Generating Weather Comments from Meteorological Simulations

Soichiro Murakami[†] **Sora Tanaka**[†] **Masatsugu Hangyo**[§] **Hidetaka Kamigaito**[†]
Kotaro Funakoshi[†] **Hiroya Takamura**[†,*] **Manabu Okumura**[†]
[†]Tokyo Institute of Technology [§]Weathernews Inc.
[*]Artificial Intelligence Research Center, AIST
{murakami,tanaka,kamigaito,funakoshi}@lr.pi.titech.ac.jp,
hangyo@wni.com, {takamura,okumura}@pi.titech.ac.jp

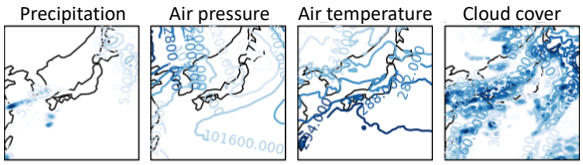
Abstract

The task of generating weather-forecast comments from meteorological simulations has the following requirements: (i) the changes in numerical values for various physical quantities need to be considered, (ii) the weather comments should be dependent on delivery time and area information, and (iii) the comments should provide useful information for users. To meet these requirements, we propose a data-to-text model that incorporates three types of encoders for numerical forecast maps, observation data, and meta-data. We also introduce *weather labels* representing weather information, such as sunny and rain, for our model to explicitly describe useful information. We conducted automatic and human evaluations. The results indicate that our model performed best against baselines in terms of informativeness. We make our code and data publicly available[1].

1 Introduction

Numerical weather prediction (NWP), a method for weather forecasting that uses mathematical models of the atmosphere, oceans, and observations, has become a mainstream tool for supporting today's weather forecasts around the world. Weather forecasters obtain numerical outputs from the simulation models and use their scientific knowledge and historical data to come up with forecast comments such as "*sunny and sometimes cloudy*". However, writing local or personalized weather comments for end users is labor intensive and requires a solid knowledge of meteorology. Therefore, the task of generating weather-forecast comments has traditionally been addressed in the field of data-to-text generation (Goldberg et al., 1994; Belz, 2007).

In this paper, we focus on the task of generating weather-forecast comments from meteorological simulations. While previous studies have mainly focused on database records and tables (Sripada et al., 2004; Liang et al., 2009), which are modified results by experts based on their local knowledge (Reiter et al., 2005), we use *raw* simulation results of NWP models as inputs for text generation. This is closer to the real-world scenarios, in which meteorological specialists describe weather comments by interpreting such numerical data. We believe it will be more helpful for less experienced forecasters. There has been little research on generating descriptions from a sequence of raw numerical data even in the data-to-text generation (Gatt and Krahmer, 2018).



Delivery time: 05:51 a.m. on 06 April, Tokyo

Today patches of blue sky will appear, but the sky will become cloudy and it will gradually start to rain in the evening. Please bring an umbrella when you go out, even if it's not raining.

Figure 1: A weather comment written by a meteorological expert and simulation results from NWP models

We illustrate the three characteristic problems of weather-comment generation in Figure 1. The first problem is that a forecaster needs to consider the changes in numerical values for different types of physical quantities. For example, the comment in this figure states that it will be raining in the evening after a sunny spell according to the changes in precipitation and cloud cover. The second problem is that weather-forecast comments are often written on the basis of meta-data such as area (e.g., Tokyo), delivery time (e.g., 05:51 a.m.), and date

[1] https://github.com/titech-nlp/pinpoint-weather

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 1462–1473
April 19 - 23, 2021.

(e.g., 06 April). For example, weather comments contain expressions that depend on their delivery time and date; comments published in the morning use "*today*", while those posted in the evening usually refer to "*tomorrow*". The third problem is that consumers place a higher priority on informativeness of weather comments and their correctness. In particular, important information such as sunny, rain, and snow should be explicitly mentioned since it will greatly affect the consumers. For example, in Figure 1, although there are several possible types of content to be described such as precipitation, cloud cover, and air pressure, the comment mainly focuses on the information on rain and umbrellas since they can affect the consumers' behavior.

To address these issues, we propose a data-to-text model for generating weather comments from simulation results of NWP models and past observation data. To tackle the first problem, we use a multi-layer perceptron (MLP) or convolutional neural network (CNN) to capture different types of physical quantities and input them to a bi-directional recurrent neural network (Bi-RNN) to take their time scales into account. For the second problem, we incorporate meta-data, such as area information, delivery time, and date, into an encoder for the meta-data. To address the third problem, we introduce *weather labels* representing weather information, such as sunny and rain, to help our model explicitly describe useful information and improve the correctness.

We conducted automatic and human evaluations to evaluate the proposed model on the task of generating Japanese weather comments from simulation results of NWP models and meteorological observation data. The results of both automatic and human evaluations indicate that our model improves the informativeness of generated comments compared with baselines.

2 Related Work

Data-to-text generation, which is the task of automatically producing descriptions from non-linguistic data (Gatt and Krahmer, 2018), has been widely used in various fields such as sports (Wiseman et al., 2017; Puduppully et al., 2019), finance (Murakami et al., 2017; Aoki et al., 2018, 2019), and medical care (Portet et al., 2009; Jing et al., 2018). Neural generation methods have been attracting increased attention in the field of data-to-text generation (Liu et al., 2018; Iso et al., 2019), although rule-based approaches have been the mainstream (Kukich, 1983; Reiter et al., 2005).

