Texylon: Dataset of Log-to-Description and Description-to-Log Generation for Text Analytics Tools

Masato Nakata, Kosuke Morita, Hirotaka Kameko, and Shinsuke Mori

Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501 Japan
nakata.masato.26m@st.kyoto-u.ac.jp

Abstract. We propose two tasks for text analytics tools, description generation and log estimation, and a dataset to solve them. Text analytics tools, popular in the digital humanities, provide researchers with various functions for texts based on natural language processing (NLP). Our first task, description generation from operational logs, helps these researchers write papers easily and accurately. Our second task, log estimation given a paper, is just the opposite and helps readers reproduce the analyses in the paper. For those tasks, we created a dataset consisting of descriptions and logs in text analytics experiments. Because our dataset is not large enough, we also propose some methods for data augmentation: swapping a value of each configuration with another value, pseudo-labeling using BERT and NER, pseudo-labeling using BERT and T5, and a combination of them. The highest BLEU score for that model in the description generation task is 36.98, and F1 score in the log generation task is around 0.7.

Keywords: Data-to-text, Text-to-data, Text analytics, Natural language processing, Digital humanities

[1]

1 Introduction

The uses of informatics have been getting more and more common for the humanities in these decades [3, *inter alia*]. Among them one of the most typical and classical usage is statistical analysis of texts, such as term frequencies/document frequencies of words, co-occurrence network, correspondence analysis for word distribution, and cluster analysis. A text analytics tool, Voyant Tools[2], has recently been awarded by Alliance of Digital Humanities Organizations, for example. In addition to such web-based tools, installation type tools are also available. Famous one is KH Coder [7,8], which provides many powerful functions

[1] This preprint has not undergone peer review (when applicable) or any post-submission improvements or corrections. The Version of Record of this contribution will be published in New Frontiers in Artificial Intelligence, Springer.

[2] https://voyant-tools.org/ accessed on 2023/Oct/20.

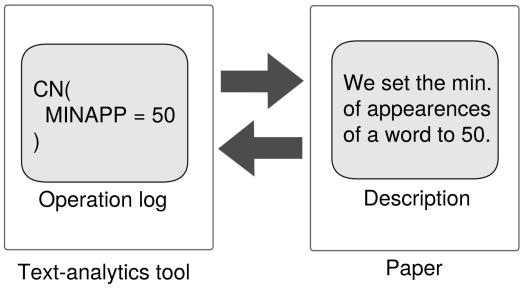

Fig. 1. Bidirectional conversion between operation logs and descriptions.

to analyze texts in various domains, including literature, historical documents, questionnaires, etc.

One of the faults in many analytics tools is that configurations are a bit complicated for non-specialists in statistics, so that describing what the configurations are appropriately can be hard work. Such a difficulty implies another problem: inappropriate descriptions result in the low reproducibility for readers. A proper description for a general-purpose text analytics tool needs authors/readers to have statistical knowledge.

In this background, we aim to build a model that

1. generates an appropriate description given a set of configurations to lighten the burden on the authors' side, and
2. estimates underlying configurations given a description to help readers reproduce the results;

however, no proceeding datasets have been constructed for those tasks, so this time we announce the first dataset, Texylon, which can be used for description/log generation tasks. The precise definitions of a "description" and a "log" are given in §3, but roughly speaking, a description is a part of a paper where the authors explain which operation they conducted and with which configurations. A log corresponds to the configurations for that operation.

One drawback of our dataset is the modest size nevertheless, as it is the first dataset in this domain. To relieve that problem, the second subject of our paper has appeared now — we had better augment the dataset in order to predict descriptions/logs more precisely and stably.

In this paper, first we define the tasks because we use confusing terms to name similar concepts, then introduce our dataset. We also present several methods for augmenting the data. In particular, some methods require specific language models, so we will also describe the construction of such models. Then, we will go on to the evaluation of the augmentation methods by proposing a baseline model for the log-to-description and description-to-log tasks.

