日本語追加学習は LLM の内部表現をロバストにするか：コミュニティ分類における回路・SAE 分析

深澤祐援
コミューン株式会社
fukasawa.yusuke@commune.co.jp

概要

大規模言語モデルの内部メカニズム解明は重要な研究課題である. 本研究では日本語コミュニティ分類タスクを対象に, 同一アーキテクチャで日本語追加学習の有無のみが異なる日本語版 (gemma-2-2b-jpn-it) とベース版 (gemma-2-2b-it) の内部メカニズムを比較分析した. 分類性能は同等にもかかわらず, 日本語版は広範囲に分散した回路構造と Label-specific な SAE feature を多く持つ一方, ベース版は後半層に集中した回路で Shared feature に依存する傾向が見られた. さらに, ベース版が少数手掛かりへ依存し脆弱であるのに対し, 日本語版は多様な手掛かりを用い feature knockout への耐性と表記ゆれへの頑健性が高いことを示した. 日本語追加学習により, 日本語の多様な表現に対応するロバストな特徴構造の形成が促進される可能性が示唆された.

1 はじめに

大規模言語モデル (LLM) は自然言語処理の様々なタスクで高い性能を示している. しかし, LLM 内部の処理は依然ブラックボックスであり, その解明は重要課題である. 近年, Sparse Autoencoder (SAE) を用いた特徴量抽出により, LLM 内部の解釈可能な表現を取り出す研究が進展している [1]. Nikankin ら [2] は計算タスクで fine-tuning した LLM を分析し, モデルが多数の手掛かり（ヒューリスティック）で解空間を段階的に絞り込んでいる可能性を示唆した. しかし, 意味理解を要するタスクでも同様のメカニズムが働くか, また言語追加学習がこの手掛かり依存構造に影響を与えるかは未解明である.

本研究では, この問いに答えるためテキスト分類タスクを分析対象とする. 具体的には, オンラインコミュニティへの投稿文から, その投稿が属するコミュニティを推定する分類タスクを扱う. このタスクは, 投稿内容の意味を理解し, コミュニティごとの特徴（話題, 文体, 専門用語など）を捉える必要がある. データセットとしてはコミューン株式会社が運営するサービス Commune の日本語コミュニティ投稿データを用いる.

ただし, 1 つのモデルのみを分析した場合, 観察されるメカニズムがタスク由来か, モデルの言語処理能力に由来するかを切り分けることが難しい. 本研究では日本語タスクを扱うため, 日本語処理能力が異なるモデル間で比較を行えば, 言語処理能力がメカニズムに与える影響を識別できる. そこで, 同一アーキテクチャで日本語追加学習の有無のみが異なる日本語版 (gemma-2-2b-jpn-it) とベース版 (gemma-2-2b-it) の 2 モデルを比較する [3]. 日本語版は, ベース版に対して日本語テキストで fine-tuning を施したモデルである. 両モデルで共通するメカニズムはタスク由来, 差異として現れるメカニズムは日本語追加学習による変化と解釈できる. また, 回路構造の分散性/局所性や手掛かり依存度を定量化する指標を導入し, Activation Patching, SAE, feature knockout, ならびに手掛かり多様性・表記ゆれ分析を通じてモデル間で比較する. 本研究の主な貢献は, (1) 日本語追加学習により回路構造と Label-specific Feature の形成が変化することの発見, (2) feature knockout と手掛かり多様性・表記ゆれ分析により, 日本語版がロバストな冗長性を持つことを因果的に示したことである. これらの結果は, 特定言語の追加学習が同言語のタスクに対するモデルの内部戦略に影響を与えることを示唆している.

2 関連研究

2.1 LLM の解釈可能性研究

LLM の内部メカニズムを解明する Mechanistic Interpretability 研究では, Transformer を回路として捉

え特定タスクに寄与するサブグラフを発見するアプローチが主流である [4]. 間接目的語同定 [5] や誘導ヘッド [6] の回路同定に始まり, 近年では感情分析 [7] や知識想起 [8] など意味理解を伴うタスクへも分析が広がっている. 主要な手法には Activation Patching [9], Logit Lens [10], SAE [11, 1] がある. また, Prakash ら [12] は fine-tuning が既存回路を増幅するのみで構造は変化しないとする増幅仮説を提唱したが, 言語適応でも同様かは未検証である. これらの研究は英語中心の設定であり, 日本語の意味理解を要する分類タスクでの比較分析は相対的に少ない.

