

テキスト平易化事例を説明する最小編集操作列の自動生成

山口大地¹ 宮田玲¹ 藤田篤² 梶原智之³ 佐藤理史¹

¹ 名古屋大学大学院工学研究科 ² 情報通信研究機構

³ 愛媛大学大学院理工学研究科

yamaguchi.daichi.e4@es.mail.nagoya-u.ac.jp

概要

テキスト平易化システムの評価手法が種々提案されてきたが、システムを詳細に分析できる評価手法の開発は進んでいない。我々はそのような評価の自動化に向け、テキスト平易化事例を説明する最小編集操作列の自動生成という新しいタスクを提案する。本稿では、このタスクを定式化するとともに、このタスクを実現するための我々の提案手法について説明する。そして、提案手法の一部を実装し、最小編集操作列を半自動で生成する実験を行った結果について報告する。

1 はじめに

テキスト平易化 (text simplification, TS) はテキストの主要な情報を保持しつつ、語彙や構造の複雑さを軽減するタスクである [1]。TS は第二言語学習者や子どもの学習支援に利用でき、また機械翻訳や文章要約のような自然言語処理タスクの改善のための前処理として利用することが期待できる [2, 3]。

TS システムの評価には自動評価と人手評価が広く使われてきた。自動評価では、SARI [4] や BLEU [5]、FKGL [6] などの評価指標が用いられてきた。人手評価では、流暢性、妥当性、平易度の観点から、それぞれ3~5段階での評価が行われてきた [7, 8, 9]。しかし、これらの評価はいずれもシステムの振る舞いや特徴を一つの評価尺度に集約してしまうものであるため、評価結果に基づいてシステムの振る舞いや特徴を説明することは難しい。TS のエンドユーザーは難解文を理解できないことが想定されるため、TS システムがどのような編集操作を適用したかを説明することは重要である。従って、そのような説明を可能にする評価手法が必要である。

TS システムが適用した編集操作を説明する評価方法は分析的評価と呼ばれる [10]。これも人手評価の一つではあるが、一連の編集操作の適用前後の文

難解文

Scientists who study the brain have found it is tough.

編集操作1 ↓ Scientists → Researchers

Researchers who study the brain have found it is tough.

編集操作2 ↓ φ → that

Researchers who study the brain have found that it is tough.

編集操作3 ↓ Researchers who study the brain

平易文 → Brain researchers

Brain researchers have found that it is tough.

図 1: 最小編集操作列の生成例

対に対して単一のスコアを与えるような人手評価とは異なる。山口らの手法 [10] では、図 1 に示すように、難解文と平易文の対を最小編集操作列に分解し、各編集操作を平易化方略体系に基づいて分類する。平易化方略体系は、語や句といった文法的な単位に対する形式的な操作である表層方略と、平易化の観点からの意味・内容の変更操作である内容方略からなる。例えば、図 1 の “Scientists ⇒ Researchers” の編集は、{ 表層方略：語置換, 内容方略：上位語への言い換え } に分類される。

分析的評価における編集操作の分類は TS タスクに依存する。しかし、TS 事例を説明する最小編集操作列の自動生成の手法は、TS に限らず、機械翻訳の前編集や後編集、文章要約、文法誤り訂正などの分析にも利用できる可能性がある。そこで我々は、分析的評価の自動化に向けて、他のタスクにも利用できる可能性のある最小編集操作列の自動生成に取り組んでいる。本稿では、このタスクの詳細な説明と我々のアプローチについて報告する。

2 問題の定式化

所与の文対 (X_a, X_b) に対し、次の条件を満たす編集操作列 $\mathbf{E} = (E_1, \dots, E_n)$ を (X_a, X_b) に対する最小編集操作列と呼ぶ。

$$X_i = E_i(X_{i-1}) \text{ s.t. } d(X_{i-1}, X_i) = 1$$

ただし、 $X_0 = X_a$ 、 $X_n = X_b$ である。 X_i は文法的な適切性を保つ文である。 E_i は引数の入力文 X_{i-1} のある箇所を別の表現に編集する関数であり、返り値の文 X_i を X_{i-1} に対する編集適用文と呼ぶ。 $d(A, B)$ は A から B を生成するために必要な最小編集操作の回数を返す関数である。なお、最小編集操作とは、書き換え後の文法的な適切性を保ちつつ、これ以上分解できない編集操作とする [11]。本稿でも、この操作的な定義を踏襲し、最小性に関する概念的・言語学的定義は与えないが、原則として、文字単位の置換・追加・削除ではなく、文法的な単位での編集操作とする。

