

# テキストアナリティクスツールの操作ログからの 実験設定の説明文生成

森田康介<sup>1</sup> 西村太一<sup>1</sup> 亀甲博貴<sup>2</sup> 森信介<sup>2</sup>

<sup>1</sup> 京都大学大学院 情報学研究科 <sup>2</sup> 京都大学 学術情報メディアセンター  
{morita.kosuke.45c,nishimura.taichi.43x}@st.kyoto-u.ac.jp  
{kameko,forest}@i.kyoto-u.ac.jp

## 概要

本研究では、テキストアナリティクスツールの操作ログと実験設定の説明文の相互生成を目的とする。人文科学分野において広く使用されているKH Coderを使用した論文を対象に収集し、論文の実験設定の記述と実際のツールの操作ログを再現したものからなるデータセットを構築した。また、このデータセットを用いて論文中の記述から操作ログを推定するモデルを構築し、アノテーションしていない論文に適用することやデータの値を置換することで自動的にデータセットを拡張した。これらを用いて、操作ログと説明文を相互生成するモデルを構築した。

## 1 はじめに

科学技術論文の実験においてその実験設定を正しく記述し残すことは、論文を書いた本人だけでなく論文を読んだ人間も行った実験の再現が容易になることから重要である。近年では、あらゆる分野で論文の再現性が問題となる事例が発生しており [1, 2], ますますその重要性が高まっている。

科学実験においてはツールを使用して研究することが多いが、一般的にツールの利用にあたってはどのような処理が行われたかを示す操作ログ（以下ログと表記）が出力される。実験に使用したツールのログから行なった操作の説明文の生成ができれば、論文の書き手に対する執筆の補助になり、読み手の実行内容の理解度と再現性が高まる。反対に、論文に記されている操作の説明文からツールのログの生成ができれば、実行内容の再現がより容易になる。

本研究では、様々なツールの中でもテキストアナリティクスツールに焦点を当てる。テキストアナリティクスとは、文章を解析して情報を取り出す分析

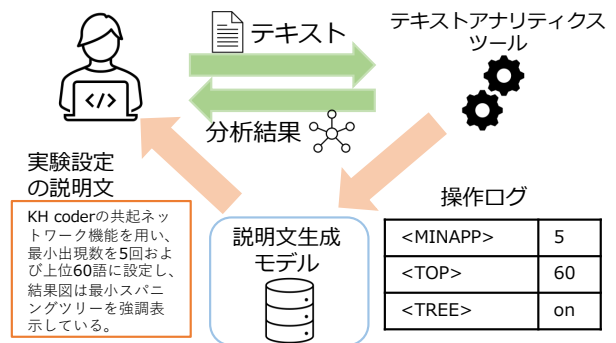


図1 利用シーンの例。

のことである。テキストアナリティクスツールは人文科学の分野で頻繁に使用されているため、我々はその操作の説明文を生成することで貢献ができると考えた。利用シーンの例を図1に示した。

本研究の課題のように、グラフや表などのデータとして記述されたものについてテキストで説明するタスクはData-to-Textと呼ばれる [3]。元来は規則やテンプレートをによる手法が考えられてきた [4, 5] が、自然な文を自動で生成するためには、膨大な規則の構築が必要である。そのため、近年では深層学習を利用した手法が目立っている [6, 7]。深層学習モデルが十分な性能を発揮するためには大規模なデータセットを用意しなければならず、その作成にはコストがかかるという問題がある。これを解決するために擬似的にデータを生成してデータセットを自動で拡張する手法がとられてきた [8, 9, 10]。

本研究ではテキストアナリティクスツールを利用した際の仮想的なログを定義し、データセットを構築する。さらに、3種のデータ拡張手法を導入する。その後、T5 [11] を学習させ、ログから操作の説明文の生成と説明文からのログの生成を行う。生成した説明文とログについて、自動評価尺度による評価を行う。ここで、データセットの拡張の有無やその手法による評価結果の違いについて考察する。

## 2 タスクの定義

### 2.1 使用するツール

本研究では、テキストアナリティクスツールとして KH Coder [12] を用いる。KH Coder を使用した文献は累計で 5,686 件に上っており、社会学や文学をはじめとした人文情報学の幅広い分野で使用されているツールである。

### 2.2 アノテーション

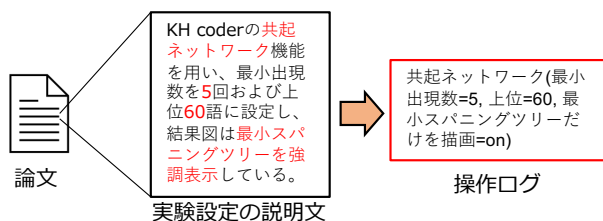


図2 アノテーションの例.

