

# Neural KWIC : KWIC形式用例からの 動的単語埋め込みの獲得と分析

島田 真緒<sup>1,a)</sup> 木山 朔<sup>1</sup> 凌 志棟<sup>1</sup> 小町 守<sup>1</sup> 小木曾 智信<sup>2</sup> 高村 大也<sup>3</sup> 持橋 大地<sup>2,4</sup>

**概要**：本研究では、KWIC によって検索された対象単語の文脈表現を変換する関数を学習し、事前学習済み言語モデルに基づく動的単語埋め込みを近似して分析する手法を提案する。KWIC (Keyword-in-Context) とは、コーパス内において指定したキーワードとその前後の文脈を固定長で局所的に取り出す文脈付きの索引の形式であり、言語学的分析において広く用いられている。しかし、実応用上の場面においては、大量の KWIC 用例の全容を解釈することは困難である。そこで本研究では、検索された KWIC 用例から単語埋め込みを動的に計算し、埋め込み空間上で KWIC を可視化することにより、解釈性を向上することを目的とする。変換関数の学習時には、入力として KWIC 形式で取得した文脈単語列を用い、事前学習済みの静的単語埋め込みで符号化した上で、出現順に連結した表現を採用する。これらの埋め込み表現を変換関数に入力し、正解として与えた事前学習済み言語モデルを近似する単語埋め込みを出力する。評価実験では、複数の対象単語に対してその語の出現文脈を KWIC 形式で取得し、本手法を適用した。得られた動的単語埋め込みに基づき、辞書により定義された語義数をクラスタ数として設定して可視化を行い、BERT による単語埋め込みとの一致度を定量的に評価した。実験の結果、本手法は計算効率と実用性に優れており、KWIC 形式に対応した動的単語埋め込みが高速に得られることを示した。また、埋め込み空間上での可視化を通じて、大量の KWIC 用例に対する解釈性を高められることを示した。

**キーワード**：KWIC、用例検索、単語埋め込み、高速化、可視化

## 1. はじめに

KWIC (Keyword-in-Context) とは、文書中に出現する特定のキーワードとその前後の文脈を、固定長で同時に取り出す文脈付きの索引のことである [1]。例えば、「絶対」という単語を KWIC 形式で取得した場合の例を表 1 に示す。KWIC では、対象単語そのものだけでなく、その前後に現れる文脈情報も合わせて確認できるため、文中における単語の用いられ方を直接的に把握することが可能である。このような特性から、KWIC は言語学的分析において広く用いられている [2-4]。しかし、実応用の場面においては、大量の KWIC 用例を一括して解釈することは容易ではない。

一方、近年の自然言語処理においては、用例間の意味的な差異を捉えるために、単語埋め込みに基づく手法が広く用いられている。単語埋め込みは、静的単語埋め込みと動的単語埋め込みの二種類に大別される。静的単語埋め込みでは、一つの単語に対して一つの固定された埋め込み表現

表 1: 中納言 [10] から抽出した「絶対」の KWIC 形式用例。

前文脈	キー	後文脈
両辺を 2 乗して 2 乗すると	絶対	値は気にしなくて良いので
で、主の顔は見えません。	絶対	的な権力者ではあっても、
みんな右。左が池なんで、	絶対	に右に打つんですよ。では
は親を大切に思っていて、	絶対	に親を裏切っても、親は親
：	：	：

が割り当てられるのに対し、動的単語埋め込みでは、同一の単語であっても用例ごとに異なる単語埋め込み表現が割り当てられる。語義曖昧性解消 [5-7] や意味変化検出 [8,9] などの研究分野では、このような動的単語埋め込みを用いて、同一単語に対して得られる複数の埋め込み表現に基づく語義の分析が行われている。

KWIC のような局所的な文脈情報から埋め込み表現を獲得する既存手法として、周辺単語の単語埋め込みの平均を入力とし、線形変換によって対象語の静的単語埋め込みを推定する手法 [11] や BERT [12] による手法が存在する。一方で、前者の手法では周辺単語の平均操作によって単語間の相対的な位置関係の情報が失われるとともに、推定される表現は各語や特徴に対して一意に定まるという性質を持つ。また、後者の手法では BERT を用いて動的な KWIC

<sup>1</sup> 一橋大学

<sup>2</sup> 国立国語研究所

<sup>3</sup> 産業技術総合研究所 人工知能研究センター

<sup>4</sup> 統計数理研究所 統計基盤数理研究系

<sup>a)</sup> mao@scl.sds.hit-u.ac.jp

形式の埋め込みが獲得できるが、数千から数万を超えることのある KWIC 用例について埋め込みを計算することは計算量が多く、また事前にコーパスすべての出現について埋め込みを計算して保存しておくことは容量の面からも不可能である。

そこで本研究では、既存手法である *à la carte* 埋め込み [11] のアルゴリズムを拡張し、動的単語埋め込みを獲得する手法を提案する。まず、KWIC 形式で取得した文脈単語の静的単語埋め込みを出現順に連結した表現を入力とし、事前学習済み言語モデルとして BERT に由来する動的単語埋め込みを教師信号として用いる。そして、この入力表現に対して一般的な形で定式化した変換関数を学習することで、BERT に基づく表現力豊かな動的単語埋め込みを近似する変換関数を学習する。(図 2) これにより、KWIC という局所的な文脈からであっても動的単語埋め込みの獲得が可能となり、埋め込み空間上での KWIC の可視化を通じて、大量の KWIC 用例に対する解釈を効率的に行うことができる。(図 1) また、本手法は BERT と比較して計算効率に優れている点も特徴である。

本研究の貢献は以下の通りである。

- KWIC 形式で取得した対象単語について、高速に動的単語埋め込みを計算する学習フレームワークを提案した。入力を KWIC 文脈の静的単語埋め込み、教師を動的単語埋め込みとして変換関数を学習することで、KWIC 形式の動的単語埋め込みの獲得を実現した。
- 定量評価によって、既存手法と比べて BERT に近い表現を獲得できていることを示した。また、変換関数は一度学習すれば、推論時には適用のみで済むため、高速に動的単語埋め込みを計算することができ、BERT と比較して計算効率に優れていることを示した。加えて、文脈単語数、品詞フィルタリング、教師埋め込みに対する前処理の影響を分析し、適切な学習設定について考察した。
- 定性分析によって、得られた動的単語埋め込みに対して、類似度行列の算出および低次元空間への可視化を行い、既存手法と比べて大量の KWIC 用例に対しても解釈が容易となることを示した。また、BERT と比較した時の品詞ごとのクラスタリング結果の傾向についても考察した。

## 2. 関連研究

### 2.1 *à la carte* 埋め込み

*à la carte* 埋め込み [11] は、事前学習済みの静的単語埋め込みを活用し、未知語や低頻度語を含めた、あらゆるテキスト特徴に対して分散表現を推定する手法である。その目的は、対象となる語や特徴が出現する文脈情報を用いて、各語・特徴に対して一意に定まる静的な埋め込み表現を獲

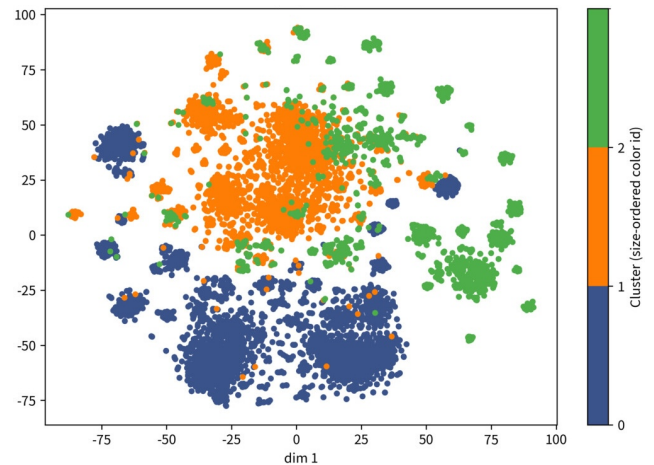


図 1: 対象単語「絶対」について、表 1 の KWIC 形式データから Neural KWIC により計算した動的単語埋め込みを *t*-SNE で可視化したもの。ラベルは BERT でのクラスタリング結果である。小さなクラスターが複数存在している様子が観察される。

得することにある。Algorithm 1 に *à la carte* 埋め込みの基本的なアルゴリズムを示す。

**入力表現:** 語彙  $\mathcal{V}$  に含まれる各単語  $w$  について、コーパス  $\mathcal{C}_{\mathcal{V}}$  から当該単語が出現する文脈集合  $\mathcal{C}_w$  を抽出する。ここで、各文脈  $c \in \mathcal{C}_w$  は、単語  $w$  が出現した位置の前後から取得された固定長の文脈であり、 $w$  の周囲に出現する単語集合である。文脈集合  $\mathcal{C}_w$  に含まれる単語の静的単語埋め込みを平均することで、文脈表現  $\mathbf{u}_w$  を次式で定義する：

$$\mathbf{u}_w = \frac{1}{|\mathcal{C}_w|} \sum_{c \in \mathcal{C}_w} \sum_{w' \in c} \mathbf{v}_{w'} \quad (1)$$

**学習段階:** 学習段階では、既知語  $w \in \mathcal{V}$  に対して、文脈表現  $\mathbf{u}_w$  から対応する静的単語埋め込み  $\mathbf{v}_w$  を予測する線形変換行列  $\mathbf{A}$  を学習する。この学習は、語彙全体に対する二乗誤差の最小化問題として定式化される。

**推論段階:** 推論段階では、対象となるテキスト特徴  $f$  が出現する文脈から、文脈単語の事前学習済み静的単語埋め込みを取得し、それらを平均することで文脈表現を構成する。この文脈表現に対して、学習済みの固定された線形変換を一度適用するだけで、任意のテキスト特徴  $f$  に対応する静的単語埋め込み  $\mathbf{v}_f$  を計算することができる。

*à la carte* 埋め込みは上述の通り、文脈単語の静的単語埋め込みを平均した表現を入力とし、各語や特徴に対して一意に定まる静的な埋め込み表現を推定することを目的としている。そのため、文脈内の語順や中心語との相対位置といった局所的な構造情報は、入力表現には明示的に含まれない。これに対し、本研究で提案する Neural KWIC は、KWIC 形式で切り出された局所的な文脈に着目し、文脈単語の静的単語埋め込みを出現順に連結することで、暗黙的に相対位置を考慮した入力表現を構成する。また、教師信号として事前学習済み言語モデルにより得られた文脈依存的な動的単語埋め込みを用いることで、文脈ごとに異なる

**Algorithm 1:** The basic à la carte feature embedding induction method. All contexts  $c$  consist of sequences of words drawn from the vocabulary  $\mathcal{V}$ . [11]

**Data :** vocabulary  $\mathcal{V}$ ; corpus  $\mathcal{C}_\mathcal{V}$ ; vectors  $\mathbf{v}_w \in \mathbb{R}^d$  for all  $w \in \mathcal{V}$ ; feature  $f$ ; corpus  $\mathcal{C}_f$  of contexts of  $f$ ;  
each context  $c \in \mathcal{C}_w$  is a fixed-length context of  $w$ , represented as a set of surrounding words

**Result:** feature embedding  $\mathbf{v}_f \in \mathbb{R}^d$

**for**  $w \in \mathcal{V}$  **do**

    let  $\mathcal{C}_w \subset \mathcal{C}_\mathcal{V}$  be the subcorpus of contexts of  $w$

$\mathbf{u}_w \leftarrow \frac{1}{|\mathcal{C}_w|} \sum_{c \in \mathcal{C}_w} \sum_{w' \in c} \mathbf{v}_{w'}$

**end**

$\mathbf{A} \leftarrow \arg \min_{\mathbf{A} \in \mathbb{R}^{d \times d}} \sum_{w \in \mathcal{V}} \|\mathbf{v}_w - \mathbf{A} \mathbf{u}_w\|_2^2$

$\mathbf{u}_f \leftarrow \frac{1}{|\mathcal{C}_f|} \sum_{c \in \mathcal{C}_f} \sum_{w \in c} \mathbf{v}_w$

$\mathbf{v}_f \leftarrow \mathbf{A} \mathbf{u}_f$

// compute each word's context embedding  $\mathbf{u}_w$

// compute context-to-feature transform  $\mathbf{A}$

// compute feature's context embedding  $\mathbf{u}_f$

// transform feature's context embedding

動的単語埋め込みを近似する点において、à la carte 埋め込みの拡張となっている。

## 2.2 知識蒸留

知識蒸留は、大規模モデルが獲得した知識を、より軽量なモデルへと転写することで、高い表現力を持つモデルの知識を維持しつつ、推論時の計算コストを削減することを目的とした学習方法である [13–17]。知識蒸留の多くは、ニューラルネットワークを生徒モデルとして用い、教師モデルの出力や中間表現との誤差を最小化することで学習を行う。一方で、蒸留対象となる表現や写像の形式は必ずしも非線形に限定されず、線形写像や低次元表現を用いた蒸留も検討されている。このような枠組みでは、蒸留された表現の解釈性や計算効率が重視される傾向にある。

本研究で提案する Neural KWIC は、事前学習済み言語モデルにより獲得された動的単語埋め込み表現を教師信号として用いる。KWIC に基づく文脈表現から、学習済みの変換行列によってその表現を近似する点において、表現蒸留の一形態ともみなすことができる。

## 3. Neural KWIC

本研究では、KWIC 形式で定義される文脈単語の静的単語埋め込みを出現順に連結し、変換関数  $g_\theta(\cdot)$  によって事前学習済み言語モデルに基づく動的単語埋め込みを近似する手法 **Neural KWIC** を提案する。Algorithm 2 に本手法の全体像を示した。

**入力表現：** まず、語彙  $\mathcal{V}$  から、KWIC を構築する対象単語集合  $\mathcal{V}' \subseteq \mathcal{V}$  を定義する。コーパス内の各出現に対して、中心語  $w_i \in \mathcal{V}'$  の左右  $k$  語ずつの文脈を切り出すことで、次のような KWIC 文脈

$$c_i = (w_{i,-k}, \dots, w_{i,-1}, w_i, w_{i,+1}, \dots, w_{i,+k}) \quad (2)$$

を構築する。ここで、 $w_{i,\ell}$  は KWIC 用例  $i$  において中心語

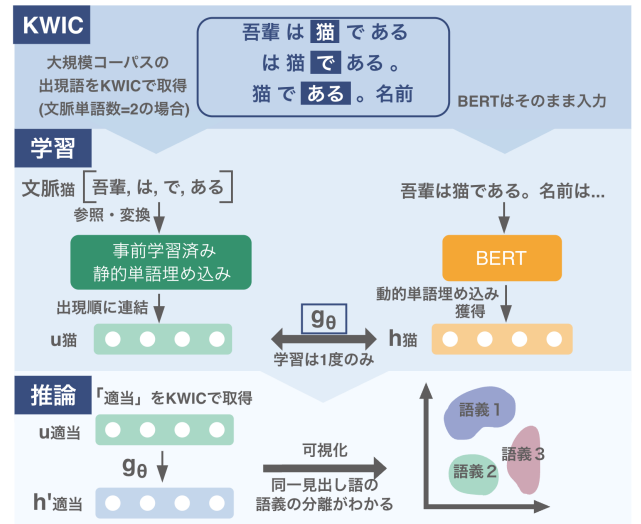


図 2: Neural KWIC: KWIC 形式の用例から事前学習済み言語モデルに基づく動的単語埋め込みを近似する提案手法の概要図。

$w_i$  から相対位置  $\ell$  に出現する単語を表し、 $\ell < 0$  は前文脈、 $\ell > 0$  は後文脈を表す。これらを集めたものを KWIC コーパス  $\mathcal{C}$  とする。各文脈単語  $w_{i,\ell}$  には、事前学習済みの静的単語埋め込み  $\mathbf{v}_{w_{i,\ell}} \in \mathbb{R}^{d_s}$  を割り当てる。すなわち、各 KWIC 文脈  $c_i$  に対して、中心語を除いた前後文脈の静的単語埋め込みを出現順序を保持したまま連結することで、

$$\mathbf{x}_{c_i} = [\mathbf{v}_{w_{i,-k}} \parallel \dots \parallel \mathbf{v}_{w_{i,-1}} \parallel \mathbf{v}_{w_{i,+1}} \parallel \dots \parallel \mathbf{v}_{w_{i,+k}}] \in \mathbb{R}^{2kd_s} \quad (3)$$

を入力表現として用いる。

**学習段階：** 学習段階では、各中心語  $w_i$  に対応する教師ベクトルとして、当該文を事前学習済み言語モデルに入力することで事前に取得した文脈依存的な動的単語埋め込み  $\mathbf{h}_{w_i}^{\text{PLM}} \in \mathbb{R}^{d_b}$  を用いる。すべての KWIC 文脈  $c_i \in \mathcal{C}$  について、入力表現  $\mathbf{x}_{c_i}$  と教師ベクトル  $\mathbf{h}_{w_i}^{\text{PLM}}$  の対応を用いて、次の二乗誤差の最小化問題を解くことで変換関数  $g_\theta(\cdot)$  を一度だけ学習する。

---

**Algorithm 2:** Neural KWIC: An Extension of the à la carte Framework for Learning a Transformation from Concatenated Static Embeddings of KWIC-based Context Words to Dynamic Contextual Embeddings

---

**Data :** vocabulary  $\mathcal{V}$ ; target word set  $\mathcal{V}' \subseteq \mathcal{V}$ ; KWIC corpus  $\mathcal{C}$  constructed for  $w \in \mathcal{V}'$  (each context  $c_i = (w_{i,-k}, \dots, w_{i,-1}, w_i, w_{i,+1}, \dots, w_{i,+k})$ ); static word embeddings  $\mathbf{v}_w \in \mathbb{R}^{d_s}$ ; contextual embeddings  $\mathbf{h}_w^{\text{PLM}} \in \mathbb{R}^{d_b}$  obtained from a pretrained language model; window size  $k$ ; feature  $f$ ; feature-context corpus  $\mathcal{C}_f \subset \mathcal{C}$ ; parametric transformation  $g_\theta(\cdot)$

**Result:** dynamic contextual embedding  $\mathbf{h}_{w_j} \in \mathbb{R}^{d_b}$

---

```
// Training (learn  $g_\theta$  once)
 $\mathbf{X} \leftarrow []$ ;  $\mathbf{Y} \leftarrow []$  // design matrix and targets
foreach  $c_i = (w_{i,-k}, \dots, w_{i,-1}, w_i, w_{i,+1}, \dots, w_{i,+k}) \in \mathcal{C}$  do
     $\mathbf{x}_{c_i} \leftarrow [\mathbf{v}_{w_{i,-k}} \parallel \dots \parallel \mathbf{v}_{w_{i,-1}} \parallel \mathbf{v}_{w_i} \parallel \dots \parallel \mathbf{v}_{w_{i,+k}}]$  // concatenate static embeddings of KWIC words
    append  $\mathbf{x}_{c_i}^\top$  to  $\mathbf{X}$ ; append  $\mathbf{h}_{w_i}^{\text{PLM}^\top}$  to  $\mathbf{Y}$ 
end
 $\theta \leftarrow \arg \min_\theta \sum_{i=1}^{|\mathcal{C}|} \|\mathbf{Y}_{i,:}^\top - g_\theta(\mathbf{X}_{i,:}^\top)\|_2^2$  // learn  $g_\theta$  (linear case reduces to à la carte)

// Inference (apply  $g_\theta$ )
 $\mathbf{x}_{c_j} \leftarrow [\mathbf{v}_{w_{j,-k}} \parallel \dots \parallel \mathbf{v}_{w_{j,-1}} \parallel \mathbf{v}_{w_{j,+1}} \parallel \dots \parallel \mathbf{v}_{w_{j,+k}}]$  // concatenated static context embeddings
 $\mathbf{h}_{w_j} \leftarrow g_\theta(\mathbf{x}_{c_j})$  // dynamic contextual embedding
```

---

$$\theta \leftarrow \arg \min_\theta \sum_{i=1}^{|\mathcal{C}|} \|\mathbf{h}_{w_i}^{\text{PLM}} - g_\theta(\mathbf{x}_{c_i})\|_2^2 \quad (4)$$

本研究では、à la carte 埋め込みと同様に

$$g_\theta(\mathbf{x}) = \mathbf{A}\mathbf{x} \quad (5)$$

を用いて  $\mathbf{A}$  を最適化した。ただし、この  $g_\theta(\mathbf{x})$  はより複雑なニューラル関数にも拡張可能である。

