Learning with Contrastive Examples for Data-to-Text Generation

Yui Uehara[*†] **Tatsuya Ishigaki**[*†] **Kasumi Aoki**[○] **Keiichi Goshima**[†◇] **Hiroshi Noji**[†]
Ichiro Kobayashi[†○] **Hiroya Takamura**[†•] **Yusuke Miyao**[†‡]

[†]National Institute of Advanced Industrial Science and Technology, Japan
[○]Ochanomizu University [•]Tokyo Institute of Technology
[◇]Waseda University [‡]The University of Tokyo

{yui.uehara, ishigaki.tatsuya, noji, takamura.hiroya}@aist.go.jp
{g1120501, koba}@is.ocha.ac.jp
keiichi.goshima@aoni.waseda.jp yusuke@is.s.u-tokyo.ac.jp

Abstract

Existing models for data-to-text tasks generate fluent but sometimes incorrect sentences e.g., "*Nikkei gains*" is generated when "*Nikkei drops*" is expected. We investigate models trained on contrastive examples, that is, incorrect sentences or terms, in addition to correct ones to reduce such errors. We first create rules to produce contrastive examples from correct ones by replacing frequent crucial terms such as "*gain*" or "*drop*". We then use learning methods with several losses that exploit contrastive examples. Experiments on the market comment generation task show that 1) exploiting contrastive examples improves the capability to generate sentences with better lexical choices, without degrading the fluency, 2) the choice of the loss function is an important factor because the performances of different metrics depend on the types of loss functions, and 3) the use of the examples produced by some specific rules further improves performance. Human evaluation also supports the effectiveness of using contrastive examples.

1 Introduction

We address the task of generating market comments from stock prices as illustrated in Fig. 1. This can be seen as a data-to-text generation task. Recently, neural data-to-text generation has been studied in a wide range of domains such as biography (Lebret et al., 2016; Liu et al., 2018), sports recap (Wiseman et al., 2017; Puduppully et al., 2019a; Puduppully et al., 2019b; Iso et al., 2019; Gong et al., 2019), and market comments (Murakami et al., 2017; Aoki et al., 2018; Aoki et al., 2019).

These models generate fluent sentences, but we often observed problematic generated sentences in terms of correctness. As shown in Fig. 1, the word *gain* is possibly generated, although the word *drop* or *rebound* is expected. The terms that express the fluctuation of stock prices are crucial because such errors could reverse the meaning of the sentence in the worst case.

Similar issues have been seen in other generation tasks, such as machine translation or summarization. The known solutions are, for example, the use of alignments between input and output (Sennrich, 2017; Arthur et al., 2016) or copy mechanisms (See et al., 2017). However, they cannot be directly applied to our task because ours treat sequences of numerical values as an input.

In this paper, we consider how to alleviate such errors by using contrastive examples, which are identical to the correct examples except for a single word: *Nikkei gained* vs. *Nikkei dropped*. Learning with such examples provides models direct signals on the words that are not to be generated in addition to those to be generated. We propose a learning framework to examine how to use such examples from the viewpoint of loss functions and rules to create contrastive examples.

Recent studies show the effectiveness of learning methods that exploit explicit negative examples in language modeling. Huang et al. (2018) introduced a margin loss to penalize sentences in a beam, assuming that the generated sentences are imperfect. Noji and Takamura (2020) used synthesized ungrammatical sentences in addition to the originals to improve the syntactic ability of language models.

[*]The first and second authors equally contributed to this work.

Proceedings of the 28th International Conference on Computational Linguistics, pages 2352–2362
Barcelona, Spain (Online), December 8-13, 2020

