時系列数値データからの概況テキストの自動生成

村上 聡一朗[†,§] 渡邉 亮彦[†,§] 宮澤 彬[‡,¶,§] 五島 圭一[†,§] 柳瀬 利彦[*] 高村 大也[†,§] 宮尾 祐介[‡,¶,§]
[†]東京工業大学 [‡]総合研究大学院大学 [*]日立製作所 [¶]国立情報学研究所 [§]産業技術総合研究所
{murakami,watanabe}@lr.pi.titech.ac.jp, goshima.k.aa@trn.dis.titech.ac.jp,
toshihiko.yanase.gm@hitachi.com, takamura@pi.titech.ac.jp,
{miyazawa-a,yusuke}@nii.ac.jp

1 はじめに

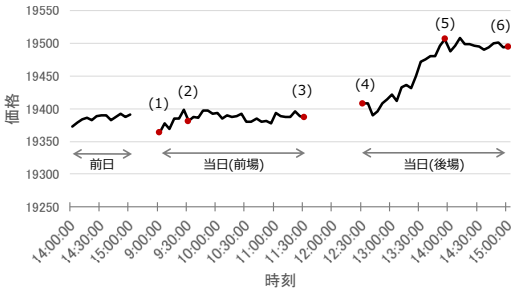
金融や医療，情報通信などの多くの分野において，様々な形式のデータを取り扱う機会が増えてきている．しかし，大規模で複雑なデータを専門知識のない人が見て解釈することは容易ではない．そのため，データの概要を説明する概況テキストを自動的に生成する技術の必要性が高まっている．時系列データの概況テキストでは，テキストが書かれる時間帯や，過去のデータの履歴によって言及すべき内容は様々である．また，数値の時系列データの場合，時系列中の数値や，過去との差分を計算した値が言及されることが多々ある．

本稿では，日経平均株価の概況テキストを生成するタスクを例として，時系列数値データの多様な特徴を抽出してテキスト化する手法を提案する．図 1 に示すように，概況テキストでは「上がる」「下がる」といった単純な特徴だけが表出されるわけではない．この例では,「続落」「反発」のように価格の履歴を参照する表現 (1，3，6),「上げに転じる」のように時系列データの変化を示す表現 (2),「始まる」「前引け」「午後」「大引け」などテキストが書かれる時間帯に依存する表現 (1，3，4，6) が見られる．また，数値に言及する場合は，価格が直接言及される (3，6) こともあれば，履歴からの差分 (3，6) や，切り上げ・切り捨てした値 (5) が用いられることもある．本研究では，機械翻訳や文書要約などで広く用いられている encoder-decoder モデル [6] をベースラインとし，上記のような多様な特徴を自動抽出してテキスト化するための encode/decode 手法を探求する．

2 関連研究

時系列データの概要を人に分かりやすく伝えるために，データからテキストを生成する様々な研究が行われている．例えば，Angeli ら [1] は，時系列の気象情報から天気予報の概況を説明するテキストを自動生成する研究に取り組んでいる．Gkatzia ら [3] は，一定期間毎の学習態度を記録した時系列データから学生へのフィードバックのテキストの自動生成を行っている．

Aoki ら [2] は，日経平均株価を対象に，株価がどのように変動したかを説明するテキストの生成に取り組んでいる．しかし，この研究では，図 1(3)，(5) のような終値や値上げ幅などの数値への言及を行う取り組みが行われていない．これに対し本研究では，数値の変動の概要だけでなく，入力の時系列数値データを参照した上で，実際の数値へ言及を行うテキストを生成する手法を提案する．



(1) 09:00 日経平均、続落で始まる
(2) 09:29 日経平均、上げに転じる
(3) 11:30 日経平均、続落　前引けは５円安の19386円
(4) 12:30 日経平均、午後は上昇で始まる
(5) 13:54 日経平均、上げ幅100円超える
(6) 15:00 日経平均、反発　大引けは102円高の19494円

図 1: 日経平均株価と概況テキスト

3 提案手法

近年，機械翻訳や文書要約などの様々な系列生成タスクにおいて，encoder-decoder モデル [6] を用いた手法が提案され，有用性が示されている．本研究では，時系列数値データに対する概況テキストの生成を，時系列数値データから単語系列を生成する系列生成タスクとして考え，encoder-decoder モデルを用いた手法を提案する．decoder には，テキスト生成の研究において広く使われている再帰的ニューラルネットワーク言語モデル (Recurrent Neural Network Language Model; RNNLM) を利用する．時系列数値データの概況を説明するテキストを記述するためには，数値データの短期的または長期的な変化，絶対的または相対的な変化などを捉える必要がある．また，画像の 2 値化やグレースケール，単語の分散表現や N-gram などの表現方法と同様に，数値をモデルで取り扱うための表現方法としていくつかの手法が考えられる．本研究では，時系列数値データの encode/decode 手法を複数提