2 Related Work

He et al. studied effective and efficient usages of software logs in a statistical or machine learning way [6]. They presented a survey on some topics, including the design of logging systems, log compression and parsing logs into structured data. In their paper, the definition of a log is closer to ours of a "description" (see §3), and structured data means our "logs."

They aimed for more reliable engineering. Although their approaches looked good for engineering, where a log is always formatted and well-structured, they do not work for natural language tasks.

Nivikova et al. published E2E, a dataset for natural language generation in the restaurant domain [12]. It contains a large amount of meaning representations (MRs) of restaurants and reference texts that explain about the restaurants in natural language.

Kale et al. proposed a method which can be applied to general data-to-text generation tasks [9]. They used pre-trained T5 models [15], the large language models (LLMs) of a wide range of application, and then fine-tuned them on a few shot data representing the structured data. For example, they tried ToTTo [14], which consists of Wikipedia tables paired with natural language descriptions, as a fine-tuning dataset.

3 Task Definitions

The tasks we introduce in this paper are the following two:

1. **Log-to-description**: description generation from operational logs, which helps text analytics researchers write papers easily and accurately,
2. **Description-to-log**: operational log estimation from descriptions, which helps paper readers reproduce the analyses.

Here, by an "operational log" we mean a pair composed of the following data:

- a function name — which function(s) is used in a research;
- configurations — key-value pairs that represent the parameters to this function(s).

To formulate the two tasks in a stricter manner, we define a few sets as follows. Let $\mathcal{F}_T$ be a set of all the functions in a text analytics tool T. For each function $f \in \mathcal{F}_T$, we have a set of *keys* of its configurations, denoted by $\mathcal{K}_T(f)$, and a set of possible *values* for a key $k \in \mathcal{K}_T(f)$, denoted by $\mathcal{V}_T(f, k)$. Then, a *configuration* of f is no more than an ordered pair (k, v), where $k \in \mathcal{K}_T(f)$ and $v \in \mathcal{V}_T(f, k)$.

Before we describe about configuration*s*, remember that the symbol $\coprod$ represents a disjoint union of a family of sets:

$$\coprod_{\lambda \in \Lambda} X_\lambda := \{(\lambda, x) \mid \lambda \in \Lambda, x \in X_\lambda\} \tag{1}$$

for a family of sets X_λ indexed by $\lambda \in \Lambda$. Using this symbol, the disjoint union of $\mathcal{V}_T(f,k)$ running $k \in \mathcal{K}_T(f)$ is a set

$$\mathcal{V}_T(f) := \coprod_{k \in \mathcal{K}_T(f)} \mathcal{V}_T(f,k) = \{(k,v) \mid k \in \mathcal{K}_T(f), v \in \mathcal{V}_T(f,k)\}. \tag{2}$$

Now we can define *configurations* of f as a map $c : K \to \mathcal{V}_T(f)$ for some subset $K \subset \mathcal{K}_T(f)$ with a property $c(k) \in \{k\} \times \mathcal{V}_T(f,k)$. A condition for a key $k \in \mathcal{K}_T(f)$ being *not* contained in K implies that the key k is not mentioned in a paper and its value appears set to the default value. We denote the set of configurations of f by

$$\mathcal{C}_T(f) := \{\text{configurations of } f\}, \tag{3}$$

and an *operational log*, or simply a *log*, is an element of the set

$$\mathcal{L}_T := \coprod_{f \in \mathcal{F}_T} \mathcal{C}_T(f) = \{(f,c) \mid f \in \mathcal{F}_T,\, c \in \mathcal{C}_T(f)\}. \tag{4}$$

