Matryoshka 表現学習を考慮した埋め込みモデル蒸留

小島 幹　岩﨑 雄介 山本 昌輝
株式会社デンソーテン
{motoki.kojima.j3h,yuusuke.iwasaki.j7p,masaki.yamamoto.j3h}@jpgr.denso.com

概要

Retrieval-Augmented Generation (RAG) は，外部知識を活用することで大規模言語モデルにおけるハルシネーションを抑制できる手法として注目されている．RAG ではベクトル検索を利用するケースが多いが，スマートフォンや車載器などのエッジデバイスで実現する場合，計算資源が限られている点が課題となる．本研究では，次元別ランク整合フィルタリングに基づく蒸留を提案し，次元削減してもベクトル検索の精度低下を抑制可能な埋め込みベクトルを生成できる小規模埋め込みモデルの開発を達成した．

1　はじめに

RAG[1] の性能を向上させるための重要な要素の 1 つにベクトル検索の精度向上が挙げられる．ベクトル検索の精度向上には高品質な埋め込みベクトルが重要であり，高次元長の埋め込みベクトルは豊かな意味情報を保持できるため，高い検索性能を発揮できる一方，計算コストやストレージコストが増大する．本研究では特にストレージコストの課題に着目し，ストレージコストを抑制しつつ高い検索精度を維持できる小規模埋め込みモデルの開発を目指す．

高次元埋め込みベクトルの軽量化アプローチの 1 つに，生成済み埋め込みベクトルから一部を切り出し，低次元長の埋め込みベクトルとして下流タスクに活用しても高い性能を維持できるように埋め込みモデルを学習する Matryoshka 表現学習 (MRL)[2] が知られている．MRL で学習したモデルで生成される埋め込みベクトルは，任意次元に切り出しても高いベクトル品質を維持できる．この特徴はストレージ制約の厳しいエッジデバイス上での RAG において，下流タスクに応じて求められる検索精度とストレージコストのバランスを柔軟に調整できる点で有用である．

エッジデバイスでも実行可能な軽量な RAG システムを実現する取り組みとして，EdgeRAG[3] や LightRAG[4]，MiniRAG[5] のような取り組みも報告されている．これらは RAG システム全体の最適化を目指しているが，我々の狙いは検索精度の根幹となる埋め込みベクトルの高品質化と任意次元への変更による下流タスクへの柔軟な対応に焦点を当てている点で異なる．

近年の高性能な埋め込みモデルの開発には対照学習 [6, 7] が広く用いられている．日本語特化埋め込みモデルにおいて上位の性能を持つ Ruri[8] も，2 段階の対照学習により高品質な埋め込みモデルを実現している．対照学習は高品質なテキストペアのデータを大量に学習させるほど性能が上がりやすいが，高品質な学習データの整備には多大なコストがかかる．一方，ターゲットとする埋め込みモデルが小規模である場合，蒸留を活用することで新規に高品質データを構築することなく高精度な埋め込みモデルを開発した報告 [9] もある．本研究ではエッジデバイスでの実行を前提とし，Matryoshka 特性を持つ小規模埋め込みモデルを蒸留によって開発する．

2　関連研究

2.1　Matryoshka 表現学習

MRL[2] は，低次元に切り詰めたベクトルも損失計算に含める学習により，元の埋め込みベクトルの一部を切り出して下流タスクに利用しても精度劣化抑制を可能とする学習手法である．MRL は著名な埋め込みモデル [7, 10] の学習でも採用されている．研究領域でも MRL は埋め込みベクトルに基づくタスクの軽量化で広く応用されており，スケーラブルな埋め込み計算量削減 [11, 12] やクローズドモデルへの適用 [13] などへの応用例がある．特に Wen ら [14] は，MRL とスパースベクトルを組み合わせた CSR を提案している．しかし CSR はスパース演算に対応したライブラリやハードウェアに依存し，適用できるエッジデバイスが制限されるため本研究で