**推論段階：** 推論段階では、新たな用例  $c_j$  に対して同様に入力表現  $\mathbf{x}_{c_j}$  を構成し、学習済みの変換関数  $g_\theta(\cdot)$  を適用することで、中心語  $w_j$  に対応する動的単語埋め込み

$$\mathbf{h}_{w_j} = g_\theta(\mathbf{x}_{c_j}) \quad (6)$$

を高速に計算する。

**à la carte 埋め込みとの関係** なお、本手法は à la carte 埋め込み [11] を拡張した表現学習として位置付けられる。à la carte 埋め込みは、n-gram などの表層的な単位に捉われず、任意のテキスト特徴に対する埋め込み表現を獲得可能な手法である。本研究では、この枠組みを KWIC に適用することで拡張を行った。à la carte 埋め込みでは、文脈単語の静的単語埋め込みを平均したベクトルを入力表現として用いるのに対し、Neural KWIC では、KWIC 形式で定義された文脈単語の静的単語埋め込みを出現順序を保持したまま連結したベクトルを入力表現として用いる。この違いにより、Neural KWIC は文脈内の語順および位置情報を暗黙的に反映した表現学習を可能とする。また、à la carte 埋め込みでは教師ベクトルとして静的単語埋め込み

を用いるのに対し、Neural KWIC では事前学習済み言語モデルにより得られた文脈依存的な動的単語埋め込みを教師信号として用いる。これにより、Neural KWIC は KWIC に基づく局所的な文脈情報から、静的単語埋め込みではなく、事前学習済み言語モデルが獲得した動的単語埋め込みを近似することを目的としている。本手法によって得られる動的単語埋め込みを可視化することで、大量の KWIC 用例に対しても解釈が可能となり、同一の見出し語に対する用例間の意味的な差異を直感的に把握できる。さらに、à la carte 埋め込みでは、変換関数を線形写像に限定して定義している一方、Neural KWIC では、変換関数  $g_\theta(\cdot)$  として一般性の高い形に定式化しており、線形変換を含むより広い関数クラスを扱うことが可能である。

## 4. 実験設定

### 4.1 データセット

本節では、変換関数の学習および評価に用いたデータセットの詳細について説明する。学習用データセットの構成を示すとともに、評価に用いた 11 個の対象単語について述べる。

**学習用データセット** 変換関数の学習には、現代日本語書き言葉均衡コーパス (BCCWJ) [18] の CORE データを使用した。BCCWJ CORE は新聞や書籍、雑誌、白書、ウェブテキストなど、複数のジャンルから収集された書き言葉データで構成されている。本研究では、BCCWJ CORE に含まれる複数のサブコーパスを対象として、総トークン数 1,290,060 語からなる学習コーパスを用いた。表 2 に本研



表 2: 実験に使用した BCCWJ CORE データの統計量。

レジスター	トークン数
Yahoo!知恵袋	110,280
白書	228,172
Yahoo!ブログ	117,242
書籍	234,400
雑誌	239,440
新聞	360,526

究で使用した BCCWJ CORE の各サブコーパスと、それぞれのジャンルおよびトークン数を示す。なお、学習時に用いる KWIC の文脈に NWJC2vec に含まれない語彙が含まれる場合には、該当する KWIC を学習データから除外した。

**評価用データセット** 評価対象として、文脈に応じた意味の違いを持つ、一般的に使用頻度の高い多義語の中から、11 語を対象単語として選定した。(表 3) \*<sup>1</sup>これは実応用の場面において、同一の見出し語に対する文脈依存的な意味の差異を適切に捉えられているかを検証することを目的としているためである。対象単語の品詞および語義区分は、現代国語例解辞典〔第 5 版〕[19] に基づいて定義した。対象単語の用例は、中納言\*<sup>2</sup>[10] を用いて収集した。用例検索にあたっては、前述の目的に基づき、表層形による検索ではなく、見出し語単位での検索を行った。\*<sup>3</sup>これにより、活用形や表記揺れによる影響を抑えつつ、同一見出し語に属する多様な用法を、品詞の違いを含めて網羅的に扱うことが可能となる。

## 4.2 手法

Neural KWIC の学習においては、入力 of 静的単語埋め込みと教師の動的埋め込みの両方が必要となる。本研究では入力の単語静的埋め込みとして NWJC2vec を用いる。また教師信号の事前学習済み言語モデルに基づく動的単語埋め込みとして、双方向の文脈情報を考慮可能な BERT ベースのモデルである NWJC-BERT を用いる。NWJC2vec [20] は、国立国語研究所が構築した大規模日本語ウェブコーパス「NWJC」[21] に基づいて生成された静的単語埋め込みである。本研究では CBOW [22] モデルにより学習された 300 次元の単語埋め込みを採用した。NWJC-BERT [23] は、「NWJC」を用いて事前学習された BERT ベース [12] の大規模言語モデルである。ベースラインとしては、KWIC 用例における文脈単語の NWJC2vec に基づく静的単語埋め込みを平均した表現を用いる。

本研究では、提案手法の有効性を検証するため、以下の

\*<sup>1</sup> 5.3 節での分析単語以外の語義区分については付録 A.1 に示す。

\*<sup>2</sup> <https://chunagon.ninjal.ac.jp/>

\*<sup>3</sup> 用例検索には中納言を用い、短単位検索により対象単語を語彙素単位で検索した。検索対象は固定長・可変長の両方とし、文境界をまたぐ共起を許可した。

表 3: 実験に使用した対象単語のリスト。

対象単語	辞書品詞	語義数	用例数
「生理」	名詞	2	3,055
「姿勢」	名詞	2	5,962
「問題」	名詞	5	70,087
「養生」	名詞, サ変動詞	3	340
「消化」	名詞, サ変動詞	3	1,848
「運動」	名詞, サ変動詞	3	17,498
「適当」	名詞, 形容動詞	2	3,628
「絶対」	名詞, 形容動詞, 副詞	3	10,561
「薄い」	形容詞	5	5,226
「甘い」	形容詞	7	4,889
「高い」	形容詞	8	39,501

手法を比較する。

- **変換なし+平均**: 変換関数を用いず、KWIC 文脈の単語の静的単語埋め込みを単純平均することで文脈表現を構成する手法。本研究では、性能比較における下限として位置付ける。
- **à la carte+平均**: 文脈単語の静的単語埋め込みを単純平均した表現に対して、学習した線形変換関数を適用する手法。à la carte 埋め込みに対応する設定である。
- **à la carte+SIF**: 文脈単語の静的単語埋め込みに対して SIF 重み付け [24] を行った上で平均し、線形変換関数を適用する手法。
- **à la carte+concat**: 文脈単語の静的単語埋め込みを出現順に結合した表現に対して、学習した線形変換関数を適用する手法。本研究の提案手法に該当する。
- **BERT**: 各用例を BERT に入力し、中心語に対応する動的単語埋め込みを直接取得する手法。本研究では、性能比較における上限、かつ計算効率の下限として位置付ける。

これらの比較により、(1) 文脈単語の単純な集約と連結表現の違い、(2) 変換関数の有無による影響、(3) 提案手法と BERT による動的単語埋め込みとの関係を定量的に検証する。

## 4.3 変換関数の学習

Neural KWIC の変換関数の学習においては、文脈表現の構成方法および教師ベクトルの前処理に関して、いくつかの設計選択が存在する。本研究では、これらの要素が性能および計算効率に与える影響を明らかにするため、ablation 実験を通じて体系的に比較を行う。

**文脈単語数  $k$**  まず、文脈単語数  $k$  は、KWIC に基づく文脈表現が含む情報量を直接的に制御する重要な要素である。本研究では、 $k \in \{5, 10, 20\}$  と設定し、中心語に対して前後それぞれ  $k$  語ずつの文脈を用いた場合の性能および計算効率を比較する。例えば、 $k = 5$  の場合は、中心語の

前後5語ずつをKWIC文脈として用いる。

**学習データセットにおける品詞フィルタリング** 次に、学習に用いる中心語に対する品詞フィルタリングの有無を検討する。具体的には、名詞・動詞・形容詞・副詞・形状詞といった内容語<sup>\*4</sup>のみを用いる設定と、品詞による制約を設けない設定を比較する。これは、実応用において意味的に重要な単語に学習対象を限定することで、より安定した変換関数の学習が可能かどうかを検証する目的によるものである。なお、品詞によるフィルタリングは中心語に対してのみ適用し、中心語が内容語である場合には、KWIC文脈に機能語が含まれていてもそのまま入力表現の構成に用いる。

**教師ベクトルの前処理** さらに、教師ベクトルとして用いるBERT埋め込みに対して、Zipf白色化[25]を適用する場合と適用しない場合を比較する。Zipf白色化は、BERT埋め込みに内在する頻度依存的な異方性を緩和し、用例間の差異をより明確に反映させることを目的とした前処理である。本研究では、この前処理が変換関数の学習および最終的な埋め込み表現の品質に与える影響を検証する。なお、Zipf白色化を学習時に適用した場合であっても、評価時に元の空間に戻す逆変換は行っていない。これは後述する評価指標が、埋め込み空間そのものの一致度ではなく、用例間の相対的な類似度構造に基づいて性能を測定するためである。

#### 4.4 評価

提案手法によって得られる単語埋め込みが、BERTの表現構造とどの程度整合しているのか、また計算効率の観点においてどの程度実応用上有効であるかを、以下の4つの指標を用いて定量的に評価する。さらに、同一の見出し語が持つ多義性を文脈に応じて適切に分離できているかを確認するため、取得した単語埋め込みに基づく類似度行列および中心化後の類似度行列の算出に加え、主成分分析(PCA)およびt-SNEにより2次元空間へ射影した。これらの結果に対して、現代国語例解辞典[第5版][19]に基づく品詞分類に着目した定性的な分析を行う。

##### 4.4.1 定量評価

**BERTのクラスタラベルとの一致度 (ARI)** 各手法により獲得された単語埋め込みに対して、現代国語例解辞典[第5版][19]により定義された語義数をクラスタ数としてK-meansクラスタリングを行った。提案手法とBERTにより獲得した単語埋め込みに基づくクラスタラベルとの一致度を、Adjusted Rand Index (ARI)により評価する。ARIは $[-1, 1]$ の値をとり、値が大きいほどBERTのクラスタリング結果との一致度が高いことを示す。

**用例間の相対的類似性の一致度 ( $CKA_{raw}$ ,  $CKA_{norm}$ )**

提案手法によって得られる単語埋め込みが、BERTにより得られた単語埋め込みとどの程度用例間の相対的な類似性構造を共有しているのかを評価するため、Centered Kernel Alignment (CKA)[26]を用いた。具体的には、各手法で得られた単語埋め込みから、用例度の類似度行列を計算し、中心化を行わない場合( $CKA_{raw}$ )と、平均的な類似度の影響を除去するために中心化を行った場合( $CKA_{norm}$ )のそれぞれについて、Frobenius内積に基づくCKA値を算出する。CKAは $[0, 1]$ の値をとり、値が大きいほど提案手法とBERTによる用例間の相対的な類似性構造がより一致していることを示す。

**用例間類似度の相関係数 (Spearman's  $\rho$ )** 提案手法が獲得する単語埋め込みが、BERTにより獲得した単語埋め込みとどの程度用例間の類似度構造を共有しているかを評価するため、用例間類似度の相関係数を測定した。具体的には、まず各手法で得られた単語埋め込みに対して中心化を行った。これは、高頻度に高い類似度を与える用例(ハブ)の影響を考慮することを目的としている。その後、用例間のペアをランダムに1,000組抽出し、各ペアについて各手法ごとにcosine類似度を計算した。得られた類似度に対しては、両手法が与える類似度の順位関係がどの程度一致しているかを、Spearmanの順位相関係数により評価する。Spearmanの相関係数は $[-1, 1]$ の値をとり、値が大きいほど提案手法がBERTによる相対的な類似度構造をより忠実に再現していることを示す。<sup>\*5</sup>

**推論時間** 各手法について、対象単語ごとの全用例の推論時間を測定し、計算効率の観点から性能評価を行った。本研究では、代表例として「問題」に含まれる全70,087用例に対して単語埋め込みの計算に要する総推論時間を示す<sup>\*6</sup>。なお、すべての推論時間はIntel® Xeon® Gold 6426Y上で測定した。<sup>\*7</sup>

##### 4.4.2 定性評価

**用例間の類似度行列** 各手法について、単語埋め込み間の類似度行列と、中心化を施した後の類似度行列を算出している。いずれの類似度行列においても、用例はK-meansによって取得したクラスタラベルに基づいて並べ替えられており、同一クラスタ内の平均類似度およびクラスタ間の平均類似度を併記している。

**単語埋め込みの次元圧縮** 各手法で得られた単語埋め込みを主成分分析(PCA)およびt-SNEを用いて2次元空間

<sup>\*4</sup> 内容語は、BCCWJで採用されている品詞体系に基づき、名詞・動詞・形容詞・副詞・形状詞として定義した。

<sup>\*5</sup> 付録A.3ではcosine類似度ペアの散布図を各単語ごとに可視化している。補助指標として、類似度値そのものの一致度を評価するためにPearsonの相関係数を、さらに類似度値の誤差の大きさを定量化するために平均絶対誤差(MAE)および二乗平均平方根誤差(RMSE)も併記した。

<sup>\*6</sup> 他の対象単語についても同様に総推論時間を測定しており、その結果は付録A.2に掲載している。

<sup>\*7</sup> <https://www.intel.co.jp/content/www/jp/ja/products/sku/232377/intel-xeon-gold-6426y-processor-37-5m-cache-2-50-ghz/specifications.html>

表 4: 手法や学習条件を変えた時のモデル性能と計算効率の結果。

手法	学習条件			モデル性能				計算効率
	文脈単語数	品詞 filtering	Zipf 白色化	平均 ARI	平均 CKAraw	平均 CKAnorm	平均 Spearman's $\rho$	「問題」全用例の 推論時間 [s]
BERT (ベースライン)	5	—	—	1.000	1.000	1.000	1.000	79.379
	10	—	—	1.000	1.000	1.000	1.000	88.785
	20	—	—	1.000	1.000	1.000	1.000	97.978
変換なし+平均	5	—	—	0.142	0.366	0.382	0.365	—
	10	—	—	0.120	0.321	0.341	0.322	—
	20	—	—	0.122	0.313	0.336	0.323	—
à la carte+平均	5	—	—	0.172	0.384	0.406	0.371	0.019
		—	✓	0.193	0.461	0.468	0.423	0.018
		機能語	—	0.192	0.416	0.426	0.401	0.013
		機能語	✓	0.198	0.465	0.472	0.432	0.013
	10	—	—	0.176	0.390	0.412	0.395	0.018
		—	✓	0.247	0.439	0.450	0.429	0.018
		機能語	—	0.189	0.410	0.421	0.396	0.018
		機能語	✓	0.218	0.442	0.445	0.423	0.022
	20	—	—	0.215	0.412	0.430	0.404	0.023
		—	✓	0.265	0.441	0.454	0.421	0.024
		機能語	—	0.236	0.422	0.432	0.409	0.022
		機能語	✓	0.273	0.445	0.456	0.426	0.022
à la carte+SIF	5	—	—	0.171	0.361	0.394	0.371	0.021
		—	✓	0.225	0.464	0.474	0.426	0.021
		機能語	—	0.204	0.405	0.417	0.395	0.020
		機能語	✓	0.221	0.450	0.457	0.421	0.018
	10	—	—	0.191	0.365	0.397	0.377	0.022
		—	✓	0.224	0.449	0.459	0.428	0.022
		機能語	—	0.207	0.414	0.425	0.403	0.023
		機能語	✓	0.240	0.441	0.450	0.425	0.021
	20	—	—	0.217	0.366	0.405	0.378	0.021
		—	✓	0.299	0.443	0.454	0.419	0.029
		機能語	—	0.238	0.427	0.437	0.414	0.022
		機能語	✓	0.281	0.446	0.456	0.427	0.042
à la carte+concat	5	—	—	0.239	0.536	0.566	0.511	0.082
		—	✓	0.256	0.602	0.616	0.554	0.248
		機能語	—	0.270	0.611	0.630	0.578	0.153
		機能語	✓	0.317	0.640	0.657	0.587	0.207
	10	—	—	0.218	0.545	0.575	0.515	0.256
		—	✓	0.263	0.605	0.639	0.559	0.256
		機能語	—	0.319	0.621	0.640	0.582	0.252
		機能語	✓	<b>0.375</b>	<b>0.641</b>	<b>0.661</b>	<b>0.588</b>	0.206
	20	—	—	0.203	0.544	0.573	0.509	0.457
		—	✓	0.217	0.597	0.632	0.547	0.421
		機能語	—	0.281	0.616	0.635	0.568	0.379
		機能語	✓	0.273	0.630	0.651	0.572	0.409

に射影し、クラスタ構造の分離傾向を視覚的に比較する。  
なお、PCA および  $t$ -SNE による可視化において、各用例の色分けは、NWJC-BERT によるクラスタリング結果を基準として色付けを行っている。これにより、各手法が獲得した単語埋め込み上の空間において、BERT が形成するクラスタ構造がどのように再現されているかを直接的に比較することが可能となり、各手法における点群の分離構造と色分布との対応関係を観察することで、クラスタ構造の整合性を定性的に評価することができる。

## 5. 実験結果

### 5.1 定量評価

本節では、各評価指標に基づく定量評価の結果を示す。表 4 に、全対象単語に対して平均した定量評価の結果をまとめた。<sup>\*8</sup>なお、各指標において最も性能が高い結果を太字で示している。

**モデル性能** 表 4 より、提案手法である *à la carte+concat* は、いずれの設定においても、変換なし+平均やその他の *à la carte* 系手法と比較して、全体として高い性能を示し

<sup>\*8</sup> 各対象単語ごとの定量評価の結果は付録 A.2 に示す。

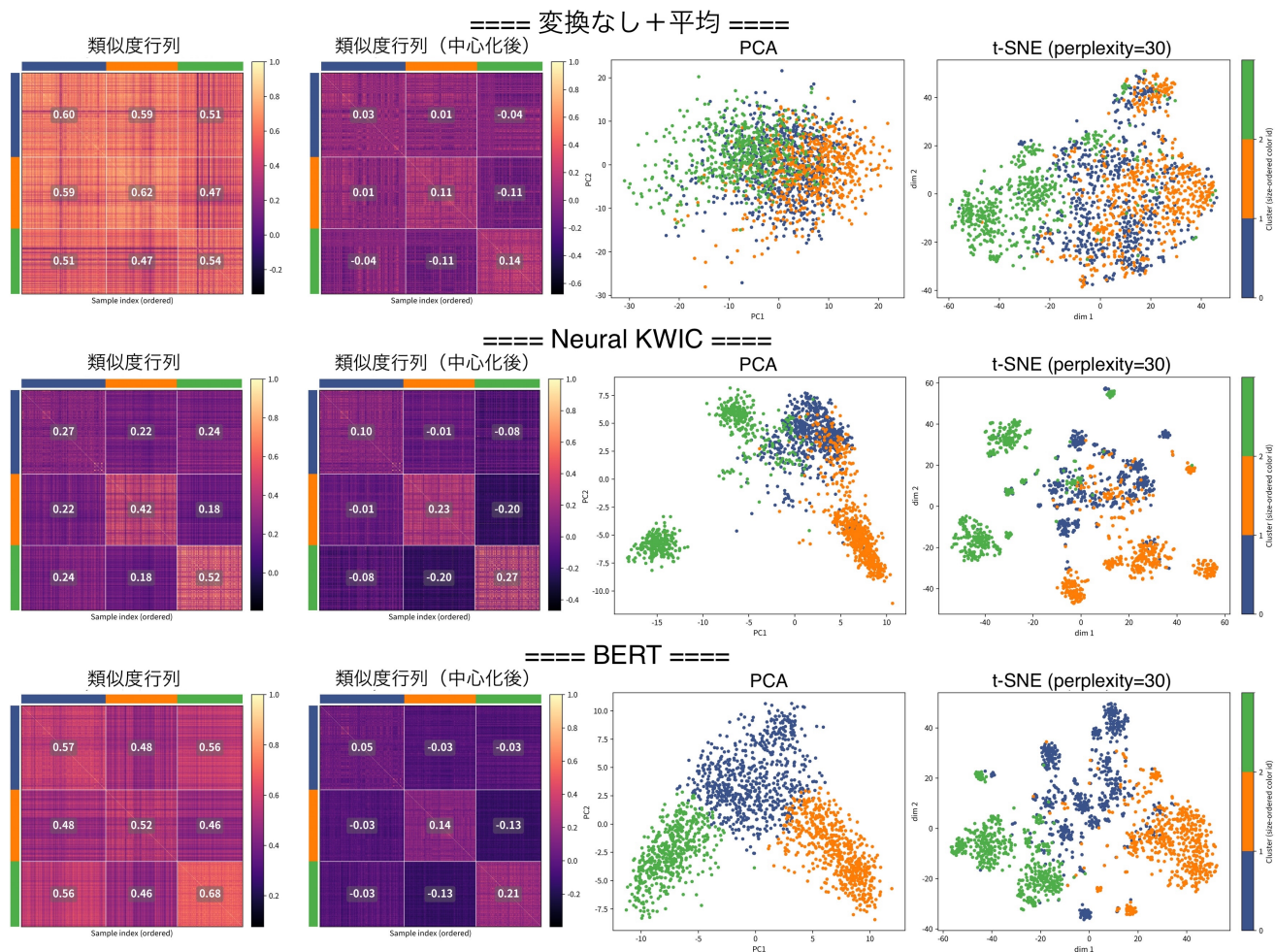


図 3: 「消化」に対するそれぞれの手法の結果：類似度行列と PCA・t-SNE による埋め込み可視化。

ていることが確認できる。変換なし+平均やその他の *à la carte* 系手法では、文脈単語の埋め込みを平均することで文脈表現を構成しているため、文脈内の語順や中心語との相対的な位置関係といった構造的情報は明示的には保持されない。これに対して提案手法では、KWIC に基づいて定義された文脈単語の埋め込みを出現順に連結することで、局所的な文脈構造を入力表現に直接反映している。このような設計の違いにより、提案手法は KWIC 用例間における文脈情報の差異をより詳細に捉え、表現することが可能であると考えられる。