The task of generating weather-forecast comments has traditionally been tackled in the field of data-to-text generation (Belz, 2007; Angeli et al., 2010; Mei et al., 2016). For example, there are efforts in generating weather-forecast comments intended for marine shipping or offshore oil facilities (Kittredge et al., 1986; Reiter et al., 2005), as well as local weather forecasts for more general use (Kerpedjiev, 1992; Liang et al., 2009).

Prior research has examined the second and third problems mentioned in Section 1 (Murakami et al., 2017; Puduppully et al., 2019). For the second problem, we need to incorporate information for time and area into a generation model to generate time-dependent expressions. For the third problem, we must carry out content selection to explicitly provide useful information, such as sunny and rain, for consumers. In the table-to-text task, which aims to generate a description from a structured table, there have been recent efforts to improve the correctness of generated texts by implicitly introducing a content-matching constraint (Wang et al., 2020), explicitly specifying the content in the table (Ma et al., 2019) or incorporating copy mechanism (Lebret et al., 2016). Nonetheless, the techniques proposed in the table-to-text task are not directly applicable to datasets consisting of raw numerical data, such as simulation results of NWP models, and texts since they rely on task-specific architectures such as the copy mechanism copying words from tables. In addition, Puduppully et al. (2019) proposed a method for generating summaries of basketball games by using the correspondence between entities in text and input tabular data extracted using the information-extraction method (Wiseman et al., 2017). However, the methods are also not applicable to datasets consisting of raw numerical data and texts because they rely heavily on a word-matching algorithm between input tables and texts. To overcome this limitation, we extract weather labels representing the content of weather information from *only* text on the basis of clue words and use them to explicitly describe the useful information.

3 Weather Data

Weather forecasters obtain the output of NWP models and past weather observations, and interpret them together to come up with weather comments.

To reproduce this, we use two types of meteorological data: a numerical forecast map and meteorological observation data.

Numerical Forecast Maps A numerical forecast map is composed of a sequence of 2D surface data extracted from the simulation results of an NWP model, which is a mathematical model of the atmosphere and oceans. In this study, we used numerical forecast maps around Japan simulated using the global spectral model (GSM), which is an NWP model, provided by the Japan Meteorological Agency[2]. The maps are updated four times a day at 0000, 0600, 1200, and 1800 Japan Standard Time. The prediction of physical quantities such as humidity and temperature up to 84 hours ahead is available and is suitable for roughly determining weather trends over a few days. In the maps, grid points are set every 20 km in the range of 20 to 50 degrees north latitude and 120 to 150 degrees east longitude. Therefore, the maps are composed of 151 $\times$ 121 grid points, where each point contains simulation results of the physical quantities corresponding to the area, such as 1021.01 hPa for air pressure.

Observation Data Since weather-forecast comments are often written in comparison with past weather (e.g., the day before), we also introduce meteorological observation data provided by the automated meteorological data acquisition system (AMeDAS), managed by Japan Meteorological Agency[3]. Specifically, we use four physical quantities: precipitation, air temperature, wind speed, and sunshine duration. These quantities are sequential data observed every ten minutes at about 1300 stations across Japan.

4 Weather-Forecast Generation

We consider weather-forecast generation as a task of generating a description from a sequence of 2D data, which is a forecast map. Since this can be viewed as video captioning (Yao et al., 2015; Long et al., 2018), we first introduce an encoder-decoder model (Sutskever et al., 2014) with an attention mechanism (Bahdanau et al., 2015).

Figure 2 shows an overview of our proposed model. It takes three types of input data: a sequence of numerical forecast maps $\boldsymbol{g} = (g_i)_{i=1}^{|\boldsymbol{g}|}$, observation data $\boldsymbol{a} = \{a_i\}_{i=1}^{|\boldsymbol{a}|}$, and meta-data for

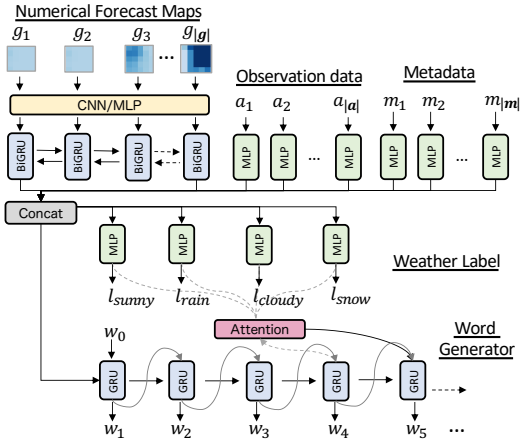


Figure 2: Neural network architecture of proposed model. For simplicity, attention mechanism for three types of input data is omitted.

comments such as delivery time and area $\boldsymbol{m} = \{m_i\}_{i=1}^{|\boldsymbol{m}|}$. Here, g_i, a_i, and m_i represent a forecast map, a numeric vector for a certain physical quantity (e.g., precipitation), and an embedding vector for specific information (e.g., area name), respectively. The output of our proposed model is a weather-forecast comment $\boldsymbol{w} = (w_i)_{i=1}^{|\boldsymbol{w}|}$ and weather labels $\boldsymbol{l} = \{l_i\}_{i=1}^{|\boldsymbol{l}|}$, where w_i and l_i are a word and a weather label, respectively.