は採用しなかった．

2.2 蒸留

埋め込みモデルの軽量化手法の 1 つとして知識蒸留が知られている．Xiaochuan ら [15] は，MRL と蒸留を組み合わせた学習手法を提案しており，正例-負例サンプリングフィルタによって誤った正例を除去することで，Retrieval タスクの性能改善を達成した．ただし，本フィルタリングは最大次元長にのみ基づくため，低次元の埋め込みベクトルに対する効果は限定的である．また，Li らは [9] は MRL も含めた 3 種の損失関数に基づく多段階の蒸留手法により，高品質な小型埋め込みモデルの Jasper を開発した．当該手法では，教師モデル側のテキストペアスコアのランキング情報に基づく損失を導入することで，検索タスクなどの精度向上を実現している．

3 提案手法

本研究では MRL と同様に複数の低次元ベクトルに基づく損失を考慮した Matryoshka 蒸留によって，Matryoshka 特性を持つ埋め込みモデルを開発する．蒸留設定および，蒸留における課題と提案手法である次元別ランク整合フィルタリングについて述べる．

3.1 蒸留設定

生徒モデル　学習対象とする生徒モデルはエッジデバイス等での実行を見据えた軽量な日本語特化モデルとするため，パラメータ数 100M 以下の ruri-v3-70m[1] とした．ruri-v3-70m は ModernBERT[16, 17] をベースとした日本語特化埋め込みモデルであり，日本語向けの埋め込みモデルベンチマークである JMTEB[18] リーダーボードにおいても高い検索精度を示す．

教師モデル　教師モデルも同様に JMTEB リーダーボード上位のうち，Apache2.0 ライセンスで利用可能なモデルから選定する．ただし MRL に基づく蒸留のため，教師モデルのベクトルを次元削減した場合の検索性能も考慮する．本来は MRL で学習されたモデルが望ましいが，現状 MRL で学習された日本語特化モデルは確認できないため，前述の条件に基づき ruri-v3-130m[2]，ruri-v3-310m[3]，plamo-embedding-1b[4] の 3 モデルを教師モデル候補とした．評価指標には，JMTEB における Retrieval タスク向けの 11 データセットで算出された nDCG@10 の平均値を採用し，生徒モデルの出力次元に基づき 384,256,128 の 3 パターンの埋め込み次元で評価する．表 1 に教師モデル候補において埋め込みベクトルを各次元へ削減した際の検索精度を示す．なお，本評価環境では ruri-v3-70m 等の Retrieval スコアは公表値と一致しなかったため，本稿のスコアは全て本評価環境における結果を記載している．本結果から次元削減時の性能も考慮し総合的な検索品質が高い ruri-v3-130m を教師モデルとして選定した．

表 1　JMTEB による次元削減時の検索精度評価

		各次元の nDCG@10		
教師モデル候補	出力次元	384	256	128
ruri-v3-70m(生徒)	384	72.03	70.68	65.45
ruri-v3-130m	512	**73.54**	**72.28**	67.14
ruri-v3-310m	768	72.37	71.81	**68.19**
plamo-embedding-1b	2048	61.46	59.51	53.27

データセット　蒸留のデータセットには日本語情報検索タスク向けに作成された JFWIR[19] を利用する．JFWIR はクエリ 1 件につき 1 件の正例と 32 件の負例 (ハードネガティブ) からなる，約 6,400 万件のテキストペアで構成されている．本研究では本データセットの先頭 1,000 万件のうち，正例ペアのスコアが 0.6 以上の約 800 万件を利用し，ハードネガティブはスコア上位 7 件を使用した．

3.2 Matryoshka 蒸留のランキングエラー

Matryoshka 蒸留の課題として，学習過程で教師モデルの埋め込みベクトルを次元削減し低次元に切り詰めた場合，ベクトル表現力の低下により正例ペアの類似度スコアが負例ペアに劣るランキングエラーが発生する場合がある．このようなランキングエラーは教師ベクトルが次元削減によって低品質なベクトルとなったと考えられ，教師信号としては不適切な特徴表現となり，モデルの性能を低下させる可能性がある．

