埋め込み表現の内在次元を測る

片岩拓也 [1]　趙羽風 [2]　大木哲史 [1,3]
[1] 静岡大学　[2] 北陸先端技術大学院大学 [3] 理研 AIP
kataiwa.takuya.23@shizuoka.ac.jp yfzhao@jaist.ac.jp
ohki@sec.inf.shizuoka.ac.jp

概要

本研究では，言語の埋め込み表現である単語ベクトルや埋め込み層について，表現に必要十分な次元である内在次元 (Intrinsic Dimension; ID) を計測し，その冗長度合いを定量評価する．具体的には，(1) Word2Vec や GloVe などの小規模モデルの埋め込みが持つ ID を推定し，(2) Pythia 系列を代表とする大規模言語モデルの埋め込み層における ID をスケール別・学習過程別に解析する．実験の結果，埋め込み空間が外在的な次元に比べ低い次元の多様体上に分布する傾向が見られた．また，モデル規模の拡大に伴う冗長率の変化や，学習初期における急激な ID の形成が見られた．

1　はじめに

大規模言語モデル (**L**arge **L**anguage **M**odels; LLMs) が自然言語処理の各種タスクで高精度な性能を実現する一方で，モデル内部の埋め込み層や中間表現の次元については，十分に解明されていない部分が多い．特に，埋め込みベクトルが外在的に何百〜何千次元もの大きさを持ちながら，実際にはどの程度の有効な次元で情報を表現するのかは，LLM の効率化を考えるうえで重要な問いである．もし埋め込み層が実質的に低次元の多様体上にあるならば，学習・推論コストやモデルサイズの削減に大きな示唆を与え得るからである．

本稿では，**内在次元 (Intrinsic Dimension; ID)** を用いて埋め込み空間の表現に必要十分な次元を考察する．ここでいう ID とは，データが本質的に取りうる自由度，すなわち，データが実際に存在している多様体の次元を表す指標である．一方，**外在次元（Extrinsic Dimension; ED）**は，埋め込みベクトルが表面的に割り当てられている次元数[1)]を指す．ID と ED の関係に関しては，これまでにも上田ら [1] が文を単位にした埋め込み空間の ID を TwoNN 法で推定した事例がある．具体的には，学習済み埋め込みを用いて文の埋め込み表現を作成し，その文間のユークリッド距離に基づいて ID を測定したところ，ED が数百次元に及ぶにも関わらず，実質的には十数次元程度に凝縮されている可能性が示唆された．しかし，近年主流となった LLM の初期層をはじめとする特定の層に対しては，モデル規模や学習過程が ID に及ぼす影響についての検証が十分に行われていない．本稿では，小規模・大規模言語モデル[2)]の埋め込み双方を対象にそれぞれの ID の特徴を比較することで，埋め込み空間の ID をより包括的に分析することを目的とする．

本稿では，以下の 2 つの RQ を設定し，小規模・大規模言語モデルの埋め込みの ID と冗長率を実証的に調査する．

RQ1　ED と ID の間に生じる差はどの程度顕著か，また何によって左右されるのか？
RQ2　LLM の埋め込み層の ID は，パラメータ数・埋め込み次元といったモデル規模や学習過程によってどのように変化・安定化するか？

まずは，RQ1 に関して，単語埋め込みを対象に，実際に ID と ED の乖離がどの程度顕著なのかを定量的に分析する（第 3.2 節）．次に，RQ2 に対して，モデル規模の拡大や学習過程が，ID の値に影響を与えるかどうかを追跡調査する．LLM の埋め込み層として Pythia シリーズ [2] を取り上げ，(a) モデル規模別に有効に使われている次元を定量化する冗長率を測定し，モデル規模がどのように ID に影響を与えるかを検証する（第 3.3 節）．そして (b) 同一モデルのチェックポイントを取得し，事前訓練の過程の中で ID の推移を調べる（第 3.4 節）．この二段構成