最初に、少量のデータセットを手で作成する。まず、ユーザーが変更可能な設定を予めリストアップした。次に、KH Coder を用いた論文のリスト<sup>1)</sup>から 2021 年の論文をダウンロードした。そして、これらの論文からテキストマイニングを行った際の KH Coder での操作の説明にあたる文を抽出した。例を図 2 の実験設定の説明文に示した。共起ネットワークか対応分析を実施している論文がほとんどであったため、どちらかの説明文のみを収集した。

その後、設定のリストをもとに説明文に対応するログを手で作成した。ログの先頭部分には分析方法を記述し、続く括弧の中には各設定項目に対する値が「(設定項目)=(値)」という形式でコンマによって区切って記述されるようにログの記述方法を定めた。ログの例を図 2 に示した。ここで作成したデータセットの概要を表 1 に記載する。

	共起ネットワーク	対応分析	合計
論文数	192	38	230
データ数	213	40	253

表1 手動作成したデータセットの概要.

## 3 データ拡張手法

深層学習モデルが十分な性能を発揮するためには大量のデータが必要となるが、人手でのアノテーションは費用や時間の点で高コストである。その問題に対処するためにデータ拡張を実施する。本研究

では、(1) 擬似ラベルに基づく手法、(2) 値の入れ替えによる手法、(3) 2つの手法の併用、の3種の拡張手法を用いる。本章では各手法について記述する。

### 3.1 擬似ラベリング

ここでは、新たにダウンロードした論文ファイルに固有表現認識とテキスト分類によって自動的なアノテーションを行う。

#### 3.1.1 固有表現認識

ここでは、実験設定の情報を説明文から抽出する固有表現認識 (NER) を行う。まず、手動作成のデータセットの説明文を MeCab [13] を用いて単語分割した。そして、説明文に含まれる実験設定を固有表現とし、IOB 形式でアノテーションを施した。付録 A にその固有表現の種類と数の詳細を示す。

Flair [14] を用いて固有表現認識モデルを作成した。学習にあたり、Flair 内の日本語の文字単位の事前学習済みの埋め込みベクトルを使用した。テストセットにおける予測結果の詳細を付録 B.1 に示す。

得られたモデルを利用し、人手でダウンロードした 2020 年の論文と Web スクレイピングによってダウンロードした 2019 年以前と 2022 年の論文計 2,132 件に対して固有表現認識を行なった。まず、論文の pdf ファイルを pdfminer<sup>2)</sup> で読み込み、spaCy<sup>3)</sup> の文分割機能を用いて自動で文単位に分割する。その後、分割した文それぞれに対して NER のモデルで固有表現を予測し、定義したログの形へ整形した。

Flair の固有表現の予測においては確信度を示すスコアも出力される。ノイズが多いデータはモデルの性能向上を妨げるとされていることから [15]、しきい値を 0.5 に設定し、それより小さい確信度の固有表現を除外した。

#### 3.1.2 テキスト分類

次に、文から分析方法を予測するモデルを作成する。固有表現抽出では分析とは無関係の文からも固有表現が予測される場合がある。そのため、共起ネットワーク、対応分析、それ以外の 3 クラスの分類タスクとして扱う。2020 年の 18 件の論文から固有表現が予測された 205 文に対してその分析方法を手でアノテーションした。さらに、これらのデータに手動作成したデータを加え、データセットを