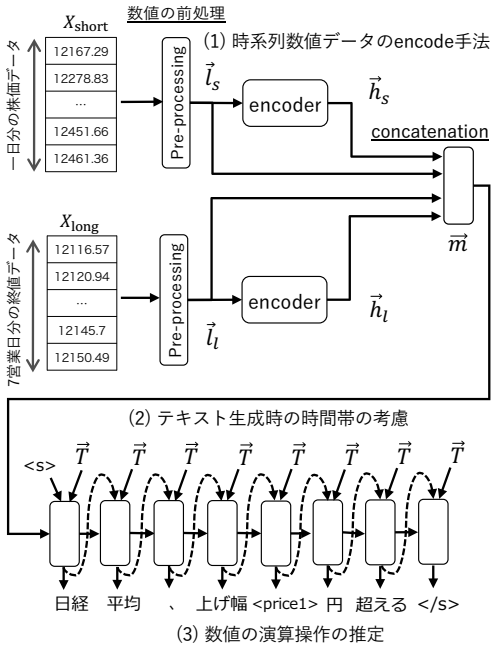


図 2: モデルの概要図

案し，実験を通して各手法の有効性を比較検討する．

本研究で提案するモデルの概要を図 2 に示す．本研究では，日経平均株価を時系列数値データの例として扱う．提案モデルでは，短期的または長期的な数値の変動を捉えるために，短期的な時系列数値データとして，一日分の株価データ[1] $X_{\text{short}} = (x_{s_0}, x_{s_1}, \cdots, x_{s_{61}})$，長期的な時系列数値データとして，7 営業日分の終値 $X_{\text{long}} = (x_{l_0}, x_{l_1}, \cdots, x_{l_6})$ を入力として利用する．また，encode には，一日分の株価データ X_{short} と 7 営業日分の終値データ X_{long} をそれぞれ encode する 2 つの encoder を用いる．decode 時には，各 encoder の出力及び前処理した数値ベクトルを受け取り，数値の変化を概況するテキスト (単語列) を RNNLM で推定する．また，時系列数値データの概況テキストにおいて，テキストが書かれる時間帯によって言及する内容や使われる表現が変わるため，decode 時に，テキストを記述する時間帯の情報を利用する．概況テキストの生成時に実際の数値や変動幅に言及する際には，入力した時系列数値データ中から正しい数値を出力するための演算操作を推定し，計算することで数値の出力を行う．以降では，(1) 時系列数値データの encode 手法，(2) テキスト生成時の時間帯の考慮，(3) 数値の演算操作の推定について詳細を説明する．

3.1 時系列数値データの encode 手法

株価データを encoder へ入力する際には，前処理した値を用いる．使用する前処理手法の式を以下に示す;

$$x_{\text{norm}_i} = \frac{2 \times x_i - (\bar{x}_{\max} + \bar{x}_{\min})}{\bar{x}_{\max} - \bar{x}_{\min}}, \tag{1}$$

$$x_{\text{std}_i} = \frac{x_i - \mu}{\sigma}, \tag{2}$$

$$x_{\text{move}_i} = x_i - r_i. \tag{3}$$

式 (1) では，入力データ X 中の最大値 $\bar{x}_{\max}$，最小値 $\bar{x}_{\min}$ を用いて $[-1, 1]$ へ正規化を行う．式 (2) では，学習に使用する全株価データ X_{all} 内の平均値 μ，標準偏差 σ を用いて標準化を行う．式 (3) では，前日の終値からの価格の変動を捉えるために，前日の終値 r_i から各タイムステップの価格 x_i の差分を計算する．

前処理は，一日分の株価データ X_{short} に対して行い，数値ベクトル $\vec{l}_s$ を作成する．7 営業日分の終値データ X_{long} に対しても同様に前処理を行い，数値ベクトル $\vec{l}_l$ を作成する．decoder の初期状態の設定時には，encoder の出力状態ベクトル $\vec{h}_s$，$\vec{h}_l$ と前処理した数値ベクトル $\vec{l}_s$，$\vec{l}_l$ を結合した multi-level representation ベクトル $\vec{m}(= \vec{h}_s \oplus \vec{h}_l \oplus \vec{l}_s \oplus \vec{l}_l)$ を使用する．