次元削減によってどれほどのサンプルが影響を受けるかを確認するため，JFWIR データセットにおいて次元削減時のランキングエラー数を調査した．ruri-v3-130m で生成した埋め込みベクトルについて，各次元長に削減した時の正例ペアの類似度スコアが負例ペアに劣るサンプルの割合を図 1 に示す．Top-K とはランキングエラーの判断条件であ

1） https://huggingface.co/cl-nagoya/ruri-v3-70m
2） https://huggingface.co/cl-nagoya/ruri-v3-130m
3） https://huggingface.co/cl-nagoya/ruri-v3-310m
4） https://huggingface.co/pfnet/plamo-embedding-1b

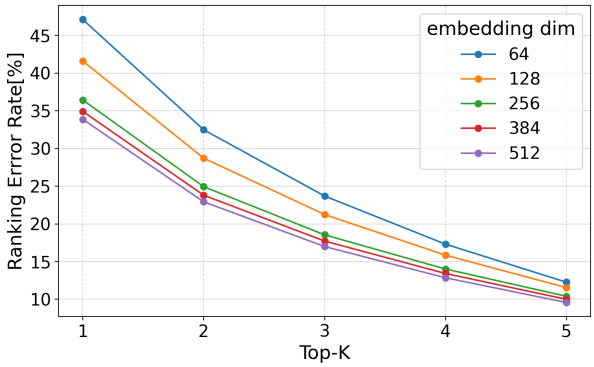

図 1　各次元における閾値別ランキングエラー割合

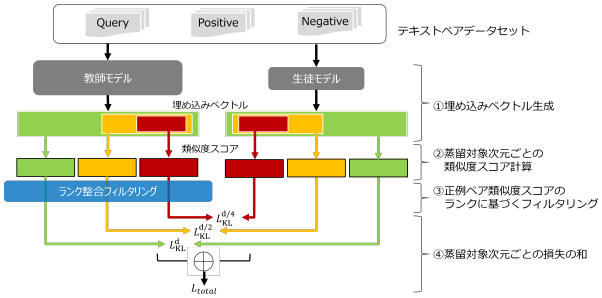

図 2　次元別ランク整合フィルタリングの概要図

り，正例ペアの類似度スコアが負例ペアの上位 K 件以内に入らないサンプルをランキングエラーとする．調査の結果，約 35〜42 %のサンプルが次元削減に伴い教師信号として不適切となることを確認した．特に 128 次元以下の低次元においてランキングエラーが比較的顕著であり，低次元の埋め込みベクトルに対する蒸留の品質低下や学習の不安定化が懸念される．

3.3　次元別ランク整合フィルタリング

3.2 節で述べたランキングエラーによる学習への影響を軽減するため，先行研究 [15, 20] の学習データフィルタリングに着想を得て，蒸留対象次元ごとに蒸留データをフィルタリングする次元別ランク整合フィルタリングを提案する．提案手法では各蒸留対象次元におけるランキングエラーを考慮し，各次元の損失計算において教師ベクトルの正例ペア類似度スコアが負例ペアに劣るサンプルを損失計算から除外するフィルタリングを導入する．図 2 に提案手法の概要を示す．これにより各次元において正しいランキングが維持された教師サンプルのみが学習に利用されるため，低次元の埋め込みベクトルに対しても高品質な教師知識の伝達が期待できる．

3.4　損失関数

提案手法における損失関数には，蒸留における標準的な損失関数である KL ダイバージェンス損失 [21] を採用し，蒸留対象次元ごとに計算した損失の和を最小化する．この損失関数により，生徒モデルによる類似度スコアから導出される確率分布を教師モデルの確率分布に近づけるように学習する．

