

Investigating the Representation of Backchannels and Fillers in Fine-tuned Language Models

Yu Wang¹◇ Leyi Lao²◇ Langchu Huang²
Gabriel Skantze³ Yang Xu²◇ Hendrik Buschmeier¹◇

¹ Bielefeld University, Bielefeld, Germany

² Southern University of Science and Technology, Shenzhen, China

³ KTH Royal Institute of Technology, Stockholm, Sweden

Abstract

Backchannels and fillers are important linguistic expressions in dialogue, but often treated as ‘noise’ to be bypassed in modern transformer-based language models (LMs). Here, we study how they are represented in LMs using three fine-tuning strategies on three dialogue corpora in English and Japanese, in which backchannels and fillers are both preserved and annotated. This allows us to investigate how fine-tuning can help LMs learn these representations. We first apply clustering analysis to the learnt representation of backchannels and fillers, and find increased silhouette scores in representations from fine-tuned models, which suggests that fine-tuning enables LMs to distinguish the nuanced semantic variation in different backchannel and filler use. We also employ natural language generation metrics and qualitative analyses to verify that utterances produced by fine-tuned LMs resemble those produced by humans more closely. Our findings suggest the potential for transforming general LMs into conversational LMs that can produce human-like language more adequately.

1 Introduction

In everyday conversations, such as the one below from the Switchboard corpus (Godfrey and Holliman, 1993, 3508:7–12), backchannels (e.g., *uh-huh*) and fillers (e.g., *uh*) are ubiquitous:

A: Well, we have saved our newspapers for years and years –

B: *Uh-huh*

A: – because the, *uh*, Boy Scouts –

B: *Uh-huh*

A: – our boys have been involved in have, *uh*, had a huge recycling bin . . .

They play an important role in managing the flow of conversation and are an important way for interlocutors to negotiate the common ground (Clark,

1996). Since backchannels/fillers usually do not convey information directly and have only pragmatic functions (e.g., expressing affirmation or disagreement to the previous utterance; Jucker and Smith, 1998), they are considered semantically ‘bleached’ elements (Fuller, 2003) and deemed optional due to their lack concrete meanings (Schourup, 1999). As a result, the NLP community often treats them similarly to stop words, excluding them during pre-processing as a method to “clean” the data and improve accuracy (e.g., see Sarica and Luo, 2021). In dependency parsing for spoken dialogue data, for example, studies report that excluding backchannels/fillers in the Switchboard corpus (Godfrey and Holliman, 1993) can significantly improve parsing accuracy such as (Charniak and Johnson, 2001; Jørgensen, 2007; Dobrovolic and Martinc, 2018).

However, previous studies have highlighted that backchannels and fillers can contain rich contextual meaning in dialogue: backchannels such as *yeah* or *okay*, for example, serve as feedback in response to preceding speech (Clark, 1996, pp. 32). Fillers, such as *uh* and *um*, are used as signals of disfluency (Rose, 2015) and reflect the speaker’s cognitive processing when searching for the next word (Clark and Fox Tree, 2002), thus indicating the speaker’s cognitive load (e.g., see Table 1 of Berthold and Jameson, 1999 and Rose, 2015). Fillers can thus be regarded as an important signal to the listener that the speaker needs some time to complete the utterance (and wants to hold the turn; Ball, 1975). Further studies using qualitative approaches show that backchannels, as a source of feedback, play an important role in the incremental updating of dialogue from a semantic perspective, controlling the flow of information during conversation (Bergey and DeDeo, 2024) and improving the mutual understanding between the interlocutors in order to reach a joint goal (Howes and Eshghi, 2021).

In order to figure out the potential of transforming language models (LMs) into conversational

◇/◆ Equal contribution.

LMs, which can utilise backchannels and fillers, it is essential to establish how effectively they can be learnt and represented. An obstacle for answering this question is that language data used in NLP is often text-based (Liesenfeld and Dingemane, 2022; Dingemane and Liesenfeld, 2022) and that well-annotated conversation data is often insufficient in size and quality. Consequently, there is limited contextual information on backchannels/fillers for LMs to learn from. This might be one reason why LM-based automatic speech recognition (ASR) systems perform poorly in recognising turn taking/holding, a practice which is largely moderated by backchannels and fillers. Moreover, due to the absence of backchannels and fillers during the pre-training phase, most LMs have limited knowledge of these linguistic units. Consequently, the content generated by these LMs tends to be text-like and differs in form from natural dialogue¹, a reason, why NLP failed to “[put] natural in natural language processing” (Chrupała, 2023). Consequently LMs lack the ability to act as competent dialogue agents that can, for example, produce (i) appropriate backchannels as feedback to user utterances, and are therefore not considered attentive enough to meet human user expectations (Buschmeier and Kopp, 2018), and (ii) natural fillers in their utterances, which are vital for their role in organising speech and their communicative functions in spoken language understanding (Dinkar et al., 2022). As a result, an LM which ignores backchannels and fillers does not clean the data, it strips the model of the social cues necessary for natural and fluid interaction (Edlund et al., 2008), and consequently cannot be used for building a competent conversational agent.

In this study, we therefore address **backchannels/fillers as important linguistic phenomena** that are not well reflected in language models and specifically investigate the issue of learning the representations of backchannels and fillers in LMs. Fine-tuning is an approach that has proven effective in tackling the issue of dialogue phenomena in LMs (see, e.g., Noble and Maraev, 2021). Although there has been a great deal of work attempting to answer the question of how the representation of linguistic knowledge in language models, such as

¹In our opinion, this case is also applicable to LLM such as LLaMA-3-8B, as evidenced in our observation shown in the examples dialogues 1 to 4 in Figure 9 in the Appendix, during the NLG task, the pre-trained LLM barely generates any backchannels and fillers, which should be considered as an important sign that backchannels/fillers are not well represented in the pre-trained language model.

BERT (Devlin et al., 2019) and GPT2 (Radford et al., 2019), is altered after fine-tuning, we believe that work makes a meaningful contribution, as an initial study trying to answer how and how well LMs can represent backchannels and fillers after fine-tuning. Our general research question (**RQ1**) and three more specific research questions are:

RQ1: Does fine-tuning improve the representations of backchannels and fillers in modern transformer-based language models, such as BERT, GPT-2 and a larger language model such as LLaMA-3 8B, Qwen-3 8B?

RQ2: What role does contextual information play when we try to obtain the representations of backchannels/fillers from LMs?

RQ3: Which of the studied LMs can benefit more from fine-tuning?

RQ4: Do various fine-tuning strategies make a difference in learning the representation of backchannels/fillers?

2 Related Work

Backchannels/Fillers as Discourse Markers: Backchannels/fillers are considered to be discourse marker (Jucker and Smith, 1998). They are semantically ‘bleached’ elements in conversation, such as *oh*, *yeah*, *uh*, *uhm* which can be either fillers or backchannels – depending on whether they are within the dialogue as a sign of disfluency, or stand alone or at the beginning of an utterance, which are then usually taken as feedback to the previous utterance.

As indicated by Fuller (2003), Fox Tree (2010) and Skantze (2021), backchannels/fillers have the following two properties: first of all, they indicate the turn relations of utterances and thus play a role in conversation management, e.g., *uh* and *uhm* in speech, as a signal of disfluency, can indicate turn-holding and pause; secondly, they are ‘optional’, i.e., deleting them from the utterance won’t change its truth conditional meaning. Although fillers and backchannels are considered semantically bleached elements in dialogue, they are considered important for managing the flow of the dialogue and play an important role in ‘grounding’ processes: reflecting the attentiveness of the listeners during the interaction, confirming listeners’ understanding state as well as establishing common ground (Clark, 1996;

Buschmeier and Kopp, 2018). Moreover, backchannels/fillers are linguistic universals, as shown in the survey by Dingemans and Liesenfeld (2022), they tend to be the most frequent expressions in spoken language distribution.

Given the rich roles backchannels/fillers play in dialogue, some studies look into the prediction and generation of backchannels/fillers. For example, Skantze (2017) reports that an LSTM-based model can predict the occurrence of backchannels in dialogue. Ruede et al. (2017) show that using word embeddings in a speech model can improve the accuracy of backchannel detection. Amer et al. (2023) proposes a transformer-based pipeline to predict backchannels and further use the predicted backchannels as an index of the agreement among interlocutors. Wang et al. (2022) build three language models to generate fillers in clean speech and evaluate how this helps to improve the naturalness of the generated speech.

Fine-tuning and Representation: Fine-tuning is an important step to adapt a pre-trained model to novel downstream tasks and learn representations that are important for downstream tasks. It has been consistently reported that fine-tuning can improve LMs’ representation capabilities at different levels of linguistic representation. Mosbach et al. (2020), for example, use three sentence level classification tasks selected from the GLUE benchmark (Wang et al., 2018) as the fine-tuning tasks on BERT (Devlin et al., 2019), and find that fine-tuning can indeed affect the representation of linguistic knowledge in language models, especially the last hidden layers. It has been approved in many previous studies that fine-tuning is indispensable for an LM to perform well in different downstream tasks (e.g., Noble and Maraev, 2021; Merchant et al., 2020). A classical way to evaluate a fine-tuned LM is to use probing techniques to investigate the meaning representation of the hidden layer weights in the fine-tuned model, which can either be a supervised method (*building a classifier to report accuracy*), or an unsupervised method (*using clustering to report clustering quality before and after fine-tuning*) (see, e.g., Zhou and Srikumar, 2021; Mosbach et al., 2020). However, what is unknown is how language inputs that are less represented in the original language model would be represented in the fine-tuned language model². We will try to answer this in our

²We observed that backchannels and fillers have high token IDs, which indicates that they are not included in the original

case study of backchannels/fillers, mainly through the use of clustering after the fine-tuning process.

3 Methodology

3.1 Data Selection

In order to find datasets suitable for this study, we focussed our search specifically on datasets of transcribed spoken dialogue where fillers and backchannels are properly annotated. In order to lower the complexity of the task, the number of interlocutors per dialogue were limited to two. We identified the following three datasets to meet our needs: **Switchboard** (Godfrey and Holliman, 1993) and **MapTask** (Anderson et al., 1991), both English; and the **BTSJ 1000 Person Japanese Natural Conversation Corpus** (Usami, 2023), which is in Japanese. Combined, the two English datasets are the same size as the Japanese dataset (about 150 000 utterances).

We selected backchannels/fillers based on Todd (2019) and Pihan et al. (2024), which include data of different fillers and backchannels for dialogical interaction in English and Japanese. We report details of the selected backchannels/fillers in Appendix A.

3.2 Task definition

In order to learn the representation of backchannels/fillers in language models, we chose to fine-tune existing language models that do not have (or have limited) knowledge of backchannels/fillers. For fine-tuning we use downstream tasks where the models can learn the contextual information of these linguistic items. We use and compare three different fine-tuning tasks: masking (**MASK**), next token prediction (**NTP**) and turn taking prediction (**TTP**). These are described in the following.

Masking (MASK) Masked language modelling encourages models to utilize bidirectional context to build contextualized token representations. Inspired by the pre-training objective of BERT (Devlin et al., 2019)³, we adapt this method to the conversational setting by selectively masking backchannels/fillers.

Let $\mathbf{X} = (x_1, x_2, \dots, x_T)$ be a tokenized sequence of utterances drawn from a dialogue, where x_t is the t^{th} token and T denotes the sequence length. We first identify all matching spans $\mathbf{S} = \{s_1, s_2, \dots, s_N\}$, where each s_n is a continuous token subsequence

vocabulary of the tokenizer, or the tokenizer encountered the backchannels and fillers only in a late phase of pre-training.

³Although BERT is not an autoregressive model like GPT-2 or LLaMA, it is still considered a language model.

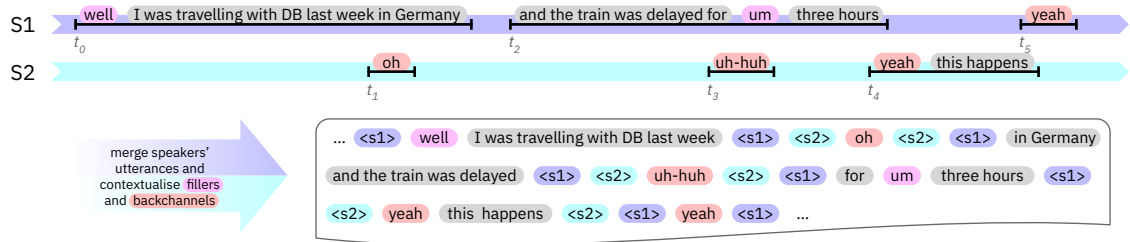


Figure 1: A dialogue example which shows turn-taking and contains various backchannels/fillers. The *um* from the speaker S1 is a **filler** while the *oh*, *uh-huh*, and *yeah* from the speaker S2 are **backchannels**. For the **NTP** and **Masking** fine-tuning tasks we merge the utterances from both speakers. For this, we take the utterances from the both and combined them into one larger sequence, considering that the utterances from one speaker is dependent on the utterances from the other speaker. We retain speaker information by adding speaker IDs (e.g., <s1>) to let the LM know that the utterances are from two different sources. Merging is not required for the **TTP** task. **TurnGPT** takes the utterances from both speakers in a linear-time order as input and predicts turn-taking probabilities.

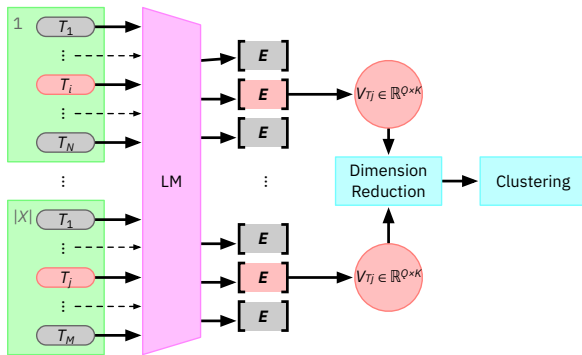


Figure 2: We select the backchannel *uh-huh* as an example to show how embeddings are obtained from language models. The pipeline consists of the following three steps: (1) taking a corpus of utterances of various lengths, which contain the backchannel *uh-huh* in, e.g., position T_i, T_j , (in total $|X|$ samples); (2) encoding the text through fine-tuning to get the contextual vector representation of the backchannel *uh-huh* (Q is the length of the backchannel tokens, K is the dimension of the selected hidden state); (3) reducing the dimension of the contextual vector and build clusters for the contextual vector of the backchannel *uh-huh*.

of backchannels or fillers, and N is the total number of such spans. For each span $s_n \in \mathbf{S}$, we define $L = \text{len}(s_n)$ and randomly apply a masking operation \mathbf{M} as follows:

$$M(s_n) = \begin{cases} [\text{MASK}]^L & (P = 0.8) \\ \text{RandomTokens}^L & (P = 0.1) \\ s_n & (P = 0.1) \end{cases}$$

where P denotes the probability of each operation (we use the default values of Devlin et al. 2019). The resulting corrupted sequence X' is then encoded into contextualized representations $H = (h_1, h_2, \dots, h_T)$. For each masked span

$s_n \in \mathbf{S}$, the model predicts the original tokens through (W being the classification layer weights):

$$P(y_{s_n,k} | X') = \text{softmax}(Wh_{s_n,k} + b), \quad \forall k \in \{1, \dots, L\}$$

We use BERT models for **English** and **Japanese** from HuggingFace library.

By learning to predict masked backchannels or fillers from contextual discourse, the model can build a better representation. As shown in Figure 1, in order to reflect the notion that backchannels and fillers are no less different from words with substantial meaning that are conditioned by their previous contexts, in our experiment practice we merge the utterances from the two sources of speakers. One of the drawbacks of doing so is that we will miss the information of the speaker of an utterance. As a solution for the fine-tuning task input, we add two special tokens to indicate to the LM the source speakers (<s1> and <s2>, as illustrated in Figure 1). This setting is also applied to the fine-tuning strategy: next token prediction.

Next-token Prediction (NTP) We consider the general task as a language modelling task, i.e., estimating the probability of the next token (backchannel/filler or other word) given previous input.