Let us provide an example situation here. KH Coder [7,8] has a function "co-occurrence network," which is useful for the visual analysis of the word statistics. We denote it by $\mathsf{CN} \in \mathcal{F}_{\mathsf{KC}}$ (KC stands for "**K**H **C**oder"). For this function CN, the key set $\mathcal{K}_{\mathsf{KC}}(\mathsf{CN})$ contains, for example, Pos := "target parts of speech" and MinApp := "the minimum number of appearances of a word." The value sets may include these instances:

$$\text{"noun and verb"} \in \mathcal{V}_{\mathsf{KC}}(\mathsf{CN}, \mathsf{Pos}), \tag{5}$$

$$50 \in \mathcal{V}_{\mathsf{KC}}(\mathsf{CN}, \mathsf{MinApp}). \tag{6}$$

If we define a map $c : \{\mathsf{Pos}, \mathsf{MinApp}\} \to \mathcal{V}_{\mathsf{KC}}(\mathsf{CN})$ by $\mathsf{Pos} \mapsto$ "noun and verb," $\mathsf{MinApp} \mapsto 50$, we obtain configurations $c \in \mathcal{C}_{\mathsf{KC}}(\mathsf{CN})$. Suppose that we have conducted an experiment expressed by the above operational log (CN, c), and that we are about to write a paper. In this case, the "description" may be fully descriptive sentences like "We calculated the co-occurrence network by KH Coder. To this end, we set the target parts of speech to 'noun and verb', and the minimum number of appearances of a word to 50."

With these notations, our tasks can be formulated in this way:

1. **Log-to-description.** Given an operation log $\ell \in \mathcal{L}_T$, output a description d which maximizes the probability $p(d \mid \ell)$;
2. **Description-to-log.** Given a description d, output an operational log $\ell \in \mathcal{L}_T$ which maximizes the probability $p(\ell \mid d)$.

4 Dataset

To evaluate methods for the tasks and even train methods based on machine learning techniques, we constructed a dataset, which we call Texylon. Texylon

Table 1. Dataset size of the manual annotation and the data augmentations.

Dataset Type	# of logs	# of newly added
Texylon	253	–
Swapping	1,873	(+ 1,620)
BERT-NER	3,592	(+ 3,339)
BERT-T5	14,375	(+14,122)
Hybrid	5,212	(+ 4,959)

consists of manually annotated description-log pairs (d, ℓ)'s, where d is a description fetched from a real-world paper and ℓ is its operational log $\in \mathcal{L}_T$ for some text-anatytics tool T.

In this section we explain our dataset, Texylon, and then we propose four augmentation methods. Table 1 summarizes the specifications of Texylon and the augmentation results.

4.1 Data Construction

We took KH Coder [7,8] as the example of text analytics tools. We first collected papers written in Japanese published in 2021 in the list of its official web page[3]. Then annotators located the descriptions about the operations of the tool, and, given such descriptions, the annotators created their operational logs manually. Of course, there can be a potential bias to focus on the only *one* text analytics tool; however, descriptions obtained should be similar among all the tools, so our data augmentation methods (§4.2) and a model (§5.1) are applicable beyond KH Coder.

The format of the logs is in a python style like `FUNCTION(KEY1=VALUE1, ...)`. The following is an example.

```
CN(Pos="noun and verb", MinApp=50)
```

Here, `Pos` and `MinApp` $\in \mathcal{K}_{\mathsf{KC}}(\mathsf{CN})$, represent the keys defined in §3.

Since almost all the researches in the selected papers focused on co-occurrence network or correspondence analysis, we selected only the papers which conducted co-occurrence network or correspondence analysis as our first attempt.

The possible configurations $\mathcal{C}_{\mathsf{KC}}(f)$ of each function $f \in \mathcal{F}_{\mathsf{KC}}$ were known in advance at the annotation time, so it had little cost for determining the rigorous annotation standard.

Note that the above annotation is not ambiguous for human, we did not ask multiple annotators to work on the same part in order to calculate inter-annotator agreement.

[3] `https://khcoder.net/bib.html` accessed on 2023/Oct/19.