2.2 言語適応と内部表現

多言語モデルの内部表現に関する研究では, language-specific neurons の同定 [13] や, 言語ごとの表現空間の幾何学的性質 [14], 単言語・多言語モデルでの共有/固有回路の併存 [15] が報告されている. 特に Tang ら [13] は言語特異的ニューロンが入出力層付近に集中する U 字型分布を報告し, Zhang ら [15] は異言語間で類似回路が共有されることを示した. しかし, 同一アーキテクチャで追加学習の有無のみが異なるモデル間で内部回路や特徴表現を直接比較した研究は限られている.

3 回路構造の分析

本章ではまず全実験で共通のタスク設定を述べ, その後 Activation Patching による回路抽出を行う.

3.1 タスク設定

コミュニティ分類タスクのデータセットとして, Commune 上のランダムに選んだ 107 コミュニティから投稿データを収集した. 学習データは各コミュニティから 280 投稿をランダムサンプリングし, 評価データは同様に 120 投稿をサンプリングした. ただし, 異なるコミュニティ間でテキストが完全に一致する投稿 190 件を除外したため, 評価データは 12,650 件となった. 投稿の期間は 2019 年 10 月から 2025 年 9 月である. 入力テキストはタイトルと本文を改行で連結し, 512 トークンを超える場合は末尾を切り詰めた. なお, 本研究で使用するデータは個人情報を含まず, 分析は統計量のみを報告する.

対象の 2 モデルに対して分類ヘッドを追加し, 3 エポックの Full Fine-tuning を行った. 表 1 にテストデータでの性能を示す. 日本語版が Macro F1 84.2% と僅かに高い性能を示したが, ベース版も 83.5% とほぼ同等であった. なお全モデルが TFIDF+SVC を大きく上回っている. 以降の章では Fine-tuning 後のモデルを対象に分析を行う.

表 1 Fine-tuning 後のコミュニティ分類性能

モデル	Macro F1
gemma-2-2b-jpn-it	**84.2**
gemma-2-2b-it	83.5
TFIDF+SVC	67.60

3.2 Activation Patching による回路抽出

Fine-tuning 後のモデルを対象に Activation Patching を行った. まず, 評価データ中で各モデルが正しく分類できた投稿を用い, 投稿（タイトル+本文）の最終層隠れ状態を平均プーリングした埋め込みに基づいて, コサイン類似度が 0.8 以上かつラベルが異なる投稿ペア（Clean, Corrupted）を 1000 ペア抽出する. 分類ヘッドは末尾トークンの隠れ状態を入力とするため, 介入も末位置で行う. 具体的には, Corrupted 入力に対して層 l の MLP 出力または Attention 出力（各々 block 全体の出力）の末尾 position の活性を, Clean 入力時の対応する活性で置換して再推論する. Effect を次式で定義する（z_y は正解ラベル y の logit）:

$$\mathrm{Effect}(l, c, p) = \frac{z_y^{\mathrm{patched}}(l, c, p) - z_y^{\mathrm{corr}}}{z_y^{\mathrm{clean}} - z_y^{\mathrm{corr}}}. \tag{1}$$

なお, 分母が極小（$< 10^{-6}$）の場合は Effect を 0 とした. 全ペアで平均した Effect が閾値 $\tau = 0.01$ 以上のコンポーネントを回路 C として抽出する. 回路の因果的重要性として, 回路のみを保持した場合の精度比（Sufficiency）と回路を除去した場合の精度低下（Necessity）を測定した.

3.3 結果: 各モデルの回路構造の比較

図 1 に示すように, 日本語版は 28 コンポーネントが回路に含まれ, Layer 0–25 まで広範囲に分布する分散型構造を示した. 一方, ベース版は 13 コンポーネントのみが抽出され, Layer 16–25 に集中する後半集中型構造を示した. 層別の効果量分布を比較すると, 日本語版は前半・中間層で 42% を占める一方, ベース版は後半層に 96% が集中した. 効果量合計は日本語版が 2.3 倍大きく, Sufficiency/Necessity も日本語版が高い. これらの結果は, 追加学習により, より多くの層がタスクに寄与する分散的な回路が形成されることを示唆している.