1) <https://kncoder.net/bib.html>

2) <https://pdfminersix.readthedocs.io/en/latest/>

3) <https://spacy.io/>

作成する。

モデルには BERT [16] を用いた。事前学習済みモデルには東北大学乾研究室のモデル `cl-tohoku/bert-base-japanese-whole-word-masking`<sup>4)</sup> を使い、ファインチューニングを行った。データセットを学習データ: 検証データ: テストデータで 8:1:1 に分割した後、学習データで 20 エポック学習させ、そのうち検証データにおける損失が最も小さいモデルを最終的な学習済みモデルとした。この時のテストデータにおける結果の詳細は付録 B.2 に示す。

このモデルを 3.1.1 項で得られた文に適用した。この分類により、文が共起ネットワークか対応分析であると分類されたデータに対して分析方法の記述を追加したものを最終的なログとする。これにより、平均して 3,339 のデータが作成された。

## 3.2 スワップ

ここでは、先行研究 [8, 9] と同様に説明文とログの設定に関する値を置換することによりデータ拡張を行う。具体的なアルゴリズムは付録 C に示す。これにより、平均して 1,620 のデータが作成された。

## 3.3 ハイブリッド

3.1 節と 3.2 節で作成したデータセットを結合させて学習を行う。

# 4 実験

## 4.1 実験設定

モデルには T5 [11] を用いる。Data-to-Text のタスクに対して T5 の有用性は示されていることから [17]、本研究ではこれを用いることにした。

T5 は入力に `prefix` と呼ばれる語句を付加することでタスクを識別し、1つのモデルで複数のタスクを実行することが可能である [11]。本研究でも、`prefix` を用いたマルチタスク学習を行う。具体的にはログからの説明文の生成には入力の先頭に「`translate log to text:`」を、説明文からのログの生成には「`translate text to log:`」を付加する。全ての訓練データに対してこれらの2通りの入出力を作成する。さらに、シングルタスクの場合でも実験を行い比較する。

事前学習済みモデルには園部 勲氏により `transformers` に公開されている `sonoisa/t5-base-japanese`<sup>5)</sup>

を使用した。Raffel ら [11] と同様にパラメータの更新には `Adafactor` [18] を使い、学習率は `1e-3`、`scale_parameter` を `false`、`relative_step` を `false` とした。バッチサイズを 8 とし 100,000 ステップの学習を行った。1,000 ステップごとに `validation` を行い、BLEU-4 の値が最も高いモデルで評価を行う。その他のパラメータはデフォルトのものを用いた。

本研究で人手で作成したデータセットのサイズは大きくない。そのため、以下の手続きで交差検定を実施した。まず、手動で作成したデータセットをランダムにシャッフルした後、4:1 に「訓練データおよび検証データ」と「テストデータ」に分割し、交差検定を行う。さらに、前者のデータは学習データと検証データに 4:1 に分割する。この訓練データと検証データを用いてデータセットを拡張する。なお、評価結果は交差検定における平均値を示している。

BLEU に代表される自動評価尺度は人間による評価と相関があることが示されている [19]。そのため、説明文の評価指標には BLEU [20] と METEOR [21]、BERTScore [22] を用いる。また、単純な規則でログから説明文を生成するシステムも作成し比較を行う。具体的な生成方法は付録 D に示す。

ログには分析の方法と設定を表す部分が存在するが、これを分けて評価を行う。まず、生成されたログから分析方法についての部分を抜き取り、正解データに対する分類精度を一般的な分類タスクと同様に計算した。その F 値を F1-method として表に掲載した。続いて、分析の設定については括弧の中に各設定項目に対する値が「(設定項目)=(値)」という形式で区切り文字コンマにより記述されている。これら設定項目と値の組の 1 ブロックを予測とみなし、スコアを計算する。具体的には、予測のうち正解と設定項目と値が完全に一致している割合を Precision とし、正解のうち設定項目と値が予測と完全に一致している割合を Recall として計算した。例えば、生成されたログの設定項目を「(最小出現数=3, 上位=60)」、正解のログの設定項目を「(最小出現数=3, 上位=60, 最小スパニングツリーだけを描画=on)」とすると、Precision は 2/2 となり、Recall は 2/3 となる。Precision と Recall の調和平均を F1-options とし計算した。さらに、「共起ネットワーク」「対応分析」のどちらかの文字列で始まり、それに続いて「(」で始まり「)」で終わるものでなかったものをログの形式のエラーとして計測する。なお、F 値の計算にはエラーデータは除外している。