最小編集操作列の自動生成タスクでは、難解文と平易文の対を入力として、最小編集操作列を出力する。形式的には、難解文と平易文の対 (X_{comp}, X_{simp}) が与えられたとき、以下で表される最小編集操作列 \hat{E} を 1 つ生成するタスクとする。

$$\hat{E} = \arg \min_{E \in \mathcal{E}} \text{length}(E)$$

$$\text{where } \mathcal{E} = \{E \mid X_{comp} = E_n(\dots E_1(X_{simp}))\}$$

length は系列の長さを返す関数である。すなわち、 X_{comp} から X_{simp} への最小編集操作列のうち、編集とその逆の編集のようなループを含む冗長な系列を避け、長さが最小のものを生成する。

このタスクに取り組む上で特に考慮すべき点として、以下の 3 つが挙げられる。なお、これ以降、最小編集操作を “ $A \Rightarrow B$ ” と表し、2 つ以上の最小編集操作の含まれた編集操作を “ $A \Rightarrow^* B$ ” と表す。

I. 編集操作の依存関係 編集操作には依存関係、すなわち編集操作列の順序における優先性が存在する可能性がある。例えば、“tell me the truth \Rightarrow^* state the truth” は “tell me the truth \Rightarrow tell the truth” と “tell the truth \Rightarrow state the truth” で構成され、後者は前者に依存する。このような依存関係がある場合、一方の編集操作を生成しなければ他方は生成できない。

II. 編集対象箇所の入れ子 編集対象箇所が重複し、入れ子になる場合がある。図 1 の “Scientists who study the brain \Rightarrow^* Brain researchers” は “Scientists \Rightarrow Researchers” と “Researchers who study the brain \Rightarrow Brain researchers” で構成され、編集対象箇所が入れ子になる。編集対象箇所の入れ子は I の特徴に似ているが、複数の順序が存在しうる点が異なる¹⁾。

III. 表層上に現れない編集 最小編集操作が X_{comp} と X_{simp} の表層上に現れない語によって構成

1) 例に挙げた編集操作は “Scientists who study the brain \Rightarrow Brain scientists” と “scientists \Rightarrow researchers” でも実現できる。

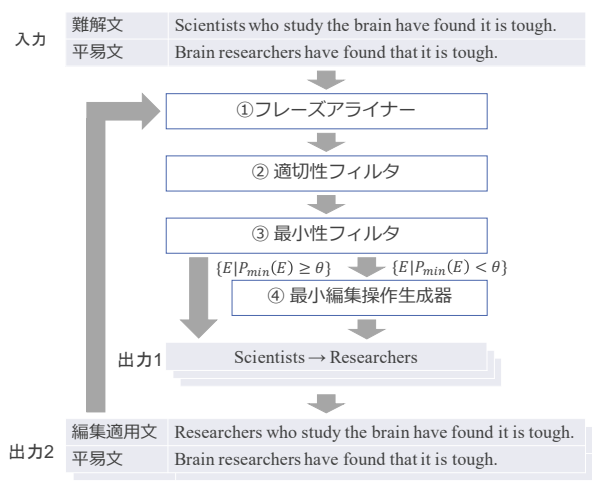


図 2: 最小編集操作列の生成手法

される場合がある。例えば、“start using \Rightarrow^* do” は “start using \Rightarrow start doing” と “start doing \Rightarrow do” から構成され、“doing” は表層上に現れない。