Let $\mathbf{X} = (x_1, x_2, \dots, x_T)$ represent a sequence of tokens, where x_t is the token at time step t (either a filler/backchannel or a regular word with substantial meaning) and T is the length of the sequence. A pre-trained language model $f(\cdot)$ is fine-tuned to estimate the probability of the next token x_{t+1} given all previous tokens x_1, x_2, \dots, x_t . Thus the probability of the next token x_{t+1} given

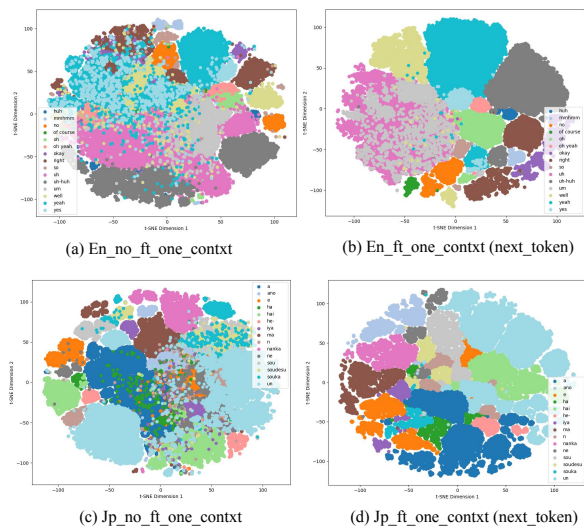


Figure 3: t-SNE plots of the English and Japanese backchannel/filler embeddings from the LLaMA-3 model (a, c) before fine-tuning, and (b, d) after **NTP** fine-tuning. Setting: one-context.

the previous tokens x_1, x_2, \dots, x_t is expressed as:

$$P(x_{t+1} | x_1, x_2, \dots, x_t) = f(x_1, x_2, \dots, x_t; \theta),$$

where θ are the model parameters that will be adjusted during the fine-tuning process.

For this fine-tuning task, we use the Japanese (Zhao and Sawada, 2021; Sawada et al., 2024) and English (Radford et al., 2019) GPT-2 models, as well as the multilingual LLaMA-3 8B (AI@Meta, 2024) and Qwen-3 8B (Yang et al., 2025) models.

Turn-taking Prediction (TTP) Our third fine-tuning task is turn-taking prediction. The task is chosen due to the close relation between the use of backchannels/fillers and turn-taking/turn-holding. Specifically, we use the framework TurnGPT (Ekstedt and Skantze, 2020), a language model based on GPT-2 and designed for the prediction of turn-shifts in spoken dialogue. A formal definition of the turn-taking prediction task is as follows.

Let $\mathbf{X} = (x_1, x_2, \dots, x_T)$ denote a sequence of tokens consisting of the linear ordering of utterances from both interlocutors (e.g., most tokens are from **interlocutor A** while **interlocutor B** produces backchannels/fillers, where x_t is the t^{th} token, and T is the sequence length. The fine-tuning task is then to estimate the probability distribution $P(y^* | \mathbf{X})$, where $y \in \{1, 2, \dots, T\}$ indicates the likelihood of a turn-taking event occurring after token x_y . The final predicted turn-taking location is then based on:

$$y^* = \arg \max_y P(y | \mathbf{X})$$

Similar to the **NTP** task, training TurnGPT also requires speaker ID (e.g., <s1>) for each utterance in order to reflect turn shifts. The details on our data pre-processing are summarised in Appendix B.

3.3 Experimental Set-Up

Our experiment workflow is shown in Figure 2. The chosen LMs are fine-tuned in advance based on the three different fine-tuning tasks⁴. For the **NTP** and **MASK** tasks, we selected 80% of the conversation data for fine-tuning and the remaining 20% for subsequent generative evaluation. Conversely, for the **TTP** task, all datasets were partitioned into training, validation, and testing sets exclusively for the fine-tuning process. For the evaluation of the representations of backchannels/fillers, we used all the conversation data.

To illustrate the workflow, we use the backchannel *uh-huh* as an example. In the initial stage, we pass through all input samples $|\mathcal{X}|$ and filter out all utterances that contain *uh-huh*. We add special token <ds> to mark and represent the backchannel (the backchannel *uh-huh* is transcribed as <ds> *uh-huh* <ds>). We then encode the corresponding utterances through the different hidden layers of a fine-tuned LM, obtaining the representation vector of *uh-huh* from the final hidden layer. Next, we check the dimensions of the representation matrix. Given that some of the backchannels/fillers can consist of more than one token (as *uh-huh* in our example), we then apply dimension reduction, simply taking the weighted average value of one dimension of the matrix so that the matrix can be levelled down to a vector representation (embedding). With the embeddings of the backchannels/fillers ‘*uh-huh*’, we examine its representation via clustering. For BERT and GPT-2 models used in the fine-tuning task, the dimension of the hidden layer is 768 for English GPT-2 model and 1024 for the Japanese GPT-2 model. TurnGPT is based on the GPT-2, thus shares the same dimension. A special case is LLaMA-3 8B and Qwen-3 8B as their hidden layer dimension is 4096. For computational reasons, we reduce the dimensionality of the obtained embeddings from all of the models to 100 using Principal Component Analysis (PCA; Abdi and Williams,

⁴Note that for our analysis, when we extract the embeddings of backchannels/fillers for the pre-trained LMs without fine-tuning, the pipeline works in the same way.

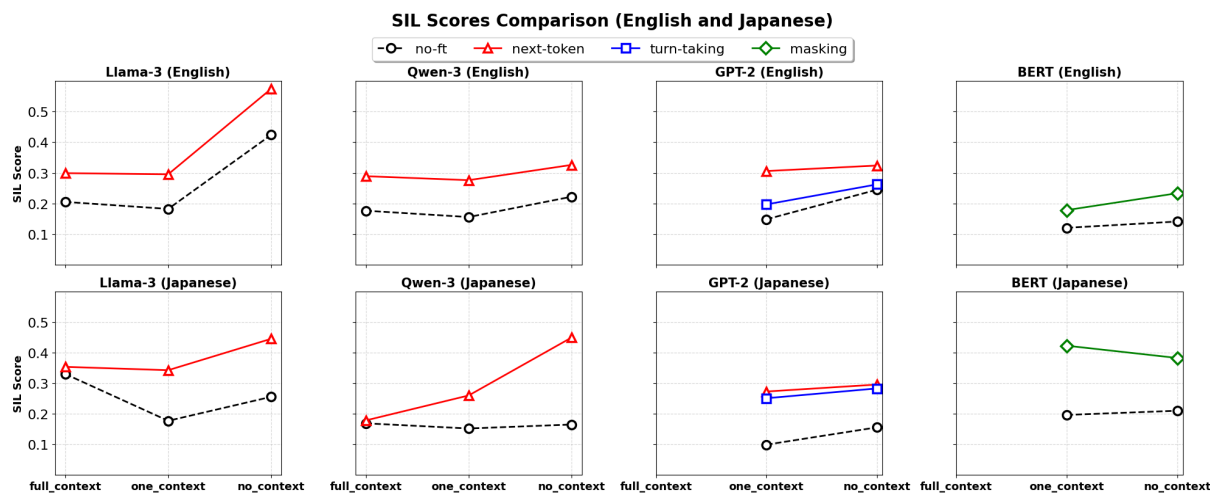


Figure 4: Average silhouette (SIL) scores of backchannels/fillers before and after masking, and two other fine-tuning methods. **NTP** (red line) is applied to GPT-2 based models, Qwen-3 8B and LLaMA-3 8B, while **TTP** (blue line) is only used for GPT-2 based models. For BERT, fine-tuning is performed using the **MASK** (green line) approach. The full-context setting is not available for the GPT-2 and BERT based models given the model’s input size limitations. Dashed lines are results without fine-tuning (“no-ft”), solid lines are those obtained after fine-tuning. The robustness test on the results are summarised in Table 6 in the Appendix.

2010). See Appendix C for technical details of the fine-tuning (e.g., GPU run time, use of LoRA for parameter optimisation when fine-tuning the LLaMA model Hu et al., 2022).

Moreover, for extracting the embeddings of backchannels/fillers after fine-tuning, we have three different context settings: (i) no context information; (ii) one context information; (iii) full context information. In the first setting, when we encounter a backchannel/filler in an utterance, only the utterance containing that backchannel/filler is fed to obtain its embedding. In the second setting, we use the previous and subsequent utterances of a backchannel/filler to build the context before obtaining its embedding. In the third setting, we combine all previous utterances when we encounter an utterance containing a backchannel/filler, using this combination as the input to obtain the embedding of that backchannel/filler. This setting is only applicable to the LLaMA-3 8B and Qwen-3 8B models since both have no input length limitation.

4 Analysis and Results

Overall observation We chose the top 15 most frequent backchannels/fillers in our Japanese and English data to check how the representation of backchannels/fillers changes after fine-tuning. Among the 15 selected backchannels/fillers in each language, typical examples are: the ones indicating

positive or negative feedback as the reference objects, and signals for turn-holding (in English, *yes*, *uh*, *yeah*, etc.; in Japanese, examples are ‘はい’ (hai, ‘yes’), ‘うん’ (un, ‘yeah’), ‘ああ’ (aa, ‘ah’), etc.).

We report the preliminary results using t-SNE visualizations (van der Maaten and Hinton, 2008). The embeddings of the 15 selected backchannels/fillers in each language are the input data. These embeddings are extracted from the last hidden layer of the LMs, before and after fine-tuning, as shown in Figure 2. We have selected t-SNE visualisations from the LLaMA-3 model with a single context setting to demonstrate how fine-tuning (in this instance, NTP) modifies the representation of backchannels/fillers within the language models (LMs). Figure 3a shows that for English data, when we obtain the embeddings of a selected backchannel/filler from the pre-trained LLaMA-3 model, the distinction between different backchannels/fillers are not clear enough. We believe that the large overlap of different data points is due to the fact that the pre-trained LMs have limited knowledge of the backchannel/filler and thus will assign random values to the encountered backchannel/filler. After fine-tuning, a clearer distinction among the embeddings of different backchannels/fillers starts to appear in the English data (see Figure 3b). The effect of fine-tuning for Japanese data is similar. After fine-tuning distinctions between different backchan-

nels/fillers emerge as clearer boundaries among different colours appear in the t-SNE visualization (Figure 3c and 3d). The t-SNE visualisations from other LMs and different settings can be found in Appendix D.5 (Figures 10 to 19).

Analysis Using K-Means Clustering As a further analysis, we look at all the corresponding embeddings and apply k-means clustering (from `scikit-learn`) individually for backchannels/fillers in the English and Japanese data. This analysis is motivated by the fact that backchannels/fillers, as functional words, often have different pragmatic functions (e.g., indicating agreement, hesitation, etc.; [Petukhova and Bunt 2009](#)). A similar idea was discussed in annotation work, which revealed that the same backchannel/filler can serve several different communicative functions ([Figuerola et al., 2022](#), fig. 4). In terms of our study targets, for example, the Japanese backchannel/filler ‘うん’ (un, ‘yeah’) can indicate both ‘confirmation’ and ‘hesitation’. When LMs have limited knowledge of backchannels/fillers and are asked to generate their representation, models will usually give random vector values to them, which will lead to two possible clustering results: either a large or a small k value. Therefore, if fine-tuning can improve the representation learning of backchannels/fillers, we should expect that the clustering effect of their embeddings will be more salient after fine-tuning. That is to say, in general we should see an increase in k value, which can indicate that fine-tuned LM’s meaning representation can reflect different pragmatic functions of backchannels/fillers if k was small; in contrast, if the k value is initially large, it should be smaller after fine-tuning.

We first look at how our proposed approaches can distinguish the meaning representations of different backchannels/fillers. We perform k -means clustering and select the associated centroids (embedding values) of the backchannels/fillers before and after fine-tuning (**NTP** and **MASK**). The centroids are later averaged to produce a single representative embedding for each backchannel/filler. We select the top-15 most frequent backchannels/fillers and use confusion matrices (based on Euclidean distance) to capture the difference before and after fine-tuning.

Figure 5 shows a representative confusion matrix, where Qwen-8B is fine-tuned on Japanese dialogue data under the one-context setting (see Appendix D.7 for the other confusion matrices). In the

confusion matrix, darker colours indicate greater distances and lighter colours indicate greater similarity. As can be seen, the inter-difference among the different backchannels/filler is more salient after fine-tuning. For example, the embeddings of the backchannel ‘ね’ (ne, ‘right?’, ‘hmm’, etc.) are separated from the embeddings of ‘はい’ (hai, ‘yes’, ‘mm-hmm’, etc.)

To quantitatively analyse the quality of the clustering of the embeddings of backchannels/fillers before and after fine-tuning, we use silhouette scores as a measure ([Rousseeuw, 1987](#)). Given a range of k -values, used for k -means clustering, we calculate the corresponding silhouette coefficient $s(i)$ for a data point i (a backchannel/filler):

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))},$$

where $a(i)$ is the average Euclidean distance between the embedding of the backchannel/filler i and all other embeddings in the same cluster as i . $b(i)$ is the minimum average Euclidean distance from embedding i to all embeddings in any other cluster, $-1 \leq s(i) \leq 1$. The **silhouette score (SC)** for a clustering is then measured as the average silhouette coefficient over all n embeddings:

$$SC = \frac{1}{n} \sum_{i=1}^n s(i),$$

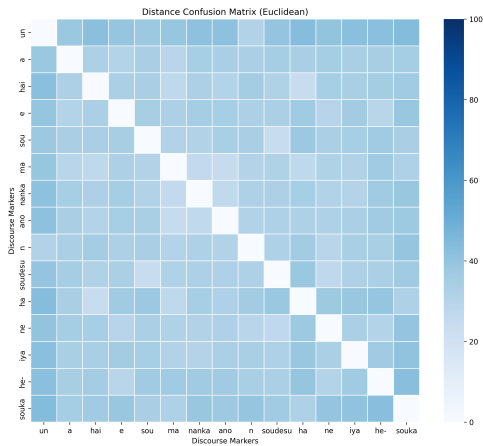
the higher the value, the better the clustering quality.

The final result is summarised in Figure 4. In all of the four selected LMs, a general tendency we can observe is that with different fine-tuning strategies, the average silhouette score increases. Detailed statistics can be found in in Tables 7 to 14 in Appendix D.6.

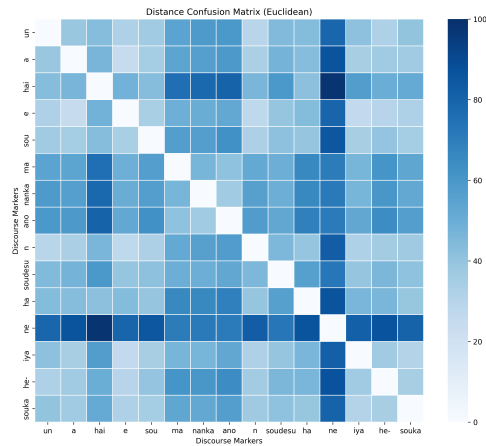
5 Discussion

In Section 4, the t-SNE visualization (Figure 3) reveals that fine-tuning LMs will change the representation of backchannels/fillers in a positive way. The analysis using k -means clustering and silhouette scores further shows that fine-tuning has beneficial effects on the representation of backchannels/fillers. Here, we further discuss these results in light of our research questions.

RQ1: Can the representations for backchannels/fillers in modern language models be learned/improved through fine-tuning? From Figure 4 (as well as Tables 7 to 14 in the Appendix),



(a) no_ft, one_context



(b) ft, one_context (NTP)

Figure 5: Distance matrices for the top 15 Japanese backchannels/fillers in the Qwen-3 model (a) before and (b) after fine-tuning.

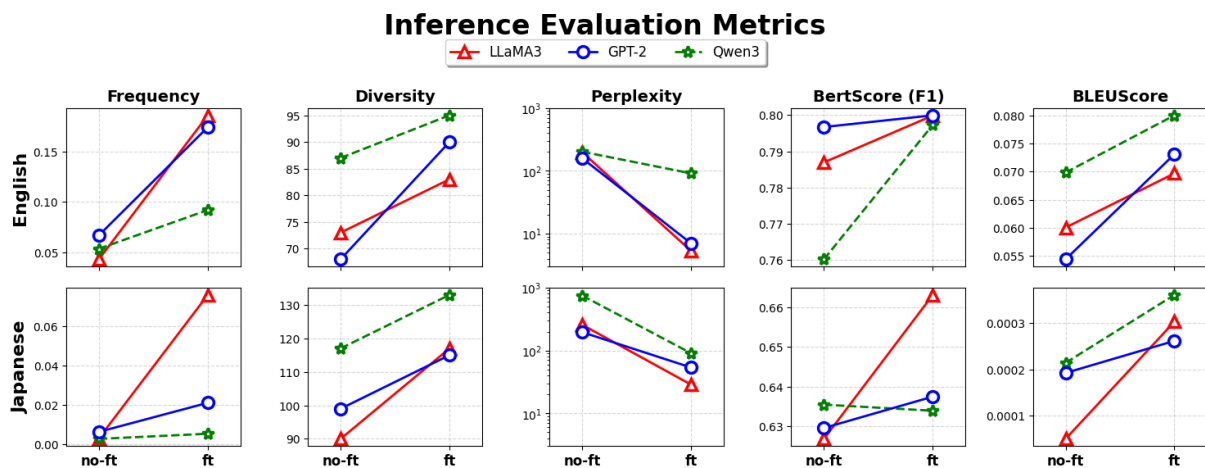


Figure 6: Evaluation metrics of generated backchannels/fillers in the English and Japanese NLG tasks. The models are evaluated before and after fine-tuning with respect to frequency, diversity, frequency-weighted perplexity, BERTScore (F1), and BLEUScore (details such as metric definition and performance results are in Section D.4, Tables 4 and 5 in the Appendix.).

we can see that fine-tuning the pre-trained LMs with dialogue data that contains contextual information on backchannels/fillers leads to a general increase of silhouette scores, which indicates a more salient clustering effect after fine-tuning and serves as crucial evidence that LMs can learn the representation of backchannels/fillers with the selected fine-tuning strategies. At the same time Figure 20 (Appendix) shows that the numbers of clusters k become more salient: slightly increasing when k was small and decreasing when k was large.