具体的には，ある次元 d における損失を L_{KL}^{d} とすると，式 1 のように表される．

$$L_{\mathrm{KL}}^{d} = \sum \boldsymbol{P}_{d}^{T} \log \frac{\boldsymbol{P}_{d}^{T}}{\boldsymbol{P}_{d}^{S}} \tag{1}$$

ここで，P_d^T, P_d^S はそれぞれ教師・生徒ベクトルを d 次元に削減したベクトルに基づき計算される確率分布であり，式 2 の通り定義される．

$$P_d^T = \frac{\exp\left(s_k^{T,d}/\tau\right)}{\sum_j \exp\left(s_j^{T,d}/\tau\right)}, \quad P_d^S = \frac{\exp\left(s_k^{S,d}/\tau\right)}{\sum_j \exp\left(s_j^{S,d}/\tau\right)} \tag{2}$$

ここで $\boldsymbol{s}^{T,d}, \boldsymbol{s}^{S,d}$ は各モデルの埋め込みを次元 d で切り出したベクトルに基づき計算される正例ペアおよび負例ペアの類似度スコア配列を示し，$\boldsymbol{s}^{d} = [s_{pos}, s_{neg1}, s_{neg2}, \ldots]$ のように表現される．τ は温度パラメータを示す．なお，本研究では類似度スコアの算出にコサイン類似度を用いる．

提案手法では，各蒸留対象次元において次元削減後も教師モデルの正例ペア順位の整合性が保持されるサンプルのみを学習に反映させるため，教師モデルの正例ペアスコアが負例ペアよりも高い場合にのみ当該サンプルの損失を計上するフィルタリングを行う．あるテキストペアのサンプルについて，各次元 d に対するマスク m^d は式 3 のように定義される．

$$m^{d} = \begin{cases} 1 & \text{if rank}\left(s_{\mathrm{pos}}^{T,d}\right) \leq K, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

ここで $s_{\mathrm{pos}}^{T,d}$ は教師モデルの正例ペアスコアを示し，K は $\boldsymbol{s}^{T,d}$ における $s_{\mathrm{pos}}^{T,d}$ の順位閾値を示す．すなわち K が大きいほど正例ペアスコアのランキングエラーを許容することになる．

最終的な損失 L_{total} は式 4 の通り，蒸留対象次元ごとに計算し，マスクに基づく損失の和を最小化する．なお，D は蒸留対象次元集合を示す．

$$L_{total} = \sum_{d \in D} m^{d} L_{\mathrm{KL}}^{d} \tag{4}$$

表 2　各学習パラメータにおける次元ごとの検索精度評価結果

手法	蒸留対象次元	ランキング閾値 Top-K	nDCG@10 384	256	128	64	全次元平均
対照学習 (ruri-v3-70m)	-	-	72.03	70.68	65.45	57.36	66.38
Matryoshka 蒸留	384,256,128	1	**72.16**	71.02	67.31	58.28	67.19
		3	71.85	**71.25**	67.93	59.89	**67.73**
		5	71.75	70.95	**68.04**	**60.07**	67.70
		-	71.58	71.01	67.87	60.00	67.62
蒸留	384	1	**72.56**	71.10	65.67	56.21	**67.39**
		3	72.43	71.19	**67.21**	58.45	67.32
		5	71.41	**71.33**	67.09	58.45	67.35
		-	72.23	71.24	67.18	**58.51**	67.29

4　実験

3 節で述べた提案手法に基づき学習した埋め込みモデルに対し，Retrieval タスクによる評価により提案手法の効果を検証する．

4.1　学習設定

学習には SentenceTransformers[22] を用い，バッチサイズ 256，学習率 5e-6，Warmup0.1 で 1 エポックの学習を行った．蒸留時の温度パラメータは 0.01，ランク整合フィルタリングのランキング閾値 (Top-K) は 1,3,5, フィルタなしの 4 パターン，蒸留対象次元は 384,256,128 と 384 のみの 2 パターンとした．