4) <https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking>

5) <https://huggingface.co/sonoisa/t5-base-japanese>



	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	BERT-P	BERT-R	BERT-F1
ルールベース	11.12	6.96	4.59	3.29	21.66	<b>0.791</b>	0.708	<b>0.747</b>
追加なし	36.55	<b>24.49</b>	<b>18.49</b>	<b>15.09</b>	<b>31.40</b>	0.735	<b>0.742</b>	0.737
追加なし (マルチタスク)	<b>36.98</b>	24.47	18.20	14.64	30.35	0.734	0.739	0.736
擬似ラベリング	33.32	21.82	15.84	12.40	27.66	0.729	0.724	0.726
擬似ラベリング (マルチタスク)	33.07	21.33	15.32	11.89	27.81	0.730	0.727	0.727
スワップ	35.36	23.46	17.43	14.08	29.85	0.720	0.735	0.727
スワップ (マルチタスク)	34.06	22.24	16.20	12.85	29.39	0.723	0.733	0.727
ハイブリッド	34.45	22.83	16.90	13.48	30.03	0.731	0.736	0.732
ハイブリッド (マルチタスク)	35.84	23.85	17.43	13.74	29.40	0.742	0.736	0.738

表2 ログから生成した説明文の評価.

## 4.2 ログからの説明文生成

表2に実験の結果を示す. まず, ルールベースにとT5による結果を比較するとBLEUでは大きい差ができてることがわかる. 一方で, BERTScoreのPrecisionはルールベースによる生成が最も高くなった. この結果からルールベースによる生成では情報を的確に伝えられているが, 論文で記述される文章とは似ていないと考えられる.

次に, データセットを拡張する場合としない場合を比較すると, データ拡張を行わない場合がスコアが高くなった. データ拡張手法によってスコアが向上しなかった理由として, 追加したデータの質が考えられる. 擬似ラベリングでは論文ファイルを新たにダウンロードして自動でアノテーションを行なった. このファイルの読み込み時にテキスト順序が本来のものでない場合や本文中にないpdfファイルの記号がテキストとして入る場合があり, これらがノイズになっていたと考えられる. また, ダウンロードした論文によって読点を表す記号が異なるため文分割を使用した, この精度が高くないということも理由の1つとしてあげられる. そして, スワップでは文字列の置換を行うが置換すべき対象ではない文字列まで置換されてしまうことがあるため, それが学習の妨げとなってしまった可能性がある.

データ拡張手法内では全体的に見てスコアが最も高いのはハイブリッドだった. これは擬似ラベリングではBERTscoreのPrecisionが高く, スワップではRecallが高くなっているという特徴があり, ハイブリッドはこの両手法の長所を取り入れてより良い学習が可能になった結果だと考えられる.

そして, マルチタスク学習を行うことでスコアが上がる場合と下がる場合があった. この結果からタスクに特化したモデルを得るためにマルチタスク学習を行うことが必ずしも良いとは限らないと考えられ, これはRaffelら[11]の結果と一致している.

## 4.3 説明文からのログ生成

	F1-method	F1-options	エラー数
追加なし	0.9442	<b>0.7379</b>	8
追加なし (マルチタスク)	0.9616	0.7106	3
擬似ラベリング	0.9514	0.6925	1
擬似ラベリング (マルチタスク)	0.9427	0.7173	0
スワップ	0.9399	0.6823	2
スワップ (マルチタスク)	0.9603	0.6979	4
ハイブリッド	0.9337	0.6973	2
ハイブリッド (マルチタスク)	<b>0.9672</b>	0.7035	1

表3 説明文から生成したログの評価.

結果を表3に掲載した. まず, F1-methodにおいては擬似ラベリングのスコアが最も高くなった. 次に, F1-optionsにおいては拡張しない場合のスコアが最も高かった. 一方, エラーが多かったのはデータ拡張をしない場合とスワップによる拡張の場合だった. この結果からエラー数の減少に対するデータ拡張の有用性が確認できた.

また, マルチタスク学習を行うことによりスコアが下がりエラー数が減少する傾向が確認できた. こちらもマルチタスク学習を行うことが必ずしも良いとは限らないと考えられる結果となった.

