大きなモデルは小さなモデルの良い教師となり得るか? —日本語算術タスクにおける検証—

山下 修平 [1] 内出 隼人 [1] 斉藤 辰彦 [1]
[1] 三菱電機株式会社
{Yamashita.Shuhei@bc, Uchide.Hayato@dy, Saito.Tatsuhiko@db}
.MitsubishiElectric.co.jp

概要

大規模言語モデル (LLM) の出力した推論過程を用いた知識蒸留は, 小型モデルに高度な推論能力を付与するための有力な手法の 1 つであり, 様々な学習条件が検討されている. 英語を対象とした既存研究では, 教師モデルのサイズ・精度が蒸留の効果に単調に反映されないことが指摘されているが, 日本語においても同様の傾向が成立するかは明らかでない. そこで本研究では, 日本語の算術タスクを対象として, 教師モデルのサイズが蒸留に与える影響を検証した. 実験の結果, 英語と同様に, 蒸留の効果は教師モデルのサイズや精度に比例せず, 7B モデルを教師とした生徒が, 72B モデルを教師とした生徒よりも平均 3.16 ポイント上回る正答率を示した. さらに分析により, 大規模な教師の下では一貫して計算ミスが増加する傾向にあり, 教師の推論過程の形式差が蒸留後の性能に影響する可能性が示唆された.

1 はじめに

大規模言語モデル (Large Language Model; LLM) は, 高度な推論能力を備え, 数学やコーディングをはじめとする多様なタスクにおいて, 高精度な回答を生成できる [1, 2, 3]. これらのモデルは, Chain of Thought (CoT) [4] と呼ばれる多段的な推論を展開することで, 複雑な問題を解決することを特徴とする.

しかし, 推論能力の高いモデルの多くは, 大規模な計算資源を必要とするため, 適用範囲が限定的である. そこで, 精度を維持したまま, モデルを軽量化する技術が必要とされている.

知識蒸留 [5, 6] は, 大きなモデル (教師) の知識を, より小さなモデル (生徒) へと受け継ぐことで, モデルを軽量化する手法である. 特に, 教師が出力した推論過程を学習させることで, 生徒に教師の持つ推論能力を付与する手法が注目されている [2, 7].

一般に, 大規模で高精度な教師ほど, より正確で豊富な推論過程を生成できるため, その知識を蒸留した生徒の性能も向上することが期待されるが, 推論能力の蒸留を対象とした既存研究では, 教師モデルのサイズと蒸留の効果が必ずしも比例しないことが指摘されている [8, 9]. しかし, 既存研究は, これまで主に英語のみを対象としており, 日本語において同様の傾向が成立するかは明らかでない.

そこで, 本研究では, 日本語の算術タスクを対象として, 教師モデルのサイズが小さな生徒モデルの学習に与える影響を検証する. 実験では, GSM8K [10] の日本語訳データセットを用い, 4 つの異なるサイズの教師モデルが生成した推論過程を, 4 種類の生徒モデルに学習させた.

実験の結果, 英語と同様, 蒸留の効果は教師モデルの精度やサイズに比例しないことを確認した. 特に, 図 1 に示すように, 生徒モデルに依らず一貫して最小サイズの 7B モデルを教師とした際に蒸留の効果は最大化し, 72B モデルを教師とした際の正答率を平均 3.16%上回った. 分析により, 大規模な教師の下では一貫して計算ミスが増える傾向にあり, 教師モデル間での推論過程の形式差が蒸留後の性能に影響した可能性が示唆された.

2 関連研究

2.1 知識蒸留

知識蒸留 [5, 6] は, 大きなサイズのモデル (教師) の持つ知識を小さなモデル (生徒) に転移するための手法である. 伝統的な枠組みでは, 教師と生徒の出力ロジットを近づけるように学習するアプローチが主流である [5].

LLM を対象とした知識蒸留では, 教師の生成文を

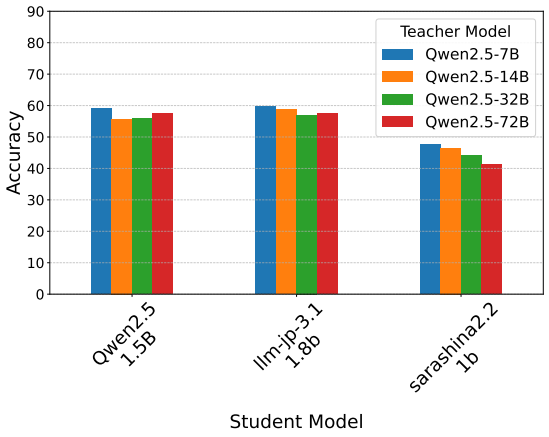


図 1: 日本語算術タスク (GSM8K の和訳) において, 異なるサイズの教師モデルの生成文を蒸留した際の生徒モデルの正答率. 高精度・大規模な教師モデルを用いることが, 必ずしも生徒モデルの推論能力の向上に寄与しないことが示唆される.