3 最小編集操作列の生成手法

3.1 考えられるアプローチ

このタスクに対するアプローチとして、アライメント手法と生成的手法の 2 つが考えられる。

アライメント手法 アライメント結果に基づいて、組み合わせ最適化問題として解くことができる。例えば、“multiple solutions \Rightarrow^* several ways” は “multiple \Rightarrow several” と “solutions \Rightarrow ways” で構成できるということをアライメント結果の組み合わせで解けば良い。この方法は、アライメント結果さえ正しければ、確実に編集操作を生成することができる。しかし、先述した 3 つの特徴を持つ編集が存在する場合、解くことができない。図 1 のように最小編集操作の生成とアライメントを繰り返し行うことで I と II の特徴を持つ編集操作を生成することはできるが、原理的に III の特徴を持つ編集操作を生成することはできない。

生成的手法 TS システムで反復改良法 [12] を用いて、難解文を少しずつ平易文に近づけることも考えられる。この方法は、原理的には全ての編集操作を生成することができる。しかし、TS システムが生成する文が入力文に対して最小編集操作を適用したものであることは保証できない。

原文と目標文の 2 文の間を補完する文を生成するテキストモーフィングというタスクが存在する [13] が、最小編集操作の制約は考慮されていない。

3.2 提案手法

本稿では、アライメント手法と生成的手法を組み合わせた手法を提案する。具体的には、図2の4つの機構で最小編集操作を逐次的に生成する。アライメント手法(図2:①~③)を主に用い、生成的手法(図2:④)でアライメント手法を補う。以下、各機構について説明する。

①**フレーズアライナー** 難解文と平易文の対を入力し、フレーズアライメントを出力する。ここで、得られたアライメントを編集操作候補とする。編集操作としては文法的な単位での操作のみを認める(2節)ため、アライメントは文法的な単位に則った出力をする必要がある。

②**適切性フィルタ** 編集操作候補の集合を入力し、編集操作として適切な編集操作候補のみを出力する。ここで、適切な編集操作とは、適用した時に文法的な適切性が保たれる編集操作である。適切である確率 P_{app} が閾値よりも高い候補を適切とする。

③**最小性フィルタ** 編集操作の集合を入力し、最小である確率 P_{min} の高い上位 M 個の編集操作を出力する。

④**最小編集操作生成器**²⁾ 編集操作を I 個入力し ($I \leq M$)、最小編集操作を O 個生成する ($O \leq I$)。最小編集操作生成器では、アライメントで原理的に生成できない編集操作を生成できる可能性がある。

4 提案手法の実装

4.1 各機構の実装の詳細

①**フレーズアライナー** Enju³⁾の出力に基づいて文法的な単位でのアライメントを出力する Arase and Tsujii [14]⁴⁾の手法を利用する。

②**適切性フィルタ** BERT [15]⁵⁾を微調整し、実装する。適切性フィルタでは、編集操作候補を独立に処理し、条件を満たす編集操作のみを出力する。図3に適切性フィルタの構成を示す。この構成は、難解文と平易文の差分と対象フレーズの差分を比較すれば、編集操作の適切性を判定できるという直感に基づいている⁶⁾。適切性フィルタに