RQ2: Role of contextual information For both fine-tuned and not fine-tuned LMs, The average silhouette score generally exhibits an inverse relationship with the context size used for embeddings

(as shown in Figure 4, exceptions are LLaMA-3 for Japanese under the fine-tuning setting, Qwen-3 without fine-tuning and BERT for Japanese under fine-tuning setting). We interpret this result as follows: When context information is added, LMs tend to treat backchannels/fillers as functional words, whose representations would be compressed into more limited regions of the semantic space. The increased context size encodes more information from the surrounding content words to the backchannels/fillers, which “dilutes” their representations to the average level.

RQ3: Differences among the selected language models In this paper, we targeted four types of language models: BERT and GPT-2 are smaller in

size (parameters) and monolingual; LLaMA-3 8B and Qwen-3 8B are much larger and multilingual. It turns out (Figure 4) that the average silhouette scores of clustering from the LLaMA-3 embeddings are best among the four selected LMs. Model size, however, does not necessarily correlate with better representation in terms of silhouette scores, e.g., the results from the fine-tuned BERT model (smallest in size) are comparable to those of the Qwen-3 model.

RQ4: Effect of different fine-tuning strategies

For the GPT-2-based models, we employed NTP and TTP fine-tuning strategies. The results show that both strategies help the LM learn backchannels/fillers, with no significant differences between the strategies. However, the embeddings learnt by NTP produce slightly larger silhouette scores (Figure 4) than those learnt by TTP. Although turn-taking is widely regarded as being highly relevant to the use of fillers and backchannels, our experiment shows that TTP does not significantly improve the learning of backchannel and filler representations compared to NTP. Another surprising result is that the MASK fine-tuning task for the Japanese BERT model makes significantly improves the representation of Japanese backchannels/fillers, comparable to the the results obtained with LLaMA-3 8B and Qwen-3-8B.

6 Further Analysis via NLG Evaluation

Besides using silhouette score to evaluate the representation learning of backchannels/fillers in LMs, we also examine generation results to provide evidence of improved representation after fine-tuning. For this evaluation, we randomly selected 20% of the utterances from both the English and Japanese corpora. The generation task is defined as follows: given two turns of a dialogue in English or Japanese, both pre-trained and fine-tuned (NTP) LMs are guided to continue writing the dialogue. Based on the generated content, we evaluate backchannels/fillers along five dimensions against the ground-truth responses: (i) frequency of backchannels/fillers, (ii) diversity of backchannel/filler types, (iii) frequency-weighted perplexity of generated backchannels/fillers, (iv) BERTScore (F1), and (v) BLEUScore. The results are summarized in Figure 6 (additional details are in Appendix D.4).

After fine-tuning, we observe a general increase in both the frequency and diversity of generated backchannels/fillers, as well as improvements in

BERTScore (except for Qwen-3 Japanese) and BLEUScore. At the same time, the frequency-weighted perplexity of backchannels/fillers decreases, offering further evidence that LMs achieve better representations of backchannels/fillers after fine-tuning. We use six illustrative examples in Figure 9 (Appendix D.3) to provide qualitative human evaluation on the generated dialogues compared to the ground truth.

As we were concerned that fine-tuning LMs in this study might undermine a LM’s general capability of language understanding, we investigate this issue with an additional task (reported in Table 3, Appendix D.2), which indicates minor side effects.

7 Conclusion

We investigate the representations of backchannels/fillers in dialogue corpora learned by transformer-based language models, through three different fine-tuning strategies, masking (MASK), next token prediction (NTP) and turn-taking prediction (TTP). The main findings are: Firstly, fine-tuning results in more salient representations of backchannels/fillers as evidenced by the increased clustering performance in semantic space. Secondly, fine-tuned LMs generate utterances that are closer to actual human dialogue, as evidenced by higher backchannel/filler frequency and diversity, lower perplexity on these tokens, and improved similarity to ground-truth conversations (higher BLEU and BERTScore).

Our findings suggest that although backchannels/fillers are typically considered semantically bleached and having only pragmatic functions, their semantic representations are affected by dedicated fine-tuning tasks that incorporate more context information – in a similar way to content words that have concrete meanings. From a broader perspective, this is a case study to investigate LMs’ capability of learning under-represented tokens in training data. In a narrow sense, we focus on LMs’ capacity of representing backchannels/fillers, which shows the potentials and challenges in developing LMs that can mimic human-like speech styles.

Limitations

Within the scope of this study, we consider the following limitations, which we believe can be further addressed in future work. First of all, at the beginning of the data selection for our experiment, we did consider including language resources such

as the German corpora Verbmobil (VM2) (Kay et al., 1992) and MUNDEX (Türk et al., 2023) in order to give our results a broader linguistic basis. However, in the end we exclude the German corpora due to their comparatively small size and less formatted annotation, which, for now, leaves us with only English and Japanese data.

Secondly, the study would be more thorough with additional tests of large language models, such as Gemini and GPT-4. This was difficult given limited computing resources, where fine-tuning language models like LLaMA-3-8B was our limit. Moreover, a further step will be analysing the representation of backchannels/fillers in different hidden layers, instead of focusing on the last hidden layer of the models. There are a number of papers that examine different layers of models to answer the interpretability question (e.g., Jawahar et al., 2019; Zhao et al., 2024). Although we have examined the representation of backchannels/fillers in different hidden layers (results in Appendix D.1), a more systematic analysis is considered as our future work.

Thirdly, in this paper, we only consider how fine-tuning tasks affect representation learning of backchannels/fillers. We did not study what kind of effects different fine-tuning techniques can bring to the representation learning of backchannels/fillers. We notice that some studies propose different fine-tuning techniques, e.g., surgical fine-tuning (Lee et al., 2023), which selects subsets of layers to perform fine-tuning while preserving weights in other layers.

Fourth, there is a big gap between language models and speech models. In speech models, even for the same backchannels/fillers, e.g., ‘uh’ in English, different representations can be expected based on the differences in voice quality, pitch, and emotional state during speech. How vocal signals of backchannels/fillers are represented in speech models is a future study we will consider.

Finally, we chose to conduct qualitative analysis on our generated dialogues from linguistic perspective to check the use of backchannels and fillers (details reported in Appendix D.3), which we believe adheres to the generally suggested practice when it comes to NLG evaluation: qualitative text analysis is recommended when the goal is to improve the system (van der Lee et al., 2021). However, we currently do not have a large-scale user study, involving comprehensive human evaluation to further support our claim. We plan recruit native speakers of English and Japanese to evaluate the use

of backchannels/fillers in the generated dialogues.

Supplementary Material

Code and data are available at https://github.com/colalao/discourse_markers (for your convenience), and as a future-proof data publication at <https://doi.org/10.5281/zenodo.19473821>.

Ethics statement

Given the scope of this study, there do not appear to be any ethical issues. All of the data and models used in this study are openly available or open weights. We checked the content of the selected dialogue data and made sure that there is no leakage of participants’ personal information such as name and ID. Refinement of text and experiment code were supported by ChaptGPT and GitHub Copilot.

Acknowledgements

The early structure of this paper benefited significantly from discussions with Lívia Qian, Erik Ekstedt and Siyang Wang from TMH Department, KTH as well as group members from CLCS lab, SUSTech. We sincerely thank all the reviewers for their feedback on the paper. This work is funded by the Deutsche Forschungsgemeinschaft (DFG): TRR 318/3 2026 – 438445824, A02. This work also acknowledges the funding from Shenzhen Science and Technology Program (No. JCYJ20240813094612017) and Guangdong Province ZJRC Program (No. 2024QN11X145). This work also acknowledges the funding from the Swedish Research Council (VR) project 2020-03812.

References

- Hervé Abdi and Lynne J. Williams. 2010. [Principal component analysis](#). *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459.
- AI@Meta. 2024. [LLaMA 3 Model Card](#).
- Ahmed Amer, Chirag Bhuvaneshwara, Gowtham K. Addluri, Mohammed M. Shaik, Vedant Bonde, and Philipp Müller. 2023. [Backchannel detection and agreement estimation from video with transformer networks](#). In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.
- Anne H. Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, and 1 others. 1991. [The HCRC map task corpus](#). *Language and Speech*, 34(4):351–366.

- Peter Ball. 1975. [Listeners' responses to filled pauses in relation to floor apportionment](#). *British Journal of Social & Clinical Psychology*, 14(4):423–424.
- Claire Augusta Bergey and Simon DeDeo. 2024. [From “um” to “yeah”: Producing, predicting, and regulating information flow in human conversation](#). *Preprint*, arXiv:2403.08890.
- André Berthold and Anthony Jameson. 1999. [Interpreting symptoms of cognitive load in speech input](#). In *Proceedings of the 7th International Conference on User Modeling*, pages 235–244, Banff, Canada. Springer.
- Hendrik Buschmeier and Stefan Kopp. 2018. [Communicative listener feedback in human-agent interaction: Artificial speakers need to be attentive and adaptive](#). In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, pages 1213–1221, Stockholm, Sweden. IFAAMAS.
- Eugene Charniak and Mark Johnson. 2001. [Edit detection and parsing for transcribed speech](#). In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, PA, USA.
- Grzegorz Chrupała. 2023. [Putting natural in natural language processing](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7820–7827, Toronto, Canada. ACL.
- Herbert H. Clark. 1996. *Using Language*. Cambridge University Press, Cambridge, UK.
- Herbert H. Clark and Jean E. Fox Tree. 2002. [Using uh and um in spontaneous speaking](#). *Cognition*, 84(1):73–111.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, MN, USA. ACL.
- Mark Dingemanse and Andreas Liesenfeld. 2022. [From text to talk: Harnessing conversational corpora for humane and diversity-aware language technology](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 5614–5633, Dublin, Ireland. ACL.
- Tanvi Dinkar, Chloé Clavel, and Ioana Vasilescu. 2022. [Fillers in spoken language understanding: Computational and psycholinguistic perspectives](#). *Traitement Automatique des Langues*, 63(3):37–62.
- Kaja Dobrovoljc and Matej Martinc. 2018. [Er . . . well, it matters, right? On the role of data representations in spoken language dependency parsing](#). In *Proceedings of the 2nd Workshop on Universal Dependencies*, pages 37–46, Brussels, Belgium. ACL.
- Jens Edlund, Joakim Gustafson, Mattias Heldner, and Anna Hjalmarsson. 2008. [Towards human-like spoken dialogue systems](#). *Speech Communication*, 50:630–645.
- Bradley Efron. 1992. [Bootstrap methods: Another look at the jackknife](#). In Samuel Kotz and Norman L. Johnson, editors, *Breakthroughs in Statistics: Methodology and Distribution*, pages 569–593. Springer, New York, NY, USA.
- Erik Ekstedt and Gabriel Skantze. 2020. [TurnGPT: a transformer-based language model for predicting turn-taking in spoken dialog](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2981–2990, Online. ACL.
- Carol Figueroa, Adaeze Adigwe, Magalie Ochs, and Gabriel Skantze. 2022. [Annotation of communicative functions of short feedback tokens in Switchboard](#). In *Proceedings of the 13th Language Resources and Evaluation Conference*, pages 1849–1859, Marseille, France. ELRA.
- John Rupert Firth. 1957. A synopsis of linguistic theory 1930–1955. In Philological Society (Great Britain), editor, *Studies in Linguistic Analysis*, pages 1–32. Blackwell, Oxford, UK.
- Jean E. Fox Tree. 2010. [Discourse markers across speakers and settings](#). *Language and Linguistics Compass*, 4(5):269–281.
- Janet M. Fuller. 2003. [The influence of speaker roles on discourse marker use](#). *Journal of Pragmatics*, 35(1):23–45.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. [Simcse: Simple contrastive learning of sentence embeddings](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. ACL.
- John J. Godfrey and Edward Holliman. 1993. [Switchboard-1 Release 2 LDC97S62](#). Linguistic Data Consortium.
- Christine Howes and Arash Eshghi. 2021. [Feedback relevance spaces: Interactional constraints on processing contexts in dynamic syntax](#). *Journal of Logic, Language and Information*, 30:331–362.
- Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. [HuBERT: Self-supervised speech representation learning by masked prediction of hidden units](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.

- Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. [What does BERT learn about the structure of language?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. ACL.
- Fredrik Jørgensen. 2007. [The effects of disfluency detection in parsing spoken language.](#) In *Proceedings of the 16th Nordic Conference of Computational Linguistics*, pages 240–244, Tartu, Estonia.
- Andreas H. Jucker and Sara W. Smith. 1998. [“And people just you know like ‘wow’”](#). Discourse markers as negotiating strategies. In Andreas H. Jucker and Yael Ziv, editors, *Discourse Markers: Description and Theory*, pages 171–201. Benjamins, Amsterdam, The Netherlands.
- Masahito Kawamori, Akira Shimazu, and Takeshi Kawabata. 1996. [A phonological study on Japanese discourse markers.](#) In *Proceedings of the Korean Society for Language and Information Conference*, pages 297–306.
- Martin Kay, Peter Norvig, and Mark Gawron. 1992. *Verbmobil: A Translation System for Face-to-Face Dialog*. University of Chicago Press, Chicago, IL, USA.
- Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. 2023. [Surgical fine-tuning improves adaptation to distribution shifts.](#) In *The Eleventh International Conference on Learning Representations*.
- Andreas Liesenfeld and Mark Dingemans. 2022. [Building and curating conversational corpora for diversity-aware language science and technology.](#) In *Proceedings of the 13th Language Resources and Evaluation Conference*, pages 1178–1192, Marseille, France. ELRA.
- Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. [What happens to BERT embeddings during fine-tuning?](#) In *Proceedings of the 3rd BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. ACL.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. [Distributed representations of words and phrases and their compositionality.](#) In *Advances in Neural Information Processing Systems*, volume 26. Curran.
- Marius Mosbach, Anna Khokhlova, Michael A. Hedderich, and Dietrich Klakow. 2020. [On the interplay between fine-tuning and sentence-level probing for linguistic knowledge in pre-trained transformers.](#) In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2502–2516, Online. ACL.
- Bill Noble and Vladislav Maraev. 2021. [Large-scale text pre-training helps with dialogue act recognition, but not without fine-tuning.](#) In *Proceedings of the 14th International Conference on Computational Semantics*, pages 166–172, Groningen, The Netherlands (online). ACL.
- Toshiki Onishi, Naoki Azuma, Shunichi Kinoshita, Ryo Ishii, Atsushi Fukayama, Takao Nakamura, and Akihiro Miyata. 2023. [Prediction of various backchannel utterances based on multimodal information.](#) In *Proceedings of the 23rd ACM International Conference on Intelligent Virtual Agents*, pages 47:1–4, Würzburg, Germany. ACM.
- Volha Petukhova and Harry Bunt. 2009. [Towards a multidimensional semantics of discourse markers in spoken dialogue.](#) In *Proceedings of the 8th International Conference on Computational Semantics*, pages 157–168, Tilburg, The Netherlands. ACL.
- Ildiko Pilan, Laurent Prévot, Hendrik Buschmeier, and Pierre Lison. 2024. [Conversational feedback in scripted versus spontaneous dialogues: A comparative analysis.](#) In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 440–457, Kyoto, Japan. ACL.
- Livia Qian and Gabriel Skantze. 2024. [Joint learning of context and feedback embeddings in spoken dialogue.](#) In *Proceedings of Interspeech 2024*, pages 2955–2959, Kos, Greece. ISCA.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. [Robust speech recognition via large-scale weak supervision.](#) In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners.](#) Technical report, OpenAI.
- Ralph L. Rose. 2015. [Um and uh as differential delay markers: The role of contextual factors.](#) In *Proceedings of the 7th Workshop on Disfluency in Spontaneous Speech*, pages 73–76, Edinburgh, UK.
- Peter J. Rousseeuw. 1987. [Silhouettes: A graphical aid to the interpretation and validation of cluster analysis.](#) *Computational and Applied Mathematics*, 20:53–65.
- Robin Ruede, Markus Müller, Sebastian Stüker, and Alex Waibel. 2017. [Enhancing backchannel prediction using word embeddings.](#) In *Proceedings of Interspeech 2017*, pages 879–883, Stockholm, Sweden. ISCA.
- Serhad Sarica and Jianxi Luo. 2021. [Stopwords in technical language processing.](#) *PLOS ONE*, 16(8):1–13.
- Kei Sawada, Tianyu Zhao, Makoto Shing, Kentaro Mitsui, Akio Kaga, Yukiya Hono, Toshiaki Wakatsuki, and Koh Mitsuda. 2024. [Release of pre-trained models for the Japanese language.](#) In *Proceedings of the*