## 5 おわりに

本研究では, テキストアナリティクスの際に出力されるログと説明文の相互生成を行なった. これを深層学習的モデルで行うにあたって, 少量のデータセットを拡張するための固有表現認識とクラス分類による自動アノテーション手法と値の置換による拡張手法を提案した. 実験において, データセットを拡張しない場合と各拡張手法を自動評価尺度によって比較した. その結果, 指標によってはデータ拡張をすることでスコアが向上することが確認できた.

本研究ではログを仮想的に定義したが, その他の一般的なツールにおいても本研究と同じ形式でアノテーションを施すことによって同様の生成手法・データ拡張手法をとることができるという点で応用可能性があると考えられ, 研究の余地がある. そのため, 様々なデータへの適用や適切な拡張手法の考案は今後の課題である.

## 参考文献

- [1] Jeremy Freese and David Peterson. Replication in social science. **Annual review of sociology**, Vol. 43, pp. 147–165, 2017.
- [2] Scott E Maxwell, Michael Y Lau, and George S Howard. Is psychology suffering from a replication crisis? what does “failure to replicate” really mean? **American Psychologist**, Vol. 70, No. 6, p. 487, 2015.
- [3] Thiago Castro Ferreira, Chris van der Lee, Emiel Van Miltenburg, and Emiel Krahrmer. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. **arXiv preprint arXiv:1908.09022**, 2019.
- [4] Eli Goldberg, Norbert Driedger, and Richard I Kittredge. Using natural-language processing to produce weather forecasts. **IEEE Expert**, Vol. 9, No. 2, pp. 45–53, 1994.
- [5] Kees Van Deemter, Mariët Theune, and Emiel Krahrmer. Real versus template-based natural language generation: A false opposition? **Computational linguistics**, Vol. 31, No. 1, pp. 15–24, 2005.
- [6] Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. Table-to-text generation by structure-aware seq2seq learning. In **Thirty-Second AAAI Conference on Artificial Intelligence**, 2018.
- [7] Ratish Puduppully, Li Dong, and Mirella Lapata. Data-to-text generation with content selection and planning. In **Proceedings of the AAAI conference on artificial intelligence**, Vol. 33, pp. 6908–6915, 2019.
- [8] Ernie Chang, Xiaoyu Shen, Dawei Zhu, Vera Demberg, and Hui Su. Neural data-to-text generation with lm-based text augmentation. **arXiv preprint arXiv:2102.03556**, 2021.
- [9] Ernie Chang, Vera Demberg, and Alex Marin. Jointly improving language understanding and generation with quality-weighted weak supervision of automatic labeling. In **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume**, pp. 818–829, Online, April 2021. Association for Computational Linguistics.
- [10] Chris van der Lee, Thiago Castro Ferreira, Chris Emery, Travis Wiltshire, and Emiel Krahrmer. Neural data-to-text generation based on small datasets: Comparing the added value of two semi-supervised learning approaches on top of a large language model. **arXiv preprint arXiv:2207.06839**, 2022.
- [11] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. **J. Mach. Learn. Res.**, Vol. 21, No. 140, pp. 1–67, 2020.
- [12] 樋口耕一. テキスト型データの計量的分析—2つのアプローチの峻別と統合—. **理論と方法**, Vol. 19, No. 1, pp. 101–115, 2004.
- [13] Taku Kudo. Mecab: Yet another part-of-speech and morphological analyzer. <http://mecab.sourceforge.net/>, 2005.
- [14] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In **NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)**, pp. 54–59, 2019.
- [15] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. **IEEE transactions on neural networks and learning systems**, Vol. 25, No. 5, pp. 845–869, 2013.
- [16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.
- [17] Mihir Kale and Abhinav Rastogi. Text-to-text pre-training for data-to-text tasks. **arXiv preprint arXiv:2005.10433**, 2020.
- [18] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In **International Conference on Machine Learning**, pp. 4596–4604. PMLR, 2018.
- [19] Ehud Reiter and Anja Belz. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. **Computational Linguistics**, Vol. 35, No. 4, pp. 529–558, December 2009.
- [20] 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**, pp. 311–318, 2002.
- [21] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In **Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization**, pp. 65–72, 2005.
- [22] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. **arXiv preprint arXiv:1904.09675**, 2019.