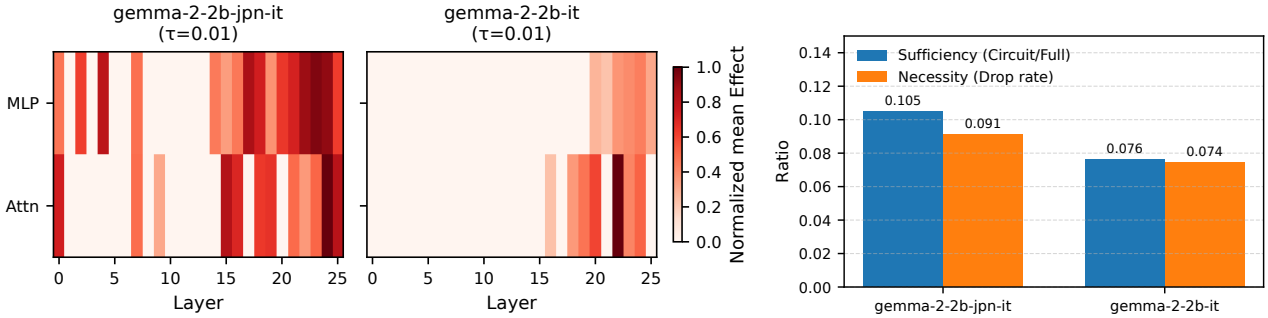

図 1 (Left) Activation Patching により抽出した回路の層 × コンポーネント分布（コンポーネント単位で可視化, 色は各モデル内で最大値正規化した mean Effect）. (Right) 抽出回路の Sufficiency と Necessity.

4 特徴量分析

4.1 実験設定：SAE による Feature 分解

LLM の中間表現は重畳しているため, Sparse Autoencoder（SAE）[16] を用いて解釈可能な特徴量に分解する. 各層の残差ストリームから SAE を学習し, 各特徴量がラベルに対してどの程度特異的に活性化するかを Cohen's d で測定した. SAE の説明分散は両モデルでほぼ同等であり, 全層平均で約 89% であった. 各特徴量について, Cohen's $d > 1.0$ となるラベル数を **label variety** と定義する. variety=0 はどのラベルにも強く反応しない弱い特徴, variety=1 は 1 ラベルにのみ強く反応する **Label-specific Feature**, variety≥2 は複数ラベルに反応する **Shared Feature** と解釈できる.

4.2 結果：Label-specific Feature の傾向

日本語版は全層で Label Variety（Cohen's $d > 1.0$ となるラベル数）がわずかに低い傾向が見られた（0.76 vs 0.77）. この差は小さいが, 表 2 に示すように高 Cohen's d feature の絶対数では日本語版が 65% 多く（1490 vs 904）, そのうち単一ラベル特化率（=Label-specific）も高い（58.3% vs 53.8%）. すなわち, Label Variety の平均値に大きな差はないものの, Label-specific Feature の絶対数において日本語版が優位であり, 追加学習によりこうした特徴の形成が促進される可能性が示唆される.

表 2 Label-specific Feature の統計

モデル	高 d Feature 数	単一ラベル特化率
日本語版	1490	58.3%
ベース版	904	53.8%

4.3 Knockout 実験

SAE により抽出した feature が因果的に重要か, またモデル間でロバスト性に差があるかを検証するため, 各ラベルについて Cohen's d が大きい上位 20 feature を選び, 対象層の残差ストリーム上でそれらの活性化を 0 にする feature knockout を行った. Control として, 同数の feature をランダムに選んで knockout した場合は精度変化が ±0.1% 以内であり, 介入対象 feature が意味のある構造を捉えていることを確認した.

7 分野は, 各モデルの SAE において Cohen's d が大きく活性が顕著であったラベルの中から, ドメインの多様性を確保するよう選定した. 表 3 に, 7 分野のコミュニティに対する Recall 低下率を示す. ベース版は特定分野でほぼ完全に崩壊する一方, 日本語版は平均低下が -58.1% に抑えられ, 相対的に頑健であった. これは日本語版が Label-specific Feature を多く持ち, それらが冗長に配置されているため, 上位 feature を破壊しても残りの feature でラベル特定が可能であることを示唆する.

表 3 Knockout 実験: ターゲットラベル Recall の低下（top-20 feature, zero ablation）. 値は knockout 前後の Recall の絶対差（ポイント）.

