

# 数値予報マップからの天気予報コメントの自動生成

村上 聡一朗<sup>†</sup>

笹野 遼平<sup>‡</sup>

高村 大也<sup>‡,§</sup>

奥村 学<sup>‡</sup>

<sup>†</sup> 東京工業大学 大学院総合理工学研究科

<sup>‡</sup> 東京工業大学 科学技術創成研究院

<sup>§</sup> 国立研究開発法人科学技術振興機構さきがけ

<sup>†</sup> murakami@lr.pi.titech.ac.jp <sup>‡</sup>{sasano,takamura,oku}@pi.titech.ac.jp

## 1 はじめに

天気予報情報サイトでは、図1のように1日の気象情報を表す表形式の時系列データと共に、表データには含まれない雷や雨雲の情報や外出時に気をつけるポイントを解説する天気予報コメントが配信されている。これらのテキストは、気象予報モデルを大規模コンピュータでシミュレーションして得られる時系列の数値予報マップ等を参考にして、気象の専門家が記述している。天気予報は、全国の数十から数百のエリアごとに1日に複数回更新されることが一般的であり、コメント更新に必要なコストは大きい。そのため、本研究では、時系列数値予報マップから自動的に天気予報コメントを生成することを考える。

近年、機械翻訳 [2] や画像キャプション生成 [8] などの様々な系列生成タスクにおいて、encoder-decoder モデル [6] を用いた研究が数多く報告されており、成果を挙げている。数値予報マップは、気温や湿度などの予測数値を時間・格子点ごとに格納した時系列の面データである。本研究では、天気予報コメントの自動生成を、入力の時系列面データからテキスト (単語系列) を出力する系列生成タスクとして考える。時系列面データから単語系列を生成する取り組みとして、動画キャプション生成の研究が挙げられる [7]。本研究では、動画キャプション生成の研究に倣い、時系列の数値予報マップからテキストを生成する手法を提案する。

本研究で取り組む天気予報コメント生成と動画キャプション生成の違いとして、入力データ中の考慮すべき要素の数とコメントを配信する日時や対象のエリアによって記述する内容が異なること等が挙げられる。多くの動画キャプションの研究では、各タイムステップのフレームを赤、青、緑の3つの要素を含む画像として表現し、特徴抽出器を用いることでフレーム中の特徴を認識している。時系列数値予報マップの場合、各タイムステップには、気温や気圧、雲量や降水量など、より多くの要素 (物理量) が含まれている。しかし、時系列数値予報マップのように、数多くの要素を含む時系列面データから、encoder-decoder モデルを用いてテキスト生成を行う例はいままでに報告されておらず、入力データの適切な encode 手法は明らかで



図 1: 表形式データと天気予報コメントの例

表 1: 天気予報コメントの例

配信時間	エリア	コメント
(1) 2014/03/17 14:27:37	新宿区	明日は春本番のポカポカ陽気
(2) 2014/12/22 00:06:19	熊本	今日 (月) は路面状態に注意

はない。また、天気予報コメントは、コメントを配信する時間や季節によって記述する内容が異なる。例えば、表1のように、朝や日付が変わった直後などは今日の天気について言及し、午後や夕方の場合は、明日の天気について言及することが多い。表1の(1)のように、3月中旬に配信されたコメントでは、「春本番」などの季節特有の表現が使われる。そのため、天気予報コメントの生成時には、数値予報マップだけではなく、日時や季節などのメタ情報も考慮する必要がある。

本研究では、コメント配信時の日時などを考慮した数値予報マップから天気予報コメントを自動生成する手法を提案する。encoder-decoder モデルを用いて、時系列数値予報マップのようなデータからテキストを生成する手法はいくつか考えられるため、本研究ではいくつかの encode 手法の提案を行い、実験的に手法の比較検討を行う。

## 2 関連研究

気象データから天気予報を解説するテキストを生成する研究はいくつか行われている。Belzら [3] は、気象シミュレータから出力された海上の風の予測値を用い、時刻ごとの海上の風について説明するテキストを生成する研究に取り組んでいる。この研究では、解説テキストを生成する対象として風の予報のみを取り扱い、気象予報士などの専門家に向けたテキストの生成を行っている。これに対し本研究では、数値予報マッ