For the numerical maps, we use either an MLP or CNN to extract numeric features related to physical quantities, such as air pressure, and input each map to a Bi-RNN to take their sequential information into account. We also incorporate meteorological observation data $\boldsymbol{a}$ and meta-data $\boldsymbol{m}$ by encoding with an MLP. We use the MLP to predict weather labels $\boldsymbol{l}$ from the output of the encoders. We use the RNN language model (RNNLM) (Mikolov et al., 2010) to generate words $\boldsymbol{w}$.

We explain our proposed model and how we introduce meta-data and weather labels into it in the following sections.

4.1 Extracting Numerical Maps for Areas

A weather forecaster writes a weather comment for a specific area by referring to weather data corresponding to the area. With this in mind, we extract a g_i for each area, which has 5 $\times$ 5 grid points, from an larger map that has 151 $\times$ 121 grid points on the basis of latitude and longitude. The extracted map will be a map of 100 square kilometers around the area. Thus, a sequence of numerical forecast maps $\boldsymbol{g}$, for a specific area (e.g., Tokyo) can be acquired, as shown in Figure 3, which shows changes for ten

[2] https://www.jma.go.jp/jma/indexe.html
[3] http://www.jma.go.jp/en/amedas/

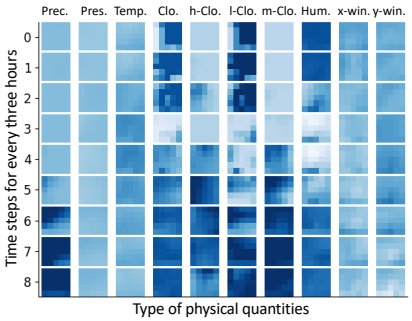


Figure 3: Sequence of numerical forecast maps for specific area, extracted from larger map. Types of physical quantities are as follows: precipitation (Prec), air pressure (Pres), air temperature (Temp), cloud cover (Clo), high-level cloud cover (h-Clo), low-level cloud cover (l-Clo), medium-level cloud cover (m-Clo), humidity (Hum), and wind direction (x-win, y-win).

types of physical quantities (such as precipitation and cloud cover) predicted with an NWP model every three hours up to 24 hours ahead.[4]

4.2 Encoding Numerical Forecast Maps

The task of generating text from sequences of 2D data can be regarded as video captioning. Thus, we use either a CNN or MLP to capture the numeric features of forecast maps. We compared the following two encoders in our experiments.

CNN-based Encoding A CNN is widely used in video captioning to extract visual features from each frame of a video. For example, the CNN encoder for color images has three channels for red, green, and blue. We used ten channels, each channel corresponds to one of the physical quantities (e.g., precipitation), shown in Figure 3, to take the relationships among them into consideration.

MLP-based Encoding In image recognition, a CNN has an advantage over an MLP in terms of translation invariance. However, since we use a map for a specific area, which is extracted from a larger map and whose center is always that area, we hypothesize that a model that takes into account the absolute position on the map is more suitable. Therefore, we use an MLP, which has $10 \times 5 \times 5$ units, to extract the features from forecast maps.

By applying the encoding method to a g_i for each time-step t, we obtain an output vector h_i^g. Next, we sequentially input vector h_i^g to a Bi-RNN to capture value changes for physical quantities in the sequence. As a result, we have the output vector h^g that represents time-series changes in $\boldsymbol{g}$ by concatenating the hidden states of the Bi-RNN:

$$h^g = [h_1^g; h_{|\boldsymbol{g}|}^g], \tag{1}$$

where $[;]$ represents a vector concatenation.

4.3 Encoding Observation Data

With regard to $\boldsymbol{a}$, which are a set of observed values for physical quantities (e.g., sun duration), we use an MLP, which performed best in a study by Murakami et al. (2017). We apply the MLP to each a_i to obtain feature vectors h_i^a. These vectors are concatenated to create the representation that captures the characteristics of meteorological observation data:

$$h^a = [h_1^a; h_2^a; \cdots; h_{|\boldsymbol{a}|}^a]. \tag{2}$$

4.4 Introducing Meta-Data

Since weather forecasters often take their own local knowledge and time information into account when writing weather comments, we incorporate the meta-data for weather comments (such as delivery time) to generate word expressions that depend on the date and time for a comment, e.g., "*today*". Specifically, we create an m_i on the basis of the delivery time and area name (such as *5 a.m.* and *Tokyo*, as shown in Figure 1). We also encode the vector by using the MLP and concatenate its output vector h_i^m. Thus, we obtain a vector h^m that captures the meta-data $\boldsymbol{m}$:

$$h^m = [h_1^m; h_2^m; \cdots; h_{|\boldsymbol{m}|}^m]. \tag{3}$$

Finally, we set the initial hidden state s_0 of both decoders as follows:

$$s_0 = \mathrm{ReLU}(\mathrm{MLP}([h^g; h^a; h^m])). \tag{4}$$

4.5 Weather Labels

In this task, the informativeness of weather comments and their correctness are a key factor. However, since neural-generation models often struggle to capture long-term dependencies, they lose important information included in input data. This shortcoming limits their application to the real world. Thus, we propose a method for explicitly specifying the *content* to be mentioned to help our proposed model correctly describe useful information,

[4]The shade of color for physical quantities indicates the magnitude of predicted values. The predicted values are standardized with all values for a year. For example, the dark color for precipitation around 15–18 hours later in Figure 3 indicates that the predicted amount of precipitation is relatively high.

