同一の原文書に対する複数の翻訳文書間で対応する言語単位対の自動抽出

本田友乃 [1]　藤田篤 [2]
[1] 東京大学大学院教育学研究科　[2] 情報通信研究機構
tomono20@g.ecc.u-tokyo.ac.jp　atsushi.fujita@nict.go.jp

概要

機械翻訳の研究や産業翻訳において、誤りのない翻訳間の品質における差異を分析的に記述することは重要である。同一の原文書に対する異なる翻訳間に観察される差異を言語表現に基づいて記述するための枠組みは、翻訳文書対の分割と分割された単位対の分類という 2 つの工程からなるスキームとして整理されている [1] ものの、差異を効率的に分析するためには、分割・分類の自動化が課題である。そこで我々は、同一の原文書に対する複数の翻訳文書間で対応する言語単位対の自動抽出に取り組んだ。

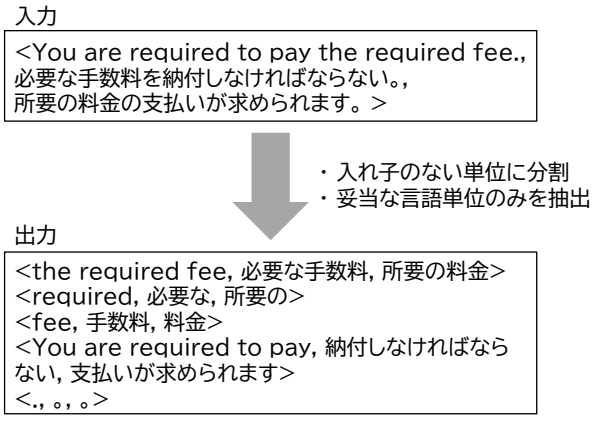

図 1　自動抽出タスクにおける入出力の例

1　はじめに

機械翻訳 (MT) 技術の発展に伴い、産業翻訳においても MT の利用、特に MT の出力 (以下、MT 訳) を人手で編集するポストエディット (MT+PE) の導入 [2] が加速している。ポストエディットの作業は MT 訳に依存するため、MT+PE は人手による翻訳 (HT) と比べて劣る [3] という指摘がある。一方で、品質の観点から MT+PE は HT と遜色ないという結論を出している研究が多い [4] という分析もある。しかし、言語表現に基づいて HT と MT+PE との差異を分析的に記述する研究は行われておらず、MT+PE において到達可能な品質も明らかにされていない。

翻訳の品質評価では、翻訳を可能な限り体系的かつ客観的に診断することが重要とされており [5]、体系性と客観性を保った翻訳評価の枠組みとして、Multidimensional Quality Metrics (MQM) [6] をはじめとしたエラー分析のスキームが多く提案されている。しかし、このような評価スキームはあくまで誤りに着目して作られているため、誤りのない翻訳間に生じている品質上の差異を分析的に診断するという形での評価は行われていない。現状では MT 訳には誤りが多く、品質面では HT には至らない [7, 8] ものの、MT の性能の向上に鑑みると、今後は、誤り以外の観点から MT 訳を診断・評価することも必要である。したがって、HT、MT、MT+PE などの産出工程の異なる翻訳を対象として、既存の尺度では十分に捉えられていない翻訳間の差異を分析的に記述することは、MT 技術の発展の観点からも、産業翻訳における翻訳技術の導入の観点からも重要である。

こうした課題を背景に、原文書を同じくする独立に産出された異なる翻訳文書間の差異を分析的に記述するためのスキーム (以下、差異分析スキーム) が、開発 [9, 1] されてきた。差異分析スキームは一般に、(1) 所与の翻訳文書対を対応する言語単位対へと階層的に分割し、(2) 得られた個々の言語単位対の差異を分類する、という 2 つの工程からなる。ただし、既存のスキームは分割・分類の作業を全て人手で行うことを想定して設計されているため、多数の翻訳文書対に適用して分析することは作業負担を考慮すると困難である。