表 2: 数値予報マップの概要

配信時間	6 時間毎に更新 (1 日 4 回: 00, 06, 12, 18UTC)
予報時間	84 時間予報
領域	北緯 20 度から 50 度, 東経 120 度から 150 度
物理量 (11 種類)	気圧, 海面更正気圧, 東西風, 南北風, 気温, 相対湿度, 積算降水量, 上層雲量, 中層雲量, 低層雲量, 全雲量

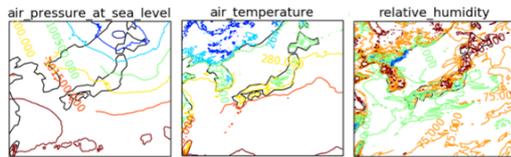


図 2: 数値予報マップの例

に含まれる気温や気圧, 風, 降水量等の複数の物理量の予測値を入力として扱い, より一般的な天気予報の解説テキストを生成することを目的としている.

Angeli ら [1] は, WEATHERGOV データ [5] を用い, エリア毎の気温や降水確率, 風速等の気象予報データから天気予報の解説テキストを生成する研究を行っている. 前述の通り, 天気予報コメントは, コメントを配信する時間や季節によって記述する内容が異なることが一般的である. しかし, この研究で使っている WEATHERGOV データには, 季節や曜日等の時間に関する情報は含まれていない. 一方, 本研究では, 時間に関する情報を用い, 季節や曜日, 月等を考慮した天気予報コメントを生成する手法を提案する.

### 3 数値予報マップの概要

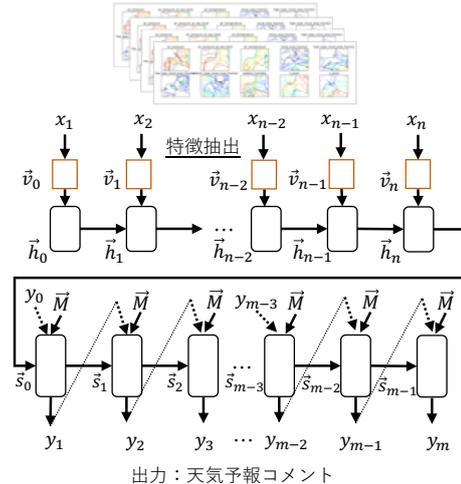
本研究で使用する数値予報マップは, 気象予報モデルを大規模コンピュータで三次元シミュレーションした結果から, 地表部分を取り出した二次元面データである. 本研究では, 全球数値予報モデルを用いて計算された数値予報マップを使用する.

表 2 に本研究で使用する日本周辺の数値予報マップの概要を示す. 数値予報マップは, 気圧や気温, 風向きなどの各物理量の 1 時間ごとの予測数値が 84 時間先まで格納された時系列数値データである. 気象庁が作成している日本周辺の数値予報マップには, 北緯 20 度から 50 度, 東経 120 度から 150 度の範囲で 20km ごとに格子点が設定されており, 各格子点には各物理量の予測数値が格納されている. 図 2 に日本周辺の地図上に可視化した数値予報マップの例を示す. 図 2 中の各物理量の色は予測数値の大きさを表している.

### 4 提案手法

本研究で提案するモデルの概要を図 3 に示す. 本研究では, 時系列数値予報マップ  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  を入力とし, 天気予報コメント  $\mathbf{y} = (y_1, y_2, \dots, y_m)$  を出力とする encoder-decoder モデルを提案する. ここで,  $x_i$  は各時刻の数値予報マップ,  $y_j$  は出力単語を

入力: エリア毎の時系列数値予報マップ



出力: 天気予報コメント

図 3: モデルの概要図. ここで,  $\vec{v}_i$  は各時刻の数値予報マップの特徴ベクトル,  $\vec{M}$  はメタ情報の埋め込みベクトル,  $\vec{h}_i$ ,  $\vec{s}_j$  はそれぞれ encoder, decoder の隠れ状態を表す.

表す. decoder には, 再帰的ニューラルネットワーク言語モデルを用いる. 複数の物理量を含む数値予報マップの encode 手法としていくつか考えられるため, 本研究では, 2 つの手法の提案を行い, 実験的に手法の比較検討を行う. 以降では, エリア毎の数値予報マップの抽出手法, 2 つの数値予報マップの encode 手法, メタ情報の考慮手法について詳細に説明する.

