. on pp. 32, 33, 85, 104, 112, 132).

Zhang, Dong, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu (2023a). *SpeechGPT: Empowering Large Language Models with Intrinsic Cross-Modal Conversational Abilities*. arXiv: 2305.11000 [cs.CL] (cit. on pp. 32, 104, 135, 136).

Zhang, Susan, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer (2022). *OPT: Open Pre-trained Transformer Language Models*. arXiv: 2205.01068 [cs.CL] (cit. on pp. 9, 101).

Zhang, Xin, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu (2024). "SpeechTokenizer: Unified Speech Tokenizer for Speech Language Models". In: *The Twelfth International Conference on Learning Representations* (cit. on p. 132).

Zhang, Yizhe, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan (2019). "DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation". In: *CoRR* abs/1911.00536. arXiv: 1911.00536 (cit. on pp. 61, 72).

Zhang, Yu, Wei Han, James Qin, et al. (2023b). *Google USM: Scaling Automatic Speech Recognition Beyond 100 Languages*. arXiv: 2303.01037 [cs.CL] (cit. on p. 104).

Zhu, Yaoming, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu (2018). "Texygen: A benchmarking platform for text generation models". In: *SIGIR*, 1097–1100 (cit. on p. 27).

List of Figures

List of Tables