正解ラベルとして生徒を学習させる蒸留が広く用いられる [11]. この枠組みにより, 単なるモデル軽量化に留まらず, 教師が示すタスク固有の解き方や出力形式を生徒に学習させ, 対象タスクにおける性能を向上させる手法として注目されている [12].

2.2 推論過程の蒸留と教師サイズ

LLM の推論能力を生徒へ転移する方法として, 教師が生成した推論過程を蒸留する手法が提案されている [4, 7]. 一般に, 大規模な教師ほど正確な推論過程を生成できるため, 蒸留の効果が高まることが期待される. 一方で, 既存研究では, 生徒と教師の能力差が大きい場合, 生徒が教師の推論過程を適切に学習できず, 蒸留の効果が限定的となることが報告されている [8, 9].

既存研究は, これまで主に英語のみを対象としており, 日本語において同様の傾向が成立するかは明らかでない. 本研究では, 日本語算術タスクを対象として, 推論過程の蒸留と教師サイズの影響を検証する.

3 実験

3.1 教師データの生成

算術タスク GSM8K の訓練データの問題文 7,473 件を, 翻訳性能の高い Llama 3.3 Swallow 70B [13] [1] で日本語訳し, 教師モデルによる回答を生成する. 教師モデルとして, 日本語性能の高い Qwen2.5-{7B, 14B, 32B, 72B}-Instruct を用いる. 教師モデルに CoT reasoning を促す同一のプロンプトを与え, 0-shot かつ greedy decoding で回答を生成する (付録 B 参照).

本研究では, 教師モデル間で生成される推論過程の違いが蒸留に与える影響を検証するため, すべてのサイズの教師モデルが正答した問題 5,000 件を学習データとして用いる.

3.2 生徒モデルの学習

生徒モデルとして, 日本語性能の高い小型モデルである, Qwen2.5-{0.5, 1.5}B, sarashina2.2-1b [14], llm-jp-3.1-1.8b [15] を用いる. Instruction tuning による学習の干渉を避けるため, 生徒モデルにはいずれもベースモデルを用いる.

学習は, 教師モデルの生成文を正解ラベルとした, フルパラメータ更新の Supervised Fine-Tuning (SFT) により行う. 学習時のハイパーパラメータの詳細は, 付録 C に一覧する.

3.3 学習後のモデルの評価

GPT-4o により日本語訳された GSM8K の評価データ [2] を用いて, 生徒モデルの回答の正答率を評価する. モデルの生成文の末尾から正規表現を用いて数値を抽出後, 表記ゆれ等による誤分類を避けるため, 正答 (数値) との絶対誤差が 10^{-5} 以下の場合に正解と判定する. 学習した生徒モデルの回答生成は, 教師データ生成時と同様のプロンプトを用い, 0-shot かつ greedy decoding で行う.

また, 蒸留の効果を示すため, 学習前の生徒モデルの正答率を評価する. ただし, ベースモデルは指示追従性が低いため, 4-shot かつ repetition penalty を 1.1 に設定して回答を生成させる.

3.4 結果

表 1 に, 蒸留後の各生徒モデルの正答率を示す. 教師生成文を蒸留することで, sarashina2.2-1b を除く生徒モデルにおいて, ベースモデル (4-shot) からの正答率が大きく改善した [3].

いずれの生徒モデルにおいても, 最小サイズの

1) https://huggingface.co/tokyotech-llm/Llama-3.3-Swallow-70B-Instruct-v0.4

2) https://huggingface.co/datasets/SakanaAI/gsm8k-ja-test_250-1319

3) 学習前の sarashina2.2-1b と llm-jp-3.1-1.8b は, repetition penalty を設定することで, 冗長な繰り返しが抑制され正答率が 40 ポイント以上向上した.

表 1: GSM8K 訓練データ (日本語) の問題文に対する教師モデルの出力を学習した生徒モデルの, GSM8K 評価データ (日本語) における正答率. 教師モデルは instruction tuning 済みのモデル, 生徒モデルはベースモデルを用いた.