- 2024 *Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13898–13905, Torino, Italy. ICCL and ELRA.
- Lawrence Schourup. 1999. *Discourse markers*. *Lingua*, 107(3):227–265.
- Gabriel Skantze. 2017. *Towards a general, continuous model of turn-taking in spoken dialogue using LSTM recurrent neural networks*. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 220–230, Saarbrücken, Germany. ACL.
- Gabriel Skantze. 2021. *Turn-taking in conversational systems and human-robot interaction: a review*. *Computer Speech & Language*, 67:101178.
- Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. *Dialogue act modeling for automatic tagging and recognition of conversational speech*. *Computational Linguistics*, 26:339–373.
- James Allen Todd. 2019. “It has the ability to make the other person feel comfortable”: L1 Japanese speakers’ folk descriptions of aizuchi. *Lingua*, 230:102737.
- Olcay Türk, Petra Wagner, Hendrik Buschmeier, Angela Grimminger, Yu Wang, and Stefan Lazarov. 2023. *MUNDEX: A multimodal corpus for the study of the understanding of explanations*. In *Book of Abstracts of the 1st International Multimodal Communication Symposium*.
- Mayumi Usami, editor. 2023. *Building of a Japanese 1000 Person Natural Conversation Corpus for Pragmatic Analyses and Its Multilateral Studies, and NINJAL Institute-Based Projects: Multiple Approaches to Analyzing the Communication of Japanese Language Learners*. NINJAL.
- Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Kraemer. 2021. *Human evaluation of automatically generated text: Current trends and best practice guidelines*. *Computer Speech & Language*, 67:101151.
- Laurens van der Maaten and Geoffrey Hinton. 2008. *Visualizing data using t-SNE*. *Journal of Machine Learning Research*, 86:2579–2605.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. *GLUE: A multi-task benchmark and analysis platform for natural language understanding*. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. ACL.
- Siyang Wang, Joakim Gustafson, and Éva Székely. 2022. *Evaluating sampling-based filler insertion with spontaneous TTS*. In *Proceedings of the 13th Language Resources and Evaluation Conference*, pages 1960–1969, Marseille, France. ELRA.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, and 1 others. 2025. *Qwen3 Technical Report*. *Preprint*, arXiv:2505.09388.
- Tianyu Zhao and Kei Sawada. 2021. *rinna/japanese-gpt2-medium*. Huggingface.
- Zheng Zhao, Yftah Ziser, and Shay B Cohen. 2024. *Layer by layer: Uncovering where multi-task learning happens in instruction-tuned large language models*. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15195–15214, Miami, FL, USA. ACL.
- Yichu Zhou and Vivek Srikumar. 2021. *DirectProbe: Studying representations without classifiers*. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5070–5083, Online. ACL.

Table 1: The top 15 selected English backchannels/fillers and their combined frequency of occurrence in **Switchboard** and **MapTask** (from a total number of 127 672 backchannels/fillers).

Backchannel/Filler	Occurrence
<i>uh</i>	24.56%
<i>yeah</i>	17.36%
<i>uh-huh</i>	13.46%
<i>well</i>	7.21%
<i>right</i>	6.70%
<i>oh</i>	5.81%
<i>um</i>	5.49%
<i>okay</i>	3.76%
<i>no</i>	2.63%
<i>yes</i>	1.88%
<i>so</i>	1.08%
<i>oh yeah</i>	0.93%
<i>huh</i>	0.74%
<i>mmhmm</i>	0.71%
<i>of course</i>	0.59%

A Data Statistics

In this section, we provide details on the backchannels/fillers used in this paper, see Table 1 (English) and Table 2 (Japanese). We selected the 15 most frequent backchannels/fillers for our data analysis.

Table 1 shows the backchannels/fillers in the Switchboard (Godfrey and Holliman, 1993) and MapTask (Anderson et al., 1991) corpora that we selected as our study objects.

Table 2 shows the backchannels/fillers in the BTSJ 1000 Person Japanese Natural Conversation Corpus (BTSJ) (Usami, 2023) that we selected as our study objects. The Japanese data was selected based on prior guidance and analyses (Kawamori et al., 1996; Onishi et al., 2023). In the table, the backchannel/filler is used to represent its different variants, for example ‘うん’ (un, ‘yeah’) includes ‘うん、’ (un,) and ‘うん。’ (un), etc.

B Data Pre-Processing

As we observed in our data, fillers are usually surrounded by others words with substantial meaning, backchannels on the other hand, usually stand out as an independent utterance. Therefore, our data preprocessing aims to contextualize the backchannels/fillers that are not surrounded by other words so that their meaning can be reflected by the other words, i.e., distributional semantics (Firth, 1957, pp. 1–32, Mikolov et al., 2013). Figure 1 illustrates our concept.

We noticed that some of the backchannels/fillers can be subsequences of other words (e.g., *um* is

a subsequence in ‘maximum’), which affects the tokenization of the data when we add backchannels/fillers to the vocabulary as special tokens. In addition, some backchannels/fillers can be ambiguous, for example, ‘well’ and ‘right’ in English can have substantial meaning, but also serve as feedback, expressing either a positive or negative attitude towards the previous utterance. The same applies to the Japanese word ‘ちよっと’ (‘chotto’), which can mean ‘a little’ or be used as a backchannel/filler to indicate hesitation, for example. Our solution to this problem is to add a special token to backchannels/fillers so that the tokenizer will not mistake them. This practice is based on our observation in the data that for the backchannels/fillers which are ambiguous, the use of these words as backchannels/fillers usually occur as the first word in the utterance, followed by a comma as a filler (e.g., ‘well,’ or co-occur with other backchannels/fillers (e.g., *okay*, *right*).

C Implementation Details

We used a total of eight L40 48G GPUs for our experiments, with one GPU assigned to each task. For the BERT model, the runtime for each task was approximately 3 hours for both English and Japanese, which includes fine-tuning and extracting embeddings for the clustering tasks. For fine-tuning GPT-2 Japanese and English, each task of the Japanese experiment took about 2 hours, which includes fine-tuning and extracting embeddings for the clustering tasks. For English, the tasks took about 1 hour.

For the fine-tuning of TurnGPT tasks for Japanese and English, we used the following parameters: batch size 4; weight decay 0.01; dropout rate 0.3; learning rate 0.0005. A total of 15 epochs were used to complete the fine-tuning tasks. After each epoch, a checkpoint (model) was generated, saving the weight parameters gained during training. The model with the minimum loss value was chosen as the final model for estimating the probability of the turn transition potential.

For LLaMA-3 8B and Qwen-3 8B, we used LoRA to accelerate fine-tuning. The parameters for LoRA were set as follows. The rank of the low-rank matrices is 16, and a dropout rate of 0.1 is used to the LoRA layers to improve regularization. LoRA is applied to the ‘q_proj’ and ‘v_proj’ layers within the model’s attention mechanism. For LLaMA-3 8B, each task took about 8 hours for the Japanese data and about 7 hours for the English data. In

Table 2: The top 15 selected Japanese backchannels/fillers (with variants), their transcription, and frequency of occurrence in the **BTSJ** corpus (from a total number of 170 898 backchannels/fillers).

Backchannel/Filler	Variants	Transcription	Occurrence
うん	‘うんうん’, ‘うんうんうん’, ‘ううん’, ‘うーん’, ‘うんー’	‘un’	23.18%
はい	‘はいー’, ‘はいい’	‘hai’	17.41%
あ	‘ああ’, ‘あああ’, ‘あー’, ‘あっ’	‘a’	8.67%
え	‘ええ’, ‘えええ’, ‘えー’, ‘えっ’	‘e’	6.08%
そう	‘そうそう’, ‘そうそうそう’, ‘そーう’, ‘そーう’	‘sou’	5.25%
ま	‘まー’, ‘まあ’, ‘まあー’	‘ma’	4.91%
なんか	‘なんかー’, ‘なんかね’, ‘なんかねー’	‘nanka’	4.73%
あの	‘あのー’, ‘あのね’	‘ano’	4.25%
ん	‘んー’	‘n’	2.67%
そうです	‘そうですね’, ‘そうですねー’, ‘そうですよ’, ‘そうですよね’, ‘そうですよねー’, ‘そーうです’, ‘そーうです’	‘soudesu’	2.40%
は	‘はは’, ‘ははは’, ‘はー’, ‘はあ’, ‘はあー’, ‘はっ’	‘ha’	2.24%
ね	‘ねー’	‘ne’	2.19%
いや	‘いやいや’, ‘いやいやいや’, ‘いやー’	‘iya’	1.77%
へー	—	‘he-’	1.65%
そうか	—	‘souka’	1.63%

contrast, the Qwen-3 8B model required around 15 and 10 hours for the same tasks, respectively.

For k -means clustering, we first standardized the obtained embeddings and applied PCA (Principal Component Analysis) to reduce the dimensionality to 100, facilitating subsequent clustering operations. The number of clusters k ranged from 2 to 15, and the optimal k was selected based on the highest silhouette score achieved.

C.1 Other Method for Obtaining Meaning Representation

Parallel to fine-tuning, there are also some other methods to acquire the meaning representations. In terms of learning the representation of backchannels/fillers in LMs, one important method is contrastive learning (Gao et al., 2021). To the best of our knowledge, the only highly relevant work for our study is by Qian and Skantze (2024), who use contrastive learning methods to test how the speech models HuBert (Hsu et al., 2021) and Whisper (Radford et al., 2023), as well as the language model BERT can represent the different functions of feedback signals (namely backchannels). Their results show that the learnt embeddings can carry information about different functions a backchannel possesses (see Figure 3 in Qian and Skantze, 2024 for details). In this paper, we focus on fine-tuning only as our method to acquire the representation of backchannels/fillers. We leave contrastive learning aside, since, unlike in Qian and Skantze (2024), the types of backchannels/fillers are much larger and thus induce much higher computational costs. The detailed reason for not using contrastive learning

to get the representation of backchannels/fillers is mainly due to its computational cost and uncertainty (the setting and objects of study are different and simpler in Qian and Skantze, 2024). As an example, in the Japanese data, although we listed the most frequent backchannels/fillers in our paper, we also have the tail examples (those examples with very few occurrence, less than 50 times in the whole dataset) and together we have more than 80 types of backchannels/fillers.

The most important part of contrastive learning is its negative sampling mechanism. In negative sampling, given a positive example (a natural utterance with a backchannel/filler), several negative samples are generated (synthesised utterances in which the original backchannels/fillers are replaced with random ones). These negative samples are then used jointly in training to move positive examples closer to each other. In Qian and Skantze (2024) this was doable and worked well because the candidate negative examples are just selected from the backchannels which are classified as feedback, thus limited negative sample candidates and therefore much lower computational cost. Moreover, both positive and negative samples are feedback but with different function types so the experimental results are quite controllable.

D Further Supporting Results

D.1 Representation of Backchannels/Fillers From LMs’ Different Hidden Layers

In the main text, we evaluate the improvement of representations by applying clustering analysis on

the weights extracted from the last hidden layers of the LMs. Here we include additional experiments to further investigate the improvement from different layers before and after fine-tuning. To reflect the difference from different hidden layers, for LLaMA-3 8B, we selected layers 8, 16, 24, 32, for Qwen-3 8B, we selected layers 9, 18, 27, 36, for GPT-2 English and BERT models, we selected layers 4, 8, 12, for GPT-2 Japanese model, we selected layers 6, 12, 18, 24. The results are shown in Figure 7 and Figure 8.

First of all, it turns out that improvement of representation in different selected layers, as quantified by silhouette score, is in general observable in the fine-tuned models in both settings. Second, before fine-tuning silhouette scores are generally higher in the shallow layer than the deeper layers, which indicates that shallow layers have better representations of backchannels/fillers compared to deeper layers. Fine-tuning seems to break this tendency. For example, fine-tuned Llama-3 8b models, under two settings, have improved representation in their deeper hidden layers. Similar patterns can also be seen in Japanese BERT, GPT models and Qwen-3b for English.

D.2 General Performance of Fine-tuned LMs

Our biggest concern toward the experimental results reported in this study is whether the fine-tuning tasks, used to leverage the usage of backchannels and fillers in LMs, has the potential to undermine an LM’s general performance. We therefore investigate this issue by checking the LMs’ capability of doing a dialogue act prediction task, which is, in principle, feasible on the **Switchboard** and **MapTask** datasets. A dialogue act is an annotation label on utterance(s) of a dialogue that indicates the communicative functions of the utterance. The communicative functions include ‘*statement*’, ‘*agreement/accept*’, ‘*wh-question*’, ‘*backchannel/acknowledge*’, etc. (Stolcke et al., 2000). We specifically compare the accuracy on dialogue act prediction task before and after fine-tuning and examine whether performance changes. The results are summarised in the Table 3. As can be seen, BERT, GPT-2, and Qwen-3 exhibit a slight decrease in accuracy after fine-tuning, while LLaMA-3 shows improvements. These results suggest that our fine-tuning strategy does not compromise the language understanding ability of models and may, in some cases, even yield modest gains.

Table 3: Accuracy on the dialogue act classification task (MapTask dataset) before and after fine-tuning, with linear probing. We used five training epochs for BERT and GPT-2, and three training epochs for LLaMA-3 and Qwen-3.

Model	no fine-tuning	fine-tuning
BERT	59.0%	58.5%
GPT-2	56.6%	55.7%
LLaMA-3	45.4%	51.2%
Qwen-3	45.3%	42.9%

D.3 Further Qualitative and Quantitative Analyses of Generation

As further evidence of improved representation capability of backchannels/fillers in fine-tuned LMs, in this subsections, we performs a qualitative analysis on the generation results under the **fine_tuning, no_context** and **fine_tuning, one_context** settings respectively. Both fine-tuned and non fine-tuned LMs are guided to perform an NLG task which requires them to complete the dialogue based on the given context. We selected around 4000 utterances from both English and Japanese corpora. As summarised in Tables 4 and 5, the results show that the fine-tuned LLaMA-3 model increases the usage of backchannels/fillers in the generation task (i.e., the frequency increases). Moreover, different types of backchannels/fillers are used (diversity increases). This is further and crucial evidence showing that the models do learn the representation of backchannels/fillers.

As our small scale human evaluation, here we select six representative dialogue examples (see Figure 9) generated by the LLaMA-3 8b model in English as our qualitative analysis on the LLM’s capability of generating backchannels and fillers after fine-tuning. Below is the instruction for reading the dialogue examples:

- **/A.../A and /B.../B**: the beginning and the end of the utterance(s) from speaker A, speaker B.
- **///**: marking of turn shifts.
- **<ds>... </ds>**: annotation of backchannels/fillers.
- **Input**: the incomplete dialogues used for the generation task.

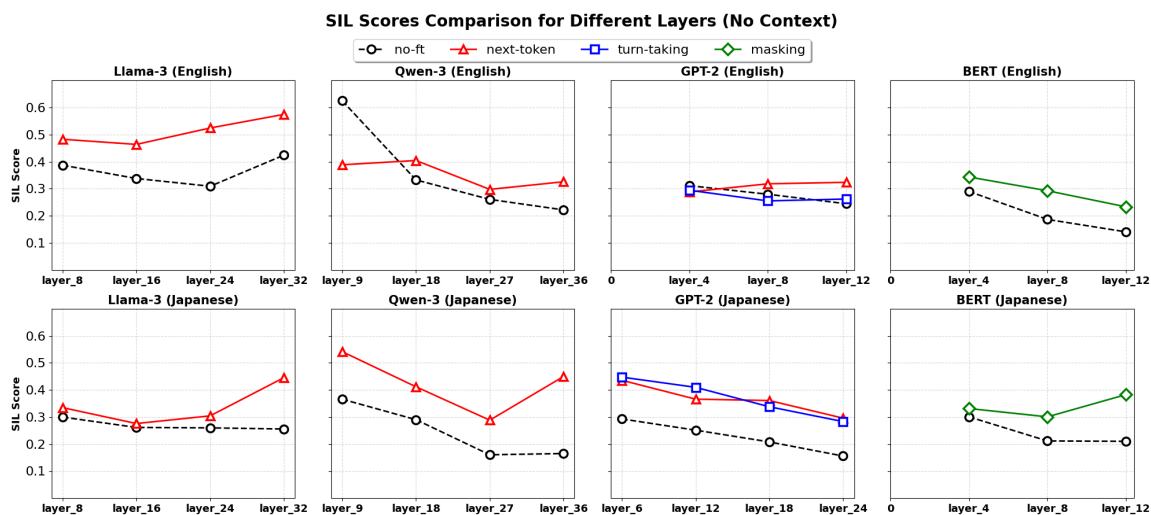


Figure 7: Improvement of the representation of backchannels/fillers on selected layers. Setting: **no-context**.