そこで我々は、差異分析スキームにおける分割の工程を「同一の原文書に対する複数の翻訳文書間で対応する言語単位対の自動抽出」というタスクとして定式化し、その自動化に取り組んだ。

2 タスクの定義

差異分析スキーム [1] における分割の工程では、言語単位 (分析単位) 対の定義と分割手順に従い、分割元の単位対を入れ子のない要素に分割する[1)]という制約のもと、所与の翻訳文書対から分析単位対へと階層的に分割することで、翻訳文書対に含まれる単位対を網羅的に抽出する。翻訳の原文書 (Source Document) を SD、当該 SD に対する異なる翻訳文書 (Target Document) を各々 TD_1、TD_2 と記す。そして、同一の原文書に対する複数の翻訳文書間で対応する言語単位対の自動抽出を次のように定式化する。

入力: $\langle SD, TD_1, TD_2 \rangle$ の組

出力: 入力に内包されている階層的な言語単位の組の集合 $\{(s, t_1, t_2) \mid s \sqsubset SD, t_1 \sqsubset TD_1, t_2 \sqsubset TD_2, s \equiv t_1 \equiv t_2\}$

ここで $x \sqsubset X$ は x が文書 X に内包される要素であること、記号 $\equiv$ は当該文書において文書要素が対応していることを表す。段落よりも大きい単位など、対応していることが自明な要素の組 $\langle s \sqsubset SD, t_1 \sqsubset TD_1, t_2 \sqsubset TD_2 \rangle$ を入力としても良い。分析対象はあくまで翻訳文書間の差異であるため、入出力における SD 及び $s \sqsubset SD$ は任意である。

3 使用するデータ

本研究では、産業翻訳で使用される翻訳文書を扱った MultiEnJa[2)] のデータセットから 2 文書を選択し、1 件を手法の検討のための開発用データ、もう 1 件を提案手法の性能評価のための評価用データとして用いた。各データの基礎統計を表 1 に示す[3)]。正解データは、開発用・評価用ともに差異分析スキームの分析単位の定義と分割手順に従って作成した。評価用データについては、Honda ら [1] がスキームの検証のために行った複数人による分析の結果を参照して正解を定めた。

4 提案手法

人手での分割は、$\langle SD, TD_1, TD_2 \rangle$ の 3 つ組を起点とするトップダウンな処理である。これに対して本研究では、語の対応から大きい単位対をボトムアップに組み合わせるアプローチについて検討した。本研究での提案手法は、1) 単語対の取得、2) フレーズ対の取得、3) フレーズ対の組み合わせ、4) 後処理の 4 段階の手順からなる[4)]。

表 1 自動抽出に使用した文書の基礎統計

データの種別	単位対数	文数	語数 (トークン数)		
			SD	HT	MT+PE
開発用データ	328	37	378	520	515
評価用データ	434	47	421	591	570

なお、入力となる翻訳文書対に対して、以下の前処理を行うこととした。

- 文書間の文単位での対応づけ
- メールアドレスと URL の抽象化
- 分かち書き[5)]

4.1 1) 単語対の取得

まず，次の 4 種類の方法で単語対を同定する[6)]。

(a) SD を介した TD 間の単語対 (WSPAlign [12] を使用)
(b) TD 間の単語対 (WSPAlign [12] を使用)
(c) TD 間の単語対 (OTAlign [13] の UOT を使用)
(d) TD 間の単語対 (OTAlign [13] の POT を使用)

(a) SD を介した TD 間の単語対 (WSPAlign を使用) まず、$\langle SD, TD_1 \rangle$ と $\langle SD, TD_2 \rangle$ の各文書対から、WSPAlign を用いて SD–TD 間の単語対を得る。SD → TD、TD → SD の各方向について WSPAlign を適用して単語対を取得し、その結果を、Nagata ら [14] の bidi-avg のアルゴリズムを用いて対称化する。そして、SD 中の同じ単語に対応づけられた TD_1、TD_2 の単語を対応づけて、TD 間の単語対を得る。