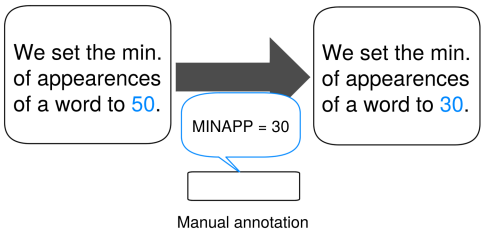

Fig. 2. Swapping, replacement of a value with another in a manually annotated dataset.

4.2 Data Augmentation

Since our task is novel and hence the size of our dataset is not large enough for machine learning, we propose four methods for augmenting the data. In the subsequent part, we explain these one by one.

Swapping The first method of data augmentation is swapping [4]. Swapping is a simple algorithm that replaces values in configurations by different existing values (Figure 2).

More precisely, we created first the list of all the configurations obtained from Texylon (§4.1) for a function f (co-occurrence network or correspondence analysis). Call this list $L_{\mathrm{ann}}(f) \subset \mathcal{C}_{\mathsf{KC}}(f)$. Next, for each paper and each function f that is used there, let its configurations be $c : K \to \mathcal{V}_{\mathsf{KC}}(f)$ for some $K \subset \mathcal{K}_{\mathsf{KC}}(f)$. For each key $k \in K$, we randomly selected a new value $v'_k \in \mathcal{V}_{\mathsf{KC}}(f, k)$ from the list $L_{\mathrm{ann}}(f)$ which belonged to the same key k. Then we collected the new key-value pairs $\{(k, v'_k)\}_{k \in K}$, and we got the new configurations $c' : k \mapsto (k, v'_k)$. We repeated this operation 1,000 times per configurations c, until we add 20 new configurations.

We made new corresponding descriptions as well by simple pattern matching. Because different keys have different values ($\mathcal{V}_{\mathsf{KC}}(f, k) \cap \mathcal{V}_{\mathsf{KC}}(f, k') = \emptyset$) in Texylon, a simple replacement was enough to produce the new descriptions unambiguously.

Pseudo-labeling Pseudo-labeling is the second method to enlarge our dataset. Here we suggest two ways for pseudo-labeling: BERT-NER and BERT-T5, as discussed below.

We prepared target texts to pseudo-label by downloading all the papers of KH Coder experiments found in the official list (as mentioned in §4.1) on October 11th, 2022. Note that Texylon was made from the papers in 2021; therefore we excluded the year 2021 from those target texts.

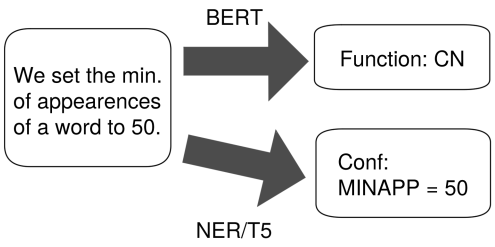

Fig. 3. Pseudo-labeling: BERT classifies functions and NER or T5 extract configurations.

BERT-NER BERT is a transformer-based language representation model proposed by Devlin et al. [5] We used BERT to classify functions ($\in \mathcal{F}_{\mathsf{KC}}$) used in papers: i) co-occurrence network, ii) correspondence analysis, and iii) others (see Figure 3). We adopted a BERT model pre-trained on Japanese Wikipedia articles[4], and then it was fine-tuned on Texylon, so that it can predict the function given a description.

On the other hand, we predicted configurations by named entity recognition (NER). Before applying NER, we tokenized the descriptions in Texylon into words with a Japanese morphological analyzer, MeCab [11]. For annotating NER labels, we preset 19 types of named entity (NE) specially for this task, each of which corresponds to each *key* in the possible configurations. Then, we annotated the *values* word-by-word in the IOB2 (inside-outside-beginning) format. For example, a label `B-PART` means the "beginning of a value" for the configuration 分析対象の品詞 (*bunseki taishou no hinshi*, target parts of speech), and a label `I-MINAPP` indicates that a word is "inside a value" for the configuration 語の最小出現数 (*go no saishou shutsugen suu*, the minimum number of appearances of a word).