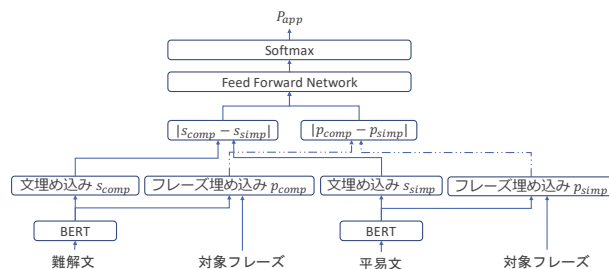


図3: 適切性フィルタの構成

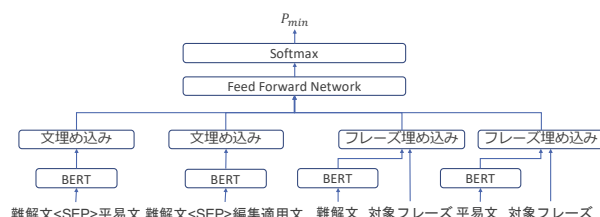


図4: 最小性フィルタの構成

は難解文と平易文、難解文と平易文の編集箇所を表す対象フレーズを入力する。文埋め込みとして文頭の CLS トークンに対して BERT が出力するベクトルを使用する。フレーズ埋め込み \mathbf{p} としては、対象フレーズの文中の位置を表す2値ベクトル系列 $\mathbf{m} (= (m_1, \dots, m_L), m_i \in \{0, 1\})$ を用い、対象フレーズに含まれるトークンの埋め込み \mathbf{h} の平均 $\mathbf{p} = \frac{1}{\sum_{i=1}^L m_i} \sum_{i=1}^L m_i \mathbf{h}_i$ を用いる。そして、難解文と平易文の文埋め込みの差分とフレーズ埋め込みの差分を直結し、全結合層に入力する。

③**最小性フィルタ** 図4に最小性フィルタの構成を示す。最小性の判断をするためには、難解文と平易文の関係、難解文と編集適用文の関係が重要である⁶⁾。最小性フィルタには、難解文と平易文を SEP トークンで結合した文と、難解文と編集適用文を SEP トークンで結合した文、難解文と平易文の編集箇所のフレーズ埋め込みを入力する。フレーズ埋め込みは適切性フィルタと同じ方法で獲得する。これら全ての埋め込みを直結し、全結合層に入力する。

4.2 フィルタの訓練と評価

各フィルタの訓練、および評価のデータは次の手順で作成した。まず Newsela⁷⁾の Popular カテゴリの3文書から難解文と平易文の対297件を抽出し、Arase and Tsujii [14]の手法を用いてフレーズアライメントを行って編集操作候補を抽出した。次に各編集操作候補に対して、第1著者が適切性を判定し、適切と判断されたもののみについて最小性を判定し

2) 検討段階のため以降の節では扱わない。

3) <https://github.com/mynlp/enju>

4) https://github.com/yukiar/phrase_alignment_cted

5) <https://huggingface.co/bert-base-cased>

6) いくつかの構成を試し、検証データで最も性能の良い構成を選択した。

7) <https://newsela.com/data>

表 1: 各フィルタに使用したデータの分布

(a) 適切性フィルタ			(b) 最小性フィルタ		
	↯適切	適切		↯最小	最小
訓練	2998	994	訓練	321	673
検証	731	219	検証	86	133
評価	915	270	評価	83	187

表 2: 各フィルタの評価結果

(a) 適切性フィルタ			(b) 最小性フィルタ		
ラベル	再現率	精度	ラベル	再現率	精度
↯適切	0.88	0.96	↯最小	0.57	0.96
適切	0.87	0.69	最小	0.99	0.84

た。このようにして得られたデータの分布を表 1 に示す⁸⁾。

評価の際には、適切性フィルタ、および最小性フィルタの閾値は 0.50 とした。両フィルタの評価結果を表 2 に示す。結果から、両フィルタは分類器としてはある程度機能していると考えられる。

5 最小編集操作列の生成実験

3~4 節で述べた機構で、最小編集操作列の生成がどの程度できるかを検証した。

5.1 実験設定

図 2 の①~③の機構のみを用いて行った。最小編集操作を 1 つ生成し、適切性と最小性を都度、人手で確認した。正しい最小編集操作が生成できた場合は、繰り返し同じ処理を行った。繰り返し処理は、正しい最小編集操作が生成できなかった場合、あるいは編集適用文と平易文が一致したときに終了した。適切性フィルタの閾値は 0.50 とした。最小性フィルタは最も P_{min} の高い編集操作を 1 つのみ出力し ($M=1$)、ビーム探索は行わないこととした。

評価データは、Newsela の Popular カテゴリから新たに 7 文書を収集して作成した。まず TER [16] が 0.5 以下の難解文と平易文の対を 0.1 刻みに 10 文対ずつ合計 50 文対抽出した。そして、各文対に対して第 1 著者が最小編集操作列を与えた。これらの TS 事例は最大 4 回の編集操作が適用されていた。