(b) TD 間の単語対 (WSPAlign を使用) SD を介さずに、$\langle TD_1, TD_2 \rangle$ に対して WSPAlign を直接適用し、

1) 「入れ子のない要素」に分割する操作とは、分割元の分析単位の統語的な構造に従って、(A) 分割後の要素同士に重なりがないように分割する、(B) できるだけ大きな単位を抽出する、という 2 つの条件を満たす操作を表す。
2) https://github.com/tntc-project/MultiEnJa
3) SD の語数は NLTK [10] を、TD (MT+PE と HT) の語数 (形態素) は MeCab 及び IPA 辞書 [11] を用いて計測した。
4) 原理的には、WSPAlign [12] を用いてフレーズ対を直接取得することも可能である。ただし、単語アラインメントのみに基づいて学習されたモデルはフレーズ対の抽出性能が低く、ファインチューニングに使用可能なデータも現時点では存在しないため、本稿では扱わない。また、提案手法の亜種として、単語アラインメントを組み合わせた上でフレーズ対を取得するという手順も考えられるが、開発用データに対する性能が低かったため、報告を割愛する。
5) SD に対しては NLTK [10] を、TD に対しては MeCab 及び IPA 辞書 [11] を使用して分かち書きを行った。ただし、分かち書きの後、正解データと照らし合わせて正解データと単位が異なる箇所は、人手による修正を施した。
6) 単語アラインメントツールの設定は、付録 A を参照されたい。

TD 間の単語対を得る。WSPAlign の設定は (a) と同じとした。

(c)・(d) TD 間の単語対 (OTAlign の UOT、または POT を使用) (b) と同じく、SD を介さずに、$\langle TD_1, TD_2 \rangle$ の単語対を直接得る。差異分析スキームが「対応なし」の単位対も分析対象としていることをふまえ、(c)・(d) では、null alignment に対応可能な OTAlign の UOT (unbalanced OT) と POT (partial OT) を単語アラインメントツールとして利用する。

4.2 2) フレーズ対の取得

1) で得た (a)〜(d) それぞれの単語対の集合からフレーズ対の集合を取得する。具体的には、Moses [15] の Phrase extractor[7] を利用した。その際、節などの長い単位対を抽出することを考慮して、抽出対象となるフレーズの最大長を 30 とした。

4.3 3) フレーズ対の組み合わせ

2) で得た (a)〜(d) それぞれのフレーズ対の抽出結果の和集合を取る。ただし、2) ではフレーズの表層文字列を考慮していないため、意味的に対応している可能性が高い表層上一致する単位対であっても同定できていない可能性がある。そこで、フレーズ対の集合に対して、TD 間で表層上一致する単位対 (表層アラインメント) も加える。表層アラインメントの取得には Meteor [16][8] を利用した。ただし、表層アラインメントのうち、ひらがな一文字同士の単語対は助詞が多く、正しい対応が取れている可能性が低いため、除外した。

4.4 4) 後処理

3) で得たフレーズ対の集合に対して、差異分析スキームの分析単位の定義と分割手順に合致する単位対のみを抽出するために、次の 2 つの後処理を行う。

句構造解析器を用いた絞り込み 2) のフレーズ対の取得では、階層性は保たれているものの、文法性は考慮されていないため、抽出結果には言語単位として妥当ではない単位も多く含まれる。そこで、日本語の句構造解析器である Jigg [17] を利用して、言語学的な単位対のみを絞り込む処理を行う。ただし、Jigg は、日本語の CCGBank [18] に基づく二分岐の構造であり、かつ、最小単位が形態素であることから、句構造解析結果の枝葉の中には、主語と格助詞を 1 単位とする単位や、述語を構成する一部の形態素のみからなる単位など、独立した言語単位でない単位も存在する。そこで、Jigg の解析結果に加えて、形態素情報をもとに、人手で文法的なルールを追加して不適切な単位対を同定することとした。