**計算時間** 表 4 より、提案手法の推論時間が一貫して非常に短いことが確認された。特に、対象単語の中でも用例数が最も多い「問題」(N=70,087) を対象とし、全用例に対する動的単語埋め込みの獲得に要する実行時間を測定したところ、全用例の埋め込み計算は 1 秒未満で完了しており、提案手法が極めて高速に推論可能であることが確認された。以上より、計算効率の観点において、提案手法は大規模な文脈表現モデルと比較して実用性の高い手法であることが示された。

## 5.2 変換関数学習の ablation

本節では、表 4 の結果をもとに、前後の文脈単語数、変換関数の学習に用いるデータセットのフィルタリング条件、および教師ベクトルに対する前処理の有無を個別に変更し、それぞれの構成要素が果たす役割について考察する。

**文脈単語数** 文脈単語数を  $k \in \{5, 10, 20\}$  と変化させて ablation 実験を行った結果、提案手法においては、文脈単語数の増加が一貫した性能向上に必ずしも結びつかないことが確認された。特に、文脈単語数を 20 とした場合には、10 の場合と比べて性能が低下する設定も多く見られた。一般に、KWIC 形式において文脈単語数を増やすことで、中心語周辺から得られる文脈情報量が増加し、用例間の差異を表現するための情報が豊富になると考えられる。しかし、本研究では変換関数として線形変換を採用しているため、文脈情報量を増加させた場合であっても、その情報を十分に表現しきれず、結果としてモデル性能が低下した可能性がある。さらに、学習時には教師信号の取得のために文単位で事前学習済み言語モデルへ入力しているのに対し、推論時に用いる KWIC 文脈は固定長で抽出されるため、文境界を跨ぐ文脈が含まれる場合がある。文脈単語



表 5: 「消化」の語義区分 ([19] より引用)。

語義 ID	用法	意味
1	名・サ変動詞	生体が体内で食物を吸収しやすい形に変化させること。また、その過程。「消化がいい」「消化に悪い」「消化不良」
2	名・サ変動詞	物事を十分に理解して、自分の知識とすること。「外来文化を消化する」
3	名・サ変動詞	残さずに、始末、処理すること。「日程を消化する」「在庫を消化する」

数を増加させると、このような文境界を跨ぐ文脈の割合も高まると考えられ、学習時の入力分布との差異が拡大することで性能に影響を与えた可能性もある。

**学習データセットのフィルタリング** 変換関数の学習に用いるデータセットに対して、中心語を内容語のみに限定する品詞フィルタリングを行った場合、提案手法の性能が向上する傾向が確認された。本研究では対象単語として内容語のみを扱っているため、学習時においても中心語を内容語に限定することで、学習データと評価対象との分布の整合性が高まり、変換関数がより安定した対応関係を学習できた可能性が考えられる。

**教師ベクトルの前処理** 教師ベクトルとして用いる BERT の単語埋め込みに対し、Zipf 白色化を適用する場合と適用しない場合を比較した結果、提案手法においては、白色化を適用した場合に性能が向上することが確認された。これは、教師側の埋め込み空間に内在する頻度依存的な異質性が緩和されることで、変換関数の学習がより安定したものとなったためであると考えられる。このような性質を持つ埋め込み空間では、特に同一見出し語を持つ用例間の差分が相対的に捉えにくくなる可能性がある。

### 5.3 品詞ごとの定性分析

本節では、提案手法により獲得された動的単語埋め込みが、品詞の異なる対象単語に対してどのようなクラスタ構造を形成するのかを、可視化結果に基づいて定性的に分析する。<sup>\*9</sup>

### 5.3.1 名詞・サ変動詞

名詞・サ変動詞として用いられる語の分析として、対象単語のうち「消化」を分析対象とする。

類似度行列の算出と低次元空間への可視化 「消化」に対して各手法を用いて動的単語埋め込みを獲得し、類似度行列の算出および低次元空間への可視化を行った結果を図 3 に示す。

\*9 本章では扱わなかった対象単語に関する分析については、付録 A.5 に示す。

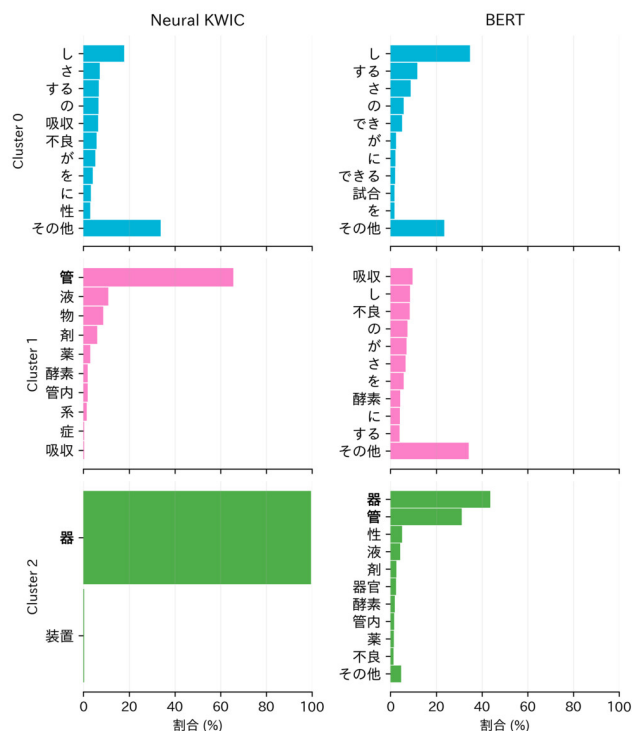


図 4: 「消化」の直後トークンの出現頻度。

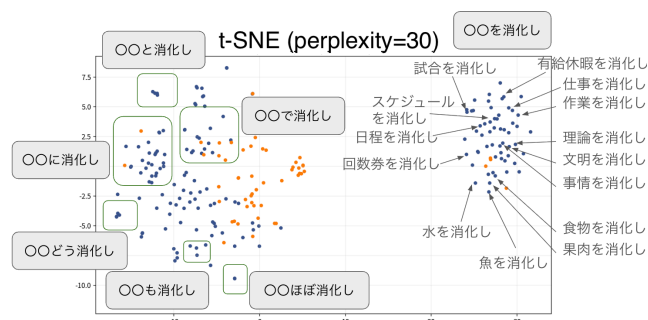


図 5: 対象単語「消化」について、「消化し」として出現する KWIC 用例のみを抽出し、拡大して t-SNE を行った可視化結果。

まず、類似度行列に着目する。比較手法である変換なし+平均埋め込みでは、クラスタ内とクラスタ間の平均類似度に大きな差が見られず、語義ごとのまとまりが明確には現れていない。一方、Neural KWIC 埋め込みおよび BERT に基づく動的単語埋め込みでは、クラスタ内の平均類似度が相対的に高く、変換関数を用いることで語義ごとのまとまりがより明確に表現されている。

次に、PCA および *t*-SNE による低次元空間に射影した可視化結果を比較する。提案手法と BERT を比較すると、BERT では同一色のクラスタが比較的単一のまとまりとして分布しているのに対し、提案手法では BERT に基づく同一色のクラスタがさらに細分化された構造を示している。例えば、PCA において緑色で示されたクラスタは、BERT では単一のまとまりとして分布している一方、提案手法では二つのクラスタに分離している。このうち一方は



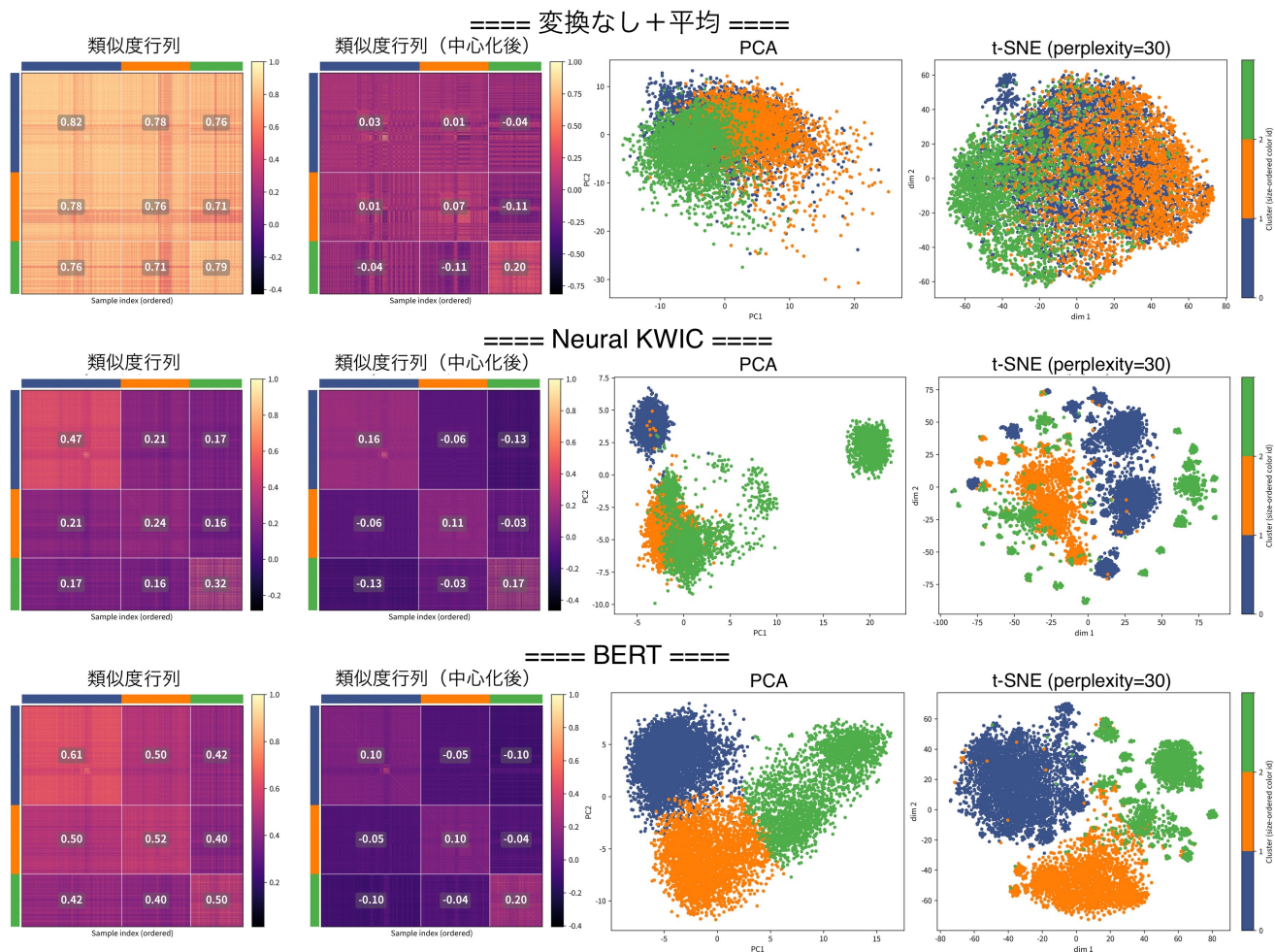


図 6: 「絶対」に対するそれぞれの手法の結果：類似度行列と PCA・t-SNE による埋め込み可視化。

青色やオレンジ色の他のクラスと明確に分離しているのに対し、もう一方は低次元空間上で比較的近接した位置に分布している。同一色のクラスがさらに細分化される傾向は、PCA よりも t-SNE による可視化において特に顕著であり、他のクラスにおいても同様の分離構造が観察される。

**クラスごとの直後トークンの分析** 以下では、提案手法において観察された、クラス内で生じている細分化構造が、どのような文脈情報に基づいて形成されているかを分析する。図 3 において各点に対応する KWIC 用例を観察すると、このような分離は対象単語の直後に出現するトークンに基づく局所的な文脈構造の違いを反映していると考えられる。この点を確認するため、図 4 に、各クラスに属する KWIC 用例について、「消化」の直後に出現するトークン<sup>\*10</sup>を集計し、出現頻度の高い順に上位 10 件を示す。また、表 5 に「消化」の語義区分を示す。

まず、提案手法における Cluster 0 に着目する。Cluster 0 には「消化し」として出現する用例が多く含まれており、「消化」が動詞として用いられている用例が主に属してい

る。このクラスでは、直後トークンの分布は一定の偏りを持ちつつも比較的分散しており、BERT においても、同様に動詞用法に対応する用例が同一クラスにまとまっており、直後トークンの分布についても概ね一致した傾向が確認できる。

次に、名詞用法に対応するクラスに着目する。提案手法では、Cluster 1 に「消化管」、Cluster 2 に「消化器」がそれぞれ頻出しており、いずれも名詞用法に対応する用例がクラスごとにまとまっている。一方、BERT では、これらの用例は同一クラスにまとめられており、表 5 に示した語義定義に照らすと、いずれも語義 ID 1 に対応していることが分かる。これに対して提案手法では、「消化管」および「消化器」といった、辞書上では同一の語義として定義されている用例が、異なるクラスに分割されている点が特徴的である。この違いは、提案手法がより局所的な文脈情報を強く反映していることを示唆している。実際に、Cluster 1 および Cluster 2 における直後トークンの分布を比較すると、BERT では分布が比較的均等であるのに対し、提案手法では特定の直後トークンが突出して出現する傾向が確認できる。特に Cluster 2 では、次のトーク

<sup>\*10</sup> トークナイズは NWJC2vec および NWJC-BERT に合わせ、MeCab-0.996 と UniDic-2.1.2 を用いて行った。

表 6: 「絶対」の語義区分 ([19] より引用)。

語義 ID	用法	意味
1	名・形容動詞	対立、比較するものがなく、それ自体としてあること。他から何も制限を受けないこと。 ⇨相対。「神は絶対的存在」「絶対の真理」「父の命令は絶対だ」
2-1	副詞	(単独、または「に」を伴って副詞的に) どういう場合でも必ず。百パーセント。「絶対勝つ」「絶対に必要だ」
2-2	副詞	(単独、または「に」を伴って副詞的に) (あとに打消の語を伴って) 決して。とうてい。「絶対許されない」「絶対に負けない」

ンが「器」である用例が約 99% を占めており、直後トークンの表層の特徴が局所的な文脈構造としてクラスタ形成に強く寄与していることが分かる。

**低次元のクラスタ構造の分析** 続いて、提案手法において「消化し」として出現しているサ変動詞用例のみを抽出し、細分化構造がより顕著であった *t*-SNE に基づいて、低次元空間におけるクラスタ構造を詳細に観察する。可視化結果を図 5 に示す。なお、本可視化では、入力に用いた KWIC をさらに短く抜粋した形で、各プロットに対応する KWIC の具体例の一部を併せて示している。

可視化結果から、「消化し」として出現する KWIC 用例は、さらに大きく二つのクラスタ構造に分離していることが確認できる。対応する KWIC を参照すると、右側のクラスタでは「○○を消化し」のように、対象単語の直前トークンとして助詞「を」を伴う構文が多く見られる一方、左側のクラスタでは、「を」以外の直前トークンを伴う用例がまとまって分布している。

さらに右側のクラスタを詳しく観察すると、語義 ID 1 に対応する「食物を消化し」「果物を消化し」、語義 ID 2 に対応する「理論を消化し」「文明を消化し」、語義 ID 3 に対応する「日程を消化し」「回数券を消化し」といった用例が、低次元空間上で近接して分布していることが確認できる。すなわち、直前トークンや直後トークンによってさらに細分化された各クラスタの内部では、表 5 に示した辞書に基づく語義区分に沿った連続的なまとまりが形成されている。

5.3.2 副詞・形容動詞

次に、副詞・形容動詞として用いられる語の分析として、対象単語のうち「絶対」を分析対象とする。

**類似度行列の算出と低次元空間への可視化** 「絶対」に対して各手法を用いて動的単語埋め込みを獲得し、類似度

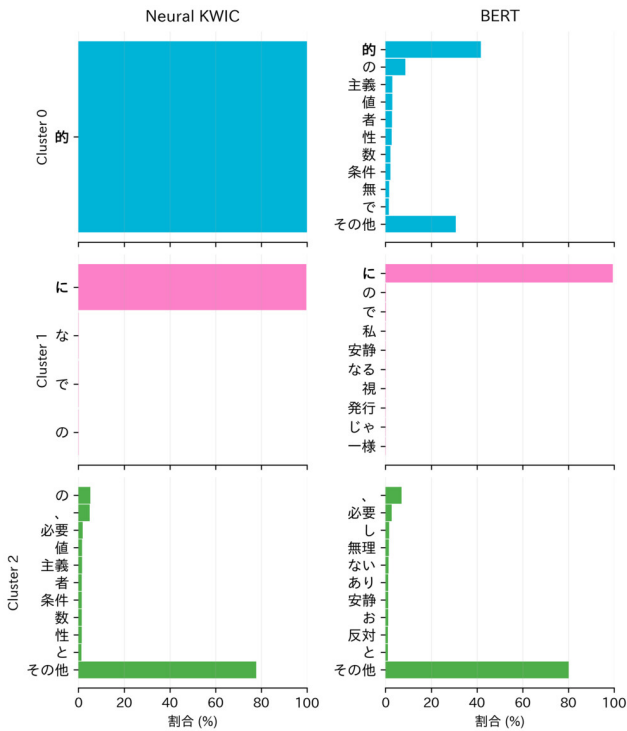


図 7: 「絶対」の直後トークンの出現頻度。

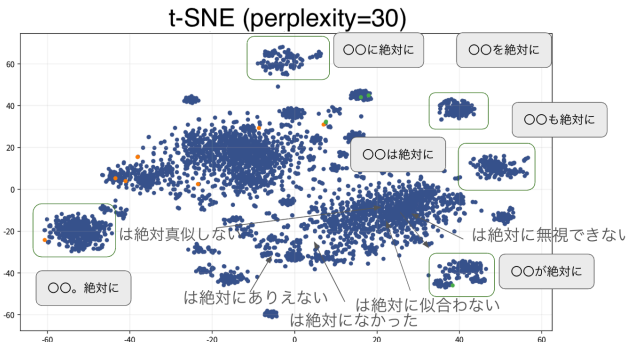


図 8: 対象単語「絶対」について、「絶対に」として出現する KWIC 用例のみを抽出し、拡大して *t*-SNE を行った可視化結果。

行列の算出および低次元空間への可視化を行った結果を図 6 に示す。類似度行列に関しては、名詞・サ変動詞の場合と同様の傾向が確認され、Neural KWIC 埋め込みおよび BERT では語義ごとのまとまりがより明確に表現されている。次に、PCA および *t*-SNE による低次元空間に射影した可視化結果を比較する。Neural KWIC 埋め込みと BERT を比較すると、例えば PCA において緑色で示されたクラスタは、BERT では比較的単一のまとまりとして分布しているのに対し、提案手法では、さらに細分化されていることが分かる。