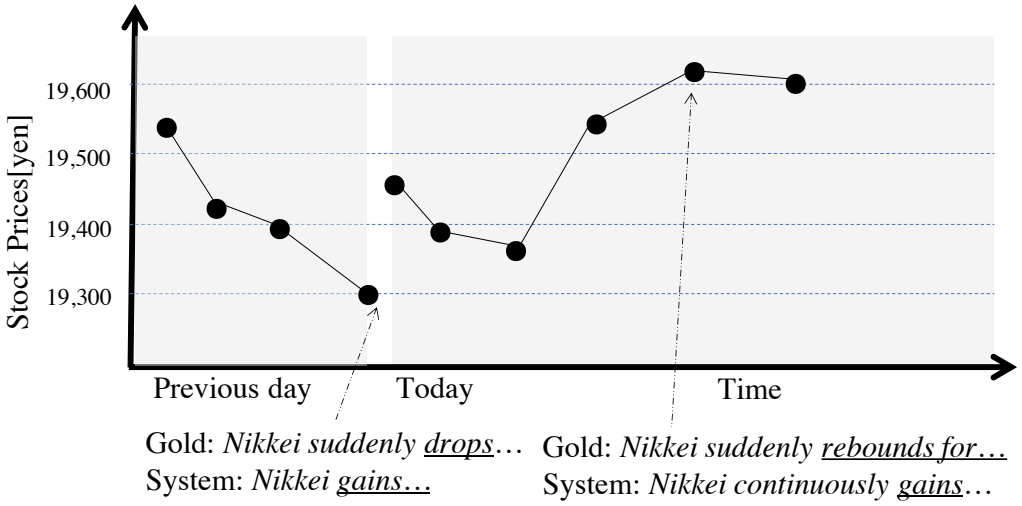

Figure 1: An example of translated gold comments and generated comments by a system. The generated comments contain an erroneous antonym (*drops* vs. *gains*) or a term that does not correctly capture the movement (*rebounds* vs. *continuously gains*).

For generation, Welleck et al. (2020) proposed a model with an unlikelihood loss to alleviate the repetition problem. Rather, we focus on improving the correctness of generated sentences, which is crucial for data-to-text tasks.

Experiments show that 1) our models can generate sentences with better lexical choice, without degrading fluency, 2) the effectiveness of the loss functions depends on the evaluation metric and we need to select an appropriate loss function based on the criteria we prioritize, and 3) from the perspective of rules for producing contrastive examples, it is more effective to replace a word with a relatively closer meaning than its antonyms. Our implementation is publicly available[1].

2 Framework

We introduce our learning framework with several losses that exploit contrastive examples. The main aim of this study is to investigate whether models can generate crucial terms more correctly if we train them with contrastive examples.

2.1 Rules for Producing Contrastive Examples

Contrastive examples are divided into two types: contrastive terms and sentences. These are used in the calculations of the losses. We first select the eight most frequent terms in the training dataset that directly indicate the fluctuation of the stock price and define them as crucial terms. We extract combinations of pairs of these eight terms as the rules to produce contrastive terms from a crucial term. We take advantage of wider applicability by defining crucial words in this simple strategy. We show the rules in Table 1. We create contrastive sentences by replacing the crucial terms in the dataset. Note that we exclude the rules that produce ungrammatical sentences. We use a Japanese dataset in the experiments. All the rules in Table 1 are either single noun-to-noun or adjective-to-adjective conversions, and these terms do not have plural or inflected forms. Thus, simply replacing the terms by the rules rarely produces ungrammatical sentences. 77.3% of the sentences (12,589 out of 16,276) in the training dataset contain one or more of the eight terms.

Rules
continual rise (続伸) ⇒ {continual fall, rebound, turn down},
continual fall (続落) ⇒ {continual rise, rebound, turn down},
rebound (反発) ⇒ {continual rise, continual fall, turn down},
turn down (反落) ⇒ {continual rise, continual fall, rebound},
gain (上げ幅) ⇒ {loss},
loss (下げ幅) ⇒ {gain},
high (高) ⇒ {low},
low (安) ⇒ {high}

Table 1: The rules to produce contrastive examples. The terms in brackets are the originals in Japanese.

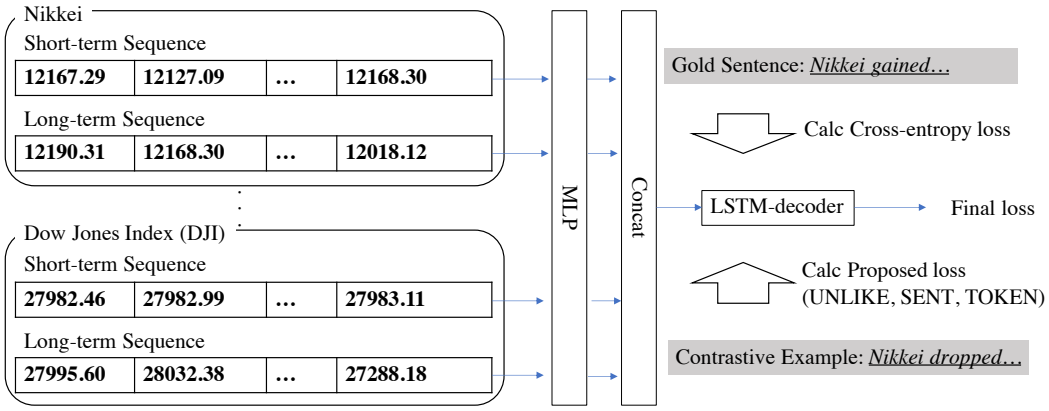

Figure 2: Learning methods that exploit contrastive examples in addition to gold sentences. We compare three different losses that take into account contrastive examples.