教師モデル	生徒モデル				
	Qwen2.5-0.5B	Qwen2.5-1.5B	sarashina2.2-1b	llm-jp-3.1-1.8b	平均
Qwen2.5-7B	**33.30**	**59.12**	**47.71**	**59.77**	**49.98**
Qwen2.5-14B	32.09	55.75	46.49	58.84	48.29
Qwen2.5-32B	30.21	55.94	44.15	56.97	46.82
Qwen2.5-72B	30.96	57.53	41.35	57.44	46.82
4-shot	4.12	11.69	46.77	43.12	-

7B モデルを教師とした際に最も性能は向上し, 72B モデルを教師とした場合と比べて平均で 3.16 ポイント上回った. また, 14B モデルを教師とした場合, Qwen2.5-1.5B を除く 3 つの生徒モデルで 7B 教師に次ぐ性能を示した.

以上より, 教師モデルのサイズ増大に伴って, 教師自体の性能は向上する (付録 D 節参照) にもかかわらず, 蒸留後の生徒モデルの性能は教師サイズに対して単調に向上しないことが確認できた.

4 分析

本節では, 蒸留後の生徒モデルの性能さが生じる要因を明らかにするため, 生徒モデルの誤答パターン, 教師データの推論過程の差異の観点から分析する.

4.1 エラー分析

4.1.1 分析方法

Wei ら [4] や Wang ら [16] に倣い, 生徒の誤答を以下の 3 つのカテゴリに分類する.

(A) Semantic Misunderstanding: 問題文の意味理解や条件, 数量関係の解釈の誤りに起因する誤答.
(B) Step Missing Error: 正答に必要な中間ステップの欠落や根拠のない論理の飛躍による誤答.
(C) Calculation Error: 立式は妥当だが, 計算・代入・変形などの操作のミスによる誤答.

誤答カテゴリの判定には, GPT-4o [17] を用いる. GPT-4o には, 問題文, 生徒モデルの回答, および参照情報として GSM8K の模範回答 (英語) を与え, 上記のカテゴリのうち最も適切なものを 1 つ選択させる. また, 誤りが複数存在する場合は,「最も早い段階で発生した主要な誤り」を対象とし, 高次の誤りを優先して分類するために, (A) > (B) > (C) の順に優先順位を与える. API 利用コストの制約上, 本分析では 7B および 72B の 2 つの教師を用いて学習した生徒モデルの回答のみ評価した.

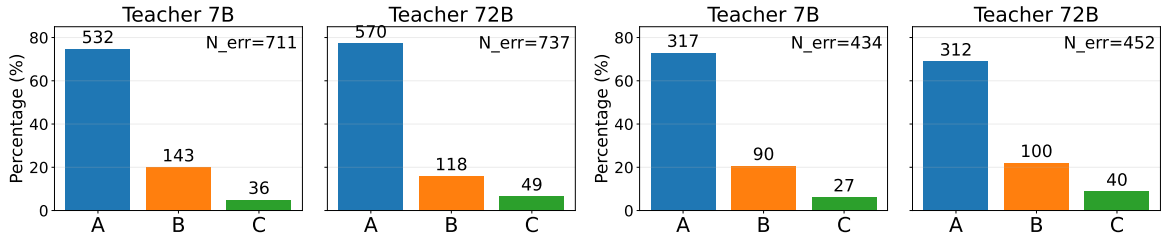
4.1.2 考察

図 2 に, 各生徒モデルの誤答カテゴリの分布を示す. 棒の高さは各モデルの誤答内訳の割合, 棒上の数値は誤答件数を表す.

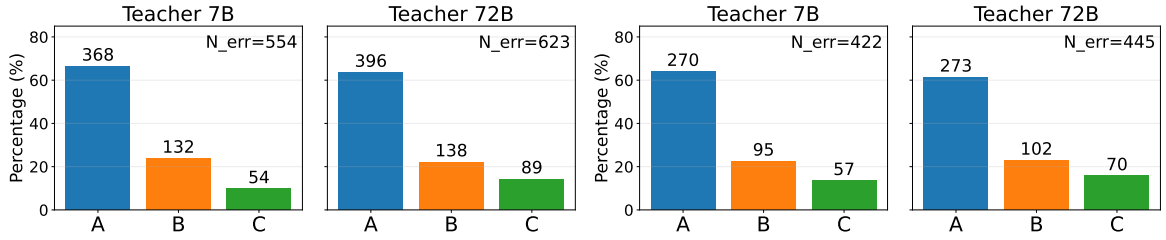
Semantic Misunderstanding すべての条件で, 最も多い誤答カテゴリであり, 誤答の大半を占めた. Qwen2.5-0.5B と sarashina2.2-1b では, 72B 教師の下で増加する一方で, Qwen2.5-1.5B および llm-jp-3.1-1.8b では大きな変化は見られなかった. このことから, 意味理解に起因する誤りに関しては, 教師サイズの影響が生徒モデルのサイズや能力に依存して現れる可能性が示唆される.