分野	jpn-it (L24) ΔRecall	it (L22) ΔRecall
メーカー系	-88.6	-66.7
就活系	-84.2	-87.5
食品系	-62.5	-95.0
自動車系	-20.0	-92.5
研究開発系	-51.7	-29.2
ホビー系	-28.3	-67.5
不動産系	-71.7	-80.8
平均	-58.1	-74.2

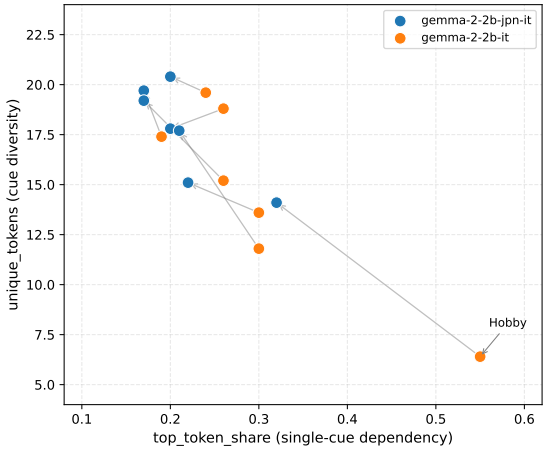

図 2　手掛かり多様性の比較. X 軸: 単一手がかり依存度, Y 軸: 発火トークンの多様性.

5　ロバスト性の要因分析

ここまで, 日本語版は分散的な回路構造と Label-specific feature を多く持つこと, そして feature knockout に対して相対的に頑健であることを示した. 本章では「なぜ日本語追加学習によりロバストな内部表現が形成されるのか」を説明するため, 文脈の手掛かり多様性と, 表記ゆれに対する冗長性を定量化する.

5.1　手掛かり多様性

前章の Knockout 実験で用いた feature が, どのような表層手がかり（トークン）に反応しているかを分析した. Knockout 実験の対象とした 7 分野について, 効果量が最大であった層（日本語版: Layer 24, ベース版: Layer 22）に限定して分析を行う.

各 feature について, 評価データ中で強く発火したトークンを収集した. ある feature が特定のトークンのみで強く発火する場合, その feature を knockout すると当該トークンを含む入力の分類が一様に失敗する. 一方, 複数の異なるトークンで発火する feature は, 1 つの手がかりが失われても他で補える. これらを意味する 2 指標を用いた: (i) **top_token_share**（最頻発火トークンの占有率, 高いほど単一依存）, (ii) **unique_tokens**（発火トークンの種類数, 多いほど多様）. 図 2 に示すように, 日本語版は多様な手がかりを用いる傾向がある. 特にホビー系ではベース版の top_token_share が 0.55 と高く, 日本語版（0.32）の約 1.7 倍であった.

表 4　表記ゆれに対する冗長性（平均生存率）

top-k	jpn-it	it	差分
20	**0.579**	0.501	+0.078
50	**0.473**	0.395	+0.078
100	**0.373**	0.314	+0.059

5.2　表記ゆれに対する冗長性

日本語追加学習がこの多様性をもたらすメカニズムとして, 同一概念に対する複数の表記（例: ミーティング/MTG/会議）をそれぞれ異なる feature で捕捉し, 相互に補償し合う可能性が考えられる. この仮説を検証するため, 50 概念 × 3–5 表記のペアを手動で用意して, 各表記を含む文を入力して SAE により top-k feature を抽出し, ある表記の feature 集合を除外したときに別表記の feature がどれだけ残るか（生存率）を計算した. 生存率が高いほど, 各表記が独自の feature を持ち, 相互に補償できることを意味する. 表 4 に示すように, 日本語版は $k = 20, 50, 100$ のいずれでも生存率が高く, 表記ゆれに対して feature-level の冗長性が高いことが確認された.

6　まとめ

本研究では, LLM によるコミュニティ分類の内部メカニズムを, Activation Patching, SAE, feature knockout, ならびに手掛かり多様性・表記ゆれ分析を通じて分析した. Gemma 2 2B をベースとして日本語追加学習の有無のみが異なる 2 モデルを比較した結果, 以下の知見が得られた. 日本語版は広範囲に分散した回路構造を持ち, Label-specific な SAE feature を多く形成する. 一方, ベース版は後半層に集中した回路構造を持ち, Shared feature に依存する. Feature knockout に対しても日本語版は相対的に頑健であり, 手掛かり多様性の分析からベース版が単一トークンに強く依存する傾向が示された. 日本語特有の表記ゆれに対する分析では, 日本語版は異なる表記に対して独立した feature を持ち, 1 つの表記を除外しても他の表記で補償できる冗長性を示した. これらの結果は, 日本語追加学習が日本語の多様な表現に対応するロバストな特徴構造の形成を促進することを示唆している. 今後は他言語や別タスクでの検証が必要である.