2.2 Learning Methods

In this subsection, we introduce our learning methods, which exploit contrastive examples, in addition to a baseline model, which does not use contrastive examples. The overview of our method is shown in Fig. 2.

2.2.1 Baseline with Cross-entropy Loss (BASE)

We use Aoki et al. (2018)'s model as a base model. This is an encoder-decoder, in which the encoder separately encodes 10 indices such as the Nikkei or Dow Jones Industrial Average, and then the LSTM-based decoder generates a market comment as a sequence of words. Each index has prices that are tracked every five minutes and is represented as two different sequences of numerical values; short-term and long-term. A short-term sequence consists of previous N prices in a day. A long-term sequence consists of the closing prices of M preceding trading days. These sequences are converted to fixed-length vectors by using three layers of Multi-Layer Perceptron (MLP). The concatenated vector of all the vectors is then fed into the decoder to generate a market comment. We train this baseline by using the cross-entropy loss. We propose to apply three different losses that take into account contrastive examples, as explained in the following.

2.2.2 Unlikelihood Loss (UNLIKE)

This is recently proposed by Welleck et al. (2020) for reducing repetitions in generation tasks. They used this loss to penalize choosing words generated before. Instead, we use this loss to penalize choosing contrastive terms to improve correctness. Given a sentence $\mathbf{x}$, we calculate the unlikelihood loss as:

$$\sum_{x_i \in \mathbf{x}} -\log p(x_i|x_{1:i-1}) + \sum_{x_i^* \in con(x_i)} g(x_i^*), \quad (1)$$

[1] https://github.com/aistairc/contrastive_data2text

$$g(x_i^*) = -\alpha \log(1 - p(x_i^*|x_{1:i-1})), \tag{2}$$

where $con(x_i)$ returns contrastive terms of x_i by using the rules in Table 1. α balances the importance between the two terms, where the first term learns the language model from the correct tokens, while the second term penalizes the contrastive terms. We finetuned α on the validation dataset.

2.2.3 Sentence-level Margin Loss (SENT)

This attempts to guarantee a certain margin of log-likelihoods between a sentence $\mathbf{x}$ and its contrastive sentence $\mathbf{x}^*$ as follows:

$$\max(0, \delta - (\log p(\mathbf{x}) - \log p(\mathbf{x}^*))), \tag{3}$$

where δ is the margin between the log-likelihoods of $\mathbf{x}$ and $\mathbf{x}^*$. This was originally proposed for analyzing the syntactic abilities of language models (Noji and Takamura, 2020). This loss is useful for developing better language models. However, the token-level supervision is missing, which may provide a more direct signal to learn a clear contrast between correct and contrastive terms. Regarding the training, we also use cross-entropy loss. For each batch, we first use only the original sentences to optimize the parameters by minimizing the cross-entropy loss. We then generate a set of contrastive sentences from the original sentences that contain at least one crucial term. If a single sentence contains multiple crucial terms, we randomly select one of them. We sample a certain number of pairs to further update the parameters by using the averaged sentence-level margin loss over the pairs in the batch. We set the number to half the size of the batch in our experiments.

2.2.4 Token-level Margin Loss (TOKEN)

Noji and Takamura (2020) also use a combination of the previous two by replacing $g(x_i^*)$ in Eq. (1) as:

$$g(x_i^*) = \max(0, \delta - (\log p(x_i|x_{1:i-1}) - \log p(x_i^*|x_{1:i-1}))).$$

This loss attempts to take advantage of both the sentence-level margin loss in terms of language modelling and the unlikelihood loss in terms of strong token-level supervision for contrastive terms.

3 Experiments

In this section, we describe the dataset used for our experiments, the ways to finetune hyperparameters and metrics for automatic evaluation in addition to manual evaluation by a human judge.

3.1 Dataset

We use the dataset preprocessed by Aoki et al. (2018). The dataset consists of 10 market indices[2] and Nikkei and corresponding market comments extracted from Nikkei Quick News. Specifically, these numerical sequences are seven stock market indices retrieved from the ThomsonReutersDataScopeSelect[3] (see Aoki et al. (2018) and their publicly available implementation[4] for details). The statistics of the dataset are shown on Table 2. We follow the task proposed by Aoki et al. (2018) to generate market comments forthe Nikkei, using the numerical sequences of the Nikkei and the other nine additional indices.