Label	Clue words
SUNNY	晴れ (*sunny*), 日差し (*sunlight*), 青空 (*blue sky*)
RAIN	雨 (*rain*), 大雨 (*heavy rain*), にわか雨 (*shower*)
CLOUDY	曇り (*cloudy*), 曇 (*cloudy*), 雲 (*cloud*)
SNOW	雪 (*snow*), 吹雪 (*blizzard*), 小雪 (*light snowfall*)

Table 1: Example weather labels and their corresponding clue words. English translations are given in parentheses. All examples are shown in Appendix A.

inspired by the recent success of faithful data-to-text generation (Ma et al., 2019; Wang et al., 2020) and the controllability of neural models with explicit labels (Aoki et al., 2019). However, since most data-to-text datasets do not typically include content plans, we introduce a simple approach for extracting them from a text as a pseudo reference.

Since consumers are primarily interested in weather information such as sunny and rain, we define such content in weather comments as *weather labels*. Specifically, we introduce four types of weather labels: SUNNY, RAIN, CLOUDY and SNOW. To extract weather labels from weather comments, we define clue words for each weather label, as shown in Table 1. Our strategy is to explicitly match the clue words and words in weather comments. For example, the weather comment in Figure 1 includes the clues words "*blue sky*", "*cloudy*", and "*rain*", so three weather labels, SUNNY, CLOUDY, and RAIN can be associated with the comment. The method for labeling text is very simple, but we found that it works in most cases[5]. The method can also avoid the following two issues in weather-comment generation. First, it is almost impossible to explicitly associate a comment with continuous numerical data such as meteorological data, although the table-to-text task can easily do this, as discussed in Section 2. Second, we need expert knowledge if we annotate content to be mentioned in input data without reference text.

To determine content to be mentioned before text generation begins, we introduce a binary classifier for each weather label, as shown in Figure 2. The classifier is based on an MLP. We train the classifier with the weather labels extracted from comments in the training data. In the inference stage, each classifier predicts each weather label (e.g., l_{sunny}) from the three types of input data ($\boldsymbol{g}$, $\boldsymbol{a}$, $\boldsymbol{m}$). The word generator then generates weather comments $\boldsymbol{w}$ from the input data and those labels.

[5]We evaluated correctness of the extracted weather labels by five people on the basis of 100 comments, which are randomly extracted from development set. As a result, 96% of the weather labels were judged to be appropriate.

4.6 Word Generator

The word generator is based on a RNNLM with an attention mechanism. In addition to introducing the attention mechanism into input data (Wiseman et al., 2017; Chen et al., 2019), we also introduce the attention mechanism into the weather labels, as shown in Figure 2. The word generator is designed to take into account weather labels to explicitly describe important information in text generation through this attention mechanism.

In the word generator, the probability of producing a word w_t at t is computed by

$$p(w_t|w_{<t}, \boldsymbol{g}, \boldsymbol{a}, \boldsymbol{m}, \boldsymbol{l}) = \mathrm{softmax}_{\mathrm{w_t}}(W_s s_t^w), \quad (5)$$

$$s_t^w = \mathrm{GRU}(w_{t-1}, s_{t-1}^w, c_t), \quad (6)$$

where w_{t-1} and s_{t-1}^w are an output word and the hidden state of the word decoder at time step $t-1$, respectively. W_s is a weight matrix. Vector c_t represents the context vector at t, which is created by concatenating four context vectors $[c_t^g; c_t^a; c_t^m; c_t^l]$ constructed with the attention mechanism (Bahdanau et al., 2015) for input data ($\boldsymbol{g}, \boldsymbol{a}, \boldsymbol{m}$) and weather labels $\boldsymbol{l}$. For instance, the context vector c_t^l over $\boldsymbol{l}$ at t can be calculated as follows:

$$c_t^l = \sum_{i=1}^{|\boldsymbol{l}|} \alpha_{t,i}^l s_i^l, \quad \alpha_{t,i}^l = \frac{\exp(\eta(s_{t-1}^w, s_i^l))}{\sum_{j=1}^{|\boldsymbol{l}|} \exp(\eta(s_{t-1}^w, s_j^l))}, \quad (7)$$

where s_i^l represents the hidden state of the weather-label classifier for label l_i, and $\alpha_{t,i}^l$ is the alignment probability between the t-th output word and i-th hidden state in the classifier. We use an MLP η as a score function. Note that c_t^g, c_t^a, and c_t^m for the input data ($\boldsymbol{g}$, $\boldsymbol{a}$, $\boldsymbol{m}$) can be derived with Equation (7) in a similar manner.

5 Experiments

5.1 Setup

Dataset We used weather comments from 2014 to 2015 in Japan as the text dataset, which consists of 57,412 comments provided by Weathernews Inc. We separated the dataset into 28,555 comments from 2014 for training, 14,464 and 14,393 comments from 2015 for development and testing. For the numerical forecast maps, we collected 2,715

maps corresponding to the comments from the website[6] of the Research Institute for Sustainable Humanosphere, Kyoto University. We also separated the numerical maps into 1,344 from 2014 for training, 1,326 and 1,329 from 2015 for development and testing, respectively. Their sum does not agree with the total number of maps, 2,715 because the weather comments for development and testing are sampled from 2015, and numerical maps for different areas are often extracted from a single entire map around Japan and are used for them. Note that the weather comments and the corresponding extracted maps are unique for each area, and the comments for development and testing do not overlap with each other.