After annotating, the NER model was trained via the Flair framework [1]. We utilized the Japanese word embeddings provided with Flair by default. Furthermore, we eliminated the NE entries whose confidence scores are less than 0.5 to decrease noisy data.

BERT-T5 As in BERT-NER, we classified the target papers by BERT into three classes of functions ($\in \mathcal{F}_{\mathsf{KC}}$), namely i) co-occurrence network, ii) correspondence analysis, and iii) others (see Figure 3 again). We discarded the "others" class, and, for the classes i) and ii), we produced logs given the descriptions by a T5 model. T5 is a language model first introduced by Raffel et al. [15], with the great focus on transfer learning.

[4] `https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking` accessed on 2023/Oct/19.

Table 2. NE tags of co-occurrence network and correspondence analysis.

NE tag	Meaning	Freq.
MINAPP	Min. # of appearances of a word	79
MET	Method of calculation of co-occurrence relations	79
TOP	Display the top N co-occurrence relations	73
EXT	External variables	41
METNUM	Threshold for co-occurrence	34
BOLD	Bolder line for stronger co-occurrence relations	25
BUBBLE	Bubble chart	25
SUBG	Type of community detection	21
PART	Target parts of speech	13
UNIT	Unit of analysis	10
TYPE	Type of co-occurrence relations	10
TREE	Display only MST	8
MINDOC	Min. document frequency	8
DIF	Analyze only words with prominent differences	6
ORI	Display only word labels far from the origin	3
MAXAPP	Max. # of appearances of a word	3
STAN	Standardize coefficients	3
BUBSIZE	Size of a bubble chart	1
MAXDOC	Max. document frequency	1
Total		443

Our model was pre-trained on Japanese Wikipedia articles[5], and the pre-trained model was transferred to our dataset Texylon.

Hybrid The final method of data augmentation, the hybrid dataset, is just the conjunction of the datasets of swapping and BERT-NER.

5 Evaluations

We evaluated our dataset Texylon and the data augmentation methods for the log-to-description task and the description-to-log task separately, to find out how our augmentation methods are useful for the tasks.

[5] `https://huggingface.co/sonoisa/t5-base-japanese` accessed on 2023/Oct/19.

5.1 Multi-task Generation Model

As a baseline model for our tasks (§3), we adopted one proposed by Kale et al. [9], which was based on T5 [15]. Here our task was defined as "translation" between descriptions and logs; hence we regarded the datasets as collections of correct translations from one language (namely "log") to another ("description"), and vice versa. We distinguished the two tasks from each other by prepending prefixes like "`Translate A to B:`," as it is suggested for a multi-task mixture [15].

To be precise, to each description-log pair (d, ℓ), we associated two data as follows that represent inputs and outputs for training the model. The first data has as an input "`Translate log to text:` ℓ" and its output is d. On the other hand, the second data is a pair of an input "`Translate text to log:` d" with an output ℓ. We passed to the model all the data which were constructed in that way.

5.2 Experiment Settings

Unless otherwise specified below, all the hyperparameters were set to the default values, or values suggested by inventors/implementers of models.

BERT-NER For the Flair framework, we set an initial learning rate to 0.5, with a batch size 8. For a BERT model, we used Adam [10] as an optimizer with a learning rate 1e-5 and the maximum number of input tokens 256. We trained a model for 20 epochs.

BERT-T5 The hyperparameters to BERT are the same as BERT-NER. For a T5 model, we adopted another optimizer, Adafactor [16], with a learning rate 1e-3, a batch size 8 and the number of steps 100,000.

Multi-task Generation Model We passed the same hyperparameters as BERT-T5 to a T5 model.