3.2 Parameters

We finetuned the margin δ for SENT and TOKEN and the parameter α of UNLIKE that balances the term for language modeling and that for penalizing contrastive terms on the validation dataset. These are selected from $\{0.01, 0.1, 1.0, 10, 100\}$. We selected the models that achieve the best in terms of the different evaluation criteria explained in the next subsection.

We set $N = 62$ for short-term sequences, and $M = 7$ for long-term sequences. Regarding the training, the mini-batch size is set to 50. We trained the models for 100 epochs and saved the parameters

[2]Specifically, the indices are Nikkei (N225), N225 Forward Transaction Index, Tokyo Stock Price Index (TOPX), S&P 500 (SPX), Dow Jones Industrial Average (DJI), FTSE 100 Index (FTSE), Hong Kong Hang Seng Index (HSI), the exchange rate of Japanese yen and the US dollar (JPYUSD), the exchange rate of Euro and Japanese yen (EURJPY), and JNIc1.

[3]https://hosted.datascope.reuters.com/DataScope/

[4]https://github.com/aistairc/market-reporter

	Train	Valid	Test
# of sentences	16,276	1,866	1,951
# of sentences with crucial terms	12,589	1,583	1,615
# of crucial terms	19,005	2,592	2,634
average # of tokens	13.17	12.77	12.69

Table 2: The statistics of the dataset.

every epoch, then selected the best model on validation dataset. We used Adam (Kingma and Ba, 2015) optimizer with the initial learning rate 0.001. Each index was converted to a 32-dimensional vector. The dimensions for the hidden layer in the encoder and the decoder were set to 256. We used 128 for the dimensions of word embeddigns. We report the averaged values of three trials with different random seeds for automatic evaluation.

3.3 Automatic Evaluation

Since we aim to improve correctness, only using BLEU (Papineni et al., 2002) is not sufficient. It is ideal to evaluate the effect of the use of contrastive pairs from various perspectives. We propose four metrics to capture how well models exploit contrastive examples and generate crucial terms.

3.3.1 Accuracy

We expect the trained model should correctly distinguish the difference between reference sentences and their contrastive sentences i.e., assign a higher probability to the reference sentences than its contrastive sentences as a direct effect of the learning with the losses that take into account contrastive examples. Therefore, following the work by Sennrich et al. (2017), we compare the likelihood of each reference sentence and those of its contrastive sentences. The winning ratio of reference sentences, which is referred to as accuracy. We denote it by A_{fluc}.

In contrast to the training setup, we use all possible contrastive sentences for this evaluation. A reference sentence wins when its likelihood is higher than those of all possible contrastive sentences.

3.4 Precision and Recall

As we explained in Sec. 2.1, the terms in Table 1 are crucial, in that these terms directly express the fluctuation of the stock price and the incorrect generation of such terms would reverse the meaning of the sentence. We therefore evaluate how accurately the models generate these terms. In particular, we calculate the precision and the recall for crucial terms. We define the **recall** (R_{fluc}) as the number of the correctly generated crucial terms divided by the number of crucial terms in the reference sentences. Similarly, we define the **precision** (P_{fluc}).

3.5 Error rate

Recall and precision defined above are not completely ideal automatic evaluation criteria because the same meaning can be expressed by other terms that are not frequent and are not listed in Table 1. We therefore propose criteria that directly capture the extent to which the model fails to generate crucial words. We define **error rate** (E_{fluc}) as the ratio of the number of sentence pairs for which the reference sentence contains one or more crucial terms in Table 1, but the generated sentence contains one or more of its contrastive terms.

Note that when calculating the scores R_{fluc}, P_{fluc}, and E_{fluc}, we exclude sentence pairs whose reference sentence contains both a crucial term and its contrastive terms.