**クラスタごとの直後トークンの分析** 同様に、このようなクラスタ構造がどのように形成されているかを詳しく分析するため、提案手法において、各クラスタに属する KWIC 用例について、直後トークンの出現頻度を集計した結果を図 7

に示す。また、表 6 に「絶対」の語義区分を示す。まず、提案手法における各クラスタ内の直後トークンに注目すると、「絶対」においても、最も頻度の高い直後トークンが突出して出現する傾向が確認できる。一方、NWJC-BERT の結果においても同様の傾向が観察され、特に Cluster 1 では、他のトークンも出現するものの、「に」が約 99% を占めている。

この結果は、表 6 に示すように、副詞用法の「絶対」が多くの場合「に」を伴う形で出現することと整合的である。すなわち、「絶対」の直後トークンが「に」である用例を同一クラスタにまとめることで、副詞用法と名詞・形容動詞用法とを分離しやすくなる。その結果として、副詞か否かという品詞レベルの区別においては、提案手法と NWJC-BERT のクラスタリング結果の一致度が他の対象単語と比較して高くなった可能性が示唆される。

前節で述べたように、提案手法はまず直後トークンに基づくクラスタ分割が生じ、KWIC 形式で与えられる局所的な文脈よりもさらに局所的な文脈情報から強い影響を受けて単語埋め込みを構成する傾向がある。この性質は「絶対」において、品詞の分離に寄与し、結果として NWJC-BERT の性能に近づく要因になっていると考えられる。

**低次元のクラスタ構造の分析** 続いて、提案手法において「絶対に」として出現している副詞用法のみを抽出し、t-SNE に基づいて低次元空間におけるクラスタ構造を詳細に観察する。可視化結果を図 8 に示す。

可視化結果から、「絶対に」として出現する KWIC 用例は、名詞・サ変動詞の場合と同様に、対象単語の直前トークンに基づいてさらに細かくクラスタに細分化されることが分かる。例えば、「○○に絶対に」や「○○は絶対に」といった直前トークンに依存したまとまりが確認できる。名詞やサ変動詞の場合には、直後トークンおよび直前トークンによって細分化されたクラスタの内部において、辞書に基づく語義定義に沿ったまとまりが観察されていた。一方で、「絶対に」の副詞用法では、例えば「○○は絶対に」という構文としてまとまっているクラスタを詳しく観察すると、語義 ID 2-2 に対応する打消表現を伴う用例が、特定のまとまりを形成せず、クラスタ内に散在していることが分かる。副詞用法の「絶対」は、表 6 に示すように、打消表現を伴わない用法（語義 2-1）と、打消表現を伴う用法（語義 2-2）に区別される。しかし、この区別は否定表現の有無といった文全体の意味構造に強く依存するため、KWIC 形式で取得される局所的な文脈情報のみから判別することは難しい。特に、文脈単語数が少ない設定においては、人手で判別する場合であっても、語義の判断が困難な用例が多く含まれる。

BERT については、先行研究において肯定文と否定文の表現が高い類似度を示しやすいことが報告されている [27]。本分析においても、副詞用法の「絶対」に関して、肯定用

表 7: 「甘い」の語義区分（[19] より引用）。

語義 ID	用法	意味
1	形容詞	砂糖や蜜など糖分の味がある。「甘い汁粉」「甘い桃」
2	形容詞	塩気が薄い。塩辛くない。⇔辛い。「甘い塩鮭」
3	形容詞	口当たりが穏やかで、刺激が少ない。酒などにいう。⇔辛い。
4	形容詞	心がとけるように、うっとりときい。「甘いにおい」「甘い言葉」
5	形容詞	物事に対して厳しさが無い。厳格でない。⇔辛い。「評価が甘い」「世間を甘く見る」
6	形容詞	物事がしっかりしていない状態にある。「ピントが甘い」「ねじが甘い」
7	形容詞	刃物の切れ味が悪い。「切れ味が甘い」

法と否定用法が同一クラスタにまとめられる傾向が確認されており、この挙動は先行研究の知見と整合的である。また、本研究では NWJC-BERT の単語埋め込みを教師ベクトルとして学習しているため、提案手法においても同様の混同が生じ得ると考えられる。

以上より、「絶対」に関しては、直後トークンといった極めて局所的な文脈情報が品詞（副詞／名詞・形容動詞）の分離には有効に働く一方で、否定表現の有無に基づく副詞用法内部の語義区別については限界があることが示唆された。

5.3.3 形容詞

次に、形容詞として用いられる語の分析として、対象単語のうち「甘い」を分析対象とする。

**類似度行列の算出と低次元空間への可視化** 「甘い」に対して各手法を用いて動的単語埋め込みを獲得し、類似度行列の算出および低次元空間への可視化を行った結果を図 9 に示す。類似度行列に関しては、名詞・サ変動詞や副詞・形容動詞の場合と同様の傾向が確認され、Neural KWIC 埋め込みおよび BERT では語義ごとのまとまりが比較的明確に表現されている。次に、PCA および t-SNE による低次元空間に射影した可視化結果を比較する。「甘い」の可視化結果では、名詞・サ変動詞や副詞・形容動詞の場合と比べて、クラスタ間の境界が不明瞭になっており、低次元空間上では明確に分離したクラスタ構造というよりも、連続的な分布として可視化される傾向が観察される。

**クラスタごとの直後トークンの分析** このような傾向の要因の一つとして、「甘い」が形容詞であることに起因する直後トークンの多様性が考えられる。この点を詳しく分析するため、提案手法において各クラスタに属する KWIC 用例について、直後トークンの出現頻度を集計した結果を



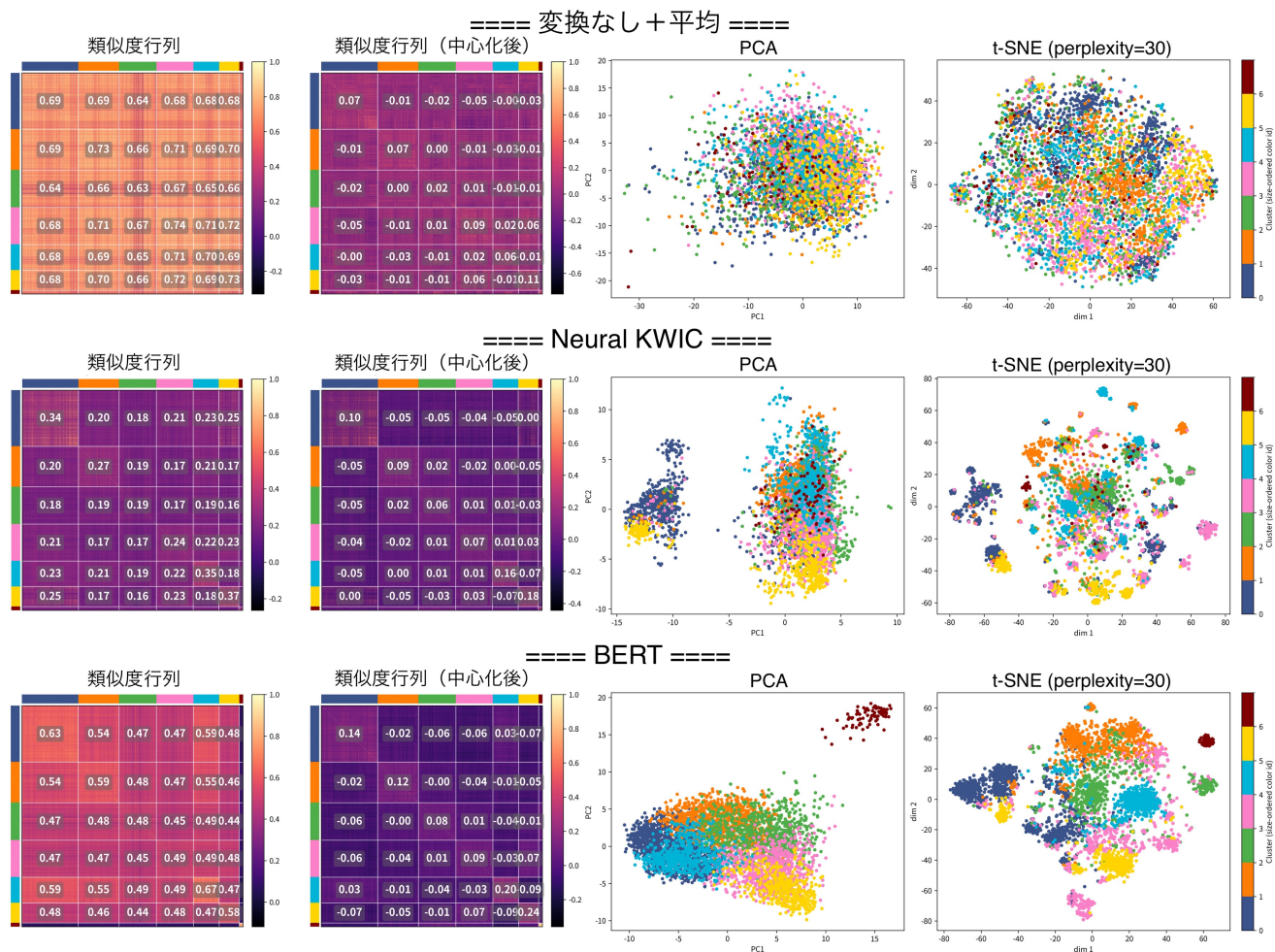


図 9: 「甘い」に対するそれぞれの手法の結果：類似度行列と PCA・t-SNE による埋め込み可視化。

図 10 に示す。また、表 7 に「甘い」の語義区分を示す。

提案手法におけるクラスタ構造を詳細に観察すると、特定の直後トークンに強く依存したクラスタも一部に確認できる。例えば、Cluster 5 では「甘い汁」として出現する用例のみからクラスタが構成されており、また Cluster 6 では「甘さ」として出現する用例がほぼ占めている。これらのクラスタでは、直後トークンが限定されているため、局所的には明確なまとまりが形成されている。一方で、その他のクラスタに注目すると、名詞・サ変動詞や副詞・形容動詞の分析で見られたような、最も頻度の高い直後トークンが突出して出現する傾向は確認できない。このことから、「甘い」の場合には、直後トークンに基づく大まかなクラスタ構成が生じにくいことが分かる。名詞やサ変動詞、副詞の場合には、対象単語の直後に出現するトークンが比較的限定されるのに対し、形容詞である「甘い」は、名詞を修飾する用法や名詞化された用法など、後続する要素の種類が多様である。そのため、直後トークンに基づく粗いクラスタ分割が生じにくく、結果として低次元空間上ではクラスタ境界が不明瞭な、連続的な分布として可視化されたと考えられる。

**低次元のクラスタ構造の分析** 最後に、提案手法において「甘さ」として出現している名詞用例のみを抽出し、t-SNE に基づいて低次元空間におけるクラスタ構造を詳細に観察する。可視化結果を図 11 に示す。

可視化結果から、「甘さ」として出現する KWIC 用例は、名詞・サ変動詞の場合と同様に、対象単語の直前トークンに基づいてさらに細かくクラスタに細分化されることが分かる。例えば、「○○な甘さ」や「○○の甘さ」といった直前トークンに依存したまとまりが確認できる。さらに、「○○の甘さ」としてまとまっている右下のクラスタを詳しく観察すると、語義 ID 1 に対応する「砂糖の甘さ」「カaramelの甘さ」、語義 ID 4 に対応する「声の甘さ」「春の甘さ」、語義 ID 5 に対応する「詰めの甘さ」「認識の甘さ」といった用例が、低次元空間上で近接して分布していることが確認できる。

以上より、形容詞である「甘い」においては、形容詞に由来する直後トークンの多様性によって全体としてクラスタ境界が不明瞭になりやすいものの、名詞・サ変動詞の場合と同様に、直前トークンや直後トークンによって各クラスタが細分化され、その内部では辞書に基づく語義区分に

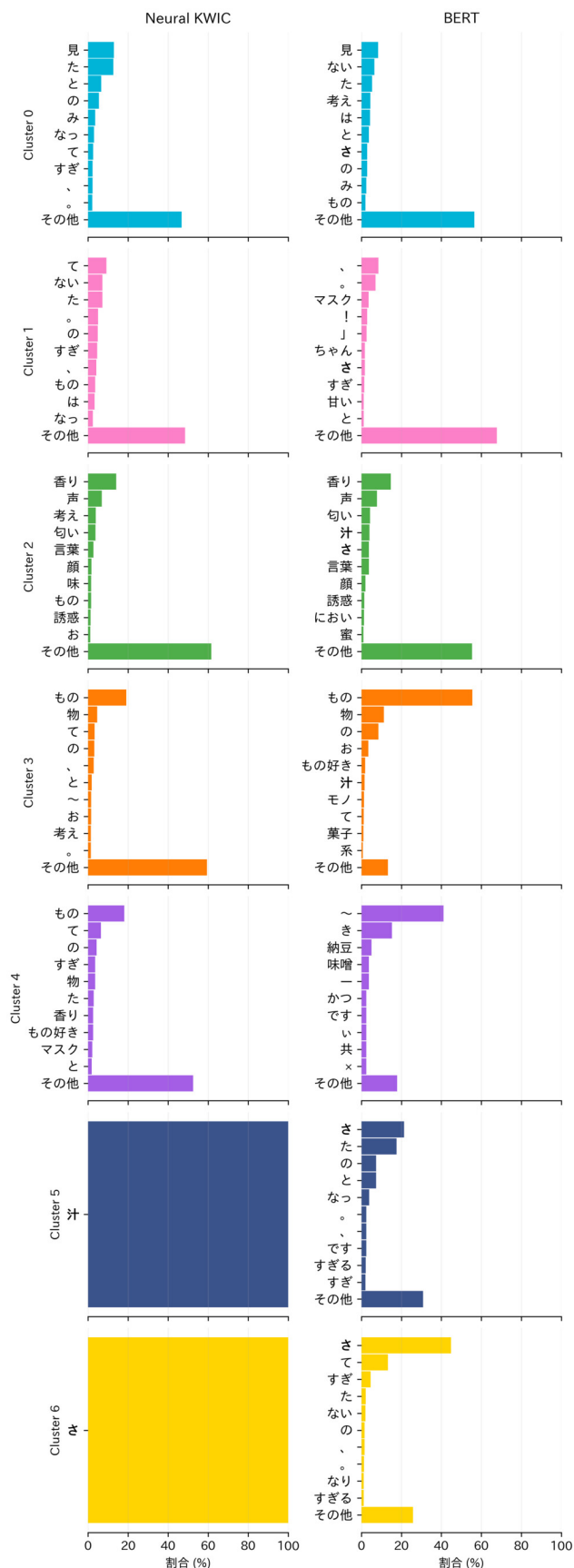


図 10: 「甘い」の直後トークンの頻度。

沿った連続的なまとまりが形成されていることが分かる。

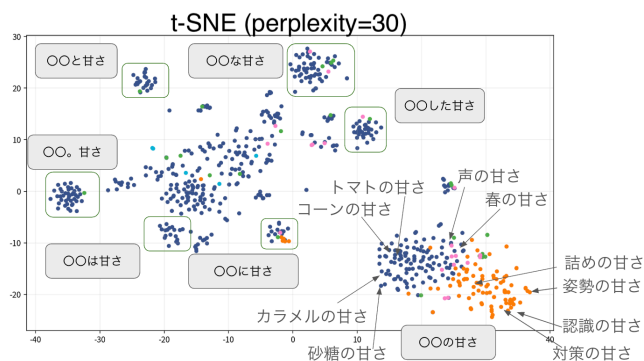


図 11: 対象単語「甘い」について、「甘さ」として出現する KWIC 用例のみを抽出し、拡大して *t*-SNE を行った可視化結果。

#### 5.3.4 クラスタ構造の傾向

各単語のクラスタ構造の分析を踏まえると、提案手法により獲得された動的単語埋め込みでは、低次元空間上で可視化を行った際に、一定の階層性を持ったクラスタ構造が形成される傾向が観察される。具体的には、以下のような段階的な構造が確認できる。

- 対象単語の直後トークンに基づく大まかなクラスタ分割 (図 4、図 7、図 10)
- 対象単語の直前トークンに基づくクラスタのさらなる細分化 (図 5、図 8、図 11)
- 各クラスタ内部における、辞書に基づいて定義された語義区分に沿った連続的なまとまり (図 5、図 8、図 11)

本手法によって獲得された動的単語埋め込みを低次元空間上で可視化すると、まず対象単語の直後トークンに基づく粗いクラスタ分割が生じ、その後、直前トークンに基づいてさらに細分化が行われることが分かる。そして、これらの細分化されたクラスタの内部では、辞書に基づいて定義された語義区分に対応する用例が、連続的な分布としてまとまっている様子が観察される。

このように、提案手法は明示的に語義分類を行うのではなく、KWIC 形式で与えられる局所的な文脈情報に基づいて階層的な構造を形成し、その内部に語義に対応する連続的な意味構造を内在的に表現している点に特徴がある。

## 6. おわりに

本研究では、a la carte 埋め込みを拡張し、KWIC 形式から動的単語埋め込みを獲得可能な手法である Neural KWIC を提案した。定量評価の結果、提案手法は既存手法と比較して BERT に近い表現能力を持ちつつ、BERT と比べて計算速度に優れていることを示した。また、定性的分析により、提案手法は対象単語の直後および直前トークンに基づいてクラスタを形成し、その内部で辞書の語義区分に対応する階層的なまとまりを示す傾向があること、さらにその挙動が品詞ごとに異なることを明らかにした。Neural



KWIC を用いて動的単語埋め込みを獲得し、低次元空間へ可視化を行うことで、大量の KWIC 用例に対しても高い解釈性が得られることを確認した。本研究を出発点として、大量の KWIC 用例を埋め込み表現に基づいて分析することにより、語義分析や用法の違いの解明をはじめとした、さまざまな言語学的研究が今後さらに発展することを期待する。

## 本研究の限界

本研究の限界として、日本語以外の言語での実験と対象単語の選定、教師ベクトル、変換関数、評価の5つについて述べる。

**日本語以外の言語での実験** 本研究で用いるデータセットは日本語のデータセットに限っているため、英語や中国語といった他の言語において Neural KWIC が有効かどうかは検証する必要がある。データセットのサイズについても言語ごとに異なることが想定されるため、複数言語で Neural KWIC の有効性を示す必要がある。

**対象単語の選定** 対象単語の選定については、言語学者が提示した11単語のみで分析を行っているが、これらの単語での評価で十分かどうかは明らかではない。一般的な単語について Neural KWIC を適応した場合にどのような分析が可能であり、どのような点に限界があるのかを明らかにする必要がある。

**教師ベクトル** 本研究では、事前学習済み言語モデルとして BERT を採用したが、教師ベクトルの選択が可視化結果に与える影響については十分に検討できていない。例えば、SimCSE のようにコントラスト学習により表現空間が整形された動的単語埋め込みを教師信号として用いることで、より明確なクラスタ構造が得られる可能性がある。今後は、複数の事前学習済み言語モデルを教師信号として比較し、その特性が Neural KWIC の表現構造に与える影響を体系的に分析する必要がある。

**変換関数** 本研究では変換関数として、線形変換のみを分析している。線形変換では線形の関係のみしか表現できず、非線形な変換での分析が必要である。例えば、パラメータ数の少ない NN で学習を行うなど、非線形かつ軽量な知識蒸留を行う関数と線形変換との比較を行うことが考えられる。

**評価** 本研究では評価手法として、クラスタラベルの一致度と CKA、類似度の相関係数を用いている。BERT の埋め込みとの比較であるため、1 に近ければ良い指標ではあるが、これらの指標の改善幅が実応用上どの程度の意味を持つかは明らかではない。人手評価を行い自動評価がどの程度良い評価指標であるかは調査する必要がある。

**謝辞** 本研究の一部は、JST さきがけ研究『文理融合による人と社会の変革基盤技術の共創』における研究課題

『意思決定のための自然言語処理による未来予測』（グラント番号: JPMJPR2366）および科研費基盤研究(A)『「明治150年雑誌コーパス」の整備と公開による近現代日本語の歴史的研究』（グラント番号: 25H00470）の支援を受けた。