4.2　評価設定

評価には，3.1 節と同様に，JMTEB の Retrieval タスク向けのデータセット 11 種を用い，各次元における nDCG@10 を算出する．なお，Ruri-v3 の対照学習時の設定に倣い，接頭辞として「検索クエリ: 」と「検索文書: 」をそれぞれ付与する．

4.3　実験結果と考察

蒸留モデルの評価結果を表 2 に示す．複数次元を蒸留対象とした Matryoshka 蒸留モデルの方が低次元側の検索精度も向上することが確認できた．特に 128 次元以下では約 2〜3pt の精度向上が見られる上，教師モデルを超える精度となっており，低次元も考慮した蒸留が効果的に機能していると言える．

提案手法の効果については，ランク整合フィルタリングを用いた方が各次元で検索精度が上回った．また，ランク整合フィルタリングのランキング閾値 (Top-K) が小さいほど高次元側の精度が向上し，Top-K が大きいほど低次元側の精度が向上する傾向が見られた．し特に Top-K が 1 の場合，2 手法とも 384 次元の精度が向上していることから，厳しいランク整合フィルタリングは高次元側の精度向上に寄与すると言える．これは，教師ベクトルのランキングエラーが抑えられたことでより正確な学習が可能になったためと考えられる．

Top-K の値によって精度向上が大きい次元が変化する別の要因として，フィルタリングによって学習に使われる教師データ数が次元によって偏りが発生することが考えられる．図 1 に示す通り Top-K が小さいほど厳しいフィルタリングにより，低次元側の教師データがより多く削減される．したがって Top-K が小さいほど高次元側で学習に使われる教師データ数が多くなるデータ数の偏りが発生し，相対的に高次元側の損失を重視した学習となり高次元側の精度が向上した可能性がある．

全次元平均の精度を見ると Top-K が 3 の場合が最も高く，次元全体でバランスよく精度が向上していることが分かる．この理由として，先行研究 [23] で報告されるような軽微なランク変動のノイズが正則化の効果を示した可能性がある．フィルタリングなしの場合よりも Top-K が 3 や 5 の方が全次元平均が高いことからも，明らかに劣化した教師信号は除外しつつも軽度なランキングエラーを許容したことで汎化性能があがり，低次元側も含めた全体の精度が向上したと考えられる．

5　まとめ

本研究では，下流タスク要件に合わせた柔軟なストレージコストと検索精度の調整が可能な埋め込みモデル開発が可能な次元別ランク整合フィルタリングによる蒸留を提案した．提案手法により，削減前次元も含む複数の次元長ベクトルそれぞれにおいて Retrieval タスクの精度向上を達成した．今後の課題として，フィルタリングによって生じる次元間のデータ不均衡の解消や，ランキングエラーの閾値を次元ごとに最適化する手法の検討が挙げられる．

参考文献

[1] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[2] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning, 2024.

[3] Sihang Liu, Samira Khan, and Korakit Seemakhupt. Edgerag: Online-indexed rag for edge devices, 2024.

[4] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation, 2025.

[5] Tianyu Fan, Jingyuan Wang, Xubin Ren, and Chao Huang. Minirag: Towards extremely simple retrieval-augmented generation, 2025.

[6] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. **ArXiv**, Vol. abs/2402.05672, , 2024.

[7] Fei Huang, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, Junyang Lin, Jingren Zhou, Huan Lin, Dayiheng Liu, Baosong Yang, An Yang, and Mingxin Li. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025.

[8] Hayato Tsukagoshi and Ryohei Sasano. Ruri: Japanese General Text Embeddings, 2024.

[9] Jiacheng Li, Ziyang Zeng, Dun Zhang, and Fulong Wang. Jasper and stella: distillation of sota embedding models, 2025.