分割元が表層的に一致する対の除外 差異分析スキームの分割手順には、TD 間で表層上一致する単位対は分割を停止するという条件が含まれる。そこで、Jigg による絞り込みの後、同様の後処理を行う。

5 評価

評価用データを用いて 4 節の手法の評価実験を行った。具体的には、3) のフレーズ対の組み合わせで得られる 15 通りの結果全てについて正解データとの再現率、適合率、F1 値を算出した。

5.1 評価結果

開発用データ及び評価用データに対する評価結果を表 2 に示す。開発用データに対して F1 値が最大となったのは (a)・(b) の組み合わせであった。(a)・(b) を用いた場合、評価用データに対する再現率、適合率、F1 値はそれぞれ 56.5%、59.5%、57.9% であった.

また、再現率が最も高かった (a)・(b)・(c)・(d) の組み合わせについて 4) の後処理の前後の性能を比較したところ、再現率は 81.6%から 60.1%に低下する一方、適合率は 3.5%から 51.2%に改善していた。後処理はフレーズ対の絞り込みの機能を果たしているものの、主に再現率の面で改善の余地があることが明らかになった。

5.2 エラー分析

再現率が最も高い (a)・(b)・(c)・(d) すべてのフレーズ対を組み合わせた場合の抽出結果を対象として、抽出もれと抽出誤りの双方の観点からエラー分析を行った。エラー分析では、差異分析スキームにおける分析単位の定義を参照しながら、抽出された単位対の種類を分類し、分析単位の対の種類ごとの再現率を算出した上で、個別の事例を分析した。詳細は付録 B を参照されたい。

5.2.1 抽出もれ

抽出もれには、対の一方が「対応なし」となるような単位対、及び節、語句相当、句、複合表現とい

7) https://github.com/moses-smt/mosesdecoder/
8) https://www.cs.cmu.edu/~alavie/METEOR/index.html, version 1.5

表 2　各フレーズ対の組み合わせと正解データとの再現率、適合率、F1 値 (%): 太字は各列の最大値を指す。

評価対象	開発用データ			評価用データ		
	再現率	適合率	F1 値	再現率	適合率	F1 値
(a) SD を介した TD 間の単語対 (WSPAlign [12])	52.1	68.7	59.3	33.9	55.1	41.9
(b) TD 間の単語対 (WSPAlign [12])	55.5	**74.9**	63.8	53.0	**69.5**	**60.1**
(c) TD 間の単語対 (OTAlign [13]、UOT)	42.7	61.7	50.5	39.6	53.1	45.4
(d) TD 間の単語対 (OTAlign [13]、POT)	42.1	64.2	50.9	34.3	54.6	42.1
(a)・(b)	65.2	69.0	**67.1**	56.5	59.5	57.9
(a)・(c)	61.0	63.5	62.2	50.7	49.8	50.2
(a)・(d)	59.8	64.7	62.1	49.1	51.0	50.0
(b)・(c)	60.4	68.5	64.2	57.1	58.9	58.0
(b)・(d)	60.1	70.4	64.8	56.0	60.6	58.2
(c)・(d)	43.6	60.9	50.8	40.8	51.5	45.5
(a)・(b)・(c)	**67.4**	64.2	65.8	59.7	52.5	55.9
(a)・(b)・(d)	67.1	65.5	66.3	59.2	54.1	56.6
(a)・(c)・(d)	61.0	62.7	61.8	52.1	48.8	50.4
(b)・(c)・(d)	60.4	67.6	63.8	57.6	56.9	57.2
(a)・(b)・(c)・(d)	**67.4**	63.5	65.4	**60.1**	51.2	55.3

う比較的大きな単位対が多かった。

対の一方が「対応なし」となる単位対に関しては、1) の単語対の取得において WSPAlign、OTAlign ともに陽に「対応なし」の単語を同定できるものの、2) のフレーズ対を取得する段階で隣接する表現とひとまとまりになるという原理的な問題がある。また、正解データ中の「対応なし」に対応する単位は単語単位でないものも存在するため、提案手法では抽出できなかった。