参考文献

[1] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. **Transformer Circuits Thread**, 2024.

[2] Yaniv Nikankin, Anja Reusch, Aaron Mueller, and Yonatan Belinkov. Arithmetic without algorithms: Language models solve math with a bag of heuristics. **ArXiv**, Vol. abs/2410.21272, , 2024.

[3] Gemma Team. Gemma. 2024.

[4] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. **Transformer Circuits Thread**, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[5] Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In **The Eleventh International Conference on Learning Representations**, 2023.

[6] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. **Transformer Circuits Thread**, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[7] Amartya Hatua. Mechanistic interpretability of GPT-2: Lexical and contextual layers in sentiment analysis. In **NeurIPS 2025 Workshop on Efficient Reasoning**, 2025.

[8] Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. Knowledge circuits in pretrained transformers. **CoRR**, Vol. abs/2405.17969, , 2024.

[9] Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In **Thirty-seventh Conference on Neural Information Processing Systems**, 2023.

[10] Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens, 2025.

[11] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. **Transformer Circuits Thread**, Vol. 2, , 2023.

[12] Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In **The Twelfth International Conference on Learning Representations**, 2024.

[13] Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. Language-specific neurons: The key to multilingual capabilities in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 5701–5715, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[14] Tyler A. Chang, Zhuowen Tu, and Benjamin K. Bergen. The geometry of multilingual language model representations. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, pp. 119–136, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[15] Ruochen Zhang, Mrinmaya Sachan, and Marek Rei. The same but different: Structural similarities and differences in multilingual language modeling. In **The Thirteenth International Conference on Learning Representations**, 2025.

[16] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In **The Twelfth International Conference on Learning Representations**, 2024.

A　付録

A.1　Fine-tuning 設定

表 5 に Fine-tuning のハイパーパラメータを示す. 両モデルで同一の設定を使用した.

表 5　Fine-tuning のハイパーパラメータ

項目	設定値
Learning rate	1e-5
Weight decay	0.01
Epochs	3
Batch size	2
Max length	512

A.2　SAE 学習設定

表 6 に SAE の学習設定を示す. 辞書サイズは隠れ層次元（2304）の 4 倍とした.

表 6　SAE の学習設定

項目	設定値
辞書サイズ	9216
L1 係数	0.03
学習率	1e-3
Batch size	512
Epochs	10
対象層	全層 (0–25)

A.3　Activation Sparsity（参考）

Label Variety の差が SAE の活性化パターンにも反映されているかを分析した（表 7）. Top-20 Ratio は, 活性化値の合計に対する上位 20 特徴量の割合である. 日本語版はアクティブな feature 数が多いが（568 vs 460）, Top-20 への集中度は低く（14.3% vs 16.6%）, より多くの feature を均等に使用する傾向を示した.

表 7　Activation Sparsity の比較

モデル	Active Features	Top-20 Ratio
日本語版	568	14.3%
ベース版	460	16.6%

A.4　同一層での Knockout 比較（参考）

本文の表 3 では各モデルで効果量が最大となる層（日本語版: L24, ベース版: L22）を対象としたが, 条件を統一した比較として同一層（L22）での結果を表 8 に示す. 同一層でも日本語版は平均 –67.9% に対しベース版は –74.2% と, 日本語版が相対的に頑健な傾向を維持している.

表 8　同一層（L22）での Knockout 実験

ラベル	jpn-it ΔRecall	it ΔRecall
233	-81.7	-87.5
2982	-93.3	-67.5
3013	-89.2	-80.8
448	-79.2	-95.0
777	-20.0	-92.5
88	-66.7	-66.7
891	-45.0	-29.2
平均	-67.9	-74.2

A.5　表記ゆれペアの例

表 9 に, 冗長性分析で使用した表記ゆれペアの例を示す. 全 50 概念について, 各 3–5 種類の表記とそれを含む文を手動で作成した.

表 9　表記ゆれペアの例（2 概念を抜粋）

概念	表記	入力文
ミーティング	ミーティング	今日はミーティングがあります。
	MTG	午後から MTG です。
	会議	会議の準備をしています。
	打ち合わせ	打ち合わせの時間を調整中です。
了解	了解	了解しました。
	りょうかい	りょうかいです！
	りょ	りょ！
	OK	OK です、進めてください。
	おけ	おけ〜！