## A アノテーションした固有表現

タグの種類	タグの説明	出現数
MINAPP	語の最小出現数	79
MET	共起関係の計算手法	79
TOP	共起関係の表示件数	73
EXT	外部変数	41
METNUM	共起関係のしきい値	34
BOLD	共起関係の強さによって線の太さを変える	25
BUBBLE	バブルプロットを使用	25
SUBG	サブグラフ検出の種類	21
PART	分析対象の品詞	13
UNIT	分析単位	10
TYPE	共起関係の種類	10
TREE	最小スパニングツリーのみ描画	8
MINDOC	最小文書数	8
DIF	差異が顕著な語を分析に使用	6
ORI	原点から遠い語のみラベル表示	3
MAXAPP	語の最大出現数	3
STAN	係数の標準化	3
BUBSIZE	バブルプロットの大きさ	1
MAXDOC	最大文書数	1
合計		443

表4 共起ネットワーク・対応分析における固有表現.

## B テストセットにおける予測結果

### B.1 固有表現認識

	Precision	Recall	F1-score	Support
MINAPP	0.714	1.000	0.833	20
TOP	0.947	0.833	0.923	20
MET	1.000	0.882	0.938	17
BOLD	0.222	0.250	0.235	8
EXT	0.000	0.000	0.000	8
METNUM	1.000	0.857	0.923	7
BUBBLE	0.429	0.500	0.462	6
PART	0.000	0.000	0.000	4
MINDOC	1.000	0.667	0.800	3
TREE	1.000	0.333	0.500	3
STAN	0.000	0.000	0.000	3
SUBG	0.500	0.500	0.500	2
UNIT	1.000	1.000	1.000	2
DIF	1.000	1.000	1.000	1
TYPE	0.000	0.000	0.000	1
ORI	0.000	0.000	0.000	0
micro avg	0.710	0.676	0.693	105

表5 固有表現認識モデルによる予測結果. 交差検定1回分の結果を示した. 高い頻度かつ単語数の少ない固有表現であるほど予測精度が高いという結果が得られた.

### B.2 分類モデル

	Precision	Recall	macro-F1	Support
いずれでもない	0.944	0.944	0.944	18
共起ネットワーク	1.000	0.957	0.978	47
対応分析	1.000	0.750	0.857	7

表6 BERTによる分析手法の分類の結果. 交差検定1回分の結果を示した.

## C スワップのアルゴリズム

### Algorithm 1 Swap

```

1: Input: original dataset  $D$ , value-set  $V$ 
2: Output: swapped dataset  $D_s$ 
3: for  $i = 1$  to  $len(D)$  do
4:    $l, d = D[i][\text{'log'}], D[i][\text{'description'}]$ 
5:    $n = len(l.options)$ 
6:    $cnt = 0$ 
7:   for  $itr = 1$  to 1000 do
8:     for  $j = 1$  to  $n$  do
9:        $r = random\_int(1, n)$ 
10:       $k, v = l.options[r]$ 
11:       $nv = random\_sample(V[k] \setminus v)$ 
12:       $l.options[r] = k, nv$ 
13:       $d = d.replace(v, nv)$ 
14:      if  $(l, d) \notin D_s$  then
15:         $D_s \cup (l, d)$ 
16:         $cnt += 1$ 
17:      if  $cnt = 20$  then
18:        break

```

スワップのアルゴリズム. value-set は各設定項目をキーとしてデータセット内の全ての取りうる値のリストを返す辞書である.

## D ルールベースによる生成

本研究では, ルールベースによる生成を行なった. 具体的なアルゴリズムは, ログの形式が「分析方法(設定項目=値, 設定項目=値...)」であることを用いて「(設定項目)を(値),」という文字列を繰り返し, 最後の設定項目まで達した場合「とし(分析方法)を行った.」と出力する.

例えば, 「共起ネットワーク(最小出現数=3, 上位=60, 最小スパニングツリーだけを描画=on)」というログが入力として与えられた場合, 「最小出現数を3, 上位を60, 最小スパニングツリーだけを描画をonとし共起ネットワーク分析を行った.」となる.