3.6 Human Evaluation

Human evaluation is essential to verify the effectiveness of contrastive examples. We create two datasets for human evaluation. `WHOLE` is a dataset that contains 100 randomly sampled instances from the test data. `CRUCIAL` is a dataset that contains 40 randomly sampled instances of which `BASE` and `TOKEN`

	BLEU	Accuracy (A_{fluc})	Recall (R_{fluc})	Precision (P_{fluc})	Error rate (E_{fluc})
BASE	26.01	90.04	74.78	62.27	7.69
Our models tuned on BLEU					
UNLIKE	25.54	91.17	75.17	63.10	6.79
SENT	**26.56**	90.26	75.82	62.44	7.56
TOKEN	25.90	91.10	75.01	63.25	6.91
Our models tuned on A_{fluc}					
UNLIKE	23.26	**93.02**	72.48	61.24	6.83
SENT	23.93	**91.19**	78.86	55.39	8.69
TOKEN	25.90	**91.74**	75.67	63.54	6.08
Our models tuned on R_{fluc}					
UNLIKE	25.98	91.17	**75.30**	63.10	6.79
SENT	23.93	91.19	**78.86**	55.39	8.69
TOKEN	26.07	91.46	**75.67**	61.95	7.05
Our models tuned on P_{fluc}					
UNLIKE	25.54	90.84	75.30	**63.05**	6.15
SENT	26.69	92.51	76.65	**63.62**	6.65
TOKEN	25.90	91.74	75.67	**63.54**	6.08
Our models tuned on E_{fluc}					
UNLIKE	25.54	92.51	75.30	63.05	**6.16***
SENT	25.37	89.45	75.63	63.34	**7.15***
TOKEN	25.90	91.74	75.67	63.54	**6.08***

Table 3: The automatic evaluation results. * means that the reduction in E_{fluc} from that of BASE was statistically significant by sign test ($p < 0.05$). Scores are in bold if they are better than BASE and for the metric tuned on. F_{fluc} is the harmonic mean of R_{fluc} and P_{fluc}. Higher values are better except for E_{fluc}.

generate different crucial terms. The latter enables us to directly evaluate the improvements for the crucial parts where our models attempt to reduce errors.

An expert in finance domain was asked to rank the sentences in terms of two criteria: **correctness** and **fluency** following Aoki et al. (2018). *Correctness* evaluates how well the movement of the input indices was correctly captured whereas *fluency* is based on naturalness as natural language. We allow the evaluator to equally rank two or more sentences. For each instance, we displayed the reference sentence (REF), the sentences generated by the baseline (BASE) and our model that achieves the best error rate on the validation dataset (TOKEN).

For the evaluation in terms of correctness, we also showed graphs that represent the fluctuation of each index. The evaluator checked both the generated sentences and various graphs and then ranked the sentences. The evaluator removed the instances if the evaluator could not judge correctness by using only the information from the graphs. For example, the generated sentence of *Nikkei drops caused by the mention from the Governor of the Bank of Japan* could not be strictly judged in terms of correctness because it included the writer's subjective thoughts on the reason for the drop and we cannot know the actual reason.

4 Results

In this section, we discuss the results of automatic and human evaluations.

4.1 Effectiveness of Contrastive Examples.

Table 3 shows the results of the automatic evaluation. The table is divided into six sections from the top to bottom. The first section shows the scores of BASE, and the following sections are the scores

	WHOLE	CRUCIAL
TOKEN vs. BASE	19-18	32-5
REF vs. TOKEN	37-7	18-2
REF vs. BASE	38-8	35-1

Table 4: The results of human evaluation in terms of **correctness**. WHOLE contains 100 and CRUCIAL contains 40, respectively.

	WHOLE	CRUCIAL
TOKEN vs. BASE	0-0	0-1
REF vs. TOKEN	0-0	1-0
REF vs. BASE	0-0	0-0

Table 5: The results of human evaluation in terms of **fluency**. Scores of 0-0, 1-0 and 0-1 exist because the compared methods were almost equally ranked in terms of *fluency* because most of them are fluent. WHOLE contains 100 and CRUCIAL contains 40, respectively.

of our models. For each section of our models, we finetuned the hyperparameters based on different target metrics, that is, BLEU, accuracy, recall, precision, or error rate on the validation dataset. We then evaluated them on the test set. The scores in bold are better than those of BASE in terms of the target criteria used for tuning hyperparameters.

Our models perform better in terms of all metrics except for the BLEU for UNLIKE and TOKEN. These scores show that the use of contrastive examples improves the correct generation of crucial terms. The improvements in E_{fluc} show that errors that mistakenly select the crucial terms were effectively suppressed. The reductions in the BLEU scores of UNLIKE (25.54) and TOKEN (25.90) from BASE (26.01) are only 0.47 and 0.11, respectively. We did not observe any statistical difference between them. These small reductions show that our models improve correctness without reducing BLEU.