Step Missing Error すべての条件で, 2 番目に多い誤答カテゴリだった. Qwen2.5-1.5B, sarashina2.2-1b, llm-jp-3.1-1.8b においては, 72B を教師とした場合に僅かに増加するが, Qwen2.5-0.5B が生徒の場合は減少しており一貫した傾向は見られなかった.

Calculation Error すべての条件で, 最も少ない誤答カテゴリだが, 生徒モデルによらず, 72B 教師の下で一貫して増加しており, 7B 教師を蒸留した生徒が上回った主要因の一つであることが示唆される. サンプルを確認すると, 7B 教師を蒸留した生徒は, 正答に至るまでの効率は悪い一方で, 整数のみを扱うような計算ミスのしづらい手順を踏む傾向が見られた. 一方で, 72B 教師を蒸留した生徒は, 簡潔な回答を試みる過程で小数を扱うような複雑な計算を行う必要が生じ, ミスをする例が確認できた (付録表 4 に代表的な例を示す).



(a) 生徒モデル: Qwen2.5-0.5B　　(b) 生徒モデル: Qwen2.5-1.5B



(c) 生徒モデル: sarashina2.2-1b　　(d) 生徒モデル: llm-jp-3.1-1.8b

図 2: Qwen2.5-{7B, 72B}を教師とした生徒モデルの誤答の内訳. 棒の高さは各モデルにおける誤答内訳の割合, 棒上の数値は誤答件数を表す. GPT-4o を用いて, 生徒モデルの誤答を (A) Semantic Misunderstanding, (B) Step Missing Error, (C) Calculation Error のいずれかに分類させた.

4.2　教師データのトークン長の分析

エラー分析では, 72B 教師を蒸留した生徒において, まとまった中間計算を含む推論形式が観察された. これは教師データの推論形式に原因があると推測し, 各教師データのトークン数を調査した.

図 3 に, 各教師モデルの生成文を Qwen2.5 のトークナイザを用いてトークン化した際の, 1 サンプルあたりの平均トークン数を示す. 最も蒸留効果の高かった 7B モデルによる平均トークン数が最も多く, 72B モデルと比べて平均約 30 トークン多かった. また, sarashina2.2 および llm-jp-3.1 のトークナイザを用いても同様の傾向が観察された.

したがって, 7B 教師は推論過程における記述量が他のサイズの教師モデルと比べて相対的に大きく, 中間計算が省略されにくい形式で生成されていた可能性がある. この結果は, 72B 教師の下で Calculation Error が一貫して増加したという前節の誤答分析と整合しており, 教師の推論形式の違いが蒸留後の性能に影響した可能性が示唆される.

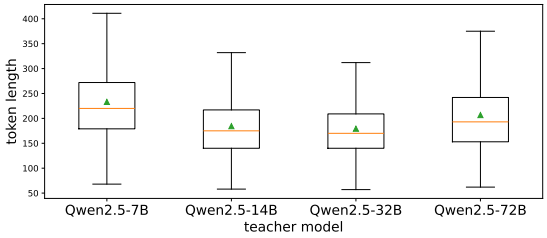


図 3: Qwen2.5-{7B, 14B, 32B, 72B}が生成した教師データのトークン数の箱ひげ図.

5　おわりに

本研究では, 日本語の算術タスクにおいて, 異なるサイズの教師モデルを用いたときの, 蒸留の効果を比較・検証した. 実験では, 英語を対象とした先行研究と同様に, 日本語においても, 小規模な教師モデルを用いた場合に蒸留の効果が最大化された. また, 分析では, 大規模な教師モデルの下で学習した生徒モデルでは計算ミスが増加する傾向が見られ, 教師データにおける推論形式と関連している可能性について考察した. 今後は, 異なるモデルやデータセットを用いて, 同様の傾向が成り立つか検証し, 教師データの推論形式を含め, 蒸留に効果的な学習設定を明らかにしていきたい.

参考文献

[1] OpenAI. Openai o1 system card, 2024.

[2] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. **arXiv preprint arXiv:2501.12948**, 2025.

[3] Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025.

[4] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. **Advances in neural information processing systems**, Vol. 35, pp. 24824–24837, 2022.

[5] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[6] 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.

[7] Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 2665–2679, Toronto, Canada, July 2023. Association for Computational Linguistics.