We used a $\boldsymbol{g}$, which consists of nine steps every three hours up to 24 hours ahead, since the comments treat weather forecasts up to the next day. Referring to delivery date and time of a comment, we aligned each $\boldsymbol{w}$ with the extracted $\boldsymbol{g}$ obtained by following the procedure in Section 4.1. We also used precipitation, air temperature, wind speed, and sunshine duration for the last 24 hours as $\boldsymbol{a}$, where each a_i consists of observed values for 24×6 steps. As $\boldsymbol{m}$, we used delivery date, time, and area name (e.g., *April*, *Monday*, *5 a.m.*, *Tokyo*).

Implementation We used a single-layer MLP and bi-directional gated recurrent unit (Bi-GRU) for the encoders and two-layer GRU for the word generator. Note that we have investigated a transformer-based model (Vaswani et al., 2017), but we found that there were no significant differences between the transformer-based model and the GRU-based model. The hidden states of our model and size of the word embeddings were both 512. We set the dimension size of the hidden vectors for the meta-data h_i^m and observation data h_i^a to 64. The model was trained using the Adam optimizer (Kingma and Ba, 2015). We applied an early stopping strategy with a minimum number of 25 epochs. We stopped training if there was no improvement in validation loss for three consecutive epochs.

Evaluation Metrics For the automatic evaluation, since reference texts written by meteorological experts generally mention important information such as sunny and rain, we used BLEU-4[7] (Papineni et al., 2002) and ROUGE-1[8] (F_1 score) (Lin, 2004) to see whether generated texts properly mention the important information as reference texts do. However, since these metrics based on word overlapping rely on the reference texts, they cannot be used to assess the correctness of the generated texts if their expressions are different from the reference texts. Thus, we also calculated precision, recall and F_1 scores of weather labels, which are extracted from the generated texts, to see how they properly describe important information in comparison with those of the reference texts.

For the human evaluation, we asked five participants to give each generated comment a score from 1 to 3 for informativeness, consistency, and grammar, where 3 is the highest. In the evaluation of informativeness, we showed the participants a human-generated comment as a reference and asked them to compare the generated text and reference. This was done because understanding complicated input data for data-to-text generation is extremely difficult for non-specialists. We randomly selected 40 comments from the test set[9]. Each comment was rated by all five participants. We used Wilcoxon signed rank test (Wilcoxon, 1945) to test the statistical significance of the difference among comparative models. The details for instructing the human raters how to evaluate each comment are provided in Appendix B.

Models We defined five models listed in Table 2 and *w/o Meta*, which does not take meta-data into account, to determine whether each component (e.g., weather label) contributes to the results. Models (1) and (2), which do not take weather labels into account, were regarded as baselines. Model (3) uses the weather labels that we proposed. To further improve the correctness of generated texts, we also introduced model (4) to investigate a content-matching constraint loss (Wang et al., 2020) that aims to constrain an embedding of input data to be close to the corresponding target text embedding. Specifically, we calculated the loss between the output embeddings of the weather-label classifier and the target text embeddings to make their representations close. We also evaluated model (5) that used oracle labels extracted from the reference text to validate the upper bound of improvement in the

[6] http://database.rish.kyoto-u.ac.jp/index-e.html

[7] https://github.com/mjpost/sacrebleu

[8] https://github.com/pltrdy/rouge

[9] To clarify effectiveness of each weather label, we randomly extracted the comments so that each label was included in at least 10 cases.

Model	Components			Word Overlap		SUNNY			RAIN			CLOUDY			SNOW		
	Enc.	Weather	CL	BLEU	ROUGE	P%	R%	F_1%	P%	R%	F_1%	P%	R%	F_1%	P%	R%	F_1%
(1)	CNN	–	–	12.7	42.8	83.5	67.6	74.7	72.8	83.6	77.8	58.5	59.8	59.0	75.2	50.1	60.2
(2)	MLP	–	–	13.0	43.5	83.2	68.4	74.9	74.6	83.5	78.8	59.8	60.3	59.9	75.7	53.3	62.3
(3)	MLP	Pred.	–	12.9	43.8	81.0	78.5	79.7	78.6	80.0	79.3	62.5	55.9	58.9	75.9	60.4	67.2
(4)	MLP	Pred.	✓	13.2	43.9	81.0	78.4	79.7	76.6	84.1	80.2	60.6	59.3	59.8	77.7	58.5	66.6
(5)	MLP	Orac.	✓	14.6	45.5	94.9	84.5	89.4	84.4	92.9	88.4	84.7	85.6	85.1	91.3	63.8	75.1

Table 2: Results of automatic evaluation on test set using BLEU, ROUGE, and correctness of each weather label extracted from its generated text in precision (P%), recall (R%), and F_1 scores (F_1%). Models are numbered (1) through (5). Models (1) and (2) are baselines. Components of each model are as follows: encoder for numerical forecast map (Enc.), weather labels (Weather), and content-matching constraint loss (CL). Weather represents whether we use weather labels extracted from generated text (Pred.) or oracle labels (Orac.) extracted from reference text. Scores were averaged over three runs.

Expression	Model (4)	w/o Meta	Δ
今日 (*Today*)	99.3	97.3	+2.0
明日 (*Tomorrow*)	95.1	91.1	+4.0
月 (*Monday*)	29.3	0.0	+29.3
火 (*Tuesday*)	29.2	0.0	+29.2
春 (*Spring*)	14.0	2.4	+11.6
夏 (*Summer*)	19.1	12.4	+6.7
BLEU	13.2	12.7	+0.5

Table 3: F_1 scores for time-dependent expressions. Each expression is accompanied by its English translation in parenthesis. Δ means difference in each score between Model (4) and w/o Meta.