## 参考文献

- [1] Hans Peter Luhn. Key word-in-context index for technical literature (kwic index). *American documentation*, Vol. 11, No. 4, pp. 288–295, 1960.
- [2] John Sinclair. *Corpus, Concordance, Collocation*. Oxford University Press, Oxford, 1991.
- [3] 前川喜久雄(編). コーパス入門, 講座日本語コーパス, 第1巻. 朝倉書店, 2013.
- [4] 中俣尚己. 「中納言」を活用したコーパス日本語研究入門. ひつじ書房, 2021.
- [5] Jiaju Du, Fanchao Qi, and Maosong Sun. Using BERT for word sense disambiguation. *arXiv preprint arXiv:1909.08358*, 2019.
- [6] Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3509–3514, Hong Kong, China, November 2019. Association for Computational Linguistics.
- [7] Boon Peng Yap, Andrew Koh, and Eng Siong Chng. Adapting BERT for word sense disambiguation with gloss selection objective and example sentences. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 41–46, Online, November 2020. Association for Computational Linguistics.
- [8] Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1384–1397, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.
- [9] Jader Martins Camboim de Sá, Marcos Da Silveira, and Cédric Pruski. Survey in characterization of semantic change. *arXiv preprint arXiv:2402.19088*, 2024.
- [10] 中村壮範, 小木曾智信. Web 版コーパス検索アプリケーション「中納言」の公開. 言語処理学会第17回年次大会発表論文集, March 2011.
- [11] Mikhail Khodak, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon Stewart, and Sanjeev Arora. A la carte embedding: Cheap but effective induction of semantic feature vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12–22, Melbourne, Australia, July 2018. Association for Computational Linguistics.
- [12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [13] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Ji-aya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF conference on computer*

- vision and pattern recognition, pp. 5008–5017, 2021.
- [14] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International journal of computer vision*, Vol. 129, No. 6, pp. 1789–1819, 2021.
- [15] Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 44, No. 6, pp. 3048–3068, 2021.
- [16] Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*, 2024.
- [17] Chuanpeng Yang, Yao Zhu, Wang Lu, Yidong Wang, Qian Chen, Chenlong Gao, Bingjie Yan, and Yiqiang Chen. Survey on knowledge distillation for large language models: Methods, evaluation, and application. *ACM Trans. Intell. Syst. Technol.*, Vol. 16, No. 6, November 2025.
- [18] Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den. Balanced corpus of contemporary written Japanese. *Lang. Resour. Eval.*, Vol. 48, No. 2, p. 345–371, June 2014.
- [19] 林巨樹, 松井栄一, 小学館辞典編集部編集. 現代国語例解辞典. 小学館, 第 5 版, 2016.
- [20] 新納浩幸, 浅原正幸, 古宮嘉那子, 佐々木稔. nwjc2vec: 国語研日本語ウェブコーパスから構築した単語の分散表現データ. 自然言語処理, Vol. 24, No. 5, pp. 705–720, 2017.
- [21] Masayuki Asahara, Kikuo Maekawa, Mizuho Imada, Sachi Kato, and Hikari Konishi. Archiving and analysing techniques of the ultra-large-scale web-based corpus project of NINJAL, Japan. *Alexandria*, Vol. 25, No. 1-2, pp. 129–148, 2014.
- [22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [23] 浅原正幸, 西内沙恵, 加藤祥. NWJC-BERT: 多義語に対するヒトと文脈化単語埋め込みの類似性判断の対照分析. 言語処理学会第 26 回年次大会発表論文集, pp. pp.961–964. 言語処理学会, 2020.
- [24] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*, 2017.
- [25] Sho Yokoi, Han Bao, Hiroto Kurita, and Hidetoshi Shimodaira. Zipfian whitening. In *Advances in Neural Information Processing Systems*, Vol. 37, pp. 122259–122291. Curran Associates, Inc., 2024.
- [26] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, Vol. 13, No. 1, pp. 795–828, 2012.
- [27] Allyson Ettinger. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, Vol. 8, pp. 34–48, 2020.

## 付 録

### A.1 対象単語ごとの語義区分

各対象単語ごとの語義区分について示す。表 A-1 には単語「生理」の語義区分を、表 A-2 には単語「姿勢」の語義区分を、表 A-3 には単語「問題」の語義区分を、表 A-4 には単語「養生」の語義区分を、表 A-5 には単語「運動」の語義区分を、表 A-6 には単語「適当」の語義区分を、表 A-7 には単語「薄い」の語義区分を、表 A-8 には単語「高い」の語義区分を示す。

表 A-1: 「生理」の語義区分（[19] より引用）。

語義 ID	用法	意味
1	名	生物が生命を維持していく上での種々の現象や機能。また、その原理。「生理現象」
2	名	月経。「生理日」「生理痛」

表 A-2: 「姿勢」の語義区分（[19] より引用）。

語義 ID	用法	意味
1	名	体の構え方。かっこう。「姿勢がいい」「姿勢を正す〔＝物事に対する態度や心構えを改める〕」
2	名	心の持ち方。態度。「前向き姿勢」「低姿勢で対処する」

表 A-3: 「問題」の語義区分（[19] より引用）。

語義 ID	用法	意味
1	名	答えを求めるための問い。「入試問題」「問題集」
2	名	批判や論争または研究の対象となる事柄。また、心にとめたり、取り上げたりすべき事柄。「失言が問題となる」「値段の高い安いは問題ではない」「問題点」「問題提起」「問題化する」
3	名	世間で注目を集めている事柄。話題。「これが問題の人物です」「最近の問題作」
4	名	面倒な事柄。厄介な事件。「問題を起こす」「大問題が持ち上がる」
5	名	ある物事にかかわる事柄。「それは良心の問題だ」「時間の問題」「程度問題」

表 A-4: 「養生」の語義区分 ([19] より引用)。

語義 ID	用法	意味
1	名・サ変	健康を維持し、その増進に努めること。摂生。「日ごろから養生に努める」
2	名・サ変	病気を治して、体をもとどおりにすること。保養。「山の温泉場で養生する」
3	名・サ変	建築などで、汚損しないように、作業するものを布や紙などで覆って保護すること。「エレベーターに養生して荷を運び入れる」「養生シート」

表 A-5: 「運動」の語義区分 ([19] より引用)。

語義 ID	用法	意味
1	名・サ変	物体が時間の経過と共にその位置を変えること。⇔静止。「太陽の公転運動」
2	名・サ変	健康保持などの目的で、体を動かすこと。「軽い運動を日課とする」「運動不足」
3	名・サ変	ある目的を達するために、いろいろな方面に働きかけること。「戦争反対のための運動を起こす」「運動費」

表 A-6: 「適当」の語義区分 ([19] より引用)。

語義 ID	用法	意味
1	名・形動	ある状態、目的、要求などに程よく当てはまること。「適当な大きさにきざむ」
2	名・形動	いいかげんなこと。大ざっぱで投げやりなさま。「面倒なので適当にあしらっておいた」

## A.2 対象単語ごとの定量評価

各対象単語ごとの定量評価の結果を示す。表 A-9 には単語「生理」の定量評価を、表 A-10 には単語「姿勢」の定量評価を、表 A-11 には単語「問題」の定量評価を、表 A-12 には単語「養生」の定量評価を、表 A-13 には単語「消化」の定量評価を、表 A-14 には単語「運動」の定量評価を、表 A-15 には単語「適当」の定量評価を、表 A-16 には単語「絶対」の定量評価を、表 A-17 には単語「甘い」の定量評価を、表 A-18 には単語「薄い」の定量評価を、表 A-19 には単語「高い」の定量評価を示す。

表 A-7: 「薄い」の語義区分 ([19] より引用)。

語義 ID	用法	意味
1	形	物の厚みが少ない。⇔厚い。「氷が薄い」「薄い本」
2	形	物の密度や濃度などが少ない。⇔濃い。「味が薄い」「影が薄い」「薄い色」
3	形	心、考え、経験などが深くない。浅い。「知識が薄い」
4	形	豊かでない。乏しい。十分でない。「利が薄い商い」「上に厚く下に薄い待遇」「選手の層が薄い」
5	形	程度が強くない。弱い。軽い。かすかである。⇔濃い。「人情が薄い」「縁が薄い」「印象が薄い」「なじみが薄い」

表 A-8: 「高い」の語義区分 ([19] より引用)。

語義 ID	用法	意味
1	形	空間的に上のほうにある。下から上までのへだたりが大きい。⇔低い。「日が高い」「高い山」「背丈が高い」「天高く馬肥ゆる秋」
2	形	音や声が大きい。また、音声の振動が多い。高音である。⇔低い。「高い笑い声」「ピッコロの高い音」
3	形	広く知れわたっている。「評判が高い」「悪名が高い」
4	形	身分や地位が上位にある。⇔低い。「高い地位につく」
5	形	品位、理想、力量、機能などがすぐれている。⇔低い。「格調が高い」「志が高い」「見識が高い」「生産性が高い」「彼を高く買っている」
6	形	高慢である。多く「お高い」の形で用いる。→お高い。
7	形	金のかかるさまである。高価である。⇔安い。「高い代金を払う」
8	形	計測器の示す目盛りが上のほうにあって、度合いが強いほうである。⇔低い。「室温が高い」「血圧が高い」。

表 A.9: 「生理」に対して手法や学習条件を変えた時のモデル性能と計算効率の結果。

手法	学習条件			モデル性能				計算効率
	文脈単語数	品詞 filtering	Zipf 白色化	ARI	CKAraw	CKAnorm	Spearman's $\rho$	全用例の 推論時間 [s]
BERT (ベースライン)	5	—	—	1.000	1.000	1.000	1.000	131.224
	10	—	—	1.000	1.000	1.000	1.000	142.635
	20	—	—	1.000	1.000	1.000	1.000	149.304
変換なし+平均	5	—	—	0.664	0.560	0.570	0.560	—
	10	—	—	0.540	0.472	0.494	0.501	—
	20	—	—	0.389	0.445	0.473	0.511	—
à la carte+平均	5	—	—	0.587	0.556	0.573	0.516	0.002
		—	✓	0.516	0.614	0.630	0.580	0.002
		機能語	—	0.605	0.587	0.597	0.563	0.002
		機能語	✓	0.587	0.635	0.648	0.614	0.002
	10	—	—	0.576	0.519	0.537	0.522	0.003
		—	✓	0.578	0.556	0.583	0.560	0.003
		機能語	—	0.592	0.536	0.549	0.522	0.004
		機能語	✓	0.625	0.572	0.594	0.576	0.004
	20	—	—	0.584	0.530	0.550	0.562	0.005
		—	✓	0.691	0.546	0.586	0.556	0.005
		機能語	—	0.595	0.536	0.553	0.559	0.005
		機能語	✓	0.637	0.559	0.588	0.575	0.002
à la carte+SIF	5	—	—	0.545	0.546	0.571	0.509	0.004
		—	✓	0.548	0.630	0.649	0.587	0.004
		機能語	—	0.594	0.584	0.594	0.551	0.001
		機能語	✓	0.587	0.633	0.645	0.599	0.004
	10	—	—	0.616	0.513	0.535	0.533	0.002
		—	✓	0.641	0.579	0.599	0.594	0.006
		機能語	—	0.611	0.549	0.558	0.557	0.005
		機能語	✓	0.692	0.587	0.600	0.604	0.005
	20	—	—	0.629	0.523	0.547	0.547	0.004
		—	✓	0.684	0.577	0.600	0.557	0.003
		機能語	—	0.666	0.561	0.571	0.564	0.003
		機能語	✓	<b>0.710</b>	0.586	0.602	0.580	0.006
à la carte+concat	5	—	—	0.550	0.672	0.691	0.641	0.005
		—	✓	0.550	0.677	0.766	0.670	0.008
		機能語	—	0.565	0.731	0.739	0.697	0.006
		機能語	✓	0.550	<b>0.735</b>	<b>0.769</b>	<b>0.705</b>	0.006
	10	—	—	0.492	0.677	0.695	0.650	0.011
		—	✓	0.492	0.683	0.762	0.675	0.010
		機能語	—	0.503	0.729	0.739	0.679	0.010
		機能語	✓	0.492	<b>0.735</b>	0.765	0.703	0.010
	20	—	—	0.202	0.670	0.685	0.641	0.017
		—	✓	0.202	0.678	0.745	0.688	0.017
		機能語	—	0.462	0.721	0.729	0.690	0.019
		機能語	✓	0.113	0.724	0.748	0.704	0.019

### A.3 用例間類似度の相関係数 (Spearman's $\rho$ ) の詳細分析

本論文では、定量的に評価するための指標の一つとして用例間類似度の相関係数 (Spearman's  $\rho$ ) を用いたが、本節では、評価指標として用いた用例間類似度の相関係数に対してより詳細な分析を行う。具体的には、得られた 1,000 個の類似度を散布図にプロットし、BERT に基づく類似度と提案手法に基づく類似度の関係性を可視化した。各散布図では、最小二乗法により回帰直線を重ねて描画している。

各対象単語ごとの用例間類似度の相関係数を示す。図 A.1 には単語「生理」の散布図を、図 A.2 には単語「姿勢」

の散布図を、図 A.3 には単語「問題」の散布図を、図 A.4 には単語「養生」の散布図を、図 A.5 には単語「消化」の散布図を、図 A.6 には単語「運動」の散布図を、図 A.7 には単語「適当」の散布図を、図 A.8 には単語「絶対」の散布図を、図 A.9 には単語「甘い」の散布図を、図 A.10 には単語「薄い」の散布図を、図 A.11 には単語「高い」の散布図を示す。

対象単語ごとの傾向を見ると、「生理」(図 A.1) において特に回帰直線の傾きが大きく、BERT と比較して類似度分布の変化度合いが大きいことが分かる。一方で、「運動」(図 A.6) では回帰直線の傾きが比較的緩やかであり、BERT 側の類似度の方がより広い分布を持つ傾向が見ら

表 A-10: 「姿勢」に対して手法や学習条件を変えた時のモデル性能と計算効率の結果。

手法	学習条件			モデル性能				計算効率
	文脈単語数	品詞 filtering	Zipf 白色化	ARI	CKAraw	CKAnorm	Spearman's $\rho$	全用例の 推論時間 [s]
BERT (ベースライン)	5	—	—	1.000	1.000	1.000	1.000	256.089
	10	—	—	1.000	1.000	1.000	1.000	287.848
	20	—	—	1.000	1.000	1.000	1.000	320.808
変換なし+平均	5	—	—	0.006	0.301	0.320	0.326	—
	10	—	—	0.021	0.301	0.326	0.296	—
	20	—	—	0.034	0.324	0.355	0.376	—
à la carte+平均	5	—	—	0.159	0.364	0.374	0.398	0.002
		—	✓	0.109	0.397	0.409	0.399	0.002
		機能語	—	0.191	0.382	0.389	0.435	0.002
		機能語	✓	0.150	0.401	0.413	0.421	0.004
	10	—	—	0.239	0.405	0.415	0.488	0.003
		—	✓	0.354	0.425	0.428	0.488	0.004
		機能語	—	0.263	0.416	0.421	0.393	0.002
		機能語	✓	0.238	0.432	0.438	0.417	0.002
	20	—	—	0.309	0.456	0.465	0.473	0.004
		—	✓	0.644	0.470	0.472	0.449	0.004
		機能語	—	0.361	0.463	0.468	0.480	0.004
		機能語	✓	0.499	0.482	0.488	0.470	0.003
à la carte+SIF	5	—	—	0.175	0.345	0.369	0.417	0.003
		—	✓	0.217	0.415	0.430	0.414	0.004
		機能語	—	0.223	0.372	0.383	0.431	0.002
		機能語	✓	0.229	0.404	0.418	0.428	0.004
	10	—	—	0.245	0.387	0.407	0.484	0.002
		—	✓	0.362	0.446	0.452	0.443	0.002
		機能語	—	0.264	0.424	0.431	0.418	0.002
		機能語	✓	0.302	0.448	0.455	0.450	0.004
	20	—	—	0.306	0.419	0.444	0.457	0.002
		—	✓	<b>0.665</b>	0.475	0.480	0.471	0.003
		機能語	—	0.346	0.467	0.473	0.482	0.003
		機能語	✓	0.461	0.491	0.498	0.493	0.003
à la carte+concat	5	—	—	0.006	0.484	0.513	0.514	0.010
		—	✓	-0.001	0.498	0.527	0.461	0.010
		機能語	—	0.135	0.605	0.616	0.592	0.010
		機能語	✓	0.003	0.569	0.579	0.527	0.010
	10	—	—	0.013	0.507	0.534	0.490	0.017
		—	✓	-0.004	0.510	0.536	0.412	0.018
		機能語	—	0.389	0.630	0.640	0.579	0.018
		機能語	✓	0.012	0.581	0.590	0.486	0.018
	20	—	—	0.050	0.519	0.546	0.550	0.035
		—	✓	-0.005	0.516	0.541	0.502	0.035
		機能語	—	0.466	<b>0.640</b>	<b>0.650</b>	<b>0.637</b>	0.032
		機能語	✓	0.006	0.584	0.593	0.570	0.032

れた。また、「養生」(図 A-4)、「適当」(図 A-7)、「絶対」(図 A-8)、では、回帰直線が  $y = x$  に近い形状を示しており、両手法間で類似度構造が比較的よく一致していることが確認できる。特に傾きが大きい「生理」について対応する可視化結果(図 A-12)を詳しく観察すると、PCA 空間において BERT 埋め込みでは2つのクラスタが比較的近接しているのに対し、提案手法では、オレンジ色で示されたクラスタが複数の小さなクラスタに分離している様子が確認される。これは、語義1(表 A-1)に対応する用例群の内部で、さらに細かな分化が生じていることを示唆しており、その結果として用例間類似度が多様化していると考えられる。

## A.4 各対象単語に対する提案手法の可視化結果

対象単語ごとに、提案手法を用いて獲得した KWIC 形式の動的単語埋め込みを、低次元空間へ射影した可視化結果を示す。図 A-12 には単語「生理」の埋め込みの可視化を、図 A-13 には単語「姿勢」の埋め込みの可視化を、図 A-14 には単語「問題」の埋め込みの可視化を、図 A-15 には単語「養生」の埋め込みの可視化を、図 A-16 には単語「運動」の埋め込みの可視化を、図 A-17 には単語「適当」の埋め込みの可視化を、図 A-18 には単語「薄い」の埋め込みの可視化を、図 A-19 には単語「高い」の埋め込みの可視化を示す。



表 A-11: 「問題」に対して手法や学習条件を変えた時のモデル性能と計算効率の結果。

手法	学習条件			モデル性能				計算効率
	文脈単語数	品詞 filtering	Zipf 白色化	ARI	CKAraw	CKAnorm	Spearman's $\rho$	全用例の 推論時間 [s]
BERT (ベースライン)	5	—	—	1.000	1.000	1.000	1.000	3011.859
	10	—	—	1.000	1.000	1.000	1.000	3346.791
	20	—	—	1.000	1.000	1.000	1.000	3792.313
変換なし+平均	5	—	—	0.058	0.315	0.328	0.310	—
	10	—	—	0.034	0.264	0.279	0.273	—
	20	—	—	0.041	0.241	0.257	0.225	—
à la carte+平均	5	—	—	0.056	0.344	0.357	0.299	0.019
		—	✓	0.058	0.377	0.386	0.319	0.018
		機能語	—	0.052	0.358	0.366	0.316	0.013
		機能語	✓	0.065	0.391	0.400	0.355	0.013
	10	—	—	0.038	0.320	0.329	0.297	0.018
		—	✓	0.034	0.334	0.338	0.306	0.018
		機能語	—	0.033	0.325	0.330	0.318	0.018
		機能語	✓	0.042	0.342	0.348	0.303	0.022
	20	—	—	0.053	0.316	0.322	0.344	0.023
		—	✓	0.049	0.318	0.321	0.332	0.024
		機能語	—	0.048	0.316	0.321	0.340	0.022
		機能語	✓	0.050	0.325	0.331	0.344	0.022
à la carte+SIF	5	—	—	0.051	0.287	0.310	0.300	0.021
		—	✓	0.065	0.349	0.361	0.318	0.021
		機能語	—	0.049	0.309	0.319	0.293	0.020
		機能語	✓	0.060	0.345	0.357	0.327	0.018
	10	—	—	0.034	0.278	0.293	0.255	0.022
		—	✓	0.040	0.317	0.322	0.268	0.022
		機能語	—	0.036	0.302	0.307	0.290	0.023
		機能語	✓	0.041	0.318	0.325	0.281	0.021
	20	—	—	0.050	0.276	0.291	0.311	0.021
		—	✓	0.043	0.304	0.308	0.322	0.029
		機能語	—	0.038	0.304	0.309	0.333	0.022
		機能語	✓	0.044	0.312	0.318	0.329	0.042
à la carte+concat	5	—	—	0.216	0.518	0.550	0.449	0.082
		—	✓	0.134	0.603	0.624	0.468	0.248
		機能語	—	0.234	0.617	0.641	0.551	0.153
		機能語	✓	0.263	<b>0.664</b>	<b>0.668</b>	0.545	0.207
	10	—	—	0.177	0.519	0.549	0.469	0.256
		—	✓	0.247	0.595	0.614	0.496	0.256
		機能語	—	0.203	0.622	0.642	0.559	0.252
		機能語	✓	<b>0.351</b>	0.656	0.660	0.553	0.206
	20	—	—	0.225	0.521	0.549	0.466	0.457
		—	✓	0.143	0.594	0.613	0.502	0.421
		機能語	—	0.287	0.622	0.640	<b>0.593</b>	0.379
		機能語	✓	0.315	0.652	0.656	0.579	0.409