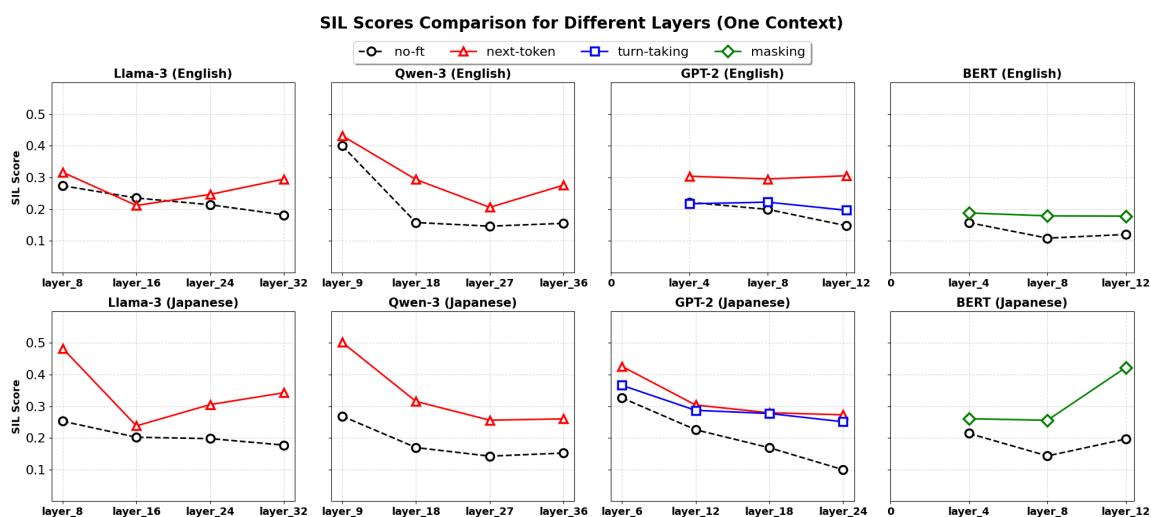


Figure 8: Improvement of the representation of backchannels/fillers on selected layers. Setting: **one-context**.

- **Ground truth:** the utterances which are the continuation of the incomplete dialogue. (**Input**)
- **Output (no_ft, one_ctx):** The generated utterances from the off-the-shelf LLaMa-3 model under the one-context setting.
- **Output (ft, one_ctx):** The generated utterances from the fine-tuned LLaMa-3 model under the one-context setting.

We notice that only under the ft, one_ctx setting (fine-tuned, one-context) does the LLaMA-3 model begin to use backchannels/fillers in its generated output, which further indicates the effectiveness of fine-tuning and context information.

The ft, one_ctx outputs of examples dialogues 1 and 4 show that the fine-tuned LLaMA-3 model can

generate feedback signals (e.g., *yeah*) to acknowledge the previous utterance. It can also use fillers to mimic disfluency or hesitation in an utterance (e.g., *um* in both examples).

In output (ft, one_ctx) of example dialogue 2, we can see that the fine-tuned LLaMA-3 model can generate a backchannel to show attentiveness to the previous speaker's utterance (the *uh-huh* in response to the utterance by speaker A).

In output (ft, one_ctx) of dialogue example 3, we can see a more complex use of fillers: the first filler *so* indicates the transition of topic, and again a filler *um* indicates disfluency or cognitive load in the production of speech. The examples shown here all support the idea that we can indeed fine-tune a LM to become a conversational agent which mimics the way humans produce speech.

A notable finding is the emergence of advanced

Dialogue 1

Input: /A <ds>yeah</ds> /A ///
/B so we went in and the financed the car for five and we didn't know that we got kind of taken just a little bit so we kind of upset that we having to you know pay this fifth year on the car because it was just not a wise thing but we learned a good lesson so you know /B ///

Ground truth: /A <ds>well</ds> five years is not that <ds>uh</ds> five years is not that bad of a problem only because you paying more interest but your payments are lower /A

Output (no_ft, one_ctx): /A this all the experience that we got in that situation yeah I just want to say that it was not just the car that we had to deal with but it was /A

Output (ft, one_ctx): /A <ds>yeah</ds> we were just <ds>um</ds> my my parents bought the car that we would drive in high school and they put like two hundred down and and I . . . /A

Dialogue 2

Input: /A but then <ds>uh</ds> we also didn't do spend <ds>uh</ds> time with them trying to trying to <ds>uh</ds> you know just family time /A ///
/B <ds>uh-huh</ds>/B ///

Ground truth: /A <ds>uh</ds> we had a at a fairly young age we started teaching them to play games and card games and so forth and and <ds>uh</ds> <ds>uh</ds> but we didn't <ds>uh</ds> do it as much or as often as <ds>uh</ds> I would have liked to I would like to do that as a as a family and /A

Output (no_ft, one_ctx): /A with the family that we have now with the family that we have now. What are you going to do with that kind of logic? I mean, I'd like to solve this problem. But I can't. Can you help me? I have a little bit of time before the family comes over. So, can you give . . . /A

Output (ft, one_ctx): /A I think <ds>uh</ds> my father was really into them and we would you know go camping and stuff and <ds>uh</ds> you know I think if you do that then it important to spend some time with them you know /A ///
/B <ds>uh-huh</ds> /B

Dialogue 3

Input: /A so it becomes <ds>uh</ds> a general battle of the w- sort of the election of the lesser of two evils I think so /A ///
/B <ds>yeah</ds> /B ///

Ground truth: /A I don't know I just <ds>uh</ds> I unhappy with it but and well II sort of semi-fortunate right now I a graduate student so I don't make that much /A

Output (no_ft, one_ctx): /A The concept of "lesser of two evils" suggests that when faced with a difficult decision, it's better to choose the option that is less undesirable. In the context of the 202 /A

Output (ft, one_ctx): /A it always seems to be that way <ds>so</ds> <ds>um</ds> what do you think is the other major issue now /A ///
/B <ds>um</ds> I I don't know I think . . . /B

Dialogue 4

Input: /A <ds>yeah</ds> <ds>well</ds> my <ds>uh</ds> my <ds>uh</ds> probably one of the biggest decisions I think that was very strengthening for our family was rather than have one child make that decision /A ///
/B <ds>uh-huh</ds> /B ///

Ground truth: /A than just delegate it I think that they they had a great deal of <ds>um</ds> all the brothers and sisters got together and they actually had a conference /A

Output (no_ft, one_ctx): /A one year. For one of one or a rather long time I am one I do one I did one I have one I will one one I had one /A

Output (ft, one_ctx): /A I think we went to church and I think we sought <ds>um</ds> the advice of those that could see it from a different perspective /A

Dialogue 5

Input: /A you don't have a stone circle? /A /// /B no /B /// /A <ds>okay</ds> do you have a stone circle on the map at all /A /// /B <ds>um</ds> i have a ruined monastery /B ///

Ground truth: /A fine <ds>um</ds> do you have any obstacle below the telephone kiosk /A

Output (no_ft, one_ctx): /A hat's good too we can explore it 1a /A

Output (ft, one_ctx): /A you have a ruined monastic <ds>uh-huh</ds> <ds>well</ds> you're going to go down and then you're going to go right and then you're going to go up over the top of the stone monument and then down again /A

Dialogue 6

Input: /A <ds>eh</ds> can you dr- actually draw round the /A ///
/B <ds>well</ds> <ds>yeah</ds> i suppose /B /// /A <ds>well</ds> if you could draw round them then/A ///

Ground truth: /B straight down to the giraffes /B

Output (no_ft, one_ctx): /B then maybe we can make a plan 1. but then i can't actually get from the other end of the rope bridge to the giraffe /B

Output (ft, one_ctx): /B i don't know whether there's a river or not i've got a river which is on the left-hand side /B

Figure 9: Six illustrative examples generated with the non fine-tuned and fine-tuned LLaMA-3 model in the one-context setting.

pragmatic competencies of backchannels and fillers in the fine-tuned LMs. As illustrated in output (ft, one_ctx) of dialogue examples 3 and 5, the fine-tuned LLaMA-3 model successfully distinguishes between the structural role of *so*, a filler which marks a sequence transition, and the cognitive signalling of the filler *um*.

Similarly, in output (ft, one_ctx) in example dialogue 5, the *uh-huh* is a backchannel which indicates confirmation and signals grounding while *well* is a filler which shows transition and framing. Furthermore, as evidenced by example dialogue 6, the fine-tuned LM does not overgenerate backchannels and fillers to complete every dialogue. Instead, it demonstrates a nuanced capacity to omit backchannels and fillers when they are not pragmatically required.

D.4 Supporting Details for NLG Evaluation Result Shown in Figure 6

Tables 4 and 5 summarize the evaluation results of models generating the next utterance based on a two-turn dialogue context, using the entire 20% evaluation split of the English and Japanese corpora (one utterance per speaker). The generated utterance is constrained to match the length of the ground-truth response. Metrics are defined as follows: (1) **Diversity** counts the number of distinct backchannel/filler types; (2) **Frequency** is the proportion of backchannel/filler tokens in the generated text, normalized by the total number of words for English and by the total number of characters for Japanese; (3) **Perplexity** is the frequency-weighted perplexity computed only on generated backchannel/filler tokens, $PPL = \exp\left(-\frac{1}{\sum_i f_i} \sum_i f_i \log p(w_i | c_i)\right)$, where w_i is a generated backchannel/filler, c_i its context, and f_i its count; (4) **BERTScore (F1)** and (5) **BLEUScore** are computed against the ground-truth continuation. Backchannels/fillers are detected using a curated lexicon with boundary-aware matching. We report results for both the pre-trained (no_ft_one) and fine-tuned (ft_one) models.

D.5 Additional t-SNE Visualisations

Figure 3 in Section 4 shows t-SNE plots of how the embeddings of backchannels/fillers in the LLaMA-3 model change with fine-tuning in the one-context setting. In Figures 10 to 19 we provide additional t-SNE visualisations for the other models and settings. One of the general trends which can be observed is that including more contextual information can

lead to better representation of backchannels/fillers (in terms of clearer borders among different ones.). For example, based on English data from Figures 10 and 11 (no-context vs. full-context), Figures 12 and 13 (no-context vs. one-context), it can be concluded that including contextual information in fine-tuning leads to better t-SNE visualisation. However, there also exist exceptions. In the Japanese data in Figures 13 and 14 (one-context vs. full-context), for example, increasing context information for fine-tuning results in poorer t-SNE visualisations.

Another issue that we noticed is that for the BERT models the improvement of the backchannels/fillers representation is not reflected well in the t-SNE visualisations: In comparison to modern LLMs (such as LLaMA-3 and Qwen-3), BERT is much more lightweight and supposedly saw less backchannels/fillers during its pre-training. However, according to Figures 18 and 19, the border among different backchannels/fillers is already quite clear before fine-tuning. This concern is not further discussed here, and could be studied in future work.

D.6 Statistics of Silhouette Scores and K-Means Values Under PCA

To ensure the statistical robustness of our evaluation, we further estimate the average silhouette scores using a bootstrap resampling approach ($n = 1000$) (Efron, 1992). As summarized in Table 6, we sample the embeddings with replacement and report the mean silhouette score alongside its 95% bootstrap confidence interval half-width. Tables 7 to 14 respectively show the silhouette scores of each backchannel/filler in both the original dimensional space and after dimensionality reduction to 100 dimensions using Principal Component Analysis (PCA), along with the average silhouette scores for English and Japanese data. A general trend is that, with fine-tuning the silhouette score increases for most of the cases. When there is no fine-tuning, adding context size (no_ctx to one_ctx to full_ctx) will also have the same effect. However, combining context size with fine-tuning as well as increasing context size will lower the silhouette score. Additionally, Figure 20 and Tables 15 to 22 summarize how the k -value changes with fine-tuning.

D.7 Additional Confusion Matrices

Figure 5 in Section 4 shows the confusion matrices of distances among the top 15 Japanese backchannels/fillers in the Qwen-3 model before and after

Table 4: Evaluation metrics for generated backchannels/fillers in the **English** NLG task.

Metric	LLaMA-3		Qwen-3		GPT-2	
	no_ft, one_ctx	ft, one_ctx	no_ft, one_ctx	ft, one_ctx	no_ft, one_ctx	ft, one_ctx
Diversity ↑	73	83	87	95	68	90
Frequency (%) ↑	4.29%	18.61%	5.33%	9.19%	6.68%	17.43%
Perplexity ↓	197.67 ± 84.75	5.30 ± 1.10	202.06 ± 92.69	91.32 ± 11.70	158.64 ± 72.19	6.98 ± 1.51
BERTScore (F1 %) ↑	78.69% ± 0.13%	79.99% ± 0.04%	76.02% ± 0.06%	79.73% ± 0.05%	79.67% ± 0.05%	79.99% ± 0.04%
BLEUScore ↑	0.0600 ± 0.0012	0.0697 ± 0.0012	0.0698 ± 0.0012	0.0800 ± 0.0013	0.0544 ± 0.0011	0.0731 ± 0.0012

Table 5: Evaluation metrics for generated backchannels/fillers in the **Japanese** NLG task.

Metric	LLaMA-3		Qwen-3		GPT-2	
	no_ft, one_ctx	ft, one_ctx	no_ft, one_ctx	ft, one_ctx	no_ft, one_ctx	ft, one_ctx
Diversity ↑	90	117	117	133	99	115
Frequency (%) ↑	0.31%	7.57%	0.27%	0.54%	0.63%	2.10%
Perplexity ↓	255.94 ± 72.48	28.51 ± 4.58	748.63 ± 31.24	90.17 ± 22.50	195.36 ± 93.86	53.51 ± 28.74
BERTScore (F1 %) ↑	62.68% ± 0.06%	66.31% ± 0.04%	63.55% ± 0.07%	63.39% ± 0.04%	62.96% ± 0.05%	63.75% ± 0.05%
BLEUScore ↑	0.00005 ± 0.00005	0.00030 ± 0.00010	0.00021 ± 0.00010	0.00036 ± 0.00010	0.00019 ± 0.00010	0.00026 ± 0.00005

fine-tuning. Here, we provide additional confusion matrices for English (LLaMA-3, Qwen-3, GPT-2, and BERT) in Figures 21 to 24 and Japanese (LLaMA-3, GPT-2, and BERT) in Figures 25 to 27.

In the confusion matrices, darker colours indicate greater distances and lighter colours indicate greater similarity. The confusion matrices for English and Japanese show that for the more recent LLM models (LLaMA-3 and Qwen-3), fine-tuning makes the differences among selected backchannels/fillers more pronounced. Change is less evident for GPT-2 and BERT.

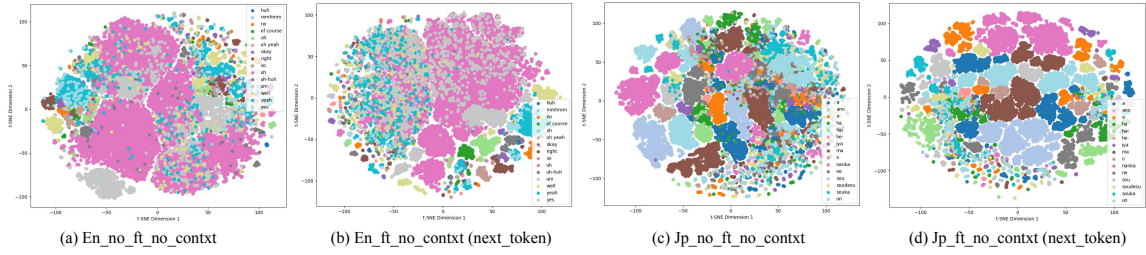


Figure 10: t-SNE plots of the backchannels/fillers embeddings from the **LLaMA-3** model (NTP). Setting: **no-context**.