1）例として，Word2Vec が 300 次元，BERT が 768 次元など

2）ここでの小規模言語モデルとは，Word2Vec などのモデル．

により，モデル規模と，学習過程という異なる視点で，動的埋め込み空間の本質的な次元性を捉える．

本稿の貢献は，(1) 小規模埋め込みおよび LLM による大規模埋め込みの双方を扱うことで，これまで断片的だった言語埋め込み空間の多様体構造を ID という観点から比較可能にした点，(2) モデル規模や学習段階ごとの変化を定量的に示すことで，LLM による埋め込みベクトルが抱える潜在的な**冗長性**を議論した点にある．これらから得られる知見は，LoRA などの埋め込み層の低ランク近似手法の理論的根拠を補強するとともに，LLM の軽量化・高速化の可能性を検証する礎となるだろう．以下，まず関連研究を概説する（第 2 章）．続いて，実験方法とその結果（第 3 章）を述べ，考察（第 4 章）を経て結論とする（第 5 章）．

2 関連研究

高次元データの本質的な次元を表す指標として，ID や局所内在次元 (**L**ocal **I**ntrinsic **D**imensionality; LID) が注目されている [3, 4]．これらは，主成分分析などの線形的な次元圧縮では捉えきれない非線形多様体上の構造を考慮するため，深層学習による特徴表現を幾何学的に解釈するうえで重要な手がかりとなる．Ansuini ら [5] は，学習済み CNN の各層の出力に対して ID を推定し，(1) 層の ED よりも小さい ID を示すこと，(2) 層を通過するにつれてさらに低次元化すること，(3) ID が大きいほど汎化性能が低下することを報告した．

文埋め込みに関しても，TwoNN [6] による ID 推定が試みられ，実際には 10 次元前後にまで圧縮されると示した [1]．LLM が人間の言語を模倣するために必要なモデル規模に関する議論について，[7] では，大規模化による性能向上と人間言語の多様性を踏まえ，どの程度のパラメータ数があれば充分に言語を扱えるのかという問いを，意味論的制約や統語論的制約を意図的に崩すことで分析されている．

また，LoRA[8] などの低ランク近似による手法は，多様体仮説に基づき，LLM の推論・学習コストを削減するアプローチである．しかし，実際にどの程度低次元化が進むのか，また学習過程でどのように変動するのかは，十分に検証されていない．したがって，埋め込み空間における ID の推移を調べることは，表現学習の理解を深め，高速化・省メモリ化の可能性を探るうえでも重要な課題となる．

3 実験方法・結果

本章では，まず LID と ID の推定方法について概説した後，埋め込みモデルを対象に 3 つの実験 (実験 1–3) を行う．また，その結果を都度示す．

3.1 LID と ID の推定

LID の最尤推定式. LID の推定方法はいくつか提案されているが，本稿では [3] に基づく最尤推定を用いる．任意の点 x に対して，十分小さい近傍内ではデータ密度が一定とみなし，その近傍内の点の数がポアソン過程に従うと仮定する．この仮定の下，点 x における推定値 $\widehat{\mathrm{LID}}_k(x)$ は式 (1) で与えられる．

$$\widehat{\mathrm{LID}}_k(x) = \left(\frac{1}{k-1} \sum_{i=1}^{k-1} \ln \frac{d_k(x)}{d_i(x)} \right)^{-1}, \tag{1}$$

ただし，$d_i(x)$ は点 x と i 番目に近い点との距離を表す．k は近傍点の数を制御するパラメータであり，本稿では $k = 5$ とする．

ID の推定式. 式 ((1)) で各点の LID を求めた後，データ全体の ID 推定には，単純に相加平均をとる方法が考えられる [3]．しかし，LID の分布は極端に大きい外れ値を含む傾向があるため，相加平均ではバイアスを生じやすい．そこで論文 [9] の議論に従い，調和平均を用いた式 2 で ID を推定する．

$$\widehat{\mathrm{ID}} = \left(\frac{1}{n} \sum_{i=1}^{n} \widehat{\mathrm{LID}}_i^{-1} \right)^{-1}, \tag{2}$$

ここで n はサンプリングした点の総数であり，本稿においてはこれは語彙数である．また，$\widehat{\mathrm{LID}}_i$ はそれぞれの点 i における推定 LID 値を指す．