複数の前処理手法を用いる場合，短期的及び長期的な時系列数値データともに前処理手法のタイプ数分の encoder を使用する．例えば，正規化と標準化を行う場合，X_{short} と X_{long} に対して 2 つずつ，計 4 つの encoder を利用する．decoder の初期状態の設定時には，それぞれの出力状態ベクトル $\vec{h}_{s_{\text{norm}}}$，$\vec{h}_{l_{\text{norm}}}$，$\vec{h}_{s_{\text{std}}}$，$\vec{h}_{l_{\text{std}}}$ と前処理した数値ベクトル $\vec{l}_{s_{\text{norm}}}$，$\vec{l}_{l_{\text{norm}}}$，$\vec{l}_{s_{\text{std}}}$，$\vec{l}_{l_{\text{std}}}$ を結合したベクトル $\vec{m}$ を用いる．

3.2 テキスト生成時の時間帯の考慮

時系列データの概況テキストは，テキストが書かれる時間帯に依って言及すべき内容は様々である．例えば株価の概況テキストの場合，一般的に，図 1 中の (1), (6) のように，取引が始まる時間帯には「前日から価格がどのように変動したか」，取引が終了する時間帯には「値上げ幅と終値はいくらか」等が言及される．

Li らが行った encoder-decoder モデルを用いた対話モデルの研究 [5] では，decode 時にある個人の人格情報を追加的に入力することで，その人の話し方や背景などの特徴を捉えた単語列を生成できることが報告されている．これらを踏まえ本研究では，decode 時の各タイムステップの状態 $\vec{s}_j$ に時間帯情報 $\vec{T}$ の付与を行い，時間帯を考慮したテキストの生成を行う．具体的には，概況テキストが配信される時間帯 (9 時，15 時等) を入力として時間帯情報埋め込みベクトル $\vec{T}$ を作成し，decode 時の各時刻の隠れ状態ベクトル $\vec{s}_j$ に時間帯情報埋め込みベクトル $\vec{T}$ を加算する．

3.3 数値の演算操作の推定

言語モデルを用いたテキスト生成において，一定の出現頻度よりも少ない単語は out-of-vocabulary(OOV) として，<unk> などのタグに置き換えられることが一般的である．しかし，数値などのバリエーションが多い単語は出現頻度が少ないため，OOV として扱わ

[1] 本研究では，5 分足の日経平均株価を用いる．従って，概況テキストが書かれる直近の時間帯から 62 タイムステップ前までの株価を一日分の時系列株価データとする．

れてしまうことがある．機械翻訳の分野では，OOVの対策として，出現頻度が少なくなりやすい固有名詞などを入力テキストからコピーを行い，単語を出力するという手法が提案されている [4]．

入力の数値データに言及するテキストでは，図 1 中の (3，6) のように，入力データに含まれる数値について直接言及することが多い．しかし，それだけではなく，履歴からの差分 (3，6) や，切り上げ・切り捨てした値 (5) が用いられることもある．そのため，入力データから数値をコピーするだけでなく,「差分の計算」等の数値の演算操作が必要となる．そこで本研究では，数値について言及する場合，入力の数値データに対する演算操作を推定し，計算した数値を出力する手法を提案する．

時系列株価データからテキスト中で言及する価格を出力できるようにするために，価格箇所の推定時に，価格の演算操作を表す汎化タグを推定し，予め定義した演算操作に基づいて価格の計算を行い，計算結果の価格で汎化タグを置換する．また，前処理として，テキスト中の価格箇所を <price1> や <price2> という汎化タグに置換する．使用する汎化タグは，言及する価格の性質に依って異なる．表 1 に事前に定義した演算操作と汎化タグを示す．例えば,「日経平均、反発 午前終値は 227 円高の 16610 円」の場合,「227」は，前日の終値 x_{l_6} と最後のタイムステップの価格 $x_{s_{61}}$ の差を表すため，<price1> に置換し,「16610」は，最後のタイムステップの価格 $x_{s_{61}}$ を表すため，<price6> に置換する．前処理時には，テキストの価格に対して全ての演算 (12 種類) を行い，正解の価格に最も近い価格を計算した演算操作の汎化タグを正解とした．

テスト時に前日の終値 x_{l_6} が 14612 円で最後のタイムステップの価格 $x_{s_{61}}$ が 14508 円の時系列数値データを入力し，モデルが「日経平均、反落で始まる 下げ幅 <price2> 円超、<price7> 円台」と推定した場合，<price2> は，前日の終値 x_{l_6} と最後のタイムステップの価格 $x_{s_{61}}$ の差を 10 の位で切り捨てた価格である「100」，<price7> は，最後のタイムステップの価格 $x_{s_{61}}$ を 100 の位で切り捨てた価格である「14500」に置換し,「日経平均、反落で始まる 下げ幅 100 円超、14500 円台」を出力テキストとする．