5.3 Cross validation

As the tasks are novel, the datasets are not large enough. Thus we followed the 5-fold cross validation to have more reliable results. First we split our dataset, Texylon, randomly into five equal-sized subsets. Then for each fold, we took one subset for the test and executed the following procedures:

1. form the training-validation subsets from the four subsets other than the test set,
2. split the training-validation subset into training/validation data with a ratio of 4/1 randomly,

3. augment the training data using the methods explained in §4.2 to have additional data,
4. train the generative model (§5.1) on the concatenation of the training data and the additional data, and
5. measure the performance of the model on the test subset.

Finally we calculated the averages of evaluation metrics.

5.4 Metrics

Log-to-Description Task We adopted the following metrics, which are widely used for natural language generation tasks, to calculate to what extent generated descriptions and reference texts coincide:

1. BLEU-N [13] — weighted geometric mean of p_n ($1 \leqslant n \leqslant N$) with the brevity penalty, where p_n is the n-gram precision,
2. METEOR [2] — weighted harmonic mean of the uni-gram precision and the uni-gram recall with penalty on adjacent uni-grams,
3. BERTScore [17] — the precision/recall/F1 scores of similarities of token embeddings.

Description-to-Log Task The metric for the description-to-log task is F1 scores, since a log is the well-structured data and it does not need any metrics for natural language evaluations, contrary to the log-to-description task. Thus we calculated the F1 scores for two components, namely for functions and for configurations. In the calculation, if an output is not in the right format (see §4.1), we ignored it; hence the F1 scores are calculated only among well-formatted logs.

5.5 Results

You can see example outputs for both the tasks in the Tables 3 and 4 respectively.

The results for log-to-description are shown in Table 5. Texylon alone achieved the highest scores for most of the metrics, and the hybrid was the second best (or higher) for all the metrics. It implies that the data augmentations may lower the accuracy in our case. We guess that the main reason is noises in the augmented data.

Table 6 shows the results for description-to-log. As we see in the table the model successfully predicted functions with a high accuracy, and there were no remarkable differences among the datasets. Moreover, the model produced few errors for most of the datasets.

We find the reasons of drawbacks be in the datasets themselves and in the models. Because the augmented data can contain a certain amount of noises, it results in errors for generated texts. The format of logs (§4.1) causes the low accuracy as well when the language models used for the data augmentation did not have a plenty of examples in the same format. Therefore, the models can have failed to recognize them.

Table 3. Examples of description generation from an input log. The rightmost column represents the English translation. We preserved grammatical or orthographic errors in English translation as much as possible.

Input	共起ネットワーク (語の最小出現数=25, 上位=60)	`CN(MinApp=25, Top=60)`
Texylon	共起ネットワーク分析では，最小出現数を 25 に設定し，描画する共起関係の絞り込みにおいては描画数を 60 に設定した．	For co-occurrence network, we set the min. # of appearances to 25, and set the # of renderings to 60 in the filter of rendered co-occurrence relations.
Swapping	次に，頻出語の最小出現数を 24 回に設定し，描画する共起関係の絞り込みにおいては，描画数を 60 に設定して，関連が特に強い語同士を線で結んだ共起ネットワーク注 2) を作成した．	Next, we set the min. # of appearances of frequent words to 24, set the # of renderings to 60 in the filter of rendered co-occurrence relations, and created a co-occurrence network (Notion 2) that links words of particularly strong relations with lines.
BERT-NER	図 12012 年版『小学校学習指導要領解説社会編』における頻出語の共起ネットワーク図 (出現回数 25 回以上・共起関係上位 60) (kh coder3 より，筆者作成)	Figure 1 figure of a co-occurrence network for frequent words (# of appearances $\geqslant$ 25, top 60 co-occurrence relations) in 2012 ver. "Course of Study for Elementary School: Social Studies" (From kh coder3, the author created)
BERT-T5	表 4 は, これらの抽出語のうち, 出現頻度が高い上位 60 語を表 4 に, 共起ネットワークを図 1 に示す.	For Table 4, among these extracted words, we show the top 60 words of highest frequency in Table 4, and a co-occurrence network in Figure 1.
Hybrid	図-24 は共起ネットワークを表す．出現数による単語の取捨選択では最小出現数 24 に設定し，描画する共起関係の絞り込みでは描画数を 60 に設定した．	Figure-24 shows a co-occurrence network. We set the min. # of appearances to 24 in the word filter by the # of appearances, and set the # of renderings to 60 in the filter of rendered co-occurrence relations.