Label	Precision	Recall	F_1 score
SUNNY	79.7	84.9	82.1
RAIN	79.9	80.5	80.2
CLOUDY	61.5	62.5	61.6
SNOW	73.9	67.1	70.3

Table 4: Results of weather-label prediction by classifier, which only performs weather-label prediction, on test set. Scores were averaged over three runs.

correctness of generated text with the labels.

A rule-based system could be also considered as a comparative model, but we could not include it since it is practically impossible to construct their *rules* without expert knowledge in meteorology.

5.2 Results

The results of the automatic evaluation are listed in Table 2. To see how our proposed model can correctly describe the content of weather forecasts, we calculated precision, recall, and F_1 scores of weather labels, which are extracted from their generated texts, by referring to weather labels extracted from human-generated texts as references. Models (3) and (4), which introduce weather labels, outperformed the baselines, which do not take into account the weather labels. In particular, we found that models (3) and (4) significantly improved F_1 scores for the sunny and snow labels by around 5% in comparison to the baselines. This suggests that incorporating the weather labels enables the model to more correctly generate texts.

In terms of BLEU and ROUGE, model (4) outperformed all other models except (5), which can use oracle labels. Interestingly, model (5) significantly improved the correctness of each weather label as in Table 2 since it can use the oracle labels, but the improvement in BLEU and ROUGE scores was limited. This implies that it is difficult to evaluate the correctness of generated texts by relying on only these metrics.

5.3 Contributions of Each Component

Next, we compared the models to investigate how each component contributes to their performance.

Encoder for numerical maps In the comparison between models (1) and (2), which use either a CNN or MLP as an encoder for numerical forecast maps, no significant differences were found.[10] This suggests that either encoder is acceptable, although we had originally expected the MLP encoder that takes into account absolute positions on the maps to be more suitable than a CNN encoder because of its simplicity.

Effect of meta-data With regard to the meta-data such as delivery time and date, for which we expected the model to generate time-dependent expressions (e.g., "*today*"), we conducted an ablation

[10] In spite of this, we used an MLP for the models to add other components since it performed slightly better than a CNN in the experiments.

Model	I	C	G	Generated comment
Ref.	–	–	–	今日は、うっすら雲が広がりやすいものの、日差しが届きます。夏至の日差しは強烈なので、紫外線・暑さ対策が欠かせません。 (*Today, the sky will easily become partly cloudy, but it will become sunny. Since the sun this summer is intense, please take precautions against the heat and ultraviolet rays.)*
(2)	2.0	3.0	3.0	今日（月）は日差しが届きますが、段々と雲が広がります。午後はニワカ雨の可能性があるので、折りたたみ傘があると安心です。 (*Today (Monday), it will be sunny then gradually become cloudy. Since there is a chance of rain in the afternoon, it will be a good idea to take a folding umbrella when you go out.)*
(4)	2.8	2.8	3.0	今日（月）は雲が広がりやすいものの、日差しが届く時間もあります。ムシムシとした暑さになるので、熱中症対策を忘れずに。 (*Today (Monday), the sky will become cloudy but will become sunny. Since it will be hot and humid, please remember to take precautions against heatstroke.)*

Table 5: Reference weather comment written by human (Ref.) and those comments generated from models (2) and (4). Weather labels extracted from the reference text are SUNNY and CLOUDY. The reference comment was posted at 00:02 a.m. on Monday June 22, 2015 for the *Toyohashi* area. Columns I, C, and G show average scores for informativeness, consistency, and grammar from human evaluation, respectively. Each example is accompanied by its English translation.

Label	Model(2)			Model(4)			# of
	Info.	Con.	Gra.	Info.	Con.	Gra.	cases
SUNNY	1.92	2.91	2.91	2.10	2.82	2.88	26
RAIN	2.02	2.93	2.92	2.13	2.88	2.90	26
CLOUDY	1.99	2.93	2.94	2.12	2.83	2.89	19
SNOW	1.88	2.95	2.92	1.95	2.91	2.94	13
Overall	1.98	2.92	2.92	2.10	2.86	2.90	40

Table 6: Results of human evaluation. Scores are averages given by five human raters. Columns Info, Con, and Gra represent informativeness, consistency, and grammar, respectively. Differences in informativeness and consistency are statistically significant at $p < 0.05$.

study to investigate whether our proposed model can properly generate these expressions in comparison with w/o Meta that does not take into account such meta-data. Specifically, we calculated F_1 scores for time-dependent expressions by using weather comments written by human as references. Table 3 shows F_1 scores for each expression in the comments generated with model (4) and w/o Meta, respectively. We found that model (4), which takes into account the meta-data, can more accurately provide time-dependent expressions than w/o Meta. This finding suggests that introducing the meta-data into a generation model improve the correctness of meta-data in generated comments.

Effect of weather labels In comparison between models (2) and (3), which do not and do use the weather labels, respectively, we found that model (3) significantly improved the F_1 scores for the weather labels extracted from their generated comments than model (2). Specifically, the recall scores significantly improved regarding the weather labels for sunny and snow. This indicates that specifying content to be mentioned, such as weather labels, helps the model to explicitly describe the information.