節、語句相当、句、複合表現という比較的大きな単位対に関しては、4) の後処理を行わない場合でも抽出もれが多かった。これは、2) のフレーズ対を取得するという手法だけでは大きな単位を適切に抽出することが困難であることを示している。

また、5.1 節で見たように、4) の後処理においても抽出もれが増える。この原因としては、句構造解析器では適切に同定できない単位、不適切だが大きな単位対が存在する場合にそれよりも小さな正解の単位対が除外されることが考えられる。

5.2.2　抽出誤り

抽出誤りは、(i) 単位・対ともに適切だが不要な対 91 件、(ii) 単位は適切だが対が適切でない 134 件、(iii) 一方または双方の単位が適切でない 24 件、の 3 種類に大別できた。このうち、(i) 及び (ii) の誤りには句の単位対が、(iii) の誤りには記号を含む単位対が多かった。

句の単位対に関しては、差異分析スキームにおける「入れ子のない要素に分割する」という操作と、句構造解析器による文構造の解析が一致しない場合が多いことが考えられる。

記号を含む単位対に関しては、分割元の単位対に句読点以外の記号が含まれる場合に、記号を先頭に置く単位など文法性を損なう単位を抽出する例が多かった。

6　おわりに

本稿では、異なる翻訳文書間の差異の記述の効率化に向けて、同一の原文書に対する複数の翻訳文書間で対応する言語単位対の自動抽出という新しいタスクを提案した。そして、単語アラインメントを起点として単語対からフレーズ対を階層的に同定する手法を実装し、一定程度の性能を達成した。また、抽出結果のエラー分析を通じて、提案手法では、句などの比較的大きな単位対の抽出が難しいことを明らかにした。

本稿で提案した言語単位対の自動抽出タスクは、異なる翻訳文書間の差異の分析だけではなく、翻訳分析における文書対の前処理や多様な単位に対応する言い換え表現の同定といった、他の研究への応用も考えられる。今後は、これらへの応用可能性を視野に入れつつ、自動抽出精度の向上に取り組む。その際、本稿で述べた手法の改良のみでなく、WSPAlign [12] をフレーズ対を同定できるようにファインチューニングして用いること、階層的なフレーズアラインメント技術 [19] を利用することなどの有用性についても調査する予定である。

参考文献

[1] Tomono Honda, Atsushi Fujita, Mayuka Yamamoto, and Kyo Kageura. Designing a metalanguage of differences between translations: A case study for English-to-Japanese translation. In **Proceedings of the 3rd Workshop on Human Evaluation of NLP Systems**, pp. 23–34, 2023.

[2] ISO/TC37. ISO 18587:2017 translation services—Post-editing of machine translation output—requirements, 2017.

[3] Spence Green, Jeffrey Heer, and Christopher D. Manning. The efficacy of human post-editing for language translation. In **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**, pp. 439–448, 2013.

[4] Maarit Koponen. Is machine translation post-editing worth the effort? A survey of research into post-editing and effort. **The Journal of Specialised Translation**, Vol. 25, pp. 131–148, 2016.

[5] Geoffrey S. Koby, Paul Fields, Daryl Hague, Arle Lommel, and Alan Melby. Defining translation quality. **Revista Tradumàtica**, Vol. 12, pp. 413–420, 2014.

[6] Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. Multidimensional Quality Metrics (MQM): A framework for declaring and describing translation quality metrics. **Revista Tradumàtica**, Vol. 12, pp. 455–463, 2014.

[7] Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. Experts, errors, and context: A large-scale study of human evaluation for machine translation. **Transactions of the Association for Computational Linguistics**, Vol. 9, pp. 1460–1474, 2021.

[8] Atsushi Fujita. Attainable text-to-text machine translation vs. translation: Issues beyond linguistic processing. In **Proceedings of Machine Translation Summit XVIII: Research Track**, pp. 215–230, 2021.