Table 4. Examples of log generation from an input description. The rightmost column represents the English translation.

Input	「失敗経験」の内容についてkhcoder(樋口,2014) の共起ネットワーク分析を行なった．解釈可能性の観点から分析設定は，ひらがなのみの名詞・動詞・形容詞・副詞・否定助動詞・形容詞 (非自立) は削除，出現数は 10 以上の語を分析対象とし，jaccard 係数が上位 50 の共起関係を描画した．	We conducted co-occurrence network analysis of khcoder (Higuchi, 2014) about "failure experience." From a point of view of interpretability, for analysis configurations, we deleted hiragana nouns, verbs, adjectives, adverbs, negative auxiliary verbs, adjectives (ancillary). The target of analysis is words with the # of appearances $\geqslant$ 10, and we rendered co-occurrence relations of top 50 Jaccard coefficient.
Texylon	共起ネットワーク (語の最小出現数=10, 描画する共起関係の選択=jaccard 係数, 上位=50)	`CN(MinApp=10, Select=jaccard coefficient, Top=50)`
Swapping	共起ネットワーク (語の最小出現数=10, 描画する共起関係の選択=jaccard 係数, 上位=50)	`CN(MinApp=10,Select=jaccard coefficient, Top=50)`
BERT-NER	共起ネットワーク (品詞による語の取捨選択=名詞・動詞・形容詞・副詞・否定助動詞・形容詞 (非自立) に削除, 語の最小出現数=10, 描画する共起関係の選択=jaccard 係数, 上位=50)	`CN(SelectByPos=delete with nouns ... adjectives (ancillary), MinApp=10, Select=jaccard coefficient, Top=50)`
BERT-T5	共起ネットワーク (語の最小出現数=10, 描画する共起関係の選択=jaccard 係数, 上位=50)	`CN(MinApp=10,Select=jaccard coefficient, Top=50)`
Hybrid	共起ネットワーク (品詞による語の取捨選択=名詞・動詞・形容詞・副詞・否定助動詞・形容詞 (非自立) に, 語の最小出現数=10, 描画する共起関係の選択=jaccard 係数, 上位=50)	`CN(SelectByPos=with nouns ... adjectives (ancillary), MinApp=10, Select=jaccard coefficient, Top=50)`

Table 5. Results for the log-to-description task.

Dataset Type	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	BERT-P	BERT-R	BERT-F1
Texylon	**36.98**	**24.47**	**18.20**	**14.64**	**30.35**	0.73	**0.74**	**0.74**
Swapping	34.06	22.24	16.20	12.85	29.39	0.72	0.73	0.73
BERT-NER	33.07	21.33	15.32	11.89	27.81	0.73	0.73	0.73
BERT-T5	24.88	15.70	10.90	8.13	23.94	**0.75**	0.71	0.72
Hybrid	35.84	23.85	17.43	13.74	29.40	0.74	**0.74**	**0.74**

Table 6. Results for the description-to-log task. F1-func for the classification of functions, and F1-conf for generating configurations. The errors are the total numbers of ill-formatted logs.

Dataset Type	F1-func	F1-conf	Errors
Texylon	0.962	0.711	3
Swapping	0.960	0.698	4
BERT-NER	0.943	**0.717**	0
BERT-T5	0.947	0.692	12
Hybrid	**0.967**	0.704	1

6 Conclusion

We introduced Texylon, the dataset of description-log pairs, and proposed four types of data augmentation of that dataset. Then we considered the multi-task generative model that converts logs to descriptions and vice versa as a baseline model to evaluate the data augmentation methods. Although Texylon itself exhibited the highest scores for log-to-description, the data augmentation methods equally high scores for description-to-log. The reasons can be that the augmented data contain noises, or that the baseline model was not suitable for our dataset(s).