We also tested a classifier that only predicts weather labels from input data to clarify the upper bound of the improvement in the correctness of each weather label extracted from its generated text. Table 4 presents the results of weather-label prediction by the classifier. According to the comparison between Tables 2 and 4, the scores of weather labels extracted from the generated texts with models (3) and (4) are approaching the upper bound by the classifier, but there is still room for improvement.

Effect of content-matching constraint To investigate the effectiveness of the content-matching constraint loss (Wang et al., 2020) for improving faithfulness of generated texts, we compared model (4), which uses this loss, with model (3), which does not. There was a slight improvement in BLEU and ROUGE scores.

5.4 Human Evaluation

Table 6 lists the results of the human evaluation, where # represents the number of cases, which includes each weather label in the evaluation set. Note that a comment may contain multiple labels. Overall, model (4), which explicitly performs content selection by using weather labels, outperformed model (2), which does not, in terms of informativeness[11]. This indicates that introducing weather labels contributes to the correctness of information included in generated texts, as we also can see from the results of the automatic evalua-

[11] Specifically, model (4) was rated more informative than model (2) in 20 of the 40 cases. 10 of them were equivalent.

tion. Model (4), however, was inferior to model (2) in terms of consistency, although the score is still significantly high. This is reasonable because more information makes it more challenging to maintain consistency. To solve this problem, it is necessary to carry out not only content selection but also content planning to specify both *what to say* and *in which order* (Wiseman et al., 2017).

Table 5 shows an example reference and weather comments generated with models (2) and (4). Both models correctly described the information on cloudy weather and sunshine, but model (2) mistakenly described rainy weather compared with the reference. In contrast, model (4) properly described all the information including hot weather and was judged as more informative than model (2). More generation examples are given in Appendix C.

6 Conclusion

In this paper, we addressed the task of generating weather comments from meteorological simulations. We proposed a data-to-text model and incorporated three types of encoders for forecast maps, observation data, and meta-data into the model. In addition, we introduced weather labels representing the content of weather information to explicitly carry out content selection and improve the correctness of information in generated comments. Experiments indicated that our model significantly improved the informativeness of generated comments and outperformed the baselines in both automatic and human evaluations.

Acknowledgements

We thank the anonymous reviewers for their insightful comments and valuable suggestions. This work was supported by JST PRESTO (Grant Number JPMJPR1655) and JST-Mirai Program Grant Number JPMJMI18BB, Japan.

References

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 502–512. Association for Computational Linguistics.

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*, pages 323–332. Association for Computational Linguistics.

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*, pages 135–139. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Anja Belz. 2007. Probabilistic generation of weather forecast texts. In *Proceedings of the 2007 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 164–171. Association for Computational Linguistics.

Shuang Chen, Jinpeng Wang, Xiaocheng Feng, Feng Jiang, Bing Qin, and Chin-Yew Lin. 2019. Enhancing neural data-to-text generation models with external background knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3022–3032. Association for Computational Linguistics.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61(1):65–170.

Eli Goldberg, Norbert Driedger, and Richard I. Kittredge. 1994. Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53.

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*, pages 2102–2113. Association for Computational Linguistics.

Baoyu Jing, Pengtao Xie, and Eric Xing. 2018. On the automatic generation of medical imaging reports. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2577–2586. Association for Computational Linguistics.

Stephan M. Kerpedjiev. 1992. Automatic generation of multimodal weather reports from datasets. In *Third*

Conference on Applied Natural Language Processing, pages 48–55. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

R. Kittredge, A. Polguere, and E. Goldberg. 1986. Synthesizing weather forecasts from formatted data. In *Proceedings of the 11th International Conference on Computational Linguistics*.

Karen Kukich. 1983. Design of a knowledge-based report generator. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pages 145–150. Association for Computational Linguistics.

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*, pages 1203–1213. Association for Computational Linguistics.

Percy Liang, Michael Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of Association for Computational Linguistics and International Joint Conference on Natural Language Processing*, pages 91–99. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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 Thirty-Second AAAI Conference on Artificial Intelligence*, pages 4881–4888.

Xiang Long, Chuang Gan, and Gerard de Melo. 2018. Video captioning with multi-faceted attention. *Transactions of the Association for Computational Linguistics*, 6:173–184.

Shuming Ma, Pengcheng Yang, Tianyu Liu, Peng Li, Jie Zhou, and Xu Sun. 2019. Key fact as pivot: A two-stage model for low resource table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2047–2057. Association for Computational Linguistics.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2016. What to talk about and how? selective generation using LSTMs with coarse-to-fine alignment. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 720–730. Association for Computational Linguistics.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association*, pages 1045–1048. International Speech Communication Association.

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 (Volume 1: Long Papers)*, pages 1374–1384. Association for Computational Linguistics.

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 of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

François Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. 2009. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7-8):789–816.

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

Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167(1-2):137–169.

Somayajulu Sripada, Ehud Reiter, and Ian Davy. 2004. Sumtime-mousam: Configurable marine weather forecast generator. *Expert Update*, 6.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020. Towards faithful neural table-to-text generation with content-matching constraints. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1072–1086. Association for Computational Linguistics.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263. Association for Computational Linguistics.

Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. 2015. Describing videos by exploiting temporal structure. In *Proceedings of the 2015 IEEE International Conference on Computer Vision*, page 4507–4515. IEEE Computer Society.

A Examples of All Clue Words

To extract weather labels from weather comments, we defined clue words for each weather label, as shown in Table 7. Note that we selected the clue words in reference to the development data.