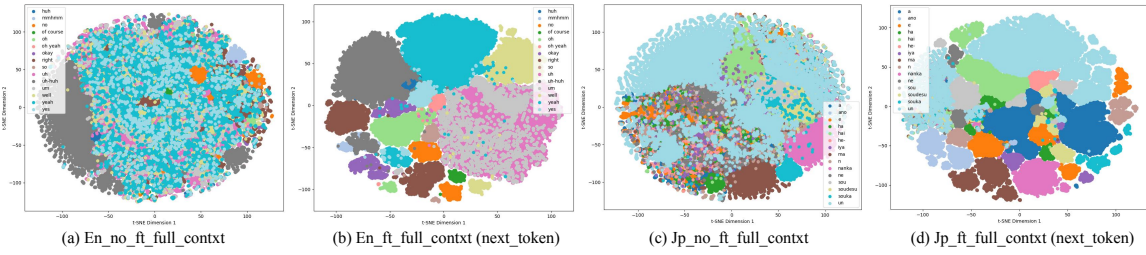


Figure 11: t-SNE plots of the backchannels/fillers embeddings from the **LLaMA-3** model (NTP). Setting: **full-context**.

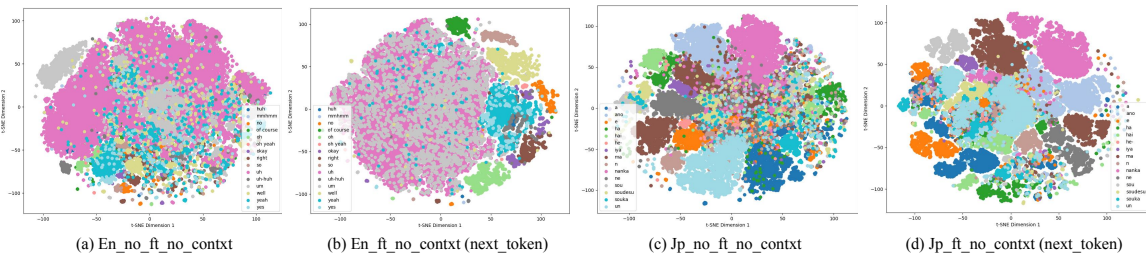


Figure 12: t-SNE plots of the backchannels/fillers embeddings from the **Qwen-3** model (NTP). Setting: **no-context**.

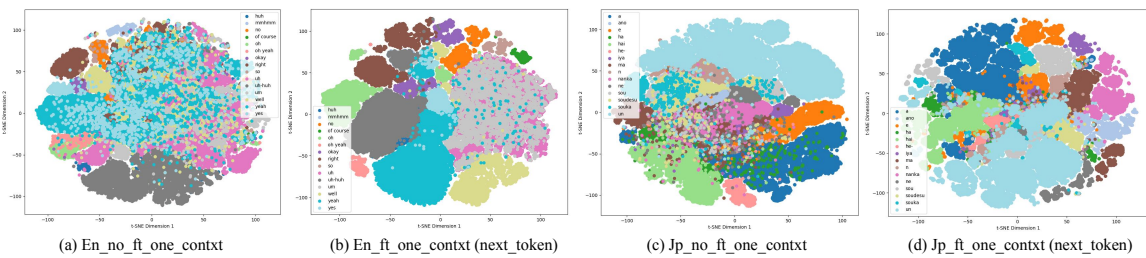


Figure 13: t-SNE plots of the backchannels/fillers embeddings from the **Qwen-3** model (NTP). Setting: **one-context**.

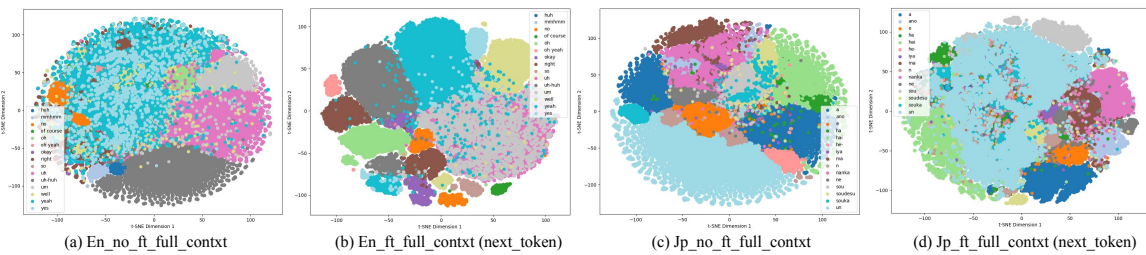


Figure 14: t-SNE plots of the backchannels/fillers embeddings from the **Qwen-3** model (NTP). Setting: **full-context**.

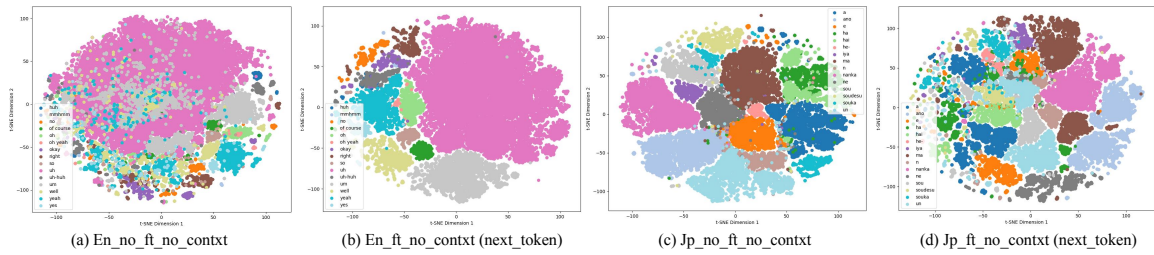


Figure 15: t-SNE plots of the backchannels/fillers embeddings from the **GPT-2** model (**NTP**). Setting: **no-context**.

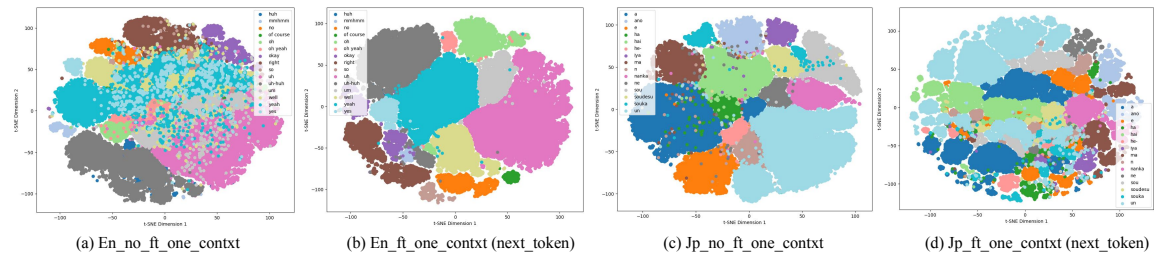


Figure 16: t-SNE plots of the backchannels/fillers embeddings from the **GPT-2** model (**NTP**). Setting: **one-context**.

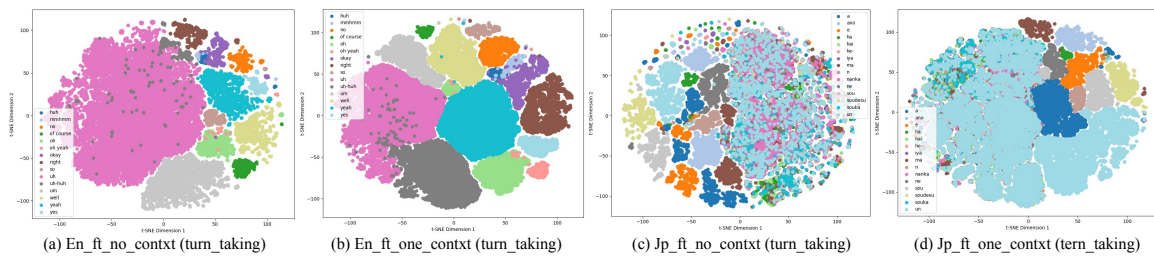


Figure 17: t-SNE plots of the backchannels/fillers embeddings from the **GPT-2** model (**TTP**). Setting: **no-** and **one-context**.

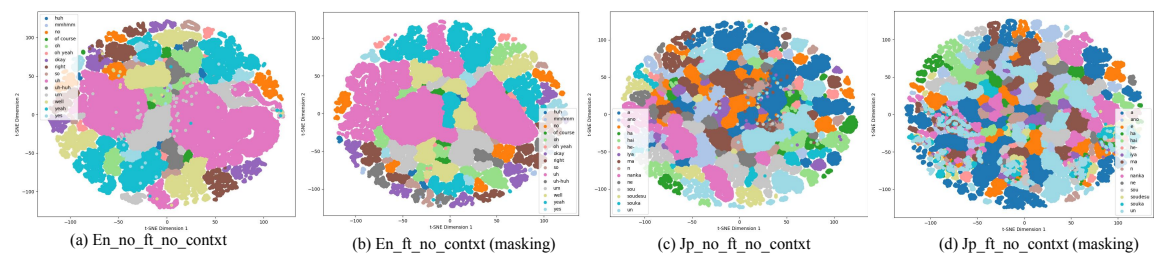


Figure 18: t-SNE plots of the backchannels/fillers embeddings from the **BERT** model (**MASK**). Setting **no-context**.

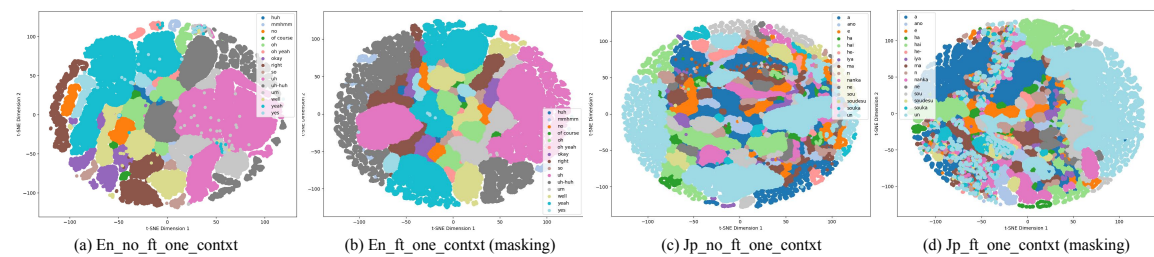


Figure 19: t-SNE plots of the backchannels/fillers embeddings from the **BERT** model (**MASK**); Setting: **one-context**.

Table 6: Evaluation results across different models and tasks. Results are reported as mean \pm 95% bootstrap confidence interval half-width. ‘Base’ denotes the original model without fine-tuning, and ‘FT’ denotes the fine-tuned model. Full-context evaluation is exclusive to LLaMA-3 and Qwen-3.

Model	Task	Lang	No-Context		One-Context		Full-Context	
			Base	FT	Base	FT	Base	FT
BERT	MASK	EN	0.144 \pm 0.011	0.241 \pm 0.008	0.213 \pm 0.008	0.391 \pm 0.017	—	—
		JP	0.213 \pm 0.008	0.391 \pm 0.017	0.197 \pm 0.004	0.429 \pm 0.012	—	—
GPT-2	NTP	EN	0.274 \pm 0.012	0.328 \pm 0.021	0.149 \pm 0.002	0.311 \pm 0.020	—	—
		JP	0.157 \pm 0.004	0.288 \pm 0.010	0.101 \pm 0.003	0.273 \pm 0.008	—	—
	TTP	EN	—	0.289 \pm 0.019	—	0.211 \pm 0.018	—	—
		JP	—	0.284 \pm 0.014	—	0.261 \pm 0.014	—	—
LLaMA-3	NTP	EN	0.450 \pm 0.009	0.588 \pm 0.011	0.183 \pm 0.002	0.291 \pm 0.027	0.210 \pm 0.020	0.301 \pm 0.018
		JP	0.257 \pm 0.009	0.450 \pm 0.014	0.179 \pm 0.003	0.335 \pm 0.010	0.318 \pm 0.021	0.408 \pm 0.037
Qwen-3	NTP	EN	0.253 \pm 0.013	0.379 \pm 0.065	0.157 \pm 0.003	0.292 \pm 0.021	0.189 \pm 0.020	0.322 \pm 0.028
		JP	0.172 \pm 0.011	0.452 \pm 0.068	0.154 \pm 0.002	0.263 \pm 0.014	0.173 \pm 0.004	0.181 \pm 0.012

Table 7: Maximum silhouette scores on the k -means clustering of the top 15 selected **English** backchannels/fillers when using the **LLaMA-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA (no_ft: no fine-tuning; ft: fine-tuning; Settings: no_ctx: no-context; one_ctx: one-context; full_ctx: full-context)

Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (NTP)		no_ft, one_ctx		ft, one_ctx (NTP)		no_ft, full_ctx		ft, full_ctx (NTP)	
	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100
uh	0.302	0.355	0.470	0.500	0.090	0.114	0.057	0.097	0.052	0.233	0.059	0.095
yeah	0.418	0.423	0.565	0.579	0.195	0.205	0.319	0.387	0.084	0.194	0.318	0.392
uh-huh	0.470	0.490	0.606	0.623	0.232	0.191	0.363	0.431	0.147	0.222	0.372	0.437
well	0.403	0.417	0.519	0.479	0.226	0.216	0.186	0.349	0.227	0.275	0.292	0.359
right	0.403	0.433	0.596	0.572	0.211	0.206	0.402	0.496	0.182	0.202	0.410	0.506
oh	0.337	0.364	0.506	0.513	0.205	0.206	0.239	0.279	0.121	0.182	0.226	0.278
um	0.335	0.374	0.482	0.516	0.109	0.127	0.076	0.149	0.054	0.102	0.074	0.144
okay	0.420	0.425	0.583	0.563	0.170	0.203	0.323	0.391	0.170	0.295	0.346	0.427
no	0.424	0.438	0.591	0.580	0.207	0.229	0.262	0.363	0.128	0.167	0.294	0.415
yes	0.397	0.422	0.600	0.598	0.159	0.191	0.267	0.358	0.155	0.280	0.276	0.372
so	0.332	0.339	0.538	0.579	0.163	0.142	0.376	0.477	0.101	0.120	0.386	0.485
oh yeah	0.525	0.359	0.676	0.620	0.183	0.192	0.089	0.141	0.154	0.208	0.078	0.135
huh	0.434	0.461	0.590	0.632	0.186	0.184	0.129	0.161	0.099	0.153	0.107	0.137
mmhmm	0.559	0.505	0.661	0.690	0.232	0.230	0.123	0.247	0.306	0.362	0.103	0.217
of course	0.350	0.376	0.522	0.580	0.081	0.091	0.060	0.102	0.034	0.079	0.048	0.087
Average \uparrow	0.407	0.424	0.567	0.575	0.177	0.182	0.218	0.295	0.134	0.205	0.226	0.299

Table 8: Maximum silhouette scores value on the k -means clustering of the top 15 selected **English** backchannels/fillers when using the **Qwen-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (NTP)		no_ft, one_ctx		ft, one_ctx (NTP)		no_ft, full_ctx		ft, full_ctx (NTP)	
	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100
uh	0.333	0.172	0.126	0.106	0.332	0.144	0.119	0.155	0.286	0.107	0.153	0.174
yeah	0.396	0.252	0.222	0.231	0.312	0.192	0.339	0.203	0.220	0.146	0.391	0.197
uh-huh	0.376	0.224	0.265	0.286	0.320	0.199	0.393	0.445	0.170	0.135	0.444	0.471
well	0.287	0.143	0.847	0.236	0.243	0.129	0.319	0.237	0.442	0.217	0.382	0.221
right	0.322	0.199	0.837	0.266	0.353	0.185	0.427	0.476	0.224	0.270	0.483	0.514
oh	0.262	0.134	0.202	0.234	0.294	0.135	0.269	0.317	0.215	0.122	0.337	0.301
um	0.316	0.156	0.147	0.132	0.295	0.111	0.178	0.222	0.298	0.130	0.202	0.226
okay	0.349	0.176	0.281	0.345	0.304	0.142	0.335	0.382	0.482	0.260	0.403	0.445
no	0.358	0.204	0.204	0.272	0.395	0.200	0.279	0.340	0.208	0.221	0.356	0.418
yes	0.291	0.207	0.825	0.278	0.313	0.132	0.236	0.354	0.521	0.209	0.351	0.219
so	0.322	0.163	0.815	0.266	0.238	0.136	0.257	0.291	0.253	0.253	0.461	0.490
oh yeah	0.417	0.478	0.909	0.742	0.295	0.154	0.083	0.128	0.198	0.110	0.075	0.128
huh	0.356	0.255	0.725	0.676	0.262	0.151	0.162	0.189	0.329	0.145	0.131	0.152
mmhmm	0.389	0.412	0.614	0.635	0.336	0.189	0.106	0.186	0.640	0.216	0.184	0.201
of course	0.272	0.155	0.312	0.185	0.274	0.130	0.202	0.213	0.185	0.099	0.199	0.180
Average \uparrow	0.336	0.222	0.489	0.326	0.304	0.155	0.247	0.276	0.311	0.176	0.304	0.289