#### 4.1 エリア毎の数値予報マップの抽出

天気予報コメントは, 全国の各エリアごとに, 周辺の気象情報を参考にして記述される. 本研究では, エリア周辺の数値予報マップと各エリアの天気予報コメントの対応付けを行った. 具体的には, 各エリアの座標情報を参照し, 数値予報マップから各エリアを中心とする  $K \times K$  のミニマップを切り出し, 各エリアの天気予報コメントと対応付けを行った.

#### 4.2 数値予報データの encode 手法

##### 4.2.1 CNN を用いた手法

画像認識や画像の説明文生成タスクでは, 入力画像の特徴を抽出する手法として畳み込みニューラルネットワーク (CNN) が広く使われている. そこで, 本研究ではまず, エリア毎の数値予報マップから, 数値の特徴や物理量間の関係を獲得するために CNN を用いて特徴抽出を行う. 画像を用いた CNN では通常, RGB の 3 チャンネル (Red, Green, Blue) を使用するが, 本研究では, 使用する物理量のタイプ数  $P$  分の入力チャンネルを用いて特徴抽出を行い, 各時刻の特徴ベクトル  $\vec{v}_i$  を生成する. 次に, 時系列順に, 各特徴ベクトル  $\vec{v}_i$  を Long Short-Term Memory (LSTM) を用いた再帰的ニューラルネットワーク (RNN) に入力し, 各タイムステップの隠れ状態  $\vec{h}_i$  を計算する. encode 処理後に, decoder の初期値として  $\vec{h}_n$  を入力する.

## 4.2.2 MLP を用いた手法

画像認識や動画キャプション生成では、位置不変性の観点から CNN を用いて特徴抽出を行うことが一般的である。しかし、本研究の場合、エリアごとに抽出したミニマップを入力としており、着目しているエリアは常にその中心に位置しているため、ミニマップ上の位置をそのまま考慮したモデルの方が適している可能性が考えられる。そのため本研究では、CNN の代替として多層パーセプトロン (MLP) を用いた特徴ベクトルの抽出も行う。具体的には、物理量ごとに  $K \times K$  の予測値を入力するために  $P \times K \times K$  個のユニットの入力層を持つ MLP に数値予報マップを入力し、各時刻の数値予報マップの特徴を表すベクトル  $\vec{v}_i$  を生成する。次に、CNN を用いる場合と同様、特徴ベクトル  $\vec{v}_i$  を RNN に入力し、 $\vec{h}_n$  を計算する。最後に、decoder の初期内部状態  $\vec{s}_0$  を、計算した  $\vec{h}_n$  で初期化する。

## 4.3 メタ情報の考慮手法

Li ら [4] の研究によると、decoder の隠れ状態にメタ情報などの追加的な情報を付与することで出力系列が変化することが報告されている。これらの知見を基に、本研究では、コメント配信時の日時などのメタ情報を decoder の隠れ状態に付与することでメタ情報を考慮した単語系列 (テキスト) の生成を行う。具体的には、考慮したいメタ情報ごとにメタ情報埋め込みベクトル  $\vec{M}$  を作成し、decoder の隠れ状態  $\vec{s}_j$  に加算する。天気予報コメントの場合、テキスト中で「今日、明日」などの時間帯や、表 1(1) 中の「春本番」のような季節特有の表現、表 1(2) 中の「今日 (月)」のように曜日、エリア名などが書かれることが多いため、月、日、時間、曜日、エリア名 (地名) ごとの埋め込みベクトル  $\vec{M}$  を用いる。

# 5 実験

## 5.1 実験設定

実験には、株式会社ウェザーニューズのピンポイント天気予報サービス<sup>1</sup>から収集した 2014 年から 2015 年の天気予報コメント 158,983 件を利用した。この内、2014 年に配信されたコメント 100,986 件を学習データ、2015 年に配信された 29,065 件、28,932 件をそれぞれ開発データ、テストデータとして使用した。また、数値予報マップのデータとして、京都大学生存圏研究所が運営する生存圏データベース<sup>2</sup>を通して公開されている気象庁作成の 2014 年から 2015 年の数値予報マップ 803 件を利用した。数値予報マップは天気予報コメントの配信時刻直前に計算されたものと対応付けを行い、使用している。テキストの形態素解析には