3.2 実験 1：単語埋め込みの ID 推定

本実験では，代表的な学習済み単語埋め込み (Word2Vec [10]，GloVe [11]，FastText [12]) を取り上げ，実際の ED よりも低次元構造を持つのかを評価する．単語埋め込みは GenSim ライブラリ [13] を用いて獲得することができる，`word2vec-google-news-300`, `glove-wiki-gigaword-300`, `fasttext-wiki-news-subwords-300`を使用する．また，正規分布からランダムサンプリングした同次元のベクトル（以下，random と呼ぶ）との比較を行うことで，言語が持つ構造性による低次元性を定量的に把握できるようにする．

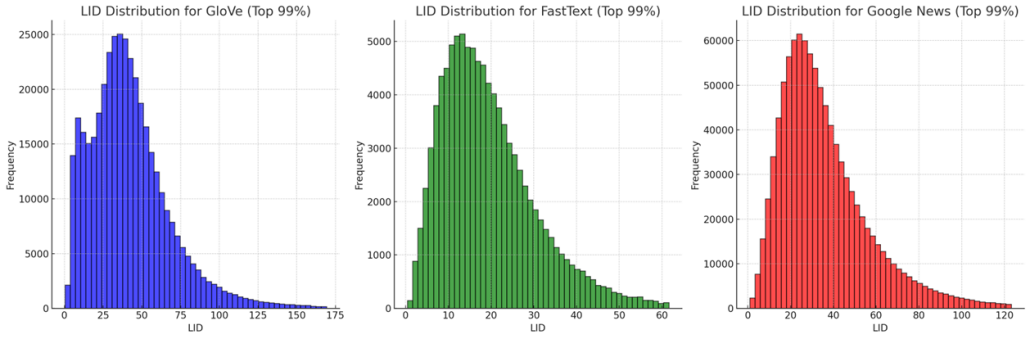

図 1　LID 値のヒストグラム（左から順に，Glove，FastText，GoogleNews）

表 1　推定 ID

統計量	GloVe	FastText	Word2Vec	Random
ID	24.77	13.19	24.75	130.3

実験手順.

1. 単語埋め込みの全単語リストからランダムサンプリングし，対応するベクトルを取得.
2. 各単語ベクトルに対し，ユークリッド距離に基づいて $k = 10$ 個の近傍点を FAISS [14] で検索.
3. 式 (1) から，各単語ベクトルの LID 値 $\widehat{\mathrm{LID}}_k$ を計算.
4. 式 (2) から，サンプル全体の ID を推定.

ベースライン（random）の設定. 埋め込みと同じ次元数 d を用い，平均 $\mathbf{0}$，共分散 $\mathbf{I}$ の正規分布から 100,000 点サンプリングして random を作成した.

結果. 表 1 に実験 3.2 によって得られた LID 値のヒストグラムを示す．また，LID 値の分布を図 1 に示す．同次元の Random ベクトルとの比較も行い，言語由来のベクトルが構造的かどうかを考察した．結果として，ID は約 10〜30 次元程度と推定された．これは ED の値 300 に対して 3〜10%に留まっており，ED の冗長性が強く示唆された.

3.3　実験 2：LLM の規模別の冗長率推定

次に，Pythia シリーズ [2] を用いて，埋め込み層の冗長率を計測する．Pythia シリーズは，同一のデータセットと学習条件でありながら，パラメータ数を 14M から 12B まで用意しているため，モデル規模に応じた ID の推移を一貫した条件下で比較できる．各モデルに対して，埋め込みベクトル次元数を ED とし，先述の手法で推定した ID を用いる．ここでは冗長率 (Redundancy) を

$$\text{Redundancy} = \frac{\text{ED} - \text{ID}}{\text{ED}} \tag{3}$$

表 2　各 Pythia モデルの冗長率 (Redundancy) と内在次元（ID）と埋め込み次元 (ED)

Model	Redundancy (%)	ID	ED
pythia-14m	72.3996	35.3284	128
pythia-70m	94.1433	29.9861	512
pythia-160m	96.4882	26.9703	768
pythia-410m	97.5635	24.9493	1024
pythia-1b	98.1822	37.2276	2048
pythia-1.4b	98.4276	32.2022	2048
pythia-2.8b	98.6649	34.1787	2560
pythia-6.9b	98.0884	78.2982	4096
pythia-12b	97.6207	121.8165	5120

と定義し，スケールの変化に伴う変化を観察する．実験 3.2 と異なり，ED が変化するため単なる ID の追跡よりも ED を考慮する必要があるからである.

