

# Forewarned is Forearmed: When Non-Sequential Embedding Turns Into an Anomaly Detector

Elys Allesiaro, Antoine Caubrière, Valentin Vielzeuf

Orange Research  
4 Rue du Clos Courtel, 35510 Cesson-Sévigné, France  
firstname.lastname@orange.com

## Abstract

This paper offers an in-depth analysis of non-sequential multimodal sentence-level embeddings, with a particular focus on the SONAR model. We demonstrate that certain embedding dimensions are sensitive to perturbations and can serve as indicators of decoding anomalies. By leveraging the consistency between successive encoding and decoding, we successfully build an accurate detector. Additionally, we explore modifying specific dimensions of interest to attempt to correct them. This work underscores the importance of understanding and analyzing the embeddings themselves to enhance the reliability of multimodal representations.

**Keywords:** SONAR, multimodal, embeddings analysis, self-consistency

## 1. Introduction

Large Language Models (LLMs) have become the premier approach for addressing a large variety of tasks in Natural Language Processing, such as dialogue understanding, summary generation, or sophisticated reasoning (Zhao et al., 2023). Moreover, this revolution has been extended to multimodal problems, with the rise of systems such as SpeechLLMs (Ji et al., 2024; Haque et al., 2019), which combine a speech encoder and an LLM decoder in the hope to process speech contents with LLM's ability. These approaches are also permitted by the principle of encoder-decoder, often exploiting cross-modal attention (Chan et al., 2016; Chen et al., 2024) However, as observed by the community, processing long sequences of tokens is expensive and processing long continuous audio input even more (Jia et al., 2025; Duquenne et al., 2023b).

It is one reason why some recent works (Duquenne et al., 2023b; Khurana et al., 2022; Feng et al., 2022) have focused on the extraction of sentence-level encoder, from both text and audio inputs. These approaches consists in extracting a single non-sequential embedding of the given input (whether it is text or speech) and therefore allow efficient semantic similarity computation and (multilingual) information retrieval (Khurana et al., 2022). Moreover, these compact and non-sequential embeddings may also be decoded, as done by SONAR (Duquenne et al., 2023b) and can help to build systems such as Large Concept Models (LCM) (team et al., 2024) or SonarLLM (Dragunov et al., 2025), by serving as a high-level concept tokenizer.

These "concept" models are promising in the sense that they show improved performance on high-level tasks, such as summary generation, and

can imply a considerable efficiency gain compared to vanilla LLMs (team et al., 2024). However, they heavily rely on the "concept" encoder. And it has been observed in (team et al., 2024; Duquenne et al., 2023a), that Sonar may be prone to instability, propagating undesirable behaviors and creating anomalies in the output, and this even when using the based decoder of Sonar (infinite repetition of a random pattern, truncation of the transcribed sentence).

These anomalies may be compared to hallucinations observed for LLMs, which is a well-explored field of research. In particular, structural hallucinations (Banerjee et al., 2025) correspond to errors inherent in the generation process itself, such as repetitive loops, premature truncation, or degenerate sequence patterns. Detecting such anomalies remains challenging. Several approaches have been proposed, often relying on decoder-level signals or multiple generations. (Ma et al., 2025) analyze log-probability distributions, showing that hallucinations exhibit distinct energy profiles; (Farquhar et al., 2024) estimate semantic entropy by clustering multiple sampled generations; and (Manakul et al., 2023) leverage self-consistency checks across diverse outputs. Although effective, these methods typically require access to internal decoding probabilities or incur high computational costs due to repeated sampling.