表 3: モデルの一覧表

Model	cnn	mlp	mlp-m	mlp-d	mlp-t	mlp-w	mlp-a	mlp-no
enc	CNN	MLP						
メ	○	○	×	○	○	○	○	×
タ	○	○	○	×	○	○	○	×
情	○	○	○	○	×	○	○	×
報	○	○	○	○	○	×	○	×
	○	○	○	○	○	○	×	×

MeCab<sup>3</sup>, MeCab の辞書には IPA 辞書, モデルの実装には Chainer<sup>4</sup>を用いた。

実験では、「今日明日の天気」の天気予報コメントを用いるため、24 時間先までの時系列数値予報マップを使用した。本研究では、時系列数値予報マップを 3 時間ごとに区切り、計 9 タイムステップの時系列数値予報マップを使用した。また、数値予報マップに含まれる 11 の物理量のうち、気圧を除く、海面更正気圧、東西風、南北風、気温、相対湿度、積算降水量、上層雲量、中層雲量、低層雲量、全雲量の計 10 の物理量を使用した。各タイムステップの積算降水量は、前タイムステップの値から差分を計算したものを使用した。数値予報マップの各値は、物理量ごとに学習データ内の平均値、標準偏差を用いて標準化を行った。

パラメータの学習には Adam を使用し、ミニバッチサイズは 80, epoch 数は 10 とした。単語埋め込みベクトルの次元は 100, メタ情報埋め込みベクトル  $\vec{M}$  の次元は 64 とした。encoder 及び decoder の隠れ状態ベクトル  $\vec{h}_i, \vec{s}_i$  の次元は 250 とした。CNN の入力チャンネル数は使用する物理量のタイプ数  $P$  と等しい 10, 畳み込みフィルタサイズは 2 とした。ミニマップのサイズ  $K$  は 5 とした。

実験では、評価指標に BLEU を使用し、各モデルによって生成されたテキストの自動評価を行った。また、生成されたテキストの内容の正確さ、テキストの流暢さを評価するために、人手による評価も行った。

自動評価では、まず、数値予報マップの encode 手法の比較を行うために、CNN を用いた手法 (cnn) と MLP を用いた手法 (mlp) の評価を行った。両モデルの decode 時には、5 つのメタ情報 (month, day, time, week, area) 全てを付与している。次に、メタ情報を付与することの有用性を確かめるために、メタ情報を用いない手法 (mlp\_no) の評価を行った。また、mlp をベースに、各メタ情報の有用性を確かめるために、各メタ情報を使用しない場合 (mlp-m, mlp-d, mlp-t, mlp-w, mlp-a) の比較を行った。表 3 に実験に使用したモデルの一覧表を示す。

人手による評価では、メタ情報としてエリア名を使用しないモデル (mlp-a), メタ情報を用いないモデル (mlp\_no) の計 2 つのモデルによって生成された 150 事例の天気予報コメントを手法名を伏せて 2 人の評価者

<sup>1</sup><http://weathernews.jp/index.html>

<sup>2</sup><http://database.rish.kyoto-u.ac.jp>

<sup>3</sup><http://taku910.github.io/mecab/>

<sup>4</sup><http://chainer.org>

表 4: 人手による評価の内訳と平均点

評価指標	点数	概要
正確さ	2	天気予報のポイントを正しく説明できている。
	1	誤りや一部の重要な情報の欠如がある。
流暢さ	2	自然言語テキストとして全く問題がない。
	1	「読みにくい」など不自然な点がある。

表 5: BLEU

Model	cnn	mlp	mlp-m	mlp-d	mlp-t	mlp-w	mlp-a	mlp_no
BLEU	0.133	0.141	<b>0.149</b>	0.142	0.127	0.095	<b>0.149</b>	0.123

に提示し<sup>5</sup>, 正確さと流暢さの2つの評価指標でそれぞれ2段階の評価を行った。表4に評価指標の概要を示す。評価者には, 正解テキストとして人手で書かれた天気予報コメントと配信日時, エリア名等のメタ情報も同様に提示した。また, 生成されたテキストの中に正解テキストから推測できない内容が含まれると評価者が判断した事例は評価対象から除外した。