結果. 表 2 に実験 3.3 で得られた結果を示す．この結果から，モデル規模が大きくなるにつれ ID は拡大し，冗長率は 90% 超〜98% 前後となり，非常に高い数値を示す．また `pythia-410m` 以降，冗長率は 98%前後を推移する．すなわち，十分に大きいモデルでは，冗長率は大きく変化しない，と考えられる.

3.4　実験 3：LLM の学習過程の ID 推定

最後に，LLM による埋め込み空間が学習過程においてどのように変形性されるのかを調査する.

モデルの学習チェックポイントを 1000 から 10000 までの区間では 1000 ステップごとに，10000 から 143000 の区間では 5000 ステップごとに取得し，それぞれに埋め込み層に対して式 (1) および式 (2)，さらに式 (3) で冗長率を推定する．これにより，学習が進むにつれて埋め込み空間の ID や冗長率がどのように変化するかを追跡する．GPU リソースの限界から，`pythia-14m`〜`pythia-1.4b` のみ実験した.

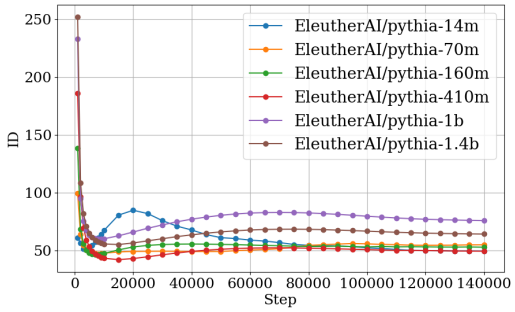

図 2　学習過程の ID の変化

表 3　トークナイザー別 ID（ED=300，Vobab Sample = 10000）

Tokenizer	ID
Word2Vec-SentencePiece	24.7846
Word2Vec-BPE	24.7275
Word2Vec-WS	27.0036
FastText-SentencePiece	11.3744
FastText-BPE	11.9805
FastText-WS	10.7492

結果. 図 2 に示す．図 2 から，学習の最初期段階で ID が急激に減少し，その後緩やかに収束する傾向が観察できる．比較的小さい `pythia-14m` で不安定な挙動は許容されている [15].

4　議論

4.1　RQ1：ED と ID の差は顕著である

単語埋め込みが 300 次元の ED に対して 10〜30 次元程度の ID を持つことは，上田らの先行研究 [1] とも大雑把には合致する．実際に，ランダムな同次元ベクトルと比較すると，確かに構造的であり，驚くほどに低いと言える．たとえば FastText は他より ID が小さいが，これがサブワード分割の影響なのか，学習データや頻度特性によるのか，より詳細に分析する必要がある．簡易的な分析として，AG-NEWS[16] をコーパスとして，トークナイザー別にモデルを訓練し，観察を行った（表 3）[3].

4.2　RQ2：冗長率は高止まりする

今回のスケール別分析では，モデル規模が拡大すると，ID は大雑把には比例して増加するが，ED の増分ほどには ID が上昇せず，冗長率が 98%ほどで高止まりする傾向が見られた．このことは，パラメータが過剰となり，効率的な埋め込み表現には貢献しない次元が多く存在することを示唆する．LLM は表現の豊かさを確保する一方で，実質的に活用されていない軸が含まれるため，ED と ID の乖離が大きくなる可能性が高い．また，冗長率が高止まりし，変化しなくなる点が存在するならば，言語を表現するのに必要十分な核となる表現の存在と，その次元性を示唆するのではないだろうか．仮に，もしこのままモデルをスケールしても冗長率が 98%から大きく変化しないなら，言語という対象はむしろ学習の余地が残るほどに複雑で情報量のあるものなのかもしれない．この観点から見ると，一見無駄に思える次元が下流タスク適応の際に柔軟性の源泉となり得る点も考慮すべきであり，端的に汎化性能と結び付ける議論には注意が必要である．