For the specific case of a "concept" encoder such as SONAR, we propose another way to identify these anomalies<sup>1</sup> by focusing on the embedding

---

<sup>1</sup>We choose to avoid the term hallucinations in this paper, as the definition is broad and the Sonar encoder-decoder model is not a chat model, targeting transcription or translation of a precise input.

itself. We summarize our contribution<sup>2</sup> as follows **(a)** We first propose an in-depth dimensional analysis of the response of SONAR embeddings to input variations (acoustic characteristics, order of words, semantics). **(b)** We then investigate the impact of some specific dimensions on the decoding performance. **(c)** It helps us to get the intuition that the textual embedding of an anomaly transcript would be far from the original embedding that was wrongly decoded and, therefore, that an anomaly detector can be implemented simply by computing the distance between those two embeddings. **(d)** And we finally show that detecting these anomalies and understanding the privileged impact of some dimensions of the embedding is an important brick in the way to correct them.

## 2. Background and Notation

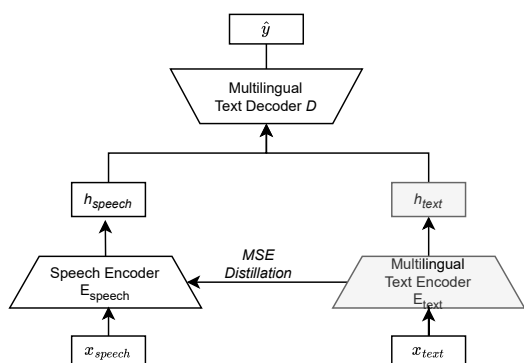


Figure 1: SONAR architecture

Before diving into our analysis, we propose an overview of the SONAR model (Duquenne et al., 2023b) and define the notation that will be used in the rest of the paper. SONAR is a unified multi-modal and multi-lingual fixed-size sentence embedding space. As explained in the Introduction, this model is particularly well suited for tasks such as semantic similarity search across different languages or modalities, as well as machine translation, both text-to-text and speech-to-text. Automatic Speech Recognition (ASR) can also be performed (even in zero-shot scenarios) by encoding speech into the SONAR embedding space and then decoding it as text in the same language. We will focus especially on this decoding regime.

From Figure 1, we can formalize the architecture. The inputs  $x_{speech}$  or  $x_{text}$  are processed, respectively, by the speech encoder  $E_{speech}$  or the text encoder  $E_{text}$  to extract a non-sequential representation  $h_{speech}$  or  $h_{text}$ . Then, this embedding is

fed to the decoder  $D$  to generate a text  $\hat{y}$ , which may be transcription or a translation of the original input.  $E_{text}$  and  $D$  are initialized from the encoder-decoder architecture of NLLB 1B (Team et al., 2022). This textual model is then finetuned on multiple tasks at the same time: translation, auto-encoding, denoising and cross-lingual similarity.  $E_{speech}$  is initialized from W2V-BERT 2.0 (Communication et al., 2023), combined with a 3-layer attention pooling transformer, in order to achieve a non-sequential representation.  $E_{speech}$  is trained by MSE distillation between  $h_{text}$  (extracted by the frozen  $E_{text}$  from ground-truth text labels) and  $h_{speech}$  (extracted from speech inputs). At inference time, the transcription process may be formally described as:

$$\hat{y} = D(E_{speech}(x_{speech})) \quad (1)$$

All our experiments are conducted using the publicly available SONAR models. For tasks related to speech transcription and text encoding, we leverage the well-known LibriSpeech dataset (Panayotov et al., 2015), specifically the test-clean subset. By applying the pipeline, we achieve a Word Error Rate (WER) of 20.2 and a BERTScore (Zhang et al., 2019) (F1) of 0.950.

## 3. How do the representations respond to variations in the input?

### 3.1. Signal perturbations

**Basic perturbations** In this section, we assess the robustness of the SONAR embeddings to input perturbations. We apply speed [0.5x;2x] (steps 0.25) and pitch shifts [-9;+9] (steps +3) to the raw input audio. For each transformation, we compute the average embeddings at dataset-level and quantify its deviation from the baseline computed on ground-truth audio.