## A.5 各対象単語の直後トークンの出現頻度

提案手法により獲得した単語埋め込みに基づいて K-means クラスタリングを行い、各クラスにおいて対象単語の直後トークンの出現頻度上位 10 件を集計した結果を示す。図 A-20 には単語「生理」の直後トークンの出現頻度を、図 A-21 には単語「姿勢」の直後トークンの出現頻度を、図 A-22 には単語「問題」の直後トークンの出現頻度を、図 A-23 には単語「養生」の直後トークンの出現頻度を、図 A-24 には単語「運動」の直後トークンの出現頻度を、図 A-25 には単語「適当」の直後トークンの出現頻度を、図 A-26 には単語「薄い」の直後トークンの出現頻度を、図 A-27 には単語「高い」の直後トークンの出現頻度を示す。

度を、図 A-27 には単語「高い」の直後トークンの出現頻度を示す。

表 A.12: 「養生」に対して手法や学習条件を変えた時のモデル性能と計算効率の結果。

手法	学習条件			モデル性能				計算効率
	文脈単語数	品詞 filtering	Zipf 白色化	ARI	CKAraw	CKAnorm	Spearman's $\rho$	全用例の 推論時間 [s]
BERT (ベースライン)	5	—	—	1.000	1.000	1.000	1.000	14.609
	10	—	—	1.000	1.000	1.000	1.000	16.304
	20	—	—	1.000	1.000	1.000	1.000	18.990
変換なし+平均	5	—	—	0.079	0.371	0.380	0.347	—
	10	—	—	0.034	0.324	0.332	0.247	—
	20	—	—	0.109	0.323	0.330	0.325	—
à la carte+平均	5	—	—	0.259	0.443	0.473	0.385	0.001
		—	✓	0.263	0.559	0.564	0.463	0.001
		機能語	—	0.266	0.498	0.508	0.415	0.001
		機能語	✓	0.258	0.546	0.552	0.452	0.001
	10	—	—	0.059	0.444	0.475	0.359	0.001
		—	✓	0.194	0.518	0.539	0.414	0.001
		機能語	—	0.089	0.475	0.489	0.380	0.002
		機能語	✓	0.113	0.503	0.519	0.390	0.001
	20	—	—	0.153	0.456	0.471	0.372	0.001
		—	✓	0.312	0.493	0.510	0.405	0.002
		機能語	—	0.214	0.460	0.468	0.380	0.002
		機能語	✓	0.183	0.481	0.492	0.400	0.002
à la carte+SIF	5	—	—	0.264	0.448	0.484	0.384	0.002
		—	✓	0.264	0.613	0.612	0.524	0.002
		機能語	—	0.273	0.520	0.524	0.439	0.002
		機能語	✓	0.257	0.580	0.577	0.498	0.002
	10	—	—	0.144	0.444	0.472	0.367	0.002
		—	✓	0.192	0.555	0.574	0.453	0.002
		機能語	—	0.159	0.500	0.509	0.401	0.001
		機能語	✓	0.143	0.533	0.544	0.430	0.001
	20	—	—	0.171	0.420	0.441	0.340	0.002
		—	✓	0.217	0.504	0.513	0.407	0.002
		機能語	—	0.177	0.468	0.472	0.383	0.001
		機能語	✓	0.174	0.492	0.496	0.411	0.001
à la carte+concat	5	—	—	0.111	0.580	0.608	0.497	0.003
		—	✓	0.197	0.665	0.679	0.582	0.003
		機能語	—	0.117	0.649	0.659	0.553	0.004
		機能語	✓	<b>0.329</b>	<b>0.691</b>	<b>0.698</b>	<b>0.588</b>	0.002
	10	—	—	0.064	0.558	0.588	0.411	0.008
		—	✓	0.255	0.637	0.652	0.474	0.006
		機能語	—	0.226	0.620	0.638	0.474	0.006
		機能語	✓	0.195	0.658	0.671	0.494	0.006
	20	—	—	0.175	0.499	0.518	0.388	0.005
		—	✓	0.064	0.549	0.566	0.388	0.007
		機能語	—	0.145	0.542	0.559	0.410	0.007
		機能語	✓	0.161	0.560	0.578	0.404	0.006

表 A.13: 「消化」に対して手法や学習条件を変えた時のモデル性能と計算効率の結果。

手法	学習条件			モデル性能				計算効率
	文脈単語数	品詞 filtering	Zipf 白色化	ARI	CKAraw	CKAnorm	Spearman's $\rho$	全用例の 推論時間 [s]
BERT (ベースライン)	5	—	—	1.000	1.000	1.000	1.000	79.379
	10	—	—	1.000	1.000	1.000	1.000	88.785
	20	—	—	1.000	1.000	1.000	1.000	97.978
変換なし+平均	5	—	—	0.155	0.453	0.480	0.475	—
	10	—	—	0.116	0.362	0.390	0.358	—
	20	—	—	0.103	0.340	0.368	0.346	—
à la carte+平均	5	—	—	0.173	0.462	0.484	0.506	0.001
		—	✓	0.373	0.607	0.615	0.613	0.001
		機能語	—	0.272	0.522	0.532	0.551	0.001
		機能語	✓	0.347	0.603	0.616	0.620	0.001
	10	—	—	0.140	0.453	0.478	0.434	0.002
		—	✓	0.311	0.564	0.585	0.545	0.001
		機能語	—	0.188	0.489	0.501	0.458	0.001
		機能語	✓	0.280	0.552	0.574	0.530	0.001
	20	—	—	0.163	0.472	0.495	0.458	0.002
		—	✓	0.313	0.550	0.584	0.553	0.001
		機能語	—	0.242	0.493	0.507	0.475	0.001
		機能語	✓	0.319	0.542	0.572	0.534	0.001
à la carte+SIF	5	—	—	0.244	0.437	0.473	0.517	0.001
		—	✓	0.277	0.621	0.633	0.634	0.001
		機能語	—	0.285	0.501	0.514	0.546	0.001
		機能語	✓	0.379	0.595	0.607	0.620	0.001
	10	—	—	0.158	0.419	0.461	0.407	0.001
		—	✓	0.357	0.595	0.608	0.558	0.001
		機能語	—	0.195	0.493	0.507	0.477	0.002
		機能語	✓	0.364	0.571	0.586	0.535	0.002
	20	—	—	0.166	0.417	0.460	0.422	0.002
		—	✓	0.394	0.573	0.589	0.538	0.002
		機能語	—	0.212	0.500	0.511	0.478	0.003
		機能語	✓	0.396	0.562	0.578	0.524	0.003
à la carte+concat	5	—	—	0.578	0.649	0.668	0.650	0.004
		—	✓	0.351	0.734	0.547	0.714	0.007
		機能語	—	<b>0.623</b>	0.670	0.697	0.649	0.005
		機能語	✓	0.320	<b>0.740</b>	<b>0.767</b>	<b>0.722</b>	0.005
	10	—	—	0.467	0.654	0.674	0.590	0.009
		—	✓	0.436	0.731	0.760	0.691	0.008
		機能語	—	0.251	0.674	0.701	0.623	0.008
		機能語	✓	0.454	0.734	0.766	0.692	0.008
	20	—	—	0.454	0.657	0.679	0.620	0.013
		—	✓	0.212	0.724	0.753	0.703	0.015
		機能語	—	0.275	0.670	0.697	0.636	0.015
		機能語	✓	0.317	0.722	0.756	0.692	0.015

表 A-14: 「運動」に対する手法や学習条件を変えた時のモデル性能と計算効率の結果。

手法	学習条件			モデル性能				計算効率
	文脈単語数	品詞 filtering	Zipf 白色化	ARI	CKAraw	CKAnorm	Spearman's $\rho$	全用例の 推論時間 [s]
BERT (ベースライン)	5	—	—	1.000	1.000	1.000	1.000	751.301
	10	—	—	1.000	1.000	1.000	1.000	841.229
	20	—	—	1.000	1.000	1.000	1.000	1004.254
変換なし+平均	5	—	—	0.051	0.297	0.323	0.287	—
	10	—	—	0.056	0.270	0.299	0.236	—
	20	—	—	0.062	0.260	0.290	0.326	—
à la carte+平均	5	—	—	0.093	0.349	0.370	0.336	0.004
		—	✓	0.177	0.410	0.419	0.390	0.006
		機能語	—	0.106	0.379	0.390	0.377	0.005
		機能語	✓	0.150	0.412	0.420	0.395	0.004
	10	—	—	0.133	0.388	0.416	0.381	0.004
		—	✓	0.348	0.440	0.456	0.400	0.007
		機能語	—	0.151	0.413	0.427	0.405	0.004
		機能語	✓	0.225	0.436	0.449	0.399	0.006
	20	—	—	0.246	0.431	0.462	0.469	0.006
		—	✓	0.317	0.478	0.502	0.495	0.004
		機能語	—	0.262	0.451	0.468	0.466	0.004
		機能語	✓	<b>0.379</b>	0.470	0.488	0.490	0.005
à la carte+SIF	5	—	—	0.127	0.344	0.375	0.373	0.005
		—	✓	0.329	0.455	0.471	0.393	0.005
		機能語	—	0.156	0.385	0.398	0.392	0.007
		機能語	✓	0.214	0.431	0.446	0.385	0.008
	10	—	—	0.206	0.374	0.410	0.376	0.005
		—	✓	0.338	0.473	0.488	0.446	0.006
		機能語	—	0.219	0.424	0.437	0.414	0.004
		機能語	✓	0.280	0.452	0.466	0.430	0.004
	20	—	—	0.308	0.392	0.440	0.434	0.004
		—	✓	0.369	0.491	0.510	0.482	0.004
		機能語	—	0.264	0.457	0.471	0.455	0.004
		機能語	✓	0.364	0.477	0.494	0.475	0.004
à la carte+concat	5	—	—	0.137	0.501	0.533	0.438	0.023
		—	✓	0.047	0.553	0.584	0.494	0.022
		機能語	—	0.335	0.588	0.603	0.542	0.022
		機能語	✓	0.319	0.612	0.618	0.548	0.022
	10	—	—	0.220	0.506	0.541	0.481	0.041
		—	✓	0.072	0.557	0.590	0.521	0.043
		機能語	—	0.308	0.598	0.614	0.548	0.043
		機能語	✓	0.325	<b>0.613</b>	<b>0.624</b>	0.567	0.043
	20	—	—	0.136	0.504	0.541	0.489	0.081
		—	✓	0.217	0.555	0.589	0.552	0.081
		機能語	—	0.349	0.595	0.613	0.543	0.077
		機能語	✓	0.348	0.606	0.619	<b>0.583</b>	0.077

表 A-15: 「適当」に対して手法や学習条件を変えた時のモデル性能と計算効率の結果。

手法	学習条件			モデル性能				計算効率
	文脈単語数	品詞 filtering	Zipf 白色化	ARI	CKAraw	CKAnorm	Spearman's $\rho$	全用例の 推論時間 [s]
BERT (ベースライン)	5	—	—	1.000	1.000	1.000	1.000	155.819
	10	—	—	1.000	1.000	1.000	1.000	174.492
	20	—	—	1.000	1.000	1.000	1.000	199.617
変換なし+平均	5	—	—	0.000	0.385	0.399	0.350	—
	10	—	—	0.118	0.355	0.378	0.395	—
	20	—	—	0.193	0.387	0.410	0.428	—
à la carte+平均	5	—	—	0.046	0.366	0.388	0.329	0.002
		—	✓	-0.001	0.453	0.466	0.349	0.002
		機能語	—	0.128	0.410	0.421	0.347	0.002
		機能語	✓	0.042	0.472	0.474	0.355	0.003
	10	—	—	0.231	0.407	0.425	0.415	0.002
		—	✓	0.258	0.424	0.433	0.406	0.004
		機能語	—	0.251	0.431	0.438	0.421	0.004
		機能語	✓	0.249	0.445	0.378	0.441	0.002
	20	—	—	0.291	0.461	0.477	0.493	0.004
		—	✓	-0.037	0.445	0.450	0.470	0.003
		機能語	—	0.297	0.472	0.479	0.505	0.005
		機能語	✓	0.314	0.470	0.475	0.487	0.003
à la carte+SIF	5	—	—	0.042	0.343	0.376	0.314	0.002
		—	✓	0.234	0.416	0.409	0.322	0.002
		機能語	—	0.210	0.395	0.405	0.332	0.002
		機能語	✓	0.179	0.413	0.402	0.312	0.002
	10	—	—	0.216	0.388	0.418	0.407	0.003
		—	✓	-0.042	0.421	0.420	0.394	0.003
		機能語	—	0.280	0.438	0.445	0.439	0.002
		機能語	✓	0.281	0.430	0.430	0.421	0.002
	20	—	—	0.251	0.422	0.463	0.482	0.004
		—	✓	0.332	0.445	0.445	0.479	0.002
		機能語	—	0.324	0.482	0.490	0.525	0.005
		機能語	✓	0.326	0.467	0.470	0.500	0.002
à la carte+concat	5	—	—	0.044	0.591	0.608	0.489	0.007
		—	✓	0.042	0.620	0.642	0.474	0.007
		機能語	—	0.042	0.661	0.672	0.566	0.007
		機能語	✓	0.083	0.631	0.656	0.490	0.007
	10	—	—	0.040	0.617	0.637	0.599	0.012
		—	✓	0.082	0.622	0.657	0.535	0.011
		機能語	—	<b>0.773</b>	0.686	0.701	0.658	0.011
		機能語	✓	0.700	0.636	0.672	0.557	0.011
	20	—	—	0.048	0.631	0.654	0.608	0.023
		—	✓	0.047	0.623	0.667	0.537	0.023
		機能語	—	0.115	<b>0.696</b>	<b>0.715</b>	<b>0.678</b>	0.023
		機能語	✓	0.048	0.639	0.682	0.563	0.023



表 A-16: 「絶対」に対して手法や学習条件を変えた時のモデル性能と計算効率の結果。

手法	学習条件			モデル性能				計算効率
	文脈単語数	品詞 filtering	Zipf 白色化	ARI	CKAraw	CKAnorm	Spearman's $\rho$	全用例の 推論時間 [s]
BERT (ベースライン)	5	—	—	1.000	1.000	1.000	1.000	488.236
	10	—	—	1.000	1.000	1.000	1.000	542.890
	20	—	—	1.000	1.000	1.000	1.000	592.819
変換なし+平均	5	—	—	0.261	0.440	0.444	0.448	—
	10	—	—	0.197	0.378	0.391	0.317	—
	20	—	—	0.173	0.351	0.367	0.289	—
à la carte+平均	5	—	—	0.196	0.408	0.438	0.463	0.003
		—	✓	0.216	0.497	0.485	0.482	0.003
		機能語	—	0.199	0.437	0.447	0.464	0.003
		機能語	✓	0.154	0.526	0.510	0.499	0.003
	10	—	—	0.176	0.393	0.418	0.373	0.003
		—	✓	0.268	0.436	0.430	0.383	0.004
		機能語	—	0.179	0.406	0.416	0.367	0.004
		機能語	✓	0.235	0.457	0.453	0.381	0.005
	20	—	—	0.168	0.392	0.410	0.327	0.003
		—	✓	0.215	0.409	0.403	0.340	0.003
		機能語	—	0.177	0.394	0.401	0.313	0.003
		機能語	✓	0.201	0.425	0.424	0.348	0.003
à la carte+SIF	5	—	—	0.159	0.351	0.392	0.400	0.004
		—	✓	0.171	0.466	0.444	0.425	0.003
		機能語	—	0.155	0.395	0.405	0.426	0.003
		機能語	✓	0.222	0.459	0.442	0.424	0.003
	10	—	—	0.151	0.341	0.379	0.320	0.003
		—	✓	0.185	0.424	0.410	0.350	0.003
		機能語	—	0.156	0.394	0.402	0.340	0.004
		機能語	✓	0.186	0.425	0.419	0.336	0.006
	20	—	—	0.150	0.328	0.371	0.295	0.006
		—	✓	0.160	0.396	0.389	0.333	0.003
		機能語	—	0.153	0.395	0.401	0.333	0.006
		機能語	✓	0.172	0.410	0.409	0.349	0.006
à la carte+concat	5	—	—	0.266	0.611	0.660	0.623	0.016
		—	✓	0.760	0.673	0.724	0.698	0.016
		機能語	—	0.119	0.664	0.689	0.656	0.016
		機能語	✓	<b>0.761</b>	0.701	0.731	0.705	0.016
	10	—	—	0.258	0.615	0.663	0.608	0.032
		—	✓	0.755	0.675	0.738	0.719	0.029
		機能語	—	0.120	0.670	0.695	0.648	0.029
		機能語	✓	0.755	0.705	0.744	<b>0.721</b>	0.029
	20	—	—	0.287	0.618	0.662	0.588	0.057
		—	✓	0.746	0.677	0.747	0.675	0.057
		機能語	—	0.135	0.674	0.696	0.603	0.054
		機能語	✓	0.746	<b>0.708</b>	<b>0.750</b>	0.661	0.054

表 A-17: 「甘い」に対して手法や学習条件を変えた時のモデル性能と計算効率の結果。

手法	学習条件			モデル性能				計算効率
	文脈単語数	品詞 filtering	Zipf 白色化	ARI	CKAraw	CKAnorm	Spearman's $\rho$	全用例の 推論時間 [s]
BERT (ベースライン)	5	—	—	1.000	1.000	1.000	1.000	209.829
	10	—	—	1.000	1.000	1.000	1.000	231.542
	20	—	—	1.000	1.000	1.000	1.000	248.388
変換なし+平均	5	—	—	0.110	0.265	0.280	0.278	—
	10	—	—	0.054	0.237	0.257	0.294	—
	20	—	—	0.065	0.225	0.251	0.225	—
à la carte+平均	5	—	—	0.126	0.275	0.306	0.283	0.002
		—	✓	0.156	0.373	0.379	0.369	0.002
		機能語	—	0.126	0.306	0.319	0.326	0.002
		機能語	✓	0.159	0.360	0.363	0.363	0.002
	10	—	—	0.102	0.295	0.332	0.369	0.002
		—	✓	0.130	0.393	0.401	0.460	0.003
		機能語	—	0.104	0.330	0.347	0.399	0.002
		機能語	✓	0.143	0.380	0.387	0.442	0.003
	20	—	—	0.145	0.319	0.357	0.348	0.002
		—	✓	0.173	0.412	0.421	0.414	0.003
		機能語	—	0.146	0.354	0.372	0.365	0.003
		機能語	✓	0.159	0.403	0.410	0.406	0.003
à la carte+SIF	5	—	—	0.081	0.241	0.293	0.301	0.002
		—	✓	0.188	0.384	0.399	0.421	0.002
		機能語	—	0.131	0.306	0.327	0.367	0.003
		機能語	✓	0.130	0.349	0.361	0.399	0.002
	10	—	—	0.111	0.249	0.312	0.354	0.002
		—	✓	0.125	0.401	0.415	0.478	0.002
		機能語	—	0.139	0.340	0.360	0.432	0.002
		機能語	✓	0.131	0.370	0.386	0.455	0.003
	20	—	—	0.132	0.229	0.318	0.328	0.003
		—	✓	0.172	0.397	0.419	0.420	0.003
		機能語	—	0.169	0.360	0.381	0.386	0.003
		機能語	✓	0.178	0.383	0.407	0.410	0.003
à la carte+concat	5	—	—	0.202	0.372	0.411	0.360	0.008
		—	✓	0.149	0.454	0.489	0.424	0.008
		機能語	—	0.200	0.441	0.469	0.431	0.008
		機能語	✓	0.221	0.489	0.509	0.461	0.008
	10	—	—	0.133	0.390	0.429	0.426	0.015
		—	✓	0.133	0.471	0.508	0.492	0.015
		機能語	—	0.147	0.462	0.488	0.488	0.015
		機能語	✓	<b>0.250</b>	<b>0.505</b>	<b>0.526</b>	<b>0.502</b>	0.015
	20	—	—	0.088	0.394	0.433	0.372	0.027
		—	✓	0.109	0.472	0.510	0.440	0.027
		機能語	—	0.153	0.466	0.492	0.425	0.026
		機能語	✓	0.179	<b>0.505</b>	<b>0.526</b>	0.457	0.026