## 5.2 実験結果

encode手法の評価, メタ情報の考慮手法の評価結果を表5, 人手による評価の内訳と平均点を表6に示す。各BLEUスコアは, 10 epoch内で開発データに対するスコアが最も高かった時のテストデータに対するスコアである。また, mlp-a, mlp\_noの各モデルの出力例と人手で記述された正解テキストを表7に示す。

encode手法の比較では, CNNとMLPを用いた手法(cnn, mlp)のBLEUスコアに大きな差は見受けられなかった。mlp\_noとメタ情報を使用するモデルmlp, mlp-a等のBLEUスコアを比較すると, メタ情報をdecode時に付与することでBLEUスコアが向上することが分かった。

人手による評価では, mlp\_noによって生成された天気予報コメントのうち, 評価指標の「正確さ」において約69%, 「流暢さ」において約99%の事例が高い評価を得た。mlp\_noとmlp-aの人手による評価結果を比較すると, メタ情報をdecode時に付与しないmlp\_noが各評価指標において高い平均点を得た。これは, mlp-aのdecode時の各時刻の隠れ状態 $\vec{s}_j$ にメタ情報埋め込みベクトル $\vec{M}$ を加算したことで, encodeされた数値予報の情報がテキストに反映されにくくなったためであると考えられる。また, BLEUによる自動評価で最も高いスコアであったmlp-aとメタ情報を使用しないモデルmlp\_noによって出力されたテキスト中の「曜日」を正しく出力できているか比較を行ったところ, mlp-aの場合, 91.1%, mlp\_noの場合, 16.5%の正解率であった。これにより, 曜日や月等の情報をdecode時に付与することで, 曜日や「春の陽気」のような季節特有の表現を正しく出力できるようになり, 単語の一致率を基にした評価指標であるBLEUのスコアが向上していたと考えられる。

<sup>5</sup>曜日情報は別途機械的に評価するため削除して使用した。

表 6: 人手による評価の内訳と平均点

	正確さ		平均点		自然さ		平均点	
	2	1			2	1		
mlp-a	153	109	1.58		234	28	1.89	
mlp_no	182	80	<b>1.69</b>		260	2	<b>1.99</b>	

表 7: 出力例

Model	配信日時	出力	平均点	
			正確さ	自然さ
正解	2015/1/9	明日はお出かけ OK	-	-
mlp-a	17:44 (金)	明日は雪舞う寒さ対策を	1.0	1.0
mlp_no		明日は穏やかな空	2.0	2.0
正解	2015/3/30	穏やかな陽気で週のスタート	-	-
mlp-a	03:18 (月)	今日は晴れて春の陽気	2.0	2.0
mlp_no		今日はお出かけ日和	2.0	2.0

## 6 結論

本研究では, 気象予報モデルをシミュレーションして得られる時系列数値予報マップから天気予報コメントを自動生成する手法を提案した。評価実験の結果, 提案手法により, 時系列の数値予報マップから人手で記述された正解テキストの内容に近い天気予報コメントを生成できることが分かった。

今後の課題として, 天気予報コメントは「今日も暑い1日となりそうです。」のように過去の天気と比較して記述されることが多いため, 過去の観測値を入力に用いることなどが挙げられる。

**謝辞** 本研究を進めるにあたり株式会社ウェザーニューズ様, 同社の萩行正嗣様からデータ提供のご協力をいただきました。ここに記して謝意を表します。また, 本研究はJST さきがけの支援を受けたものです。

## 参考文献

- [1] Gabor Angeli, Percy Liang, and Dan Klein. A simple domain-independent probabilistic approach to generation. In *Proc. of EMNLP'10*, pp. 502–512, 2010.
- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proc. of ICLR'15*, 2015.
- [3] Anja Belz. Probabilistic generation of weather forecast texts. In *Proc. of NAACL-HLT'07*, pp. 164–171, 2007.
- [4] Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. In *Proc. of ACL'16*, pp. 994–1003, 2016.
- [5] Percy Liang, Michael Jordan, and Dan Klein. Learning semantic correspondences with less supervision. In *Proc. of ACL'09*, pp. 91–99, 2009.
- [6] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Proc. of NIPS'14*, pp. 3104–3112, 2014.
- [7] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proc. of ICCV'15*, pp. 4534–4542, 2015.
- [8] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proc. of ICML'15*, pp. 2048–2057, 2015.