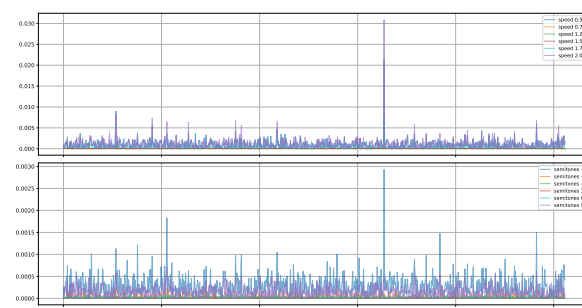


Figure 2: Absolute per-dimension deviation of SONAR embeddings under speed (top) and pitch (bottom) transformations.

For both types of perturbations, Figure 2 shows a similar pattern: deviation increases rapidly over

<sup>2</sup><https://github.com/Orange-OpenSource/sonar-speech-analysis>

all dimensions under extreme setting (e.g. speed 0.5x or 2x ; pitch  $\pm 9$  semitones) but are negligible for more realistic perturbations (e.g. speed 0.75x or 1.25x ; pitch  $\pm 3$  semitones). Since such extreme settings are unlikely for natural speech, these results suggest that SONAR embeddings are robust to small audio transformations, consistent with their semantic nature.

**Order perturbations** We perturb the temporal order of the input signal by shuffling at word level, following the word boundaries obtained by automatic alignment using Montreal Forced Aligner (MFA) (McAuliffe et al., 2017)<sup>3</sup>. We apply an n-gram shuffling procedure that consists of randomly permuting contiguous blocks of n words. A higher rank shuffles larger contiguous blocks, so only subparts of the sentence are permuted, whereas rank 1 shuffles individual words. Results are reported in table 1.

| nGram rank | WER  | BERTScore (F1) |
|------------|------|----------------|
| 6          | 46.3 | .905           |
| 5          | 49.4 | .901           |
| 4          | 55.0 | .885           |
| 3          | 62.6 | .880           |
| 2          | 73.7 | .857           |
| 1          | 87.9 | .816           |

Table 1: Decoding performance depending on the shuffling rank.

WER and BERTScore show that as the degree of semantic shuffling increases, the transcriptions tend to lose their semantic coherence. As we reduce the rank, the decoder produces more anomalies, including artifacts (e.g., repetitions of words). The system also tends to prioritize semantically plausible block of words over fidelity to the acoustic signal. These results show that the input sequence order is not encoded within the embeddings and suggest that the encoded semantics drive the decoder in constructing the output order.

**Speaker and Pitch** We perform speaker information probing experiments on the VoxCeleb dataset (Nagrani et al., 2017). The probing head is trained to simultaneously predict speaker identity and pitch information from speech SONAR embeddings. Our setup includes 1,251 unique speakers with balanced test sets (2 examples per speaker). The probing architecture consists of a shared MLP backbone followed by task-specific heads. On the test set, the probing head achieves a speaker identification accuracy of 0.026 which, while being 33x higher than the random baseline, is still very low. For pitch prediction, the model fails to capture the true variations, as evidenced by a high median er-

<sup>3</sup><https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner>

ror ( $\sim 79$  Hz) and nearly 40% of cases exceeding 100 Hz. This indicates that the system largely outputs values close to a global average rather than accurately tracking the ground-truth pitch. Thus, it confirms the weak relationship between speaker information and SONAR speech embeddings.

### 3.2. Impact of duration

We investigate the effect of input sequence duration within the SONAR embedding space by comparing the embeddings produced by  $E_{speech}$  and  $E_{text}$ . To this end, we produce a synthetic corpus of random words. We vary the sequence length from 1 to 40 by concatenating a variable number of words, each selected from a fixed vocabulary of 338 words. To generate the speech version of the sentences, we utilize coqui-tts<sup>4</sup> with the VITS Text-To-Speech model. By applying the pipeline on this random spoken words corpus, we achieve a BERTScore (F1) of 0.813 for  $E_{speech}$  embeddings and 0.884 for  $E_{text}$  embeddings.

For each of the 1024 dimensions, we analyze the monotonicity of the dimension values as a function of the sequence duration.