[9] 本田友乃, 山本真佑花, 影浦峡. 異なる翻訳間の差異を記述するためのスキームの構築. 通訳翻訳研究への招待, Vol. 24, pp. 1–21, 2022.

[10] Steven Bird, Ewan Klein, and Edward Loper. **Natural language processing with Python**. O'Reilly Media, 2009.

[11] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to Japanese morphological analysis. In **Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing**, pp. 230–237, 2004.

[12] Qiyu Wu, Masaaki Nagata, and Yoshimasa Tsuruoka. WSPAlign: Word alignment pre-training via large-scale weakly supervised span prediction. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 11084–11099, 2023.

[13] Yuki Arase, Han Bao, and Sho Yokoi. Unbalanced optimal transport for unbalanced word alignment. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 3966–3986, 2023.

[14] Masaaki Nagata, Katsuki Chousa, and Masaaki Nishino. A supervised word alignment method based on cross-language span prediction using multilingual BERT. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing**, pp. 555–565, 2020.

[15] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In **Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions**, pp. 177–180, 2007.

[16] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In **Proceedings of the Ninth Workshop on Statistical Machine Translation**, pp. 376–380, 2014.

[17] Hiroshi Noji and Yusuke Miyao. Jigg: A framework for an easy natural language processing pipeline. In **Proceedings of ACL-2016 System Demonstrations**, pp. 103–108, 2016.

[18] Sumire Uematsu, Takuya Matsuzaki, Hiroki Hanaoka, Yusuke Miyao, and Hideki Mima. Integrating multiple dependency corpora for inducing wide-coverage Japanese CCG resources. In **Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1042–1051, 2013.

[19] 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.

[20] 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, 2019.

[21] Graham Neubig. The Kyoto free translation task. http://www.phontron.com/kftt, 2011.

[22] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In **Proceedings of the 32nd International Conference on Machine Learning**, Vol. 37, pp. 957–966, 2015.

[23] Richard Sinkhorn. A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices. **The Annals of Mathematical Statistics**, Vol. 35, No. 2, pp. 876–879, 1964.

A　単語アラインメントツールの設定

WSPAlign を利用した (a)、(b) の単語アラインメントの取得では、mBERT (bert-base-multilingual-cased) [20] で訓練された事前学習済みモデルを京都フリー翻訳タスク (KFTT) のアラインメントデータ [21] でファインチューニングしたモデルを使用した。アラインメントの対称化における閾値は、Nagata ら [14] にならい 0.4 とした。

OTAlign を利用した (c)、(d) の単語アラインメントの取得では、BERT-base-uncased [20] によって学習された単語埋め込みを利用した教師なし学習のモデルを利用し、コスト関数としてコサイン距離 (cosine) を、重みとして単語埋め込みの一様分布 (uniform) [22] を指定し、Sinkhorn アルゴリズム [23] を適用した。

B　分析単位対の種類ごとの再現率

(a)・(b)・(c)・(d) のフレーズ対の抽出結果を組み合わせた自動分割の結果に対して、差異分析スキーム [1] における分析単位の定義を参照しながら、抽出された単位対の種類を分類し、分析単位の対の種類ごとの再現率を算出した。表 3 に分析単位対の種類ごとの再現率を示す。

表 3　分析単位対の種類ごとの再現率

		MT+PE							
		文相当	節	語句相当	句	複合表現	語	記号	対応なし
HT	文相当	-	-	-	-	-	-	-	-
	節	-	5/17	-	0/1	-	-	-	-
	語句相当	-	-	3/11	-	-	-	-	-
	句	-	0/2	-	12/40	3/8	1/1	-	-
	複合表現	-	-	-	9/10	38/78	11/17	1/1	0/3
	語	-	-	-	2/3	4/6	128/157	0/1	0/5
	記号	-	-	-	-	-	-	44/52	0/5
	対応なし	-	-	-	-	0/3	0/5	0/8	-