表 A-18: 「薄い」に対して手法や学習条件を変えた時のモデル性能と計算効率の結果。

手法	学習条件			モデル性能				計算効率
	文脈単語数	品詞 filtering	Zipf 白色化	ARI	CKAraw	CKAnorm	Spearman's $\rho$	全用例の 推論時間 [s]
BERT (ベースライン)	5	—	—	1.000	1.000	1.000	1.000	224.431
	10	—	—	1.000	1.000	1.000	1.000	249.182
	20	—	—	1.000	1.000	1.000	1.000	271.811
変換なし+平均	5	—	—	0.067	0.309	0.323	0.323	—
	10	—	—	0.091	0.288	0.309	0.319	—
	20	—	—	0.107	0.288	0.315	0.284	—
à la carte+平均	5	—	—	0.075	0.322	0.346	0.286	0.002
		—	✓	0.089	0.374	0.391	0.345	0.002
		機能語	—	0.067	0.334	0.352	0.318	0.002
		機能語	✓	0.099	0.379	0.394	0.349	0.002
	10	—	—	0.159	0.351	0.372	0.369	0.002
		—	✓	0.141	0.380	0.393	0.407	0.002
		機能語	—	0.144	0.355	0.367	0.359	0.003
		機能語	✓	0.150	0.384	0.396	0.410	0.003
	20	—	—	0.152	0.378	0.395	0.326	0.002
		—	✓	0.123	0.395	0.407	0.349	0.003
		機能語	—	0.146	0.381	0.390	0.338	0.003
		機能語	✓	0.148	0.402	0.411	0.352	0.002
à la carte+SIF	5	—	—	0.094	0.308	0.346	0.282	0.002
		—	✓	0.071	0.365	0.398	0.324	0.002
		機能語	—	0.081	0.345	0.366	0.296	0.002
		機能語	✓	0.060	0.363	0.386	0.331	0.002
	10	—	—	0.147	0.322	0.356	0.337	0.002
		—	✓	0.150	0.375	0.395	0.374	0.002
		機能語	—	0.140	0.366	0.380	0.355	0.004
		機能語	✓	0.137	0.375	0.391	0.377	0.004
	20	—	—	0.135	0.319	0.367	0.282	0.002
		—	✓	0.149	0.389	0.403	0.339	0.005
		機能語	—	0.168	0.388	0.400	0.337	0.003
		機能語	✓	0.144	0.395	0.408	0.353	0.003
à la carte+concat	5	—	—	0.321	0.458	0.494	0.496	0.009
		—	✓	0.292	0.554	0.569	0.543	0.009
		機能語	—	0.323	0.524	0.561	0.556	0.009
		機能語	✓	0.269	0.588	0.600	0.570	0.009
	10	—	—	0.245	0.477	0.512	0.501	0.015
		—	✓	0.112	0.569	0.587	0.548	0.016
		機能語	—	0.201	0.548	0.583	0.591	0.016
		機能語	✓	0.274	0.602	0.616	<b>0.594</b>	0.016
	20	—	—	0.289	0.485	0.520	0.496	0.030
		—	✓	0.295	0.579	0.598	0.549	0.030
		機能語	—	0.338	0.556	0.588	0.552	0.030
		機能語	✓	<b>0.377</b>	<b>0.607</b>	<b>0.621</b>	0.579	0.029

表 A-19: 「高い」に対して手法や学習条件を変えた時のモデル性能と計算効率の結果。

手法	学習条件			モデル性能				計算効率
	文脈単語数	品詞 filtering	Zipf 白色化	ARI	CKAraw	CKAnorm	Spearman's $\rho$	全用例の 推論時間 [s]
BERT (ベースライン)	5	—	—	1.000	1.000	1.000	1.000	1696.732
	10	—	—	1.000	1.000	1.000	1.000	1886.096
	20	—	—	1.000	1.000	1.000	1.000	2141.225
変換なし+平均	5	—	—	0.115	0.333	0.354	0.312	—
	10	—	—	0.061	0.278	0.300	0.305	—
	20	—	—	0.071	0.255	0.277	0.222	—
à la carte+平均	5	—	—	0.121	0.338	0.353	0.281	0.010
		—	✓	0.167	0.407	0.409	0.341	0.010
		機能語	—	0.100	0.358	0.364	0.303	0.013
		機能語	✓	0.163	0.395	0.397	0.328	0.008
	10	—	—	0.080	0.319	0.332	0.335	0.007
		—	✓	0.096	0.361	0.364	0.354	0.007
		機能語	—	0.087	0.337	0.343	0.335	0.014
		機能語	✓	0.102	0.357	0.361	0.367	0.007
	20	—	—	0.100	0.316	0.328	0.275	0.024
		—	✓	0.116	0.334	0.339	0.268	0.008
		機能語	—	0.105	0.321	0.329	0.281	0.031
		機能語	✓	0.114	0.337	0.341	0.284	0.028
à la carte+SIF	5	—	—	0.098	0.317	0.341	0.280	0.009
		—	✓	0.116	0.394	0.405	0.319	0.030
		機能語	—	0.088	0.344	0.354	0.273	0.009
		機能語	✓	0.111	0.373	0.384	0.305	0.033
	10	—	—	0.072	0.298	0.319	0.311	0.008
		—	✓	0.118	0.354	0.361	0.346	0.025
		機能語	—	0.077	0.329	0.336	0.312	0.009
		機能語	✓	0.088	0.345	0.352	0.359	0.030
	20	—	—	0.090	0.283	0.309	0.262	0.009
		—	✓	0.100	0.327	0.334	0.265	0.023
		機能語	—	0.100	0.317	0.325	0.276	0.023
		機能語	✓	0.122	0.329	0.336	0.273	0.012
à la carte+concat	5	—	—	0.201	0.458	0.486	0.466	0.046
		—	✓	0.297	0.594	0.621	0.567	0.046
		機能語	—	0.279	0.572	0.584	0.567	0.045
		機能語	✓	0.369	0.621	0.632	<b>0.598</b>	0.045
	10	—	—	0.293	0.475	0.503	0.441	0.088
		—	✓	0.317	0.601	0.625	0.581	0.088
		機能語	—	0.389	0.590	0.601	0.559	0.089
		機能語	✓	0.322	<b>0.629</b>	<b>0.638</b>	0.597	0.089
	20	—	—	0.279	0.485	0.512	0.381	0.289
		—	✓	0.360	0.595	0.621	0.485	0.169
		機能語	—	0.370	0.594	0.605	0.484	0.169
		機能語	✓	<b>0.393</b>	0.624	0.635	0.505	0.163