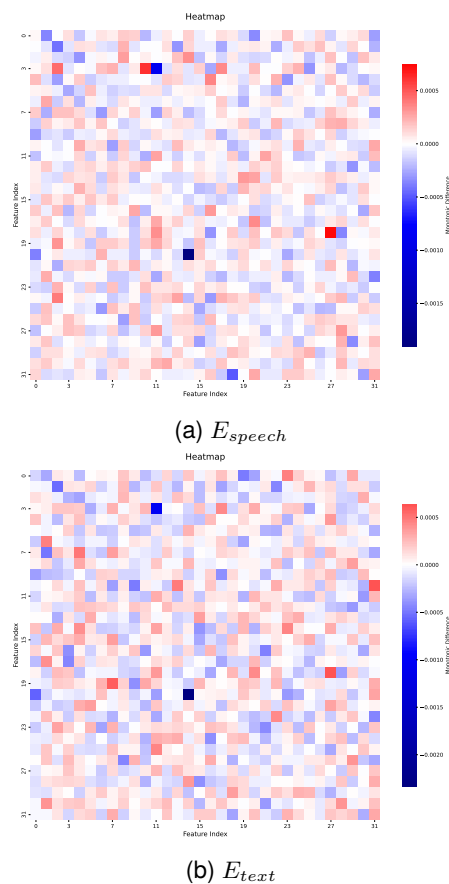


Figure 3: 32x32 monotonicity heatmaps. Each cell corresponds to a single embedding dimension.

<sup>4</sup><https://github.com/coqui-ai/TTS>

Figure 3 shows two 32x32 heatmaps, where each cell corresponds to the monotonicity of a dimension. The left heatmap corresponds to the embeddings extracted from  $E_{speech}$ , while the right one corresponds to  $E_{text}$ . These heatmaps show differences between  $E_{speech}$  and  $E_{text}$ , although a few dimensions appear to maintain similar behavior. More specifically, dimensions 107 (l3, c11) and 654 (l20, c14) seem to preserve a similar monotonicity. Such similarity between the two encoders suggests that these dimensions may encode information related to sequence length at these indices. We investigate how the value of these dimensions affects the length of the output sequence, with a focus on the dimension 654, which shows the lowest monotonicity. To assess the impact of this dimension, we perform a grid search over offsets applied to this single dimension across the entire corpus. The grid search spans the observed minimum and maximum of this dimension in the dataset, with 22 steps. For each step, we apply  $D$  and compute the WER and BERTScore. We also record the number of insertions and deletions in the automatic transcriptions. The results are reported in Figure 4.

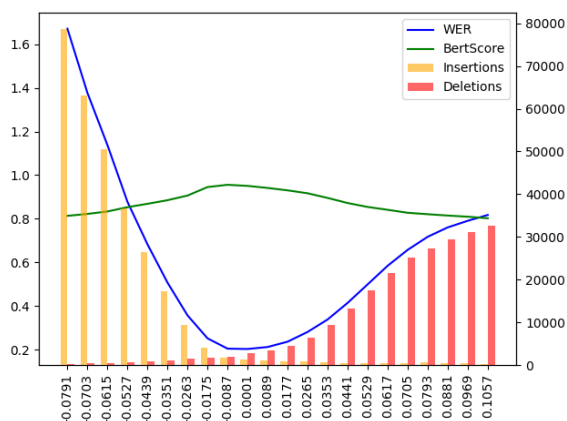


Figure 4: Effect of 654th-dimension offsets on decoding performance.

We observe a correlation between this dimension’s value and the decoder’s output length. Reducing this dimension constrains  $D$  to produce words, whereas increasing it leads  $D$  to discard most of the speech transcript. A preliminary analysis of word insertions across different offsets suggests that most insertions are repetitions of words or word sequences already present in the sentence. We also denote the existence of this kind of undesirable behaviors even when no offsets are applied.

Finally, this section leads to develop the intuition that (a) variations in embeddings may be linked to undesirable behaviors, paving the way for a detection method exploiting both text and speech embeddings and (b) that adjustments in dimensions such as 654 may lead to anomaly mitigation.