[8] Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners, 2025.

[9] Xinghao Chen, Zhijing Sun, Guo Wenjin, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, and Xiaoyu Shen. Unveiling the key factors for distilling chain-of-thought reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, **Findings of the Association for Computational Linguistics: ACL 2025**, pp. 15094–15119, Vienna, Austria, July 2025. Association for Computational Linguistics.

[10] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. **arXiv preprint arXiv:2110.14168**, 2021.

[11] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[12] 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.

[13] Swallow LLM Team. Llama 3.3 swallow 70b v0.4. https://swallow-llm.github.io/llama3.3-swallow.ja.html, 2025. accessed 2026-01-09.

[14] 高瀬翔, 李凌寒, SB Intuitions Pretraining Team. Sarashina2.2：数学・コーディングタスクの性能を向上させた日本語言語モデル. https://www.sbintuitions.co.jp/blog/entry/2025/03/06/112144, 2025. accessed 2026-01-09.

[15] LLM jp Team. Llm-jp: A cross-organizational project for the research and development of fully open japanese llms, 2024.

[16] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 2609–2634, Toronto, Canada, July 2023. Association for Computational Linguistics.

[17] OpenAI. Gpt-4o system card, 2024.

A　計算機環境

教師データの生成, 生徒モデルの学習は, NVIDIA A100 GPU を用いて行った. Qwen2.5-70B による教師データ生成時と Llama 3.3 Swallow 70B による GSM8K の和訳時のみ 4 枚を使用し, それ以外はすべて 1 枚のみを使用した.

B　プロンプト

以下に本研究で用いたプロンプトを示す. 教師データ生成時と生徒モデルによる回答生成時とで, 同一のプロンプトを用いた.

回答生成時のプロンプト

問題: {問題文}
ステップごとに考えて答えを導いてください。
解答を [answer] とするとき以下のフォーマットで出力してください。
解答: [answer]

C　学習時の設定

生徒モデルの学習時の各ハイパーパラメータの設定を表 2 に示す.

表 2: 学習時のハイパーパラメータ

ハイパーパラメータ	値
bfloat16	true
学習率	1.0×10^{-5}
学習率スケジューラ	linear
エポック数	2
per device batch size	8
gradient accumulation steps	1
max gradient norm	1.0

D　教師モデルの正答率

教師モデル Qwen2.5-{7B, 14B, 32B, 72B}の評価データにおける正答率を表 3 に示す. 教師モデルのサイズ増大に伴って, 正答率は単調に向上することが確認できる.

表 3: 教師モデルの評価データにおける正答率.

教師モデル	正答率
Qwen2.5-7B	79.98
Qwen2.5-14B	84.47
Qwen2.5-32B	86.16
Qwen2.5-72B	89.43

E　サンプル

表 4: 72B 教師を学習した生徒が Calculation Error, 7B 教師を学習した生徒が正答したサンプル.

質問文: スヴェトラーナはマフィン 1 個を作るのに、小麦粉 5 スプーン、砂糖 3 スプーン、塩 0.25 スプーンが必要です。16 個のマフィンを作るためには、乾燥材料を合計で何スプーン必要でしょうか？

72B を教師とした生徒: まず、1 個のマフィンを作るのに必要な乾燥材料の量を確認します。それは小麦粉 5 スプーン、砂糖 3 スプーン、塩 0.25 スプーンです。これらを合計すると、1 個のマフィンを作るのに必要な乾燥材料の総量は:

$$5 + 3 + 0.25 = 8.25 \text{ スプーン}$$

次に、16 個のマフィンを作るために必要な乾燥材料の総量を計算します。1 個のマフィンに必要な量が 8.25 スプーンなので、16 個のマフィンには:

$$16 \times 8.25 = 130 \text{ スプーン}$$

したがって、16 個のマフィンを作るためには、乾燥材料を合計で 130 スプーン必要です。
解答: 130

7B を教師とした生徒: まず、1 個のマフィンを作るのに必要な乾燥材料の量を確認します。

- 小麦粉: 5 スプーン
- 砂糖: 3 スプーン
- 塩: 0.25 スプーン

次に、16 個のマフィンを作るためには、それぞれの乾燥材料の量を 16 倍します。

- 小麦粉: 5 スプーン × 16 = 80 スプーン
- 砂糖: 3 スプーン × 16 = 48 スプーン
- 塩: 0.25 スプーン × 16 = 4 スプーン

最後に、これらの量を合計します。

80 スプーン + 48 スプーン + 4 スプーン = 132 スプーン

解答: [132]