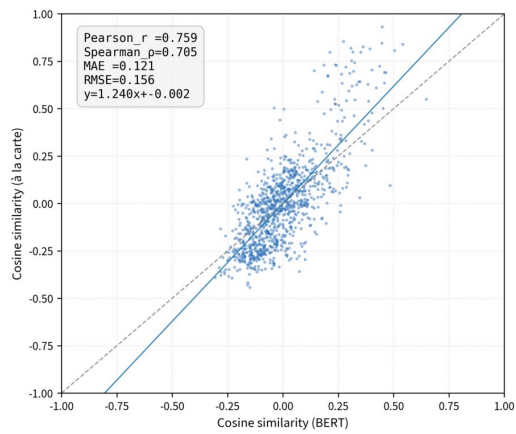


図 A-1: 「生理」の用例間類似度の分布。

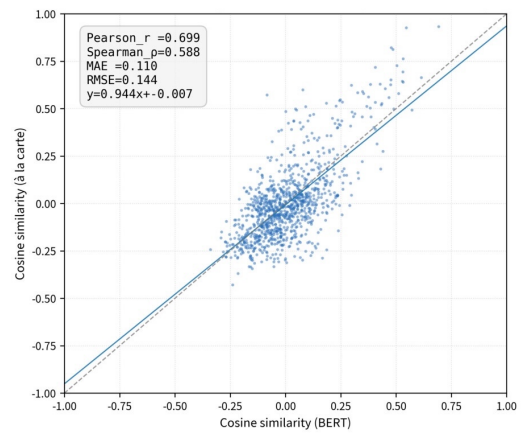


図 A-4: 「養生」の用例間類似度の分布。

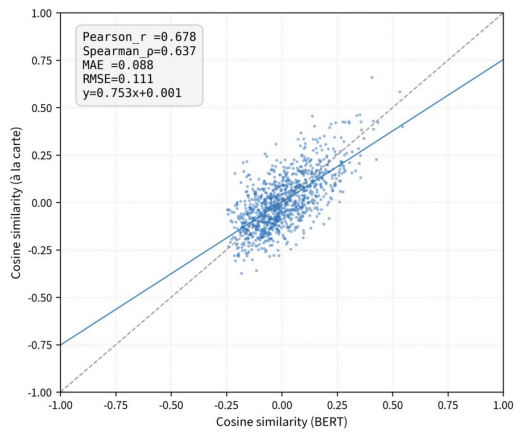


図 A-2: 「姿勢」の用例間類似度の分布。

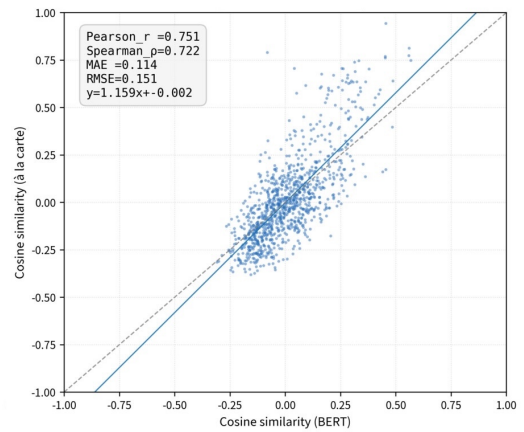


図 A-5: 「消化」の用例間類似度の分布。

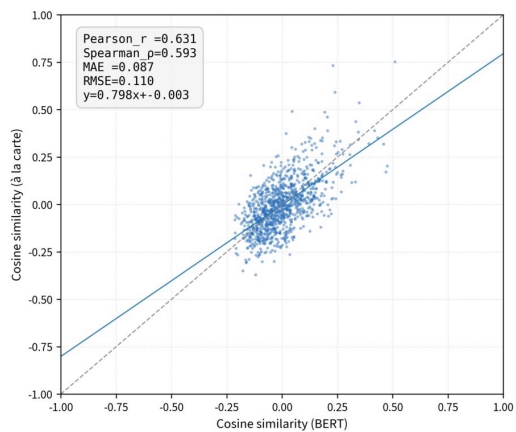


図 A-3: 「問題」の用例間類似度の分布。

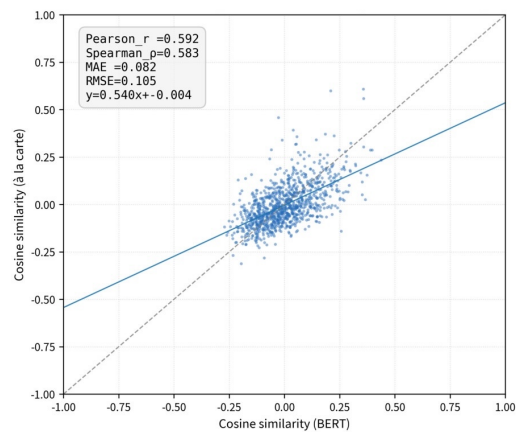


図 A-6: 「運動」の用例間類似度の分布。

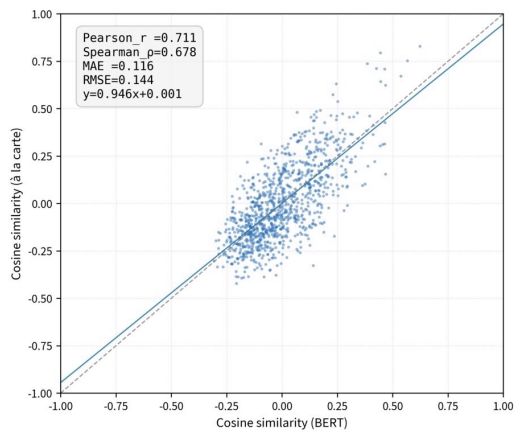


図 A-7: 「適当」の用例間類似度の分布。

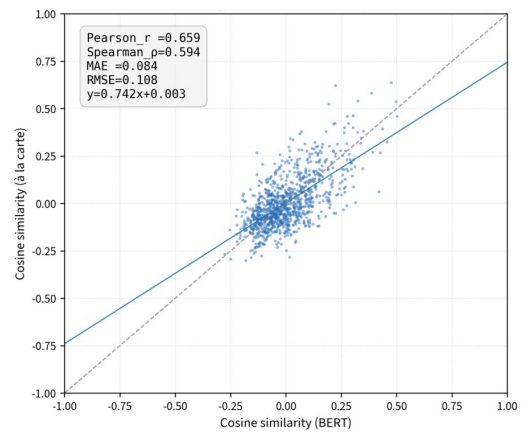


図 A-10: 「薄い」の用例間類似度の分布。

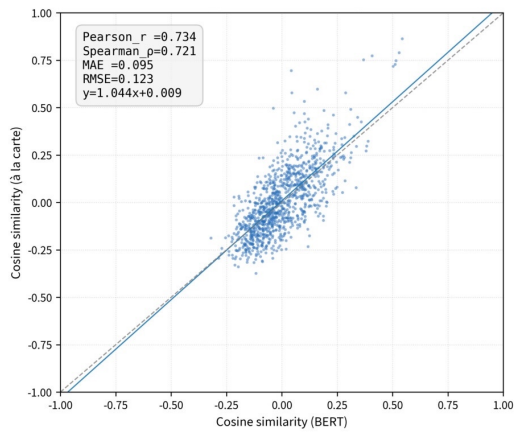


図 A-8: 「絶対」の用例間類似度の分布。

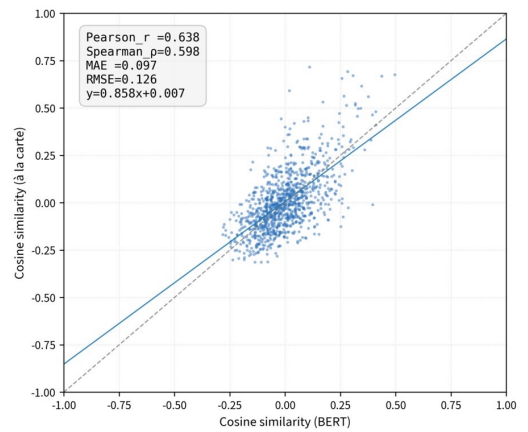


図 A-11: 「高い」の用例間類似度の分布。

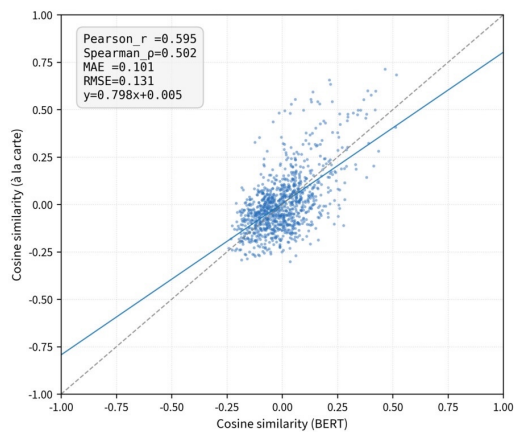


図 A-9: 「甘い」の用例間類似度の分布。

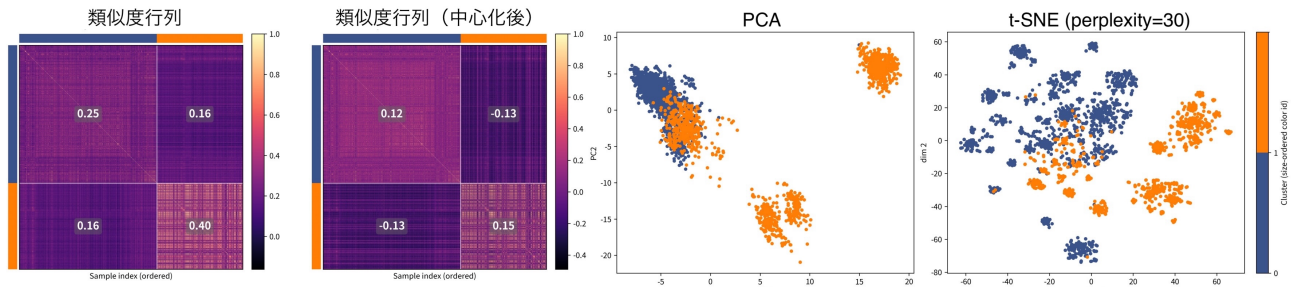


図 A-12: 「生理」に対する提案手法の結果：類似度行列と PCA・t-SNE による埋め込み可視化。

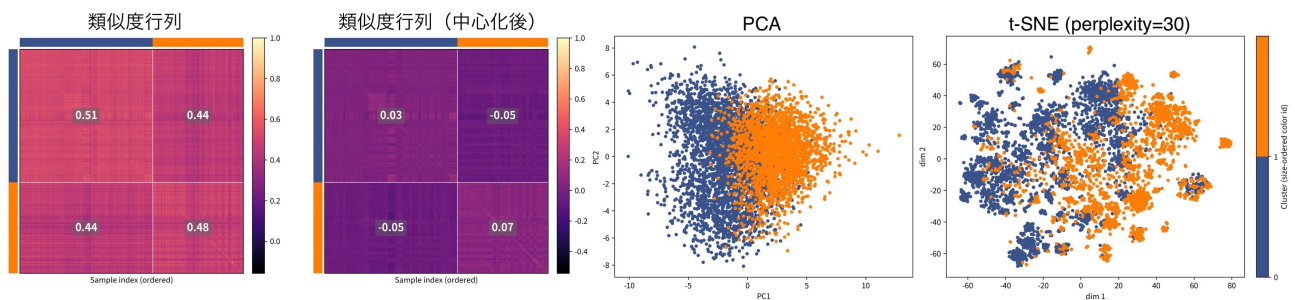


図 A-13: 「姿勢」に対する提案手法の結果：類似度行列と PCA・t-SNE による埋め込み可視化。

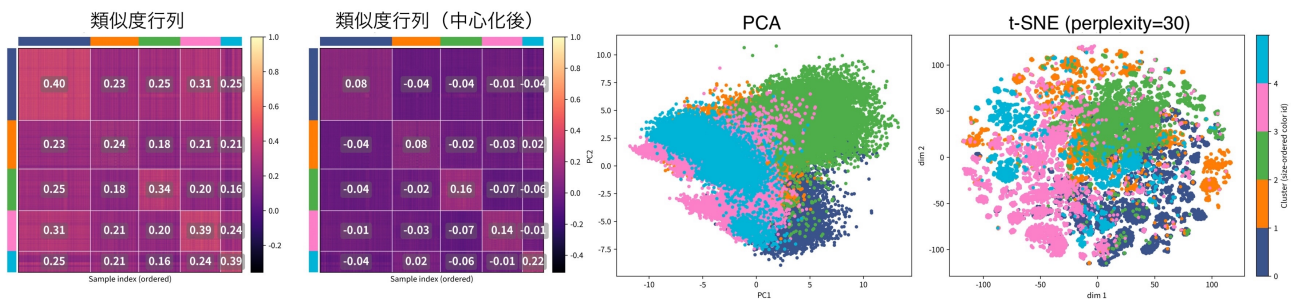


図 A-14: 「問題」に対する提案手法の結果：類似度行列と PCA・t-SNE による埋め込み可視化。

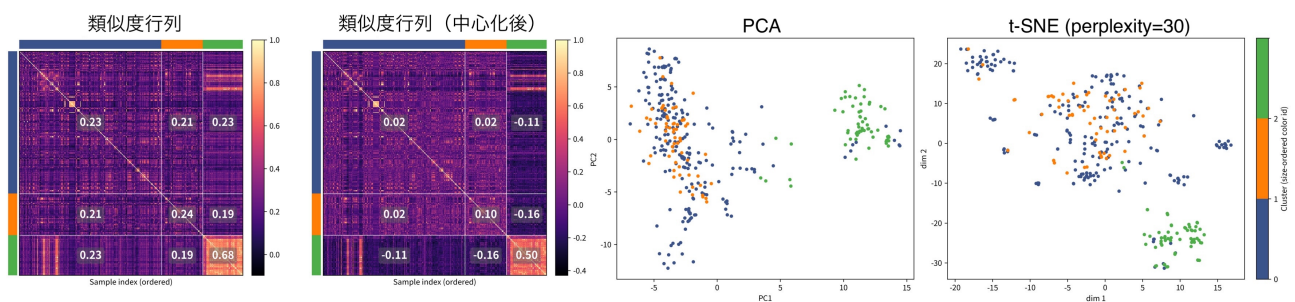


図 A-15: 「養生」に対する提案手法の結果：類似度行列と PCA・t-SNE による埋め込み可視化。



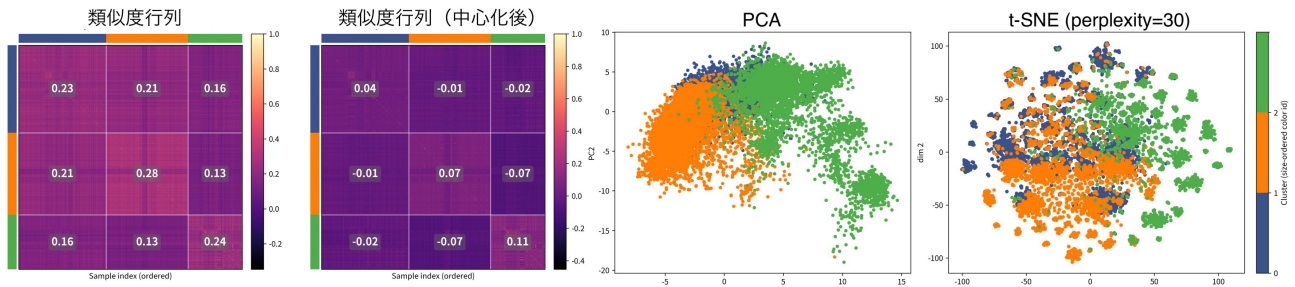


図 A-16: 「運動」に対する提案手法の結果：類似度行列と PCA・t-SNE による埋め込み可視化。

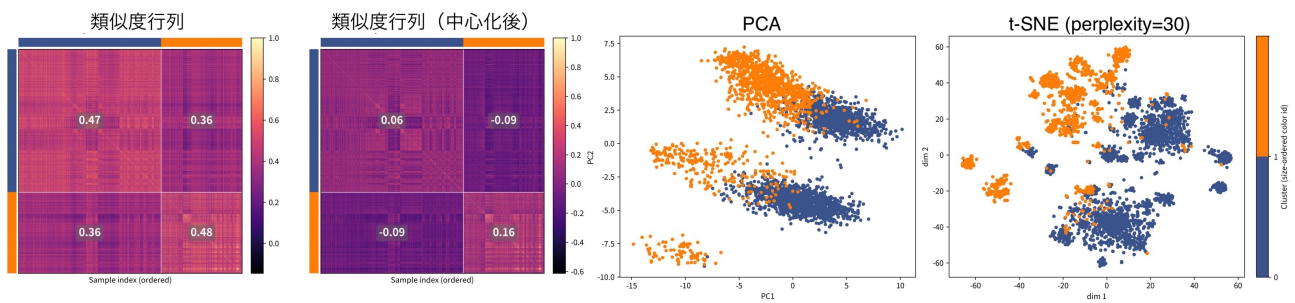


図 A-17: 「適当」に対する提案手法の結果：類似度行列と PCA・t-SNE による埋め込み可視化。

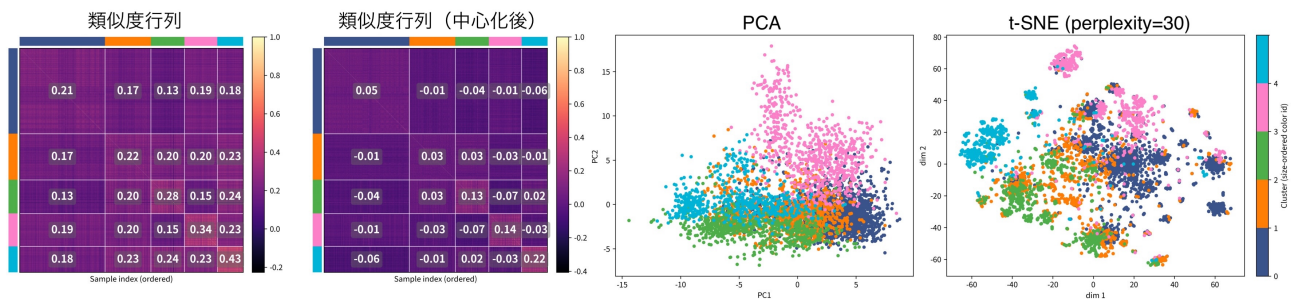


図 A-18: 「薄い」に対する提案手法の結果：類似度行列と PCA・t-SNE による埋め込み可視化。

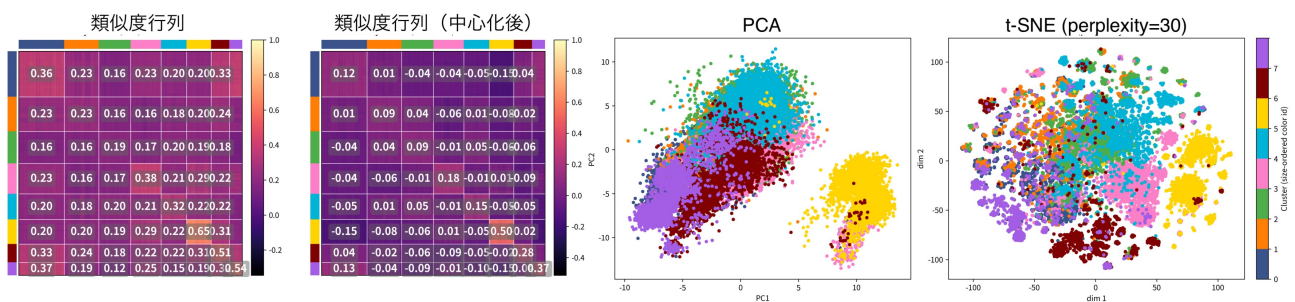


図 A-19: 「高い」に対する提案手法の結果：類似度行列と PCA・t-SNE による埋め込み可視化。



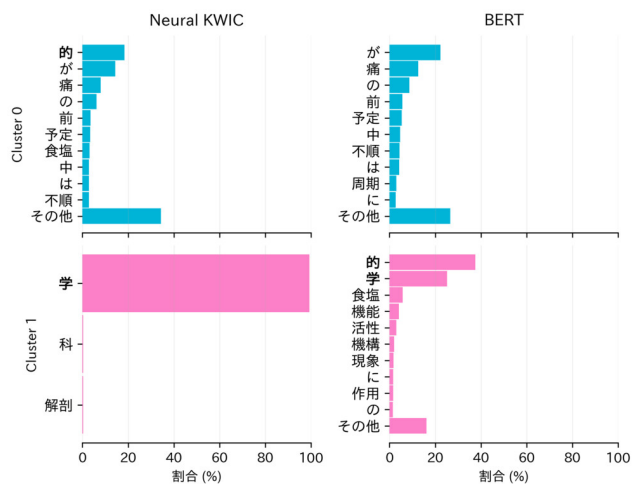


図 A-20: 「生理」の直後トークンの出現頻度。

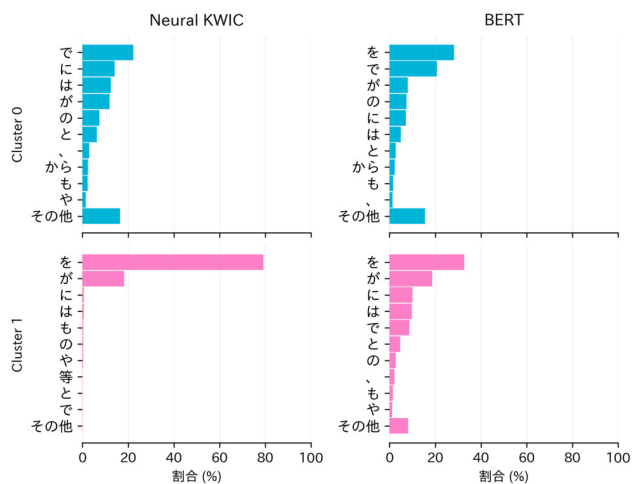


図 A-21: 「姿勢」の直後トークンの出現頻度。

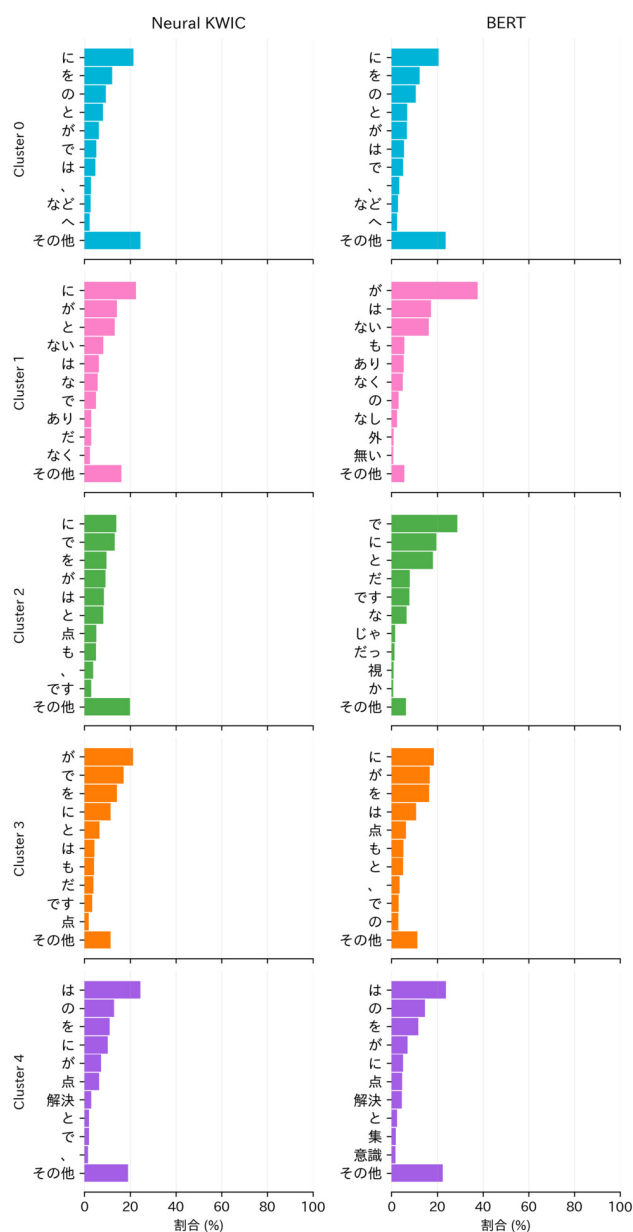


図 A-22: 「問題」の直後トークンの出現頻度。

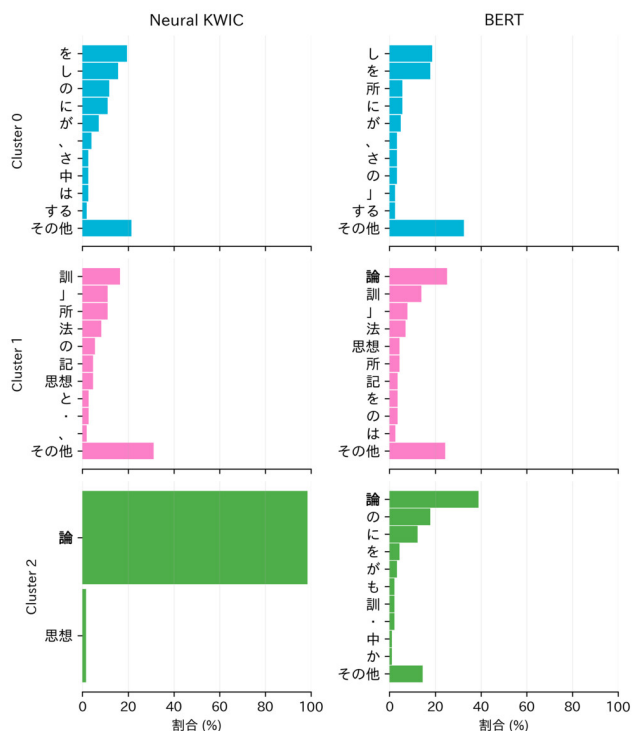


図 A-23: 「養生」の直後トークンの出現頻度。

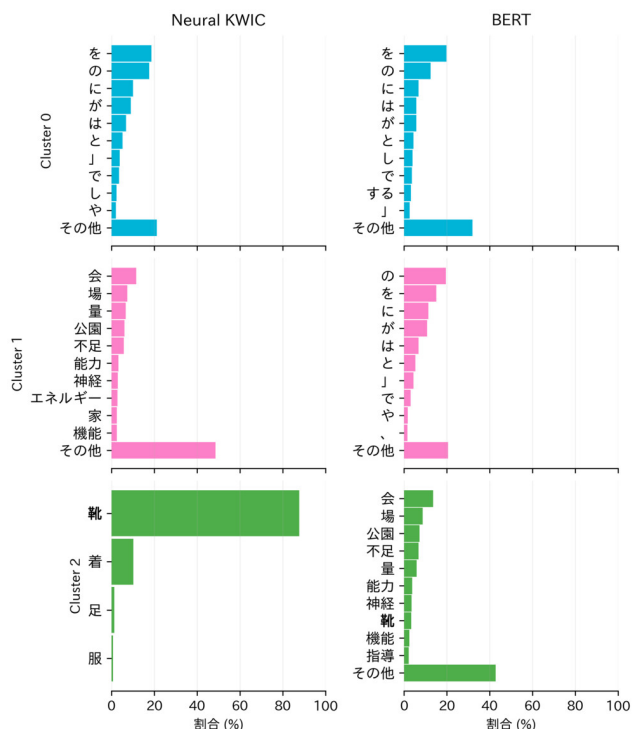


図 A-24: 「運動」の直後トークンの出現頻度。

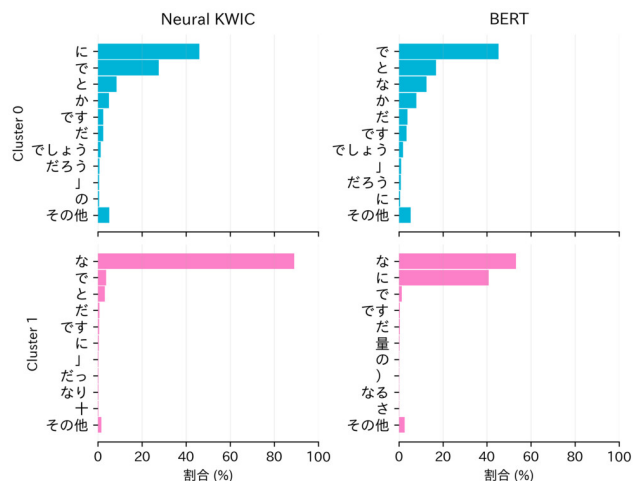


図 A-25: 「適当」の直後トークンの出現頻度。

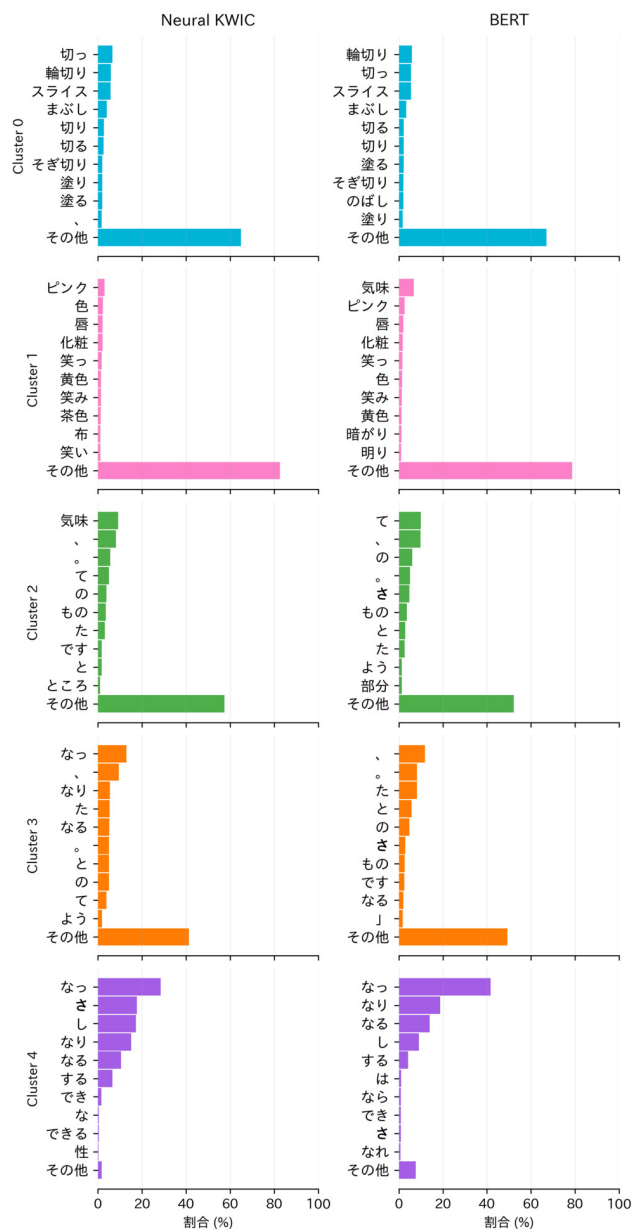


図 A-26: 「薄い」の直後トークンの出現頻度。

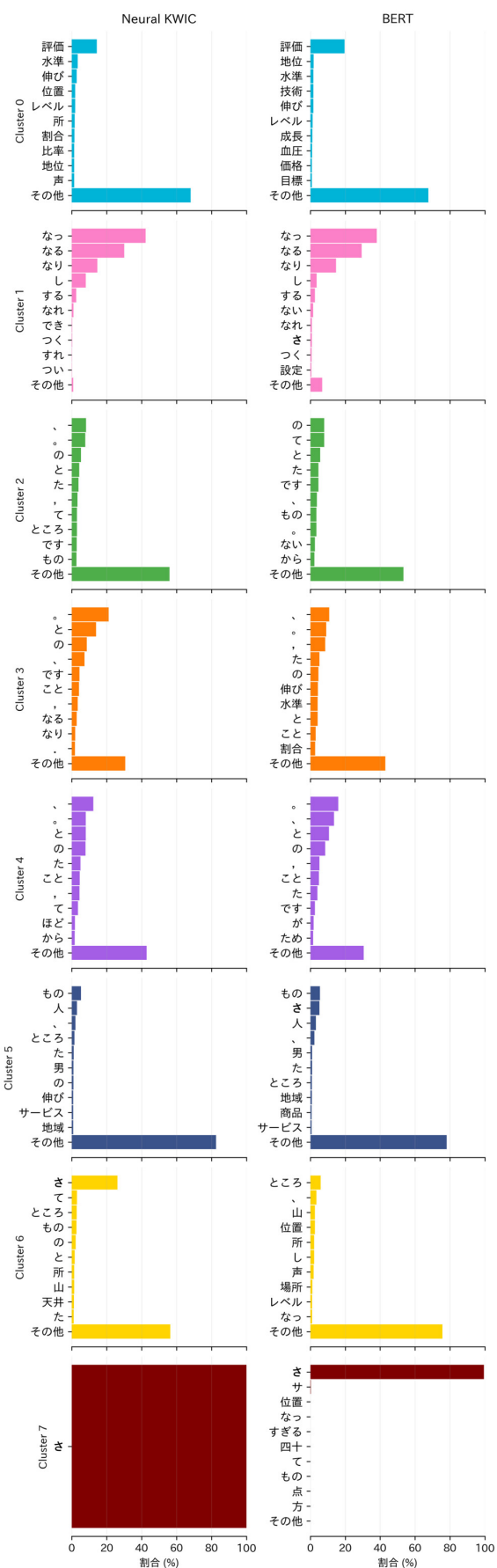


図 A.27: 「高い」の直後トークンの出現頻度。