4 実験

4.1 実験設定

実験には，時系列株価データとして IBI-Square Stocks[2]から収集した 2013 年 3 月から 2016 年 10 月までの 5 分足の日経平均株価，概況テキストとして日経 QUICK ニュース社が提供する日経平均株価ニュースのヘッドラインテキスト，計 7,351 件を利用した．ヘッドラインテキストの内，5,880 件を学習データ，730 件を開発データ，741 件を評価データとして利用した．また，テキストの形態素解析には MeCab[3]，モデルの実装には Chainer[4]を用いた．

表 1: 定義した演算操作と汎化タグ

汎化タグ	操作内容
<price1>	x_{l_6} と $x_{s_{61}}$ の差を返す
<price2>	x_{l_6} と $x_{s_{61}}$ の差を 10 の位で切り捨て
<price3>	x_{l_6} と $x_{s_{61}}$ の差を 100 の位で切り捨て
<price4>	x_{l_6} と $x_{s_{61}}$ の差を 10 の位で切り上げ
<price5>	x_{l_6} と $x_{s_{61}}$ の差を 100 の位で切り上げ
<price6>	$x_{s_{61}}$ を返す
<price7>	$x_{s_{61}}$ を 100 の位で切り捨て
<price8>	$x_{s_{61}}$ を 1000 の位で切り捨て
<price9>	$x_{s_{61}}$ を 10000 の位で切り捨て
<price10>	$x_{s_{61}}$ を 100 の位で切り上げ
<price11>	$x_{s_{61}}$ を 1000 の位で切り上げ
<price12>	$x_{s_{61}}$ を 10000 の位で切り上げ

学習時には，全 62 タイムステップから成る一日分の株価データ X_{short}，全 7 タイムステップから成る過去 7 営業日分の終値データ X_{long} と概況テキストのペアを使用する．テスト時には，株価データのみを用いて概況テキストを生成する．単語埋め込みベクトルの次元は 128，時間帯情報埋め込みベクトルの次元は 64，encoder の隠れ状態の次元は 256 とした[5]．モデルパラメータの最適化手法には Adam を使用し，ミニバッチのサイズは 100，epoch 数は 30 とした．

評価指標には，実際の株価の概況テキストと生成されたテキストの一致度合いを測る目的として BLEU を使用した．また,「続落，反発，上げに転じる」といった短期的・長期的な時系列数値データの変動を説明する表現や「始まる，前引け」などの時間帯を考慮した表現を正解テキストと比較して正しく出力できているかを評価するために，F 値による評価も行った．

実験では，まず，encode 手法の検討として，多層パーセプトロン (Multi-Layer Perceptron; MLP)，畳み込みニューラルネットワーク (Convolutional Neural Network; CNN)，再帰的ニューラルネットワーク (Recurrent Neural Network; RNN) のそれぞれを encoder としたモデル (mlp-enc，cnn-enc，rnn-enc) の比較を行う．次に入力の時系列データから短期的及び長期的な変化を捉える能力があるかを確認するために，一日分の株価データまたは 7 営業日分の終値データを使用しないモデル (-short，-long) の比較を行う．また，数値データの表現手法の有用性を確かめるために，mlp-enc モデルをベースとして，各前処理手法 (正規化，標準化，前日との差分)，multi-level representation をそれぞれ使用しないモデル (-norm，-std，-move，-multi)

[2]http://www.ibi-square.jp/index.htm

[3]http://taku910.github.io/mecab/

[4]http://chainer.org

[5]decoder の隠れ状態の次元数は，使用する前処理手法の数によって変化することに注意されたい (正規化，標準化，前日の終値との差分の 3 手法を用いる場合，$(256 \times 3) \times 2 = 1536$ である)．

表 2: BLEU

Model	baseline	mlp-enc	cnn-enc	rnn-enc	-short	-long
BLEU	0.244	0.415	0.414	0.415	0.356	0.397
Model	-std	-norm	-move	-multi	-num	-time
BLEU	**0.424**	**0.424**	0.408	0.397	0.313	0.358