Our next plan is to invent a generative model more sensible to the format. Then, we should employ a new method for data augmentation which makes better data than Texylon. We would like to build a "general" model as well in a sense that it is applicable to other text-anatytics tools than KH Coder.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., Vollgraf, R.: FLAIR: An easy-to-use framework for state-of-the-art NLP. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations). pp. 54–59. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-4010, https://aclanthology.org/N19-4010

2. Banerjee, S., Lavie, A.: 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. Association for Computational Linguistics, Ann Arbor, Michigan (Jun 2005), https://aclanthology.org/W05-0909
3. Berry, D.M.: Introduction: Understanding the Digital Humanities, pp. 1–20. Palgrave Macmillan UK, London (2012). https://doi.org/10.1057/9780230371934_1, https://doi.org/10.1057/9780230371934_1
4. Chang, E., Shen, X., Zhu, D., Demberg, V., Su, H.: Neural data-to-text generation with LM-based text augmentation. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. pp. 758–768. Association for Computational Linguistics, Online (Apr 2021). https://doi.org/10.18653/v1/2021.eacl-main.64, https://aclanthology.org/2021.eacl-main.64
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
6. He, S., He, P., Chen, Z., Yang, T., Su, Y., Lyu, M.R.: A survey on automated log analysis for reliability engineering. ACM Comput. Surv. **54**(6) (jul 2021). https://doi.org/10.1145/3460345, https://doi.org/10.1145/3460345
7. Higuchi, K.: A two-step approach to quantitative content analysis: KH Coder tutorial using anne of green gables (Part I). Ritsumeikan Social Science Review **52**, 77–91 (Dec 2016), http://hdl.handle.net/10367/8013
8. Higuchi, K.: A two-step approach to quantitative content analysis: KH Coder tutorial using anne of green gables (Part II). Ritsumeikan Social Science Review **53**, 137–147 (Jun 2017), http://hdl.handle.net/10367/8610
9. Kale, M., Rastogi, A.: Text-to-text pre-training for data-to-text tasks. In: Proceedings of the 13th International Conference on Natural Language Generation. pp. 97–102. Association for Computational Linguistics, Dublin, Ireland (Dec 2020), https://aclanthology.org/2020.inlg-1.14
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2017), https://arxiv.org/abs/1412.6980
11. Kudo, T., Yamamoto, K., Matsumoto, Y.: Applying conditional random fields to Japanese morphological analysis. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. pp. 230–237. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), https://aclanthology.org/W04-3230
12. Novikova, J., Dušek, O., Rieser, V.: The E2E dataset: New challenges for end-to-end generation. In: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue. pp. 201–206. Association for Computational Linguistics, Saarbrücken, Germany (Aug 2017). https://doi.org/10.18653/v1/W17-5525, https://aclanthology.org/W17-5525
13. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: 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. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA (Jul 2002). https://doi.org/10.3115/1073083.1073135, https://aclanthology.org/P02-1040

14. Parikh, A., Wang, X., Gehrmann, S., Faruqui, M., Dhingra, B., Yang, D., Das, D.: ToTTo: A controlled table-to-text generation dataset. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1173–1186. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.89, https://aclanthology.org/2020.emnlp-main.89
15. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research **21**(140), 1–67 (2020), http://jmlr.org/papers/v21/20-074.html
16. Shazeer, N., Stern, M.: Adafactor: Adaptive learning rates with sublinear memory cost. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 4596–4604. PMLR (10–15 Jul 2018), https://proceedings.mlr.press/v80/shazeer18a.html
17. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with bert. In: International Conference on Learning Representations (2020), https://arxiv.org/abs/1904.09675