B Details of Human Evaluation

The following are the instructions presented to evaluation raters during the human evaluation. Each comment was rated by all five participants.

We only showed the raters the weather comments written by human as references to evaluate the generated comments. This was done because understanding complicated data, which is used as input for data-to-text generation, such as numerical forecast maps, is particularly difficult for non-specialists.

Informativeness:

3: This is an ideal weather comment since it appropriately mentions important information.

2: Although some important information is missing, the information included in the comment is appropriate and is acceptable as a weather comment.

1: The information included in the comment is incorrect and inappropriate as a weather comment.

Consistency:

3: The comment is consistent on the whole and it is easy to read.

2: The comment lacks consistency in some parts and is difficult to read.

1: The comment lacks consistency on the whole and is difficult to understand.

Grammar:

3: There are no grammatical errors.

2: There are some grammatical errors, but it is understandable.

1: There are many grammatical errors, and it is difficult to understand.

C More Generation Examples

More generation examples from model (2) and (4) are shown in Table 8 and 9. In this example, we can observe that model (2) was judged as less informative on average by the five raters in comparison with model (4) since the comments generated by model (2) provide incorrect information (e.g., *rain shower* instead of *snow* in Table 8), or lack important information (e.g., *rain* in Table 9). On the other hand, model (4), which is our proposed model, properly describes these important information.

Label	Clue words
SUNNY	晴れ (*sunny*), 日差し (*sunlight*), 青空 (*blue sky*), 回復 (*improvement*), 日和 (*perfect day*), 陽気 (*weather*), 秋晴れ (*fine autumn day*), 晴天 (*fine weather*), 晴れ間 (*patch of blue sky*), 晴れる (*clear up*), 太陽 (*sun*), 五月晴れ (*beautiful weather in May*)
RAIN	雨 (*rain*), 大雨 (*heavy rain*), にわか雨 (*shower*), 雷雨 (*thunderstorm*), 暴風雨 (*storm*), 雨風 (*rain and wind*), 荒天 (*stormy weather*), 台風 (*typhoon*), 傘 (*umbrella*)
CLOUDY	曇り (*cloudy*), 曇 (*cloudy*), 雲 (*cloud*)
SNOW	雪 (*snow*), 吹雪 (*blizzard*), 小雪 (*light snowfall*), 吹雪く (*snow blowing*), ふぶく (*snow blowing*)

Table 7: Weather labels and their corresponding clue words we used to extract labels from weather comments. Corresponding English translations are also given in parenthesis.

Model	I	C	G	Generated comment
Ref.	–	–	–	今日は日差しが届いても、ゆっくり天気は下り坂になります。夕方から雪が降り出すので、外出は早めがオススメ。昨日より冷え込みが増すので、防寒が必須です。 (*Today, it will be sunny, but the weather will go downhill. Snow will start to fall in the evening. If you are going out, you should go early. Since it will be colder than yesterday, cold protection is important.)*
(2)	2.0	3.0	2.6	今日（土）は日差しが届いても、ニワカ雨の可能性があります。お出かけには折りたたみ傘があると安心です。昼間でも気温は低空飛行で体を温めるのも良さそうです。 (*Today (Saturday), it will be sunny, but there is a chance of rain shower. It will be a good idea to take a folding umbrella when you go out. Since the weather is cool even during the daytime, it will be good to warm your body up.)*
(4)	2.4	3.0	3.0	今日（土）は日差しが届いても雲が優勢の天気。ニワカ雪の可能性があるので、折りたたみ傘があると安心です。また、寒さが続くので防寒対策を万全にして下さい。 (*Today (Saturday), it will be sunny, but the weather will be predominantly cloudy. There is a chance of snow shower, so it will be a good idea to have a folding umbrella. Also, please take protective measures against the cold as the cold days continue.)*

Table 8: Reference weather comment written by human (Ref.) and those comments generated from models (2) and (4). The reference comment was posted at 00:00 a.m., on Saturday, December 26, 2015 for the *Shiranuka* area. Columns I, C, and G show average scores for informativeness, consistency, and grammar from human evaluation, respectively. Each example is accompanied by its English translation. Underline indicates incorrect information.

Model	I	C	G	Generated comment
Ref.	–	–	–	今日も変わりやすい天気です。急にザッと雨が降る可能性があるので、日差しがあっても油断できません。ムシッとする風もやや強いままです。 (*Today, the weather is changeable too. It may rain suddenly, so please be careful even if it is sunny. The humid wind is a little strong.)*
(2)	1.8	2.8	2.8	今日は雲が広がりやすいものの、日差しが届く時間もあります。この時期の紫外線は強烈なので、ＵＶ対策を忘れずにお出かけ下さい。 (*Today, the cloud will spread, but there is time sunshine appears. Since ultraviolet rays are intense these days, remember to take measures against ultraviolet rays when you go out.)*
(4)	2.6	2.8	2.8	今日は変わりやすい天気。日差しが届いても、急な雨もあります。モクモクした雲が近づいて来たら天気急変のサインです。 (*Today, the weather is changeable. It will be sunny, but it will start to rain suddenly. It is a sign of a sudden change of the weather if fuzzy clouds are approaching.)*

Table 9: Reference weather comment written by human (Ref.) and those comments generated from models (2) and (4). The reference comment was posted at 10:46 a.m., on Saturday, July 18, 2015 for the *Tokyo* area. Columns I, C, and G show average scores for informativeness, consistency, and grammar from human evaluation, respectively. Each example is accompanied by its English translation.