4.2 Comparisons between losses

The choice of the loss function is an important factor because the performance of different metrics depends on the types of loss functions. TOKEN achieved the best error rate (6.08) while SENT achieved the best scores in terms of BLEU (26.56), precision (63.62) and recall (78.86). UNLIKE achieved the best in terms of accuracy (93.02). Note that SENT is not stable because if we tuned the hyperparameter of SENT to achieve the best in terms of recall (78.86), we have to compromise the performance in terms of precision (55.39) and error rate (8.69). Similar instability can be seen for other metrics of SENT. We found that TOKEN and UNLIKE are more stable regardless of which criteria are used for tuning parameters.

Our models that use token-level supervision (UNLIKE and TOKEN) achieved better in terms of accuracy, precision and error rate than SENT, which uses sentence-level signals. SENT achieved better than UNLIKE and TOKEN in terms of BLEU, which we consider less important in this study since the correlation between BLEU scores and scores given by human judges in terms of correctness is unclear.

4.3 Results of human evaluation.

Table 4 and Table 5 show the results of human evaluation in terms of **correctness** and **fluency**, respectively. The numbers represent the counts that a method was judged better than the other. In terms of correctness, we observed statistically significant gains on CRUCIAL, where TOKEN was judged better than BASE 32 times, whereas BASE was judged better only 5 times. Furthermore, REF was judged better than TOKEN 18 times, whereas REF was judged better than BASE 35 times. Thus, the sentences generated by TOKEN are more similar to REF than those generated by BASE. These results show the usefulness of contrastive examples. We did not observe the performance reduction on WHOLE (19 vs. 18), which implies that the use of contrastive examples helps models correctly generate crucial terms without reducing the correctness of other parts.

E_{fluc}	All	Different type	Same type
BASE	7.69	7.69	7.69
UNLIKE	6.16	7.22	5.99
SENT	7.15	7.14	6.95
TOKEN	6.08	7.77	5.37

Table 6: Effectiveness of types of rules on error rates.

count	Gold	Generated crucial terms
87	**rebound (反発)**	**continuous gain (続伸)**
75	**continuous drop (続落)**	**turn down (反落)**
60	continuous drop (続落)	rebound (反発)
48	turn down (反落)	continuous gain (続伸)
38	**continuous gain (続伸)**	**rebound (反発)**
27	turn down (反落)	rebound (反発)
24	continuous gain (続伸)	turn down (反落)
16	rebound (反発)	continuous drop (続落)
15	**turn down (反落)**	**continuous drop (続落)**
13	continuous drop (続落)	continuous gain (続伸)
12	rebound (反発)	turn down (反落)
9	continuous gain (続伸)	continuous drop (続落)
6	low (安)	high (高)
4	high (高)	low (安)
3	gain (上げ幅)	reduction (下げ幅)

Table 7: The counts of errounous generations by BASE. The original Japanese terms are in brackets.

In terms of fluency, the results suggest that the use of contrastive examples did not reduce performance, as the compared methods were almost equally ranked (0 vs. 0, 0 vs. 1 or 1 vs. 0 for all pairs). The evaluator mentions that almost all sentences are fluent as natural language.

4.4 Effects of rules

We also analyze the effects of types of rules. We split the terms Table 1 into two: the terms that represent the fluctuations 1) that eventually gain e.g., "*continual rise*" or "*rebound*", and 2) that eventually drop e.g., "*continual fall*" or "*turn down*". Table 6 shows the error rates when we use only the rules that convert between the same types of terms (e.g., *continual rise* ⇒ *rebound*) and the different types of terms (e.g., *continual rise* ⇒ *continual fall*). As a result, both types of rules reduce the error rates compared to those of BASE except for TOKEN, which uses the rules of "Different type". Furthermore, the latter type of rules reduce the error rates better. This implies that the rules that convert terms to similar ones are more effective.

To further explore this result, we analyzed the output of Aoki et al. (2018)'s base model. Their model often mistakenly generates similar words to the correct ones. Table 7 shows the statistics of errors of their model. In the table, the counts of the errors that generate similar words are in bold. Most of such errors are ranked higher in this table. Their model also outputs the antonyms of the crucial term in the reference sentence, but such cases are less frequent than the cases that generate similar words. Therefore, it is a convincing result that the use of rules that replace similar words improves and the use of all rules further improves the performance. In this study, we used naive heuristics to create rules because we prioritize reducing labour costs, but it might be possible to make more effective contrastive examples based on detailed error analysis of existing models on the validation dataset.