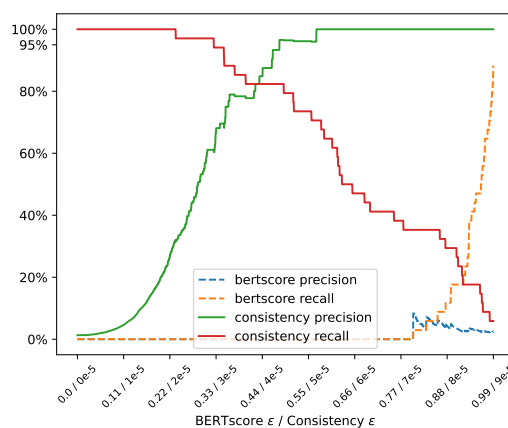


Figure 5: Precision and Recall of our anomaly detector and BERTScore baseline on the annotated LibriSpeech subset for various  $\epsilon$ .

## 4. Undesirable behaviors

### Detecting anomalies

The SONAR embedding model may be prone to undesirable behaviors, reducing its usage as a backbone for larger applications (e.g. for Large Concept Models (team et al., 2024)). Being able to automatically detect such anomalies is therefore of interest, as it may allow us to mitigate their impact on the rest of the chain.

To better our understanding of the kind of anomalies encountered, we extract the embeddings and transcriptions of the 2620 samples from the test-clean subset of LibriSpeech. Then we propose an annotation of all transcriptions<sup>5</sup>, consisting in a classification between four types of behavior: normal (e.g. usual transcription that may be paraphrased or not perfect), loop (e.g. repeating several times the same pattern of words), truncation (omitting a large part of the sentence), meaning (e.g. negating the original sentence).

Recall that embedding extraction consists of using  $E_{speech}$  or  $E_{text}$  to process a given input of the respective speech and text modalities. After getting the non-sequential embedding (let us denote it  $h_1$  here), we are using the same decoder  $D$  to generate a transcription or a translation  $\hat{y}_1$ . We propose using  $E_{text}$  to encode  $\hat{y}_1$  and obtain a second embedding  $h_2$  (which could also be decoded with  $D$  into  $\hat{y}_2$ ). Then, we compute the mean squared error between the two embeddings  $H_1$  and  $H_2$ .

$$MSE = d_{consistency} = \frac{1}{n} \sum_{i=1}^n (H_{2i} - H_{1i})^2. \quad (2)$$

In this specific case  $n = 1$ , thus we simplify  $d_{consistency} = (h_2 - h_1)^2$ . This distance is a scalar

<sup>5</sup><https://github.com/Orange-OpenSource/sonarspeech-analysis>.

and will be denoted as *consistency* in the rest of the paper. We will consider that an anomaly is detected if the *consistency* is greater than a fixed threshold  $\epsilon$ .

We report in Figure 5 the precision and recall of the designed detector for various  $\epsilon$  choices. For the rest of the experiments, we choose to set  $\epsilon = 4.39e - 5$ . We can compare our proposed approach with a naive baseline that consists of computing the WER or the BERTScore between  $\hat{y}_1$  and  $\hat{y}_2$ . This method does not allow to detect anomaly at all, while the best precision/recall compromise of  $d_{consistency}$  is 97% / 82%, enabling accurate anomaly detection.

```

 $\hat{y}_1$ 
come back come back comeback comeback comeback comeback
comeback comeback comeback comeback comeback comeback come
back comeback comeback comeback comeback comeback comeback
comeback comeback comeback
 $\hat{y}_2$ 
come back come back comeback comeback comeback comeback
comeback comeback comeback comeback comeback comeback come
back comeback comeback comeback comeback comeback comeback
BERTscore = 0.999, WER = 8%,  $d_{consistency} = 8.96e - 05$ 

```

Figure 6: Illustration of one loop anomaly from a sample of LibriSpeech test-clean. BERTscore and WER are not discriminative, while  $d_{consistency}$  is greater than the anomaly threshold.