5.2 結果

50 文対中 22 文対において、最小編集操作列の生成に成功した。表 3 に編集数ごとに正しい最小編集操作列を生成できた割合を示す⁹⁾。編集数が 3 以上

8) 訓練時のハイパーパラメータ等については付録 A を参照されたい。

9) 編集数・編集距離と成功率の関係については付録 B を参照されたい。

表 3: 編集数ごとの成功率

編集数	成功率
1	0.57 (12/21)
2	0.47 (9/19)
3	0.14 (1/7)
4	0.00 (0/3)

表 4: エラーの原因とその件数

エラーの原因	件数
フレーズアライナー	3
適切性フィルタ	16
最小性フィルタ	4
その他	5

の事例では、ほとんど成功していない¹⁰⁾。

5.3 エラー分析

表 4 にエラーの原因とその件数を示す¹¹⁾。フレーズアライナーの出力に最小編集操作が含まれていない場合は、フレーズアライナーが原因のエラーとする。フレーズアライナーのエラーに対しては、それらを補う前処理や後処理を行う必要がある。適切性フィルタが原因のエラーは全体の半分以上を占めている。16 件のうち 9 件は P_{min} の上位 5 つの中に最小編集操作が含まれていた。従って、ビーム探索によって適切性フィルタのエラーを補うことができると期待できる。最小性フィルタが原因のエラーのほとんどは訓練データ内に存在しない編集操作であった。最小性フィルタの訓練事例は 994 件と少ないため、データの拡張によって性能を改善できる可能性もある。

6 おわりに

本稿では、最小編集操作列の生成という新しいタスクについて述べた。そして、我々はこのタスクを 4 つの機構で行うことを提案した。一部を実装して行った半自動的な実験の結果、50 文対中 22 文対に対する最小編集操作列の生成に成功した。

今後はビーム探索の実装やデータの拡張などの改良を加えていく。また今後の展望として、以下が挙げられる。

- 最小編集操作生成器の作成
- TS システムの出力への適用
- 編集操作の分類器の作成
- TS システム自体の説明可能性の向上

10) 実際のシステム出力については付録 C を参照されたい。

11) これらのエラーのほとんどが 1 つ目の最小編集操作の生成時に発生した。

謝辞

本研究は JSPS 科研費（課題番号：19H05660）および KDDI 財団調査研究助成（課題名：平易な文化財情報を執筆・翻訳する技術）の支援を受けた。

参考文献

- [1] Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. Data-driven sentence simplification: Survey and benchmark. **Computational Linguistics**, Vol. 46, No. 1, pp. 135–187, 2020.
- [2] Advait Siddharthan, Ani Nenkova, and Kathleen McKeown. Syntactic simplification for improving content selection in multi-document summarization. In **Proceedings of the 20th International Conference on Computational Linguistics (COLING)**, pp. 896–902, 2004.
- [3] Sanja Štajner and Maja Popovic. Can text simplification help machine translation? In **Proceedings of the 19th Annual Conference of the European Association for Machine Translation (EAMT)**, pp. 230–242, 2016.
- [4] Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing statistical machine translation for text simplification. **Transactions of the Association for Computational Linguistics (TACL)**, Vol. 4, pp. 401–415, 2016.
- [5] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In **Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 311–318, 2002.
- [6] J. Peter Kincaid, Robert P. Fishburne Jr, Richard L. Rogers, and Brad S. Chissom. Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy enlisted personnel. Technical report, Institute for Simulation and Training, University of Central Florida, 1975.
- [7] Sanja Štajner and Sergiu Nisioi. A detailed evaluation of neural sequence-to-sequence models for in-domain and cross-domain text simplification. In **Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)**, pp. 3026–3033, 2018.
- [8] Elior Sulem, Omri Abend, and Ari Rappoport. Simple and effective text simplification using semantic and neural methods. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 162–173, 2018.
- [9] Suha S. Al-Thanyyan and Aqil M. Azmi. Automated text simplification: A survey. **ACM Computing Surveys**, Vol. 54, No. 2, pp. 1–36, 2021.
- [10] 山口大地, 島田紗裕華, 宮田玲, 佐藤理史. テキスト平易化システムの分析的評価のための平易化方略体系の構築. 言語処理学会第 28 回年次大会発表論文集 (NLP), pp. 506–510, 2022.
- [11] Rei Miyata and Atsushi Fujita. Understanding pre-editing for black-box neural machine translation. In **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)**, pp. 1539–1550, 2021.
- [12] Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. Corpora generation for grammatical error correction. 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) (NAACL-HLT)**, pp. 3291–3301, 2019.
- [13] Shaohan Huang, Yu Wu, Furu Wei, and Ming Zhou. Text morphing. **CoRR**, Vol. abs/1810.00341, pp. 1–12, 2018.
- [14] Yuki Arase and Jun'ichi Tsujii. Compositional phrase alignment and beyond. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 1611–1623, 2020.
- [15] 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) (NAACL-HLT)**, pp. 4171–4186, 2019.
- [16] Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In **Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers (AMTA)**, pp. 223–231, 2006.