[10] Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, Weiyi Wang, Zhe Li, Gus Martins, Jinhyuk Lee, Mark Sherwood, Juyeong Ji, Renjie Wu, Jingxiao Zheng, Jyotinder Singh, Abheesht Sharma, Divyashree Sreepathihalli, Aashi Jain, Adham Elarabawy, AJ Co, Andreas Doumanoglou, Babak Samari, Ben Hora, Brian Potetz, Dahun Kim, Enrique Alfonseca, Fedor Moiseev, Feng Han, Frank Palma Gomez, Gustavo Hernández Ábrego, Hesen Zhang, Hui Hui, Jay Han, Karan Gill, Ke Chen, Koert Chen, Madhuri Shanbhogue, Michael Boratko, Paul Suganthan, Sai Meher Karthik Duddu, Sandeep Mariserla, Setareh Ariafar, Shanfeng Zhang, Shijie Zhang, Simon Baumgartner, Sonam Goenka, Steve Qiu, Tanmaya Dabral, Trevor Walker, Vikram Rao, Waleed Khawaja, Wenlei Zhou, Xiaoqi Ren, Ye Xia, Yichang Chen, Yi-Ting Chen, Zhe Dong, Zhongli Ding, Francesco Visin, Gaël Liu, Jiageng Zhang, Kathleen Kenealy, Michelle Casbon, Ravin Kumar, Thomas Mesnard, Zach Gleicher, Cormac Brick, Olivier Lacombe, Adam Roberts, Qin Yin, Yunhsuan Sung, Raphael Hoffmann, Tris Warkentin, Armand Joulin, Tom Duerig, and Mojtaba Seyedhosseini. Embeddinggemma: Powerful and lightweight text representations, 2025.

[11] Xianming Li, Zongxi Li, Jing Li, Haoran Xie, and Qing Li. 2d matryoshka sentence embeddings, 2024.

[12] Shengyao Zhuang, Shuai Wang, Fabio Zheng, Bevan Koopman, and Guido Zuccon. Starbucks-v2: Improved training for 2d matryoshka embeddings, 2025.

[13] Jinsung Yoon, Raj Sinha, Sercan Ö. Arik, and Tomas Pfister. Matryoshka-adaptor: Unsupervised and supervised tuning for smaller embedding dimensions. **ArXiv**, Vol. abs/2407.20243, , 2024.

[14] Tiansheng Wen, Yifei Wang, Zequn Zeng, Zhong Peng, Yudi Su, Xinyang Liu, Bo Chen, Hongwei Liu, Stefanie Jegelka, and Chenyu You. Beyond matryoshka: Revisiting sparse coding for adaptive representation, 2025.

[15] Xiaochuan Zhang, Runqing Zhang, Xinlai Xing, Shuran Zhou, and Junliang Chen. A dense retrieval model training method combining matryoshka representation learning and knowledge distillation, 2024.

[16] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024.

[17] Hayato Tsukagoshi, Shengzhe Li, Akihiko Fukuchi, and Tomohide Shibata. ModernBERT-Ja. https://huggingface.co/collections/sbintuitions/modernbert-ja-67b68fe891132877cf67aa0a, 2025.

[18] Shengzhe Li, Masaya Ohagi, and Ryokan Ri. JMTEB: Japanese Massive Text Embedding Benchmark. https://huggingface.co/datasets/sbintuitions/JMTEB, 2024.

[19] Yuichi Tateno. Jfwir: Japanese fineweb information retrieval dataset, 2025. A large-scale Japanese information retrieval dataset with 60+ million document-query pairs.

[20] Ronay Ak, Mengyao Xu, Gabriel de Souza P. Moreira, Benedikt Schifferer, Radek Osmulski, and Even Oldridge. Nv-retriever: Improving text embedding models with effective hard-negative mining, 2025.

[21] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. Distilling dense representations for ranking using tightly-coupled teachers, 2020.

[22] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing**. Association for Computational Linguistics, 11 2019.

[23] Jiahao Xu, Wei Shao, Lihui Chen, and Lemao Liu. Distillcse: Distilled contrastive learning for sentence embeddings. In **Findings of the Association for Computational Linguistics: EMNLP 2023**, p. 8153–8165. Association for Computational Linguistics, 2023.