To explain the difference with the performance of the *consistency*, we can focus on a specific example of anomaly in Figure 6, encountered in LibriSpeech test-clean. This is a loop anomaly, repeating the term "come back" instead of the correct transcription. Both transcripts are very similar, and therefore it is not possible to predict from them that something wrong is happening. Using  $d_{consistency}$  helps to avoid this problem. An intuition that can explain this ability is that the embedding  $h_2$  can be very different from  $h_1$ , simply because  $\hat{y}_1$ , during an anomaly, is a rarely encountered input by the text encoder and therefore particularly difficult to encode.

Figure 7 may help to confirm this intuition. We observe the average consistency of the anomalies compared to that of perfectly transcribed samples. The consistency of anomalies shows that there is a great gap in some specific dimensions of the embedding, for example, 654. It therefore confirms what has been described in previous sections: certain dimensions of the embeddings have a stronger impact on the decoding. And we observe here that they are changing a lot between  $h_1$  and  $h_2$ , explaining the high  $d_{consistency}$  and the anomaly detection process.

#### Does a remedy to these anomalies exist?

As we propose an efficient and accurate anomaly detector, a straightforward usage would be to combine it with an anomaly handler/corrector. A first basic way for critical application would be to label

the output with a warning, when an anomaly is detected. Yet, another hypothesis that may need exploration is that the decoder's hyperparameters are not adapted for the inputs conducting to anomalies, and therefore, changing these very parameters would help reducing undesirable outputs. Thus we select the subset of LibriSpeech test-clean containing anomalies (which is a small dataset of around 50 samples) and we propose to perform a large grid search on the decoder hyper-parameters (temperature, length penalty, and beam size). We evaluate all configurations in terms of BERTScore. We observe no significant improvement on this subset compared to using the default configuration.

As decoder parameters do not improve anomalies, we investigate applying controlled perturbation to embeddings. By examining consistency difference between 'perfect' and anomaly embeddings (Figure 7), we identify two dimensions that deviate the most (654, 351). We include the two dimensions that deviate the least (695, 932) and two randomly chosen dimensions (17, 666). To find optimal values of these dimensions, we perform an offset grid search on the anomaly subset using 22 steps. As in Section 3.2, this spans the observed range of dimensions in the test-clean dataset. We then compute the mean consistency in Table 2. We observe that overall, the grid search allows to find values with a positive impact on consistency, compared to the baseline. Moreover, the more dimensions are prone to deviate, the greater their impact on consistency. Yet, for the specific case of loop anomalies, the opposite behavior is observed, with dimensions 932 and 695 obtaining the best consistency.

These first results confirm that modifying specific dimensions when trying to limit anomalies will

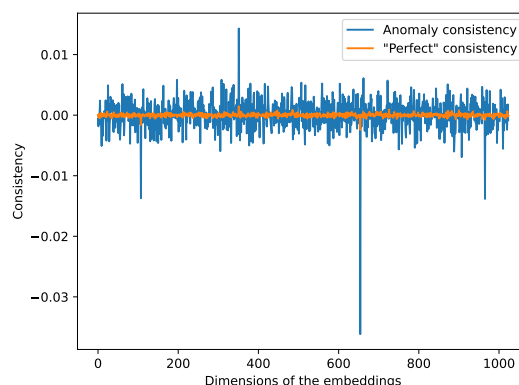


Figure 7: Consistency for two different categories: anomaly and "perfect". Anomaly consistency is computed on the average of all embeddings leading to anomalies. "Perfect" consistency is computed on the average of embeddings leading to a perfect transcript.

|                 | meaning | loop  | truncate | all   |
|-----------------|---------|-------|----------|-------|
| <i>baseline</i> | 11.98   | 8.889 | 5.991    | 6.530 |
| 2 most          | 2.918   | 4.525 | 3.730    | 3.977 |
| 2 least         | 6.551   | 3.904 | 4.540    | 4.727 |
| 2 random        | 3.540   | 5.142 | 3.971    | 4.375 |

Table 2: Best mean consistency obtained through grid search. The baseline corresponds to no offset applied. Numbers are scaled by 1e5 for readability.

have a greater impact, paving the way towards an efficient anomaly correction method. However, thinking dimensions are independent would be inaccurate, and more general approaches, notably focusing on modifying the training criterion of the SONAR distillation, may be promising.