A 各フィルタの訓練詳細

各フィルタのハイパーパラメータを表5に示す。訓練時には、BERTのパラメータも更新した。

表5: 各フィルタのハイパーパラメータ

(a) 適切性フィルタ		(b) 最小性フィルタ	
損失関数	交差エントロピー損失	損失関数	交差エントロピー損失
最適化手法	AdamW	最適化手法	AdamW
ドロップアウト率	0.10	ドロップアウト率	0.10
学習率	2.0×10^{-6}	学習率	3.0×10^{-6}
バッチサイズ	8	バッチサイズ	8
エポック数	9	エポック数	8
全結合層のサイズ	1,536	全結合層のサイズ	3,072

B 編集数・編集距離と成功率の関係

表6に編集数・編集距離と成功率の関係を示す¹²⁾。編集距離はTERの計算時に導出されるものを使用した。編集距離が大きくなるにつれて、成功率は下がっていた。しかし、編集数が1の事例は編集距離にかかわらず半分以上成功していた。

表6: 編集数・編集距離と成功率

	編集距離			
	2	3	4	≥ 5
編集数 1	0.75 (6/8)	0.50 (4/8)	0.50 (1/2)	0.67 (2/3)
2	0.50 (3/6)	0.43 (3/7)	0 (0/3)	0.67 (2/3)
3		0 (0/1)	0 (0/1)	0.20 (1/5)
4			0 (0/0)	0 (0/3)

C システム出力

表7に最小編集操作列の生成に成功した事例を示す¹³⁾。生成した編集操作に該当する部分を同じ色で示している (e.g., “Scientists \Rightarrow researchers”)。提案した構成で入れ子構造に対しても、最小編集操作列を生成できていた。表8に最小編集操作列の生成に失敗した事例を示す。

表7: 最小編集操作の生成に成功した事例

X_{comp}	Scientists who study the brain have found that, when we learn, new connections form between neurons.
X_1	researchers who study the brain have found that, when we learn, new connections form between neurons.
X_2	researchers who study the brain have found that when we learn, new connections form between neurons.
X_{simp}	Brain researchers have found that when we learn, new connections form between neurons.

表8: 最小編集操作列の生成に失敗した事例

エラーの原因	例
フレーズアライナー	X_{comp} Learning is like exercise for your brain. X_{simp} This means that learning is like exercise for your brain.
適切性フィルタ	X_{comp} Dr. Carol Dweck is a psychology professor at Stanford University in California. X_1 Dr. Carol Dweck is a psychology teaches at Stanford University in California. X_{simp} Dr. Carol Dweck teaches at Stanford University in California.
最小性フィルタ	X_{comp} The more you work it out, the stronger and smarter your network of neurons will become. X_1 The more you work it out, the stronger the connections between your neurons become. X_{simp} The more you work it out, the stronger the connections between your neurons become.
予期せぬ動作	X_{comp} And learning a new skill is hard, but won't it be worth it? X_1 And learning a new skill is hard, but won't it be worth it? X_{simp} And learning a new skill is hard, but it will be worth it.

12) 編集数は常に編集距離以下になるため、編集数が編集距離より大きい欄は空白にしている。

13) 表7、表8中の難解文と平易文の例は <https://newsela.com/read/lib-mindset-matters/id/39794/> より引用した。