In other generation tasks e.g., machine translation, Arthur et al. (2016) exemplifies an error of neural network-based models that incorrectly generates a similar word *Tunisia* instead of the correct word *Nigeria*. This error seems somewhat similar to those in our task, in which a model mistakenly generates

	Example 1
Ref	*Tokyo Stock Exchange closed its morning session with continual rise after continuous repurchases accelerated by a sudden drop* Tosho (東証) zenbike(前引け), zokushin (続伸). Kyuraku(急落)-kara(から)-no(の) kai(買い)-modoshi(戻し) tsuzuku(続く).
Base	*Nikkei* **rebounded***, and the closing price of the morning session was 15,751 yen, which is 69 yen higher higher.* Nikkei(日経) heikin(平均), **hanpatsu(反発)**, zenbike(前引け)-ha(は) 69 en(円) -daka(高)-no(の) 15,751 en(円).
Ours	*Nikkei continually rose and the closing price of the morning session is 15,751 yen, which is 69 yen higher.* Nikkei(日経) heikin(平均), zokushin(続伸). zenbike(前引け)-ha(は) 69 en(円)-daka(高)-no(の) 15,751 en(円).
	Example 2
Ref	*Nikkei stock average starts with turn down, the price of 16100 yen which is 70 yen lower.* Nikkei (日経) heikin (平均) 、 hanraku (反落) de (で) hajimaru (始まる), 70 en (円) yasu (安) no (の) 16100 en (円)
Base	*Nikkei stock average starts with turn down, and profit-taking due to the* **low prices of the US stock market** *was preceded in the market.* Nikkei (日経) heikin (平均) 、 hanraku (反落) de (で) hajimaru (始まる), **bei-kabuyasu (米株安)** de (で) rieki (利益) kakutei (確定) uri (売り) ga (が) senkou (先行)
Ours	*Nikkei stock average starts with turn down, and profit-taking is seen in the market due to the* **low prices of the US stock market.** Nikkei (日経) heikin (平均) 、 hanraku (反落) de (で) hajimaru (始まる) **beikabuyasu (米株安)** de (で) rieki (利益) kakutei (確定) uri (売り)

Table 8: Examples of generated sentences and its translations into English. The parts that do not correctly represent the fluctuation of the input indices are in **bold**.

similar incorrect crucial terms. Thus, we are interested in whether the use of contrastive examples works well for other generation tasks to improve correctness. We leave this for future work.

4.5 Example of generated sentences and error analysis

We show some generated sentences in Table 8.

In the first example, the main difference between the sentences generated by BASE and TOKEN is the lexical choice between *rebound* by BASE and *continualy rise* by TOKEN. This is a representative example to show that TOKEN generated crucial terms correctly.

In the second example, the both our model and the baseline generated the correct crucial terms (*turn down (反落)*), however, these generated wrong mentions on the US stock market (*low prices of the US stock market (米株安)*) although the US stock market actually gained. Nikkei starts after the US stock market closes and the decisions of investors for Nikkei are affected by the US stock market. Thus, Nikkei and the US stock market correlate each other for most cases. Although the base model and our models take into account both Nikkei and the US stock market in the encoder, BASE and TOKEN struggle to generate sentences that correctly express the less frequent phenomenon, that is, Nikkei turned down but the US stock gained.

A market comment often can be split into two parts where the first part describes the major fluctuation of the market (e.g., *Nikkei gained this morning*) and the second part provides supplementary information such as reason or detailed prices (e.g., *due to high prices in the US market.*). In this study, we focus on crucial terms, which are mostly observed in the first part. Thus, we observed errors in the second part of our generated sentences. In future work, it will be useful if we can develop effective contrastive examples that improve the latter half.

5 Related Work

Neural data-to-text generation has been studied in wide range of domains such as biography (Lebret et al., 2016; Liu et al., 2018), sports recap (Wiseman et al., 2017; Puduppully et al., 2019a; Puduppully et al., 2019b; Iso et al., 2019; Gong et al., 2019), and market comments (Murakami et al., 2017; Aoki et al., 2018; Aoki et al., 2019). Murakami et al. (2017) and Aoki et al. (2018) deal with sentence-level market comments generation tasks, whereas Aoki et al. (2019) generate document-level market comments that can be controlled by hand-crafted rules. We follow the most basic setup, that is, sentence-level market comments generation.