さらに，実験 3 の結果から得られた学習過程中の変化は，学習初期には，言語の主要な特性などを反映した低次元の骨格が急速に形成され，その後は細部の調整を行う微調整段階に移行して安定化することを意味するのかもしれない．こうした現象は，表現能力の確保と推論・学習コストのトレードオフに改めて注目する必要があることを示唆している．

5　おわりに

本稿では，小規模・大規模言語モデルの埋め込み空間の ID を測定し，モデル規模や学習過程に着目して冗長率を評価した．主な知見は以下の通りであり，これらは，言語モデルの埋め込み表現理解，また，言語そのものが持つ情報量の大きさについての示唆を与えるものであると考える．

- 小規模な言語モデルの埋め込みは ED=300 に対して ID が 10〜30 程度と小さく，その差が顕著．
- モデル規模が大きくなるほど ID も増加するが，ED の増大速度のほうが大きく，冗長率は 98%ほどで高止まりする．
- 学習初期に ID が急激に変化し，その後は緩やかに安定化する．

今後は，(1) 多言語や時代別の多様なコーパスを用いた比較，(2) 埋め込み層以外の注意機構や中間層の ID 測定，(3) 低ランク近似による精度・速度・メモリ効率の検証などが課題となる．(1) は特に計算的アプローチによる言語学的な知見への還元，(2)，(3) は，LLM の解釈性と効率化の可能性をさらに広げると期待する．

3)　表 3 の WS とは空白区切りを指す．

謝辞

東京大学の上田亮さん，東京理科大学の山本悠士さん議論をしていただき，貴重なご意見をいただきました．本研究の一部は JSPS 科研費 JP 23H00463, JP 23K28085, および JST ムーンショット型研究開発事業 JPMJMS2215 の助成を受けたものです．

参考文献

[1] 上田亮, 横井祥. 言語の固有次元を測る. 言語処理学会 第 30 回年次大会 発表論文集, 3 2024. 委員特別賞受賞.

[2] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. **arXiv preprint arXiv:2304.01373**, 2023.

[3] Elizaveta Levina and Peter J. Bickel. Maximum likelihood estimation of intrinsic dimension. In **Advances in Neural Information Processing Systems**, Vol. 17, pp. 777–784, 2004.

[4] Laurent Amsaleg, Oussama Chelly, Teddy Furon, Stéphane Girard, Michael E. Houle, Ken ichi Kawarabayashi, and Michael Nett. Estimating local intrinsic dimensionality. In **Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, pp. 29–38, 2015.

[5] Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. **arXiv preprint arXiv:1905.12784**, 2019.

[6] Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. **Scientific Reports**, Vol. 7, No. 1, p. 12140, 2017.

[7] 山本悠士, 上田亮, 唐木田亮, 横井祥. 人の言語を模倣するのに必要十分な言語モデルの大きさはどれだけか. NLP 若手の会 (YANS) 第 19 回シンポジウム, 2024.

[8] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In **International Conference on Learning Representations**, 2022.

[9] David J. C. MacKay and Zoubin Ghahramani. Comments on 'maximum likelihood estimation of intrinsic dimension' by e. levina and p. bickel (2004), 2005.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In **Proceedings of the International Conference on Learning Representations (ICLR 2013)**, 2013.

[11] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 1532–1543, 2014.

[12] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. **Transactions of the Association for Computational Linguistics**, Vol. 5, pp. 135–146, 2017.

[13] Radim Řehůřek and Petr Sojka. Software framework for topic modelling with large corpora. In **Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks**, pp. 45–50, Valletta, Malta, 2010. ELRA. https://radimrehurek.com/gensim/.

[14] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024.

[15] Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models, 2022.

[16] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification, 2015.