Table 9: Maximum silhouette scores value on the k -means clustering of the top 15 selected **English** backchannels/fillers when using the **GPT-2** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (NTP)		ft, no_ctx (TTP)		no_ft, one_ctx		ft, one_ctx (NTP)		ft, one_ctx (TTP)	
	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100
<i>uh</i>	0.355	0.168	0.443	0.068	0.307	0.110	0.404	0.110	0.475	0.072	0.277	0.109
<i>yeah</i>	0.423	0.206	0.351	0.378	0.356	0.147	0.365	0.160	0.408	0.413	0.277	0.132
<i>uh-huh</i>	0.490	0.282	0.727	0.391	0.271	0.310	0.680	0.172	0.489	0.460	0.299	0.194
<i>well</i>	0.417	0.170	0.414	0.282	0.440	0.311	0.400	0.136	0.332	0.276	0.286	0.123
<i>right</i>	0.433	0.219	0.291	0.388	0.315	0.289	0.406	0.149	0.498	0.487	0.288	0.352
<i>oh</i>	0.364	0.176	0.434	0.205	0.388	0.146	0.363	0.131	0.514	0.330	0.271	0.270
<i>um</i>	0.374	0.150	0.479	0.069	0.349	0.105	0.413	0.105	0.489	0.166	0.314	0.186
<i>okay</i>	0.425	0.213	0.744	0.397	0.282	0.368	0.438	0.161	0.371	0.411	0.237	0.318
<i>no</i>	0.438	0.257	0.746	0.417	0.337	0.309	0.382	0.149	0.334	0.353	0.269	0.219
<i>yes</i>	0.422	0.211	0.753	0.329	0.384	0.229	0.383	0.139	0.360	0.350	0.273	0.188
<i>so</i>	0.339	0.166	0.404	0.395	0.244	0.320	0.486	0.149	0.526	0.457	0.246	0.231
<i>oh yeah</i>	0.359	0.575	0.731	0.559	0.509	0.556	0.374	0.167	0.367	0.262	0.298	0.113
<i>huh</i>	0.461	0.246	0.761	0.337	0.293	0.170	0.366	0.150	0.501	0.194	0.296	0.217
<i>mmhmm</i>	0.505	0.481	0.779	0.549	0.512	0.529	0.442	0.214	0.289	0.268	0.260	0.193
<i>of course</i>	0.376	0.149	0.318	0.094	0.348	0.121	0.422	0.124	0.314	0.096	0.281	0.108
Average ↑	0.409	0.245	0.558	0.324	0.356	0.263	0.422	0.148	0.418	0.306	0.278	0.196

Table 10: Maximum silhouette scores value on the k -means clustering of the top 15 selected **English** backchannels/fillers when using the **BERT** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (MASK)		no_ft, one_ctx		ft, one_ctx (MASK)	
	orig.	100	orig.	100	orig.	100	orig.	100
<i>uh</i>	0.097	0.122	0.083	0.103	0.172	0.189	0.054	0.074
<i>yeah</i>	0.085	0.118	0.216	0.245	0.098	0.119	0.079	0.104
<i>uh-huh</i>	0.104	0.129	0.267	0.343	0.121	0.148	0.119	0.156
<i>well</i>	0.072	0.095	0.211	0.252	0.050	0.064	0.186	0.233
<i>right</i>	0.095	0.127	0.251	0.313	0.110	0.153	0.354	0.426
<i>oh</i>	0.079	0.097	0.225	0.256	0.090	0.113	0.075	0.093
<i>um</i>	0.091	0.117	0.144	0.178	0.192	0.206	0.078	0.112
<i>okay</i>	0.088	0.114	0.219	0.216	0.074	0.097	0.175	0.218
<i>no</i>	0.093	0.117	0.190	0.242	0.067	0.086	0.195	0.255
<i>yes</i>	0.096	0.122	0.206	0.227	0.071	0.089	0.122	0.146
<i>so</i>	0.144	0.180	0.177	0.220	0.069	0.093	0.219	0.291
<i>oh yeah</i>	0.115	0.142	0.162	0.178	0.119	0.139	0.114	0.137
<i>huh</i>	0.229	0.294	0.242	0.283	0.103	0.126	0.189	0.194
<i>mmhmm</i>	0.205	0.241	0.270	0.331	0.103	0.109	0.093	0.132
<i>of course</i>	0.069	0.096	0.088	0.111	0.050	0.073	0.088	0.103
Average ↑	0.111	0.141	0.197	0.233	0.099	0.120	0.143	0.178

Table 11: Maximum silhouette scores value on the k -means clustering of the top 15 selected **Japanese** backchannels/fillers when using the **LLaMA-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (NTP)		no_ft, one_ctx		ft, one_ctx (NTP)		no_ft, full_ctx		ft, full_ctx (NTP)	
	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100
うん (un)	0.265	0.255	0.432	0.442	0.148	0.192	0.246	0.350	0.342	0.348	0.442	0.224
あ (a)	0.264	0.213	0.422	0.427	0.147	0.177	0.237	0.341	0.328	0.300	0.149	0.419
はい (hai)	0.261	0.247	0.440	0.464	0.182	0.184	0.305	0.395	0.331	0.235	0.238	0.321
え (e)	0.290	0.262	0.427	0.430	0.144	0.186	0.251	0.378	0.311	0.233	0.402	0.261
そう (sou)	0.287	0.270	0.429	0.428	0.153	0.184	0.240	0.386	0.324	0.344	0.427	0.291
ま (ma)	0.282	0.242	0.400	0.414	0.114	0.131	0.206	0.316	0.336	0.361	0.139	0.450
なんか (nanka)	0.330	0.309	0.496	0.529	0.111	0.163	0.191	0.286	0.345	0.389	0.472	0.532
あの (ano)	0.262	0.252	0.412	0.427	0.147	0.115	0.162	0.281	0.240	0.247	0.123	0.201
ん (n)	0.291	0.261	0.429	0.428	0.129	0.162	0.307	0.412	0.142	0.339	0.198	0.269
そうです (soudesu)	0.316	0.283	0.410	0.434	0.133	0.228	0.169	0.280	0.349	0.356	0.464	0.175
は (ha)	0.228	0.210	0.352	0.355	0.116	0.164	0.242	0.343	0.265	0.316	0.359	0.315
ね (ne)	0.249	0.246	0.369	0.380	0.115	0.174	0.332	0.436	0.141	0.322	0.311	0.400
いや (iya)	0.283	0.205	0.418	0.474	0.166	0.181	0.246	0.328	0.319	0.358	0.423	0.465
へー (he-)	0.336	0.253	0.533	0.591	0.185	0.221	0.154	0.232	0.405	0.436	0.560	0.605
そうか (souka)	0.319	0.252	0.442	0.455	0.158	0.192	0.271	0.373	0.328	0.369	0.203	0.225
Average ↑	0.284	0.256	0.427	0.445	0.143	0.177	0.237	0.343	0.300	0.330	0.327	0.344

Table 12: Maximum silhouette scores value on the k -means clustering of the top 15 selected **Japanese** backchannels/fillers when using the **Qwen-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (NTP)		no_ft, one_ctx		ft, one_ctx (NTP)		no_ft, full_ctx		ft, full_ctx (NTP)	
	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100
うん (un)	0.292	0.183	0.666	0.564	0.363	0.186	0.275	0.235	0.189	0.130	0.208	0.154
あ (a)	0.272	0.172	0.157	0.535	0.344	0.151	0.225	0.208	0.291	0.188	0.165	0.168
はい (hai)	0.322	0.242	0.359	0.496	0.356	0.222	0.302	0.240	0.632	0.242	0.168	0.200
え (e)	0.291	0.115	0.740	0.578	0.371	0.134	0.198	0.211	0.213	0.126	0.179	0.154
そう (sou)	0.295	0.191	0.706	0.557	0.353	0.149	0.266	0.289	0.299	0.187	0.175	0.169
ま (ma)	0.295	0.168	0.318	0.290	0.328	0.122	0.335	0.330	0.152	0.149	0.130	0.124
なんか (nanka)	0.243	0.164	0.792	0.153	0.304	0.115	0.155	0.171	0.182	0.172	0.156	0.171
あの (ano)	0.292	0.178	0.770	0.223	0.295	0.131	0.184	0.209	0.201	0.137	0.140	0.161
ん (n)	0.286	0.151	0.739	0.167	0.360	0.155	0.286	0.241	0.252	0.174	0.190	0.144
そうです (soudesu)	0.306	0.122	0.702	0.558	0.343	0.133	0.203	0.267	0.256	0.165	0.150	0.176
は (ha)	0.337	0.227	0.335	0.299	0.365	0.219	0.340	0.313	0.320	0.183	0.204	0.224
ね (ne)	0.298	0.140	0.346	0.405	0.357	0.122	0.450	0.493	0.205	0.114	0.370	0.336
いや (iya)	0.294	0.144	0.742	0.609	0.339	0.419	0.215	0.219	0.198	0.109	0.152	0.162
へー (he-)	0.321	0.151	0.843	0.723	0.406	0.183	0.256	0.223	0.170	0.196	0.182	0.177
そうか (souka)	0.304	0.128	0.732	0.587	0.298	0.106	0.224	0.250	0.178	0.261	0.176	0.154
Average ↑	0.296	0.165	0.596	0.450	0.345	0.152	0.261	0.260	0.249	0.169	0.183	0.178

Table 13: Maximum silhouette scores value on the k -means clustering of the top 15 selected **Japanese** backchannels/fillers when using the **GPT-2** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (NTP)		ft, no_ctx (TTP)		no_ft, one_ctx		ft, one_ctx (NTP)		ft, one_ctx (TTP)	
	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100
うん (un)	0.122	0.153	0.241	0.261	0.302	0.311	0.099	0.103	0.235	0.262	0.316	0.321
あ (a)	0.114	0.119	0.193	0.229	0.256	0.249	0.083	0.081	0.238	0.241	0.292	0.283
はい (hai)	0.186	0.201	0.338	0.390	0.240	0.258	0.107	0.120	0.362	0.388	0.187	0.211
え (e)	0.156	0.150	0.265	0.269	0.293	0.295	0.097	0.087	0.258	0.293	0.319	0.320
そう (sou)	0.146	0.140	0.264	0.305	0.327	0.335	0.122	0.104	0.274	0.312	0.337	0.342
ま (ma)	0.118	0.146	0.204	0.252	0.267	0.278	0.095	0.093	0.194	0.249	0.294	0.303
なんか (nanka)	0.118	0.155	0.103	0.134	0.185	0.195	0.102	0.095	0.119	0.148	0.150	0.174
あの (ano)	0.116	0.160	0.299	0.325	0.268	0.279	0.092	0.093	0.289	0.340	0.290	0.304
ん (n)	0.107	0.133	0.237	0.278	0.275	0.286	0.082	0.079	0.263	0.296	0.299	0.296
そうです (soudesu)	0.136	0.165	0.345	0.369	0.363	0.309	0.119	0.116	0.159	0.178	0.103	0.136
は (ha)	0.135	0.160	0.274	0.313	0.199	0.227	0.121	0.128	0.292	0.330	0.258	0.214
ね (ne)	0.091	0.128	0.284	0.347	0.161	0.208	0.074	0.091	0.288	0.340	0.249	0.250
いや (iya)	0.154	0.163	0.178	0.310	0.226	0.348	0.123	0.100	0.172	0.197	0.164	0.182
へー (he-)	0.187	0.210	0.341	0.375	0.284	0.365	0.104	0.089	0.258	0.284	0.155	0.183
そうか (souka)	0.158	0.152	0.254	0.278	0.249	0.302	0.134	0.110	0.204	0.246	0.208	0.243
Average ↑	0.136	0.156	0.255	0.296	0.260	0.283	0.104	0.099	0.240	0.273	0.241	0.251

Table 14: Maximum silhouette scores value on the k -means clustering of the top 15 selected **Japanese** backchannels/fillers when using the **BERT** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (MASK)		no_ft, one_ctx		ft, one_ctx (MASK)	
	orig.	100	orig.	100	orig.	100	orig.	100
うん (un)	0.232	0.265	0.385	0.459	0.211	0.242	0.463	0.487
あ (a)	0.147	0.166	0.399	0.393	0.141	0.171	0.447	0.428
はい (hai)	0.288	0.330	0.423	0.195	0.209	0.240	0.440	0.213
え (e)	0.164	0.191	0.293	0.364	0.150	0.170	0.406	0.425
そう (sou)	0.217	0.258	0.379	0.419	0.205	0.226	0.403	0.495
ま (ma)	0.148	0.177	0.400	0.430	0.136	0.160	0.414	0.491
なんか (nanka)	0.105	0.134	0.329	0.372	0.115	0.143	0.406	0.455
あの (ano)	0.117	0.157	0.342	0.386	0.114	0.145	0.372	0.425
ん (n)	0.155	0.192	0.429	0.488	0.161	0.190	0.523	0.569
そうです (soudesu)	0.136	0.170	0.227	0.303	0.119	0.140	0.351	0.393
は (ha)	0.208	0.250	0.403	0.437	0.204	0.247	0.431	0.484
ね (ne)	0.206	0.264	0.324	0.371	0.169	0.210	0.377	0.424
いや (iya)	0.162	0.185	0.408	0.461	0.165	0.185	0.462	0.511
へー (he-)	0.160	0.176	0.249	0.287	0.152	0.180	0.081	0.104
そうか (souka)	0.203	0.237	0.346	0.375	0.258	0.296	0.410	0.436
Average ↑	0.177	0.210	0.356	0.383	0.167	0.196	0.399	0.423

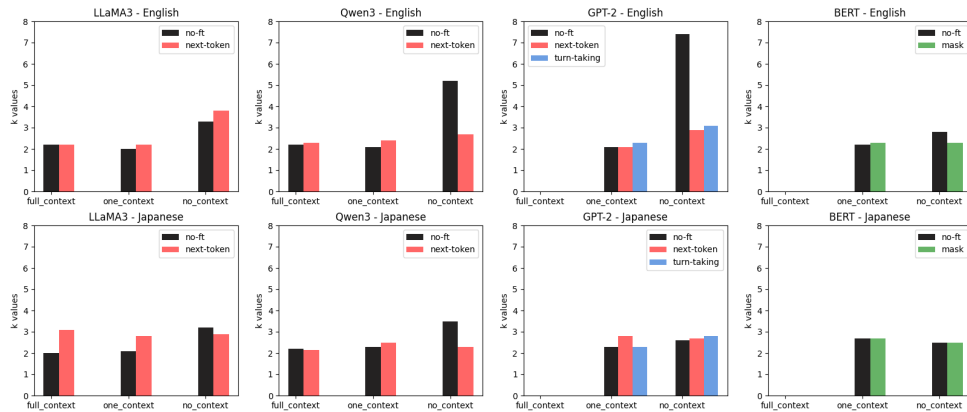


Figure 20: Change in the k -value before and after fine-tuning, as extracted from the clustering analysis of different LMs' results. A slight increasing of k after fine-tuning can generally be observed among the different fine-tuning strategies. Exceptions are the no-context setting for the LLaMA-3 model for Japanese and the Qwen-3 model for both languages. It is worth noting that a decrease of the k -value does not necessarily indicate failure, but can also mean that the clustering effect is more salient after fine-tuning.