Although each domain of data-to-text tasks has its own difficulty, these studies showed that neural end-to-end approaches such as encoder-decoders can generate fluent text. However, the sentences generated by existing models are often problematic in terms of correctness. Significant developments have been made to capture input data correctly, for example, encoders with content selection (Puduppully et al., 2019a; Gong et al., 2019), decoders with entity modeling (Iso et al., 2019; Puduppully et al., 2019b). The problem in terms of correctness is also well knwon in other generation tasks such as machine translation tasks (Sennrich, 2017; Arthur et al., 2016) or summarization (See et al., 2017). The use of an alignment dictionary (Arthur et al., 2016) or copy mechanisms (See et al., 2017) are common strategies to reduce such errors, but these are difficult to adopt for data-to-text tasks.

In this study, our models use various loss functions that take into account contrastive samples. This approach relates to recent studies that propose loss functions that use negative samples for language modeling. Huang et al. (2018) introduced a margin loss that estimates the quality of each beam-searched candidate by comparing it with the reference sentence. More recently, Noji and Takamura (2020) showed negative examples help to improve the syntactic ability of neural language models. They created negative instances from original instances by injecting a grammatical error and used them to calculate a margin loss that will be added to the cross-entropy loss. For generation, Welleck et al. (2020) proposed a model with an unlikelihood loss to alleviate the repetition problem. Although their study targets neural language models or the different problems in generation other than improving correctness, we focus on improving the generated sentences in data-to-text tasks in terms of correctness.

6 Conclusion

We presented learning methods with several losses that exploited contrastive examples for data-to-text. The results showed our methods improved the performances in terms of correctness. Human evaluation also supported the improvements for the crucial parts that our model attempted to reduce errors. Because our methods have wide applicability, we will examine their effectiveness against other models and tasks. The applicability will be wider if effective contrastive examples can be generated automatically.

Acknowledgments

This paper is based on results obtained from projects JPNP20006 and JPNP15009, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). Computational resources of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) were used.

References

Tatsuya Aoki, Akira Miyazawa, Tatsuya Ishigaki, Keiichi Goshima, Kasumi Aoki, Ichiro Kobayashi, Hiroya Takamura, and Yusuke Miyao. 2018. Generating market comments referring to external resources. In *Proceedings of the 11th International Conference on Natural Language Generation (INLG2018)*, pages 135–139.

Kasumi Aoki, Akira Miyazawa, Tatsuya Ishigaki, Tatsuya Aoki, Hiroshi Noji, Keiichi Goshima, Ichiro Kobayashi, Hiroya Takamura, and Yusuke Miyao. 2019. Controlling contents in data-to-document generation with human-designed topic labels. In *Proceedings of the 12th International Conference on Natural Language Generation (INLG2019)*, pages 323–332.

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP2016)*, pages 1557–1567.

Heng Gong, Xiaocheng Feng, Bing Qin, and Ting Liu. 2019. Table-to-text generation with effective hierarchical encoder on three dimensions (row, column and time). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP2019)*, pages 3134–3143.

Jiaji Huang, Yi Li, Wei Ping, and Liang Huang. 2018. Large margin neural language model. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP2018)*, pages 1183–1191.

Hayate Iso, Yui Uehara, Tatsuya Ishigaki, Hiroshi Noji, Eiji Aramaki, Ichiro Kobayashi, Yusuke Miyao, Naoaki Okazaki, and Hiroya Takamura. 2019. Learning to select, track, and generate for data-to-text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL2019)*, pages 1620–1629.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations 2015 (ICLR2015)*.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP2016)*, pages 1203–1213.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI2018)*, pages 4881–4888.

Soichiro Murakami, Akihiko Watanabe, Akira Miyazawa, Keiichi Goshima, Toshihiko Yanase, Hiroya Takamura, and Yusuke Miyao. 2017. Learning to generate market comments from stock prices. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL2017)*, pages 1374–1384.

Hiroshi Noji and Hiroya Takamura. 2020. An analysis of the utility of explicit negative examples to improve the syntactic abilities of neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL2020)*, pages 3375–3385.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL2002)*, pages 311–318.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019a. Data-to-text generation with content selection and planning. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI2019)*, pages 6908–6915.

Ratish Puduppully, Li Dong, and Lapata Mirella. 2019b. Data-to-text generation with entity modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL2019)*, pages 2023–2035.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL2017)*, pages 1073–1083.

Rico Sennrich. 2017. How grammatical is character-level neural machine translation? assessing MT quality with contrastive translation pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL2017)*, pages 376–382.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *2020 International Conference on Learning Representations (ICLR2020)*.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in data-to-document generation. *arXiv preprint arXiv:1707.08052*.