## 5. Conclusion

We conducted an in-depth analysis of non-sequential embeddings, highlighting strong variations in specific dimensions associated with input perturbations. Based on these insights, we developed a lightweight and accurate anomaly detection method resting on the idea that the SONAR embedding of the decoded text should be close to the first extracted embedding. We also proved that modifying only specific well-targeted dimensions of the embedding may prevent some of the anomalies. Future work may explore trainable remedies to the embedding, for instance, changing the distillation paradigm of SONAR by switching to cross-modal alignment and adding consistency as an auxiliary criterion.

## 6. Bibliographical References

- Sourav Banerjee, Ayushi Agarwal, and Saloni Singla. 2025. LLMs will always hallucinate, and we need to live with this. In *Intelligent Systems and Applications*, pages 624–648, Cham. Springer Nature Switzerland.
- William Chan, Navdeep Jaitly, Quoc Le, et al. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *ICASSP*, pages 4960–4964. IEEE.
- Zhehuai Chen, He Huang, Oleksii Hrinchuk, et al. 2024. Bestow: Efficient and streamable speech language model with the best of two worlds in gpt and t5. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 147–154.
- Seamless Communication, Loïc Barrault, Yu-An Chung, et al. 2023. [Seamlessm4t: Massively multilingual & multimodal machine translation](#).
- Nikita Dragunov, Temurbek Rahmatullaev, Elizaveta Goncharova, et al. 2025. Sonar-llm: Autoregressive transformer that thinks in sentence embeddings and speaks in tokens. *arXiv preprint arXiv:2508.05305*.
- Paul-Ambroise Duquenne, Kevin Heffernan, Alexandre Mourachko, et al. 2023a. Sonar expressive: Zero-shot expressive speech-to-speech translation.
- Paul-Ambroise Duquenne, Holger Schwenk, and Benoît Sagot. 2023b. Sonar: sentence-level multimodal and language-agnostic representations. *arXiv e-prints*, pages arXiv–2308.
- Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, et al. 2024. [Detecting hallucinations in large language models using semantic entropy](#). *Nature*, 630:625–630.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, et al. 2022. Language-agnostic bert sentence embedding. In *ACL*, pages 878–891.
- Albert Haque, Michelle Guo, Prateek Verma, et al. 2019. Audio-linguistic embeddings for spoken sentences. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7355–7359.
- Shengpeng Ji, Yifu Chen, Minghui Fang, et al. 2024. Wavchat: A survey of spoken dialogue models. *arXiv preprint*.
- Junteng Jia, Gil Keren, Wei Zhou, et al. 2025. Efficient streaming llm for speech recognition. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.
- Sameer Khurana, Antoine Laurent, and James Glass. 2022. Samu-xlsr: Semantically-aligned multimodal utterance-level cross-lingual speech representation. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1493–1504.
- Huan Ma, Jiadong Pan, Jing Liu, et al. 2025. [Semantic energy: Detecting llm hallucination beyond entropy](#).
- Potsawee Manakul, Adian Liusie, and Mark J.F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.
- Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, volume 2017, pages 498–502.

- Arsha Nagrani, Joon Son Chung, and Andrew Senior. 2017. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*.
- Vassil Panayotov, Guoguo Chen, Daniel Povey, et al. 2015. [Librispeech: An asr corpus based on public domain audio books](#). In *ICASSP*.
- LCM team, Loïc Barrault, Paul-Ambroise Duquenne, et al. 2024. Large concept models: Language modeling in a sentence representation space. *arXiv preprint arXiv:2412.08821*.
- NLLB Team, Marta R. Costa-jussà, James Cross, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.
- Tianyi Zhang, Varsha Kishore, Felix Wu, et al. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).