Table 15: k -values of optimal results on the k -means clustering of the top 15 selected **English** backchannels/fillers when using the **LLaMA-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (NTP)		no_ft, one_ctx		ft, one_ctx (NTP)		no_ft, full_ctx		ft, full_ctx (NTP)	
	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100
<i>uh</i>	2	2	2	3	2	2	2	2	2	2	2	2
<i>yeah</i>	2	2	3	3	2	2	2	2	3	2	2	2
<i>uh-huh</i>	2	2	3	3	2	2	2	2	2	2	2	2
<i>well</i>	2	2	2	2	2	2	3	2	2	2	2	2
<i>right</i>	2	2	2	2	2	2	2	2	3	2	2	2
<i>oh</i>	2	2	2	3	2	2	2	2	2	2	3	3
<i>um</i>	2	2	3	3	2	2	2	2	3	2	2	2
<i>okay</i>	2	2	2	3	2	2	2	2	3	2	2	2
<i>no</i>	2	2	2	3	2	2	2	2	3	3	2	2
<i>yes</i>	2	2	2	3	2	2	2	2	2	2	2	2
<i>so</i>	2	2	2	3	2	2	2	2	4	4	2	2
<i>oh yeah</i>	3	9	3	11	2	2	4	4	2	2	3	3
<i>huh</i>	2	2	3	3	2	2	2	2	2	2	2	2
<i>mmhmm</i>	2	15	3	9	2	2	3	3	2	2	3	3
<i>of course</i>	2	2	2	3	2	2	3	2	2	2	2	2
Average	2.07	3.33	2.40	3.80	2.00	2.00	2.33	2.20	2.47	2.20	2.07	2.20

Table 16: k -values of optimal results on the k -means clustering of the top 15 selected **English** backchannels/fillers when using the **Qwen-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (NTP)		no_ft, one_ctx		ft, one_ctx (NTP)		no_ft, full_ctx		ft, full_ctx (NTP)	
	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100
<i>uh</i>	2	2	2	2	2	2	2	2	2	2	2	2
<i>yeah</i>	2	2	4	3	2	2	2	3	2	3	2	3
<i>uh-huh</i>	2	2	2	3	2	2	2	2	2	2	2	2
<i>well</i>	2	2	2	3	2	2	2	3	2	2	2	3
<i>right</i>	2	5	2	3	2	2	2	2	2	2	2	2
<i>oh</i>	2	4	2	4	2	2	3	3	2	2	2	3
<i>um</i>	2	2	2	2	2	2	2	2	2	2	2	2
<i>okay</i>	2	2	3	3	2	2	2	2	2	3	2	2
<i>no</i>	2	14	5	3	2	2	2	2	2	3	2	2
<i>yes</i>	2	10	2	3	2	3	3	2	2	2	2	3
<i>so</i>	2	2	2	3	2	2	3	3	2	2	2	2
<i>oh yeah</i>	14	15	2	2	2	2	5	3	2	2	4	3
<i>huh</i>	2	2	2	2	2	2	2	2	2	2	2	2
<i>mmhmm</i>	2	12	2	2	2	2	4	3	2	2	2	2
<i>of course</i>	3	2	2	3	2	2	2	2	2	2	2	2
Average	2.87	5.20	2.40	2.73	2.00	2.07	2.53	2.40	2.00	2.20	2.13	2.33

Table 17: k -values of optimal results on the k -means clustering of the top 15 selected **English** backchannels/fillers when using the **GPT-2** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (NTP)		ft, no_ctx (TTP)		no_ft, one_ctx		ft, one_ctx (NTP)		ft, one_ctx (TTP)	
	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100
<i>uh</i>	2	2	2	2	2	2	2	3	2	2	2	2
<i>yeah</i>	2	2	2	2	2	2	2	2	2	2	2	2
<i>uh-huh</i>	2	4	2	3	2	2	2	2	2	2	2	2
<i>well</i>	2	7	2	2	2	2	2	2	3	3	2	4
<i>right</i>	2	15	2	3	2	2	2	2	2	2	2	2
<i>oh</i>	2	2	2	2	2	4	2	2	2	3	2	2
<i>um</i>	2	2	2	2	2	2	2	2	2	2	2	2
<i>okay</i>	2	15	2	3	2	2	2	2	2	2	3	2
<i>no</i>	2	8	2	3	2	2	2	2	2	2	2	2
<i>yes</i>	3	13	2	2	2	5	2	2	2	2	2	2
<i>so</i>	2	7	2	3	2	5	2	2	2	2	2	4
<i>oh yeah</i>	3	15	2	4	2	3	2	2	2	2	2	2
<i>huh</i>	2	2	2	2	2	4	3	2	2	2	2	2
<i>mmhmm</i>	2	15	2	2	4	7	2	2	2	2	2	2
<i>of course</i>	2	2	2	5	2	2	2	2	2	2	2	2
Average	2.13	7.40	2.00	2.87	2.13	3.07	2.07	2.07	2.13	2.13	2.07	2.27

Table 18: k -values of optimal results on the k -means clustering of the top 15 selected **English** backchannels/fillers when using the **BERT** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (MASK)		no_ft, one_ctx		ft, one_ctx (MASK)	
	orig.	100	orig.	100	orig.	100	orig.	100
<i>uh</i>	2	2	2	2	2	2	2	2
<i>yeah</i>	2	3	2	2	2	2	3	3
<i>uh-huh</i>	3	3	3	2	2	2	2	2
<i>well</i>	2	2	2	3	2	2	2	2
<i>right</i>	2	2	2	2	2	2	2	2
<i>oh</i>	4	3	2	2	2	2	3	5
<i>um</i>	2	2	2	2	2	2	2	2
<i>okay</i>	9	9	2	4	2	2	2	2
<i>no</i>	2	2	2	2	2	2	2	2
<i>yes</i>	5	2	2	2	2	2	2	2
<i>so</i>	2	2	3	3	2	3	3	2
<i>oh yeah</i>	4	3	3	3	2	2	2	2
<i>huh</i>	2	2	2	2	2	2	2	2
<i>mmhmm</i>	2	3	2	2	3	4	2	2
<i>of course</i>	2	2	2	2	2	2	2	2
Average	3.00	2.80	2.20	2.33	2.06	2.20	2.20	2.27

Table 19: k -values of optimal results on the k -means clustering of the top 15 selected **Japanese** backchannels/fillers when using the **LLaMA-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (NTP)		no_ft, one_ctx		ft, one_ctx (NTP)		no_ft, full_ctx		ft, full_ctx (NTP)	
	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100
うん (un)	3	3	2	3	2	2	2	4	2	2	2	3
あ (a)	3	3	2	3	2	2	4	3	2	2	2	2
はい (hai)	3	5	2	3	2	2	2	2	2	2	3	2
え (e)	3	3	2	3	2	2	2	4	2	2	2	6
そう (sou)	3	3	2	3	2	3	3	5	2	2	2	5
ま (ma)	3	3	2	3	2	2	2	3	2	2	3	2
なんか (nanka)	2	3	3	3	2	2	2	2	2	2	2	2
あの (ano)	2	3	2	2	2	3	2	2	2	2	3	3
ん (n)	3	3	2	2	2	2	2	2	2	2	3	3
そうです (soudesu)	3	3	4	4	2	2	3	2	2	2	2	4
は (ha)	3	4	2	3	2	2	3	3	2	2	2	3
ね (ne)	3	3	2	2	2	2	2	2	3	3	3	3
いや (iya)	2	3	2	3	2	3	2	2	2	2	2	2
へー (he-)	2	3	3	3	2	2	2	2	2	2	2	2
そうか (souka)	3	3	3	3	2	2	4	4	2	2	5	4
Average	2.73	3.20	2.33	2.87	2.00	2.13	2.53	2.80	2.13	2.00	2.53	3.07

Table 20: k -values of optimal results on the k -means clustering of the top 15 selected **Japanese** backchannels/fillers when using the **Qwen-3** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

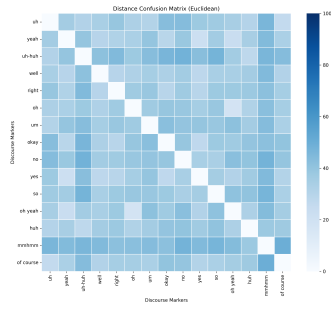
Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (NTP)		no_ft, one_ctx		ft, one_ctx (NTP)		no_ft, full_ctx		ft, full_ctx (NTP)	
	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100
うん (un)	2	3	2	2	2	2	2	3	2	4	2	2
あ (a)	2	2	2	2	2	2	2	3	2	2	2	2
はい (hai)	2	2	4	2	2	2	2	3	2	2	2	2
え (e)	2	5	2	2	2	2	2	3	3	2	2	2
そう (sou)	2	2	2	2	2	2	2	2	2	2	2	2
ま (ma)	2	2	2	2	2	3	2	2	2	2	2	4
なんか (nanka)	2	2	2	5	2	2	2	4	2	2	2	2
あの (ano)	2	2	2	2	2	2	2	2	2	2	2	2
ん (n)	2	2	2	2	2	2	2	2	2	2	2	2
そうです (soudesu)	2	15	2	2	2	2	2	2	2	2	2	2
は (ha)	2	2	4	3	2	2	2	2	2	2	2	2
ね (ne)	2	2	2	2	2	2	2	2	2	2	2	2
いや (iya)	2	3	2	2	2	2	3	3	3	3	2	2
へー (he-)	2	7	2	2	2	2	2	2	2	2	2	2
そうか (souka)	2	2	2	2	2	5	3	2	2	2	2	2
Average	2.00	3.53	2.27	2.27	2.00	2.27	2.07	2.47	2.07	2.20	2.00	2.13

Table 21: k -values of optimal results on the k -means clustering of the top 15 selected **Japanese** backchannels/fillers when using the **GPT-2** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

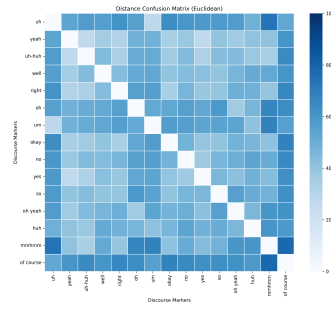
Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (NTP)		ft, no_ctx (TTP)		no_ft, one_ctx		ft, one_ctx (NTP)		ft, one_ctx (TTP)	
	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100	orig.	100
うん (un)	2	4	2	2	2	2	2	4	4	4	2	2
あ (a)	2	2	3	5	2	2	2	2	3	6	2	2
はい (hai)	2	2	2	2	3	2	2	2	2	2	2	2
え (e)	2	2	6	3	2	2	2	2	3	3	2	2
そう (sou)	2	2	2	3	2	2	2	4	2	3	2	2
ま (ma)	2	3	2	2	2	2	2	2	3	2	2	2
なんか (nanka)	2	2	2	2	2	2	2	2	2	2	2	2
あの (ano)	2	2	2	2	2	2	2	2	2	2	2	2
ん (n)	2	3	2	3	2	2	2	3	2	2	2	2
そうです (soudesu)	3	3	2	2	3	3	2	2	5	4	2	3
は (ha)	2	2	2	2	3	3	2	2	2	2	2	3
ね (ne)	3	2	2	2	4	3	2	2	3	3	2	3
いや (iya)	2	5	6	2	6	2	2	2	2	2	2	2
へー (he-)	2	3	2	2	5	2	2	2	2	2	2	3
そうか (souka)	2	2	2	6	5	11	2	2	3	3	3	3
Average	2.13	2.60	2.60	2.67	3.00	2.80	2.00	2.33	2.67	2.80	2.07	2.33

Table 22: k -values of optimal results on the k -means clustering of the top 15 selected **Japanese** backchannels/fillers when using the **BERT** model in both the original dimensional space and after dimensionality reduction to 100 dimensions using PCA.

Backchannel/Filler	no_ft, no_ctx		ft, no_ctx (MASK)		no_ft, one_ctx		ft, one_ctx (MASK)	
	orig.	100	orig.	100	orig.	100	orig.	100
うん (un)	2	2	2	3	4	2	3	3
あ (a)	3	4	4	2	3	4	4	2
はい (hai)	2	2	2	4	2	2	2	2
え (e)	4	4	2	3	5	6	2	3
そう (sou)	2	2	2	2	2	4	2	3
ま (ma)	4	4	2	2	4	4	2	3
なんか (nanka)	2	2	2	2	2	2	2	2
あの (ano)	2	2	2	2	3	3	2	2
ん (n)	3	3	2	2	2	2	2	2
そうです (soudesu)	2	2	2	3	2	2	3	4
は (ha)	2	2	3	4	2	2	5	5
ね (ne)	2	2	2	2	2	2	2	2
いや (iya)	4	2	2	2	3	2	2	2
へー (he-)	2	2	2	2	2	2	2	4
そうか (souka)	3	3	2	2	2	2	2	2
Average	2.60	2.53	2.20	2.47	2.66	2.73	2.46	2.73

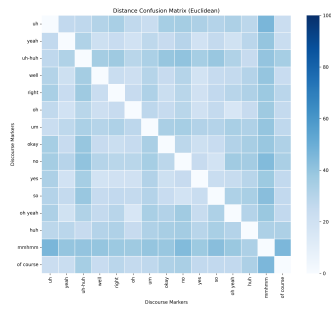


(a) no_ft, one-context

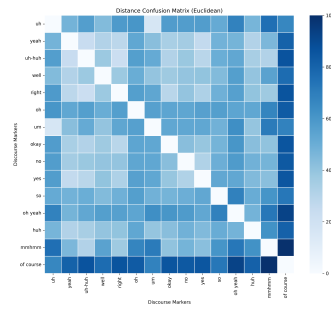


(b) ft, one-context (NTP)

Figure 21: Distance matrices for the top 15 **English** backchannels/fillers in the **LLaMA-3** model (a) before and (b) after fine-tuning.



(a) no_ft, one-context

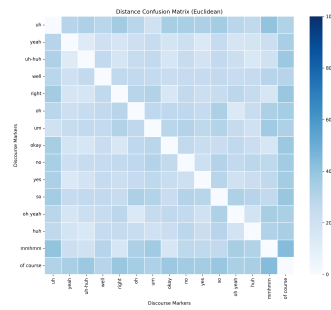


(b) ft, one-context (NTP)

Figure 22: Distance matrices for the top 15 **English** backchannels/fillers in the **Qwen-3** model (a) before and (b) after fine-tuning.

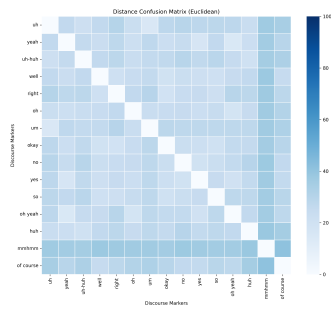


(a) no_ft, one-context

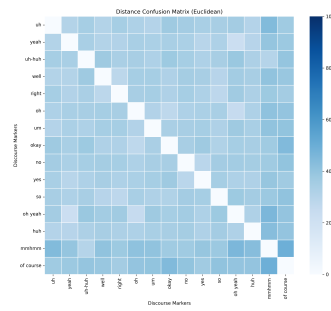


(b) ft, one-context (NTP)

Figure 23: Distance matrices for the top 15 **English** backchannels/fillers in the **GPT-2** model (a) before and (b) after fine-tuning.

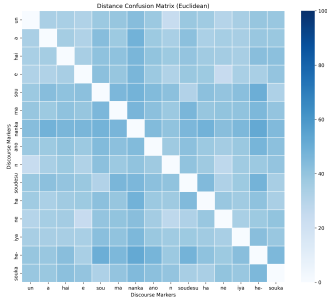


(a) no_ft, one-context

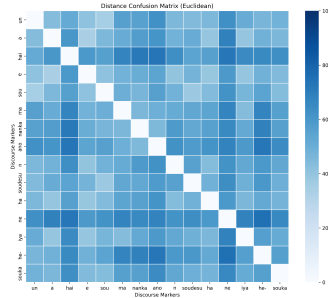


(b) ft, one-context (MASK)

Figure 24: Distance matrices for the top 15 **English** backchannels/fillers in the **BERT** model (a) before and (b) after fine-tuning.

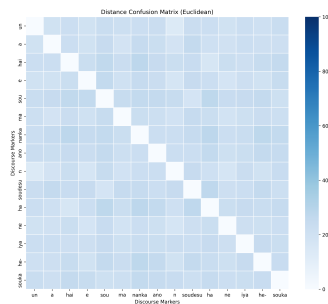


(a) no_ft, one-context

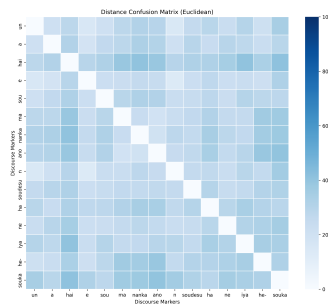


(b) ft, one-context (NTP)

Figure 25: Distance matrices for the top 15 **Japanese** backchannels/fillers in the **LLaMA-3** model (a) before and (b) after fine-tuning.

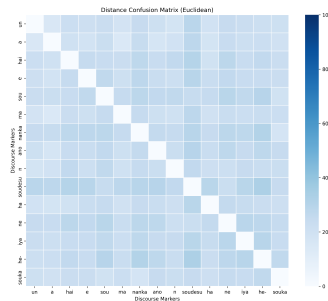


(a) no_ft, one-context

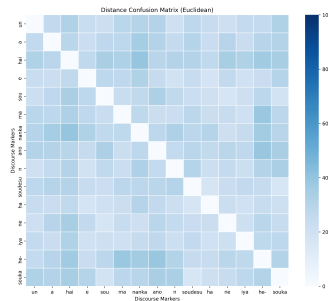


(b) ft, one-context (NTP)

Figure 26: Distance matrices for the top 15 **Japanese** backchannels/fillers in the **GPT-2** model (a) before and (b) after fine-tuning.



(a) no_ft, one-context



(b) ft, one-context (MASK)

Figure 27: Distance matrices for the top 15 **Japanese** backchannels/fillers in the **BERT** model (a) before and (b) after fine-tuning.