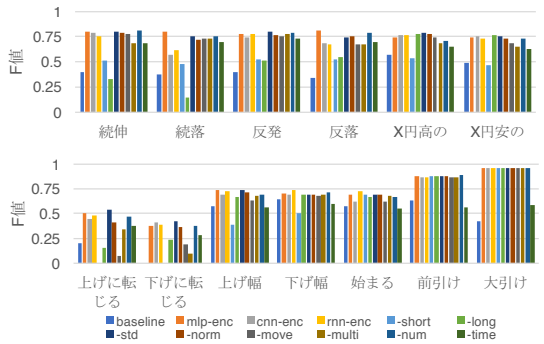


図 3: F 値

の評価を行う．最後に，数値の演算操作の推定手法，および，時間帯情報の入力手法の有用性を確かめるために，各手法を使用しないモデル (-num，-time) の評価を行う．ベースラインとして，一日分の株価データのみを入力として，encoder に MLP，前処理手法に正規化と標準化を使用するモデル (baseline) を用いた．

4.2 実験結果

実験では，時系列数値データの表現手法，数値の演算操作の推定手法，時間帯の考慮手法の評価を行う．BLEU，F 値による評価の実験結果を表 2，図 3 に示す．また，出力例を表 3 に示す．

MLP，CNN，RNN をそれぞれ encoder としたモデル (mlp-enc，cnn-enc，rnn-enc) の BLEU スコアを比較すると大きな差は見受けられなかった．短期的及び長期的な時系列数値データを両方用いたモデル (mlp-enc 等) とそれぞれ用いないモデル (-short，-long) の比較では，両方のデータを用いることで BLEU スコアが向上することが確認できた．また，両方のデータを用いたモデルは，-short 及び-long と比較して「続落，反発，上げに転じる」等の短期的及び長期的な変化を説明する表現を正しく出力できることが分かった．

数値データの表現手法の比較では，標準化または正規化を行わないモデル (-std，-norm) を用いた場合の BLEU スコアが最大となることが分かった．また，全ての数値データの表現を用いたモデル (mlp-enc 等) と-move，-multi を比較すると，BLEU スコアが低下することが分かった．これにより，当日の価格と前日の終値との差分を計算する前処理手法，encoder の出力に加え decoder に前処理した数値を直接渡すこと (multi-level representation) が入力の時系列数値データのより良い表現に寄与していることが推測できる．

-num モデルでは，数値の演算操作の推定を行わず，言語モデルから数値を “単語” として出力する．その

表 3: 出力例

Model	出力
baseline	日経平均、反落 前引けは 57 円安の 20606 円
mlp-enc	日経平均、続伸 大引けは 32 円高の 16906 円
-norm	日経平均、続伸 大引けは 32 円高の 16906 円
-num	日経平均、続伸 大引けは 28 円高の <unk> 円
-time	日経平均、続伸 前引けは 32 円高の 16906 円
正解	日経平均大引け、続伸 終値は 32 円高の 16906 円

ため，表 3 のように，<unk> として出力される事例が多く，演算操作の推定を行うモデル (mlp-enc，-std 等) よりも低い BLEU スコアとなった．また，数値の演算操作の推定を行うことにより，多くの事例で正しい価格の出力ができていることを確認できた．

-time モデルでは，decode 時の decoder の状態に時間帯情報を付与していない．図 3 の時間帯を考慮するモデル (rnn-enc，-num 等) と比較すると，時間帯に関して言及を行う表現 (始まる，前引け，大引け) を正しく出力できていないことが分かる．

5 結論

本研究では，日経平均株価を時系列数値データの例として，時系列数値データから概況テキストを自動生成する手法を提案した．時系列数値データを概況するテキストには，時系列中の数値への言及や過去の数値の変動との比較，テキストが書かれる時間帯によって言及する内容が異なる，などの特徴があり，本研究では大きく分けて 3 つの手法を提案し，有用性を示した．今後の課題として，株式市場全体の個別銘柄の中から注目すべき個別銘柄に対する概況テキストの生成，といった複数の時系列数値データを考慮した概況テキストの生成などが考えられる．

謝辞 この成果は，国立研究開発法人新エネルギー・産業技術総合開発機構 (ＮＥＤＯ) の委託業務の結果得られたものです．

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] Kasumi Aoki and Ichiro Kobayashi. Linguistic summarization using a weighted n-gram language model based on the similarity of time-series data. In *IEEE International Conference on Fuzzy Systems*, pp. 595–601, 2016.

[3] Dimitra Gkatzia, Helen Hastie, and Oliver Lemon. Comparing multi-label classification with reinforcement learning for summarisation of time-series data. In *Proc. of ACL'14*, pp. 1231–1240, 2014.

[4] Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. Pointing the unknown words. In *Proc. of ACL'16*, pp. 140–149, 2016.

[5] 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.

[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.