Vol.2020-NL-246 No.30
Vol.2020-SLP-134 No.30
2020/12/3

疑似負例を用いたData-to-Textモデルの学習

上原 由衣 * 2,a)　石垣 達也 *2,b)　青木 花純[1,c)]　能地 宏[2,d)]　五島 圭一[4,e)]　小林 一郎[1,2,f)]
宮尾 祐介[5,2,g)]　高村 大也[3,2,h)]

概要：本稿では，日経平均データなどの時系列数値データを入力とし，その値動きを説明する市況テキストを出力する data-to-text 課題を扱う．従来，data-to-text モデルは時系列数値データと正解テキストの対を用いて学習される．既存モデルによる生成文は，例えば「日経平均，続落」を出力すべき入力に対し，「日経平均，反発」と出力するなど，値動きを表す重要語について致命的なエラーを含むことがある．本研究では，このようなエラーを軽減し生成文の正しさを向上させる目的で，正解文だけでなく間違いを含む文を疑似負例として自動生成し学習時に活用する枠組みを提案する．疑似負例は「続落」「反発」といった値動きを表現する語をあらかじめ定義し，正解文中の重要語を別の重要語で置き換えることで自動生成する．疑似負例の活用によるエラー削減の効果について，疑似負例の種類，および学習時に用いる損失関数という 2 つの観点から分析する．実験より，1) 疑似負例の活用により生成文の流暢性を失うことなく正しさが向上する，2) 重視する性能指標によって選択すべき損失関数は異なる，3) 特定の規則により生成した疑似負例はより効果的に正しさの向上に寄与する，という 3 つの知見が得られた．また，人間による評価においても，負例の活用が生成文の正しさの向上に寄与することが確かめられた．

Learning with Contrastive Examples for Data-to-Text Generation

Uehara Yui[2,a)]　Ishigaki Tatsuya[2,b)]　Aoki Kasumi[1,c)]　Noji Hiroshi[2,d)]　Goshima Keiichi[4,e)]
Kobayashi Ichiro[1,2,f)]　Miyao Yusuke[5,2,g)]　Takamura Hiroya[3,2,h)]

1. はじめに

本稿では，data-to-text 課題の一例として日経平均や為替といった複数の時系列数値データを入力とし，日経平均の値動きを説明する文を出力する設定を扱う．近年活発に研究されているニューラルネットワークによる data-to-text モデルは，人物の略歴 [8], [9]，スポーツ概況 [4], [6], [13], [14], [19]，市況コメント [1], [2], [10] といった幅広いテキスト生成を対象としている．

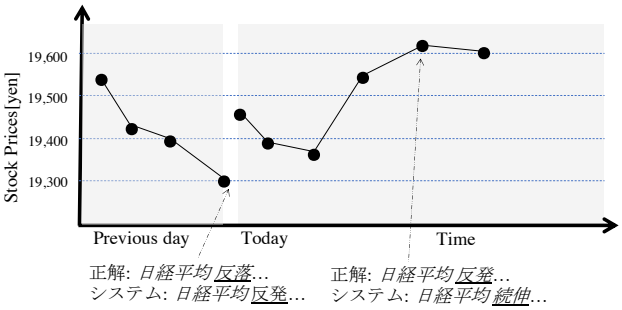

図 1　既存システムの出力例．反落と出力すべき箇所で反発を出力するなど値動きを表す重要語のエラーが見られる．

このようなモデルは流暢なテキストを出力する一方，生成文の正しさという観点からは問題のある出力が観測される．例えば，図 1 に示すように，本来「続落」や「反発」

*equal contribution.

1 お茶の水女子大学
2 産業技術総合研究所
3 東京工業大学
4 早稲田大学
5 東京大学
a) yui.uehara@aist.go.jp
b) ishigaki.tatsuya@aist.go.jp
c) g1120501@is.ocha.ac.jp
d) noji@aist.go.jp
e) keiichi.goshima@aoni.waseda.jp
f) koba@is.ocha.ac.jp
g) yusuke@is.s.u-tokyo.ac.jp
h) takamura@pi.titech.ac.jp

といった単語の出力が期待される状況において，「反落」が出力されるといった致命的なエラーを含むことがある．このようなエラーは，最悪の場合，文の意味を逆転させることもあり，生成文の正しさに大きく影響する．

同様の問題は市況テキスト生成だけにとどまらず，機械翻訳や要約などテキストを入力とする言語生成課題においても広く見られる．このような言語生成課題においては，アラインメント辞書による遷移確率 [3], [16] やコピー機構 [15] を用いて，入力と出力が意味的により等しくなるよう工夫することで，生成文の正しさを向上させる手法が存在する．一方，data-to-text 課題においては入力がテキストではない．そのため，機械翻訳や要約と異なり，入出力間の単語アラインメントやコピー機構といった方策の適用が難しい．本稿で提案する枠組みでは，正解文に加え疑似負例も用いて学習する．これにより，生成すべき文に加えて**生成すべきではない文**も教師データとして与え，生成文の正しさを向上させることを目指す．ここで，疑似負例とは「日経平均，続落」といった正解文から一語のみを置換し生成した，「日経平均，反落」といった生成すべきではない文を考える．本稿では，疑似負例の作成手法や，学習時に用いる損失関数の違いが出力文の正しさに与える影響について分析する．

負例を教師データとして用いる学習手法はこれまでにも，主に言語モデルの性能向上や分析，言語生成における単語繰り返し問題の軽減のためにいくつか提案されている．例えば，Huang ら [5] はビーム探索内で文法的に崩れた文のスコアを下げるためにマージンに基づく損失関数により負例を考慮する学習法を提案した．さらに，Noji と Takamura [11] は疑似的に作成した文法的に崩れた文を負例として，マージンに基づく損失関数を用いて言語モデルを学習することで，言語モデルがより正しく構文的な知識を捉えることを示した．言語生成課題の研究においては，同じ単語が繰り返し出力される問題を解決するために Welleck ら [18] が単語の繰り返しを負例として扱い，尤度に基づく損失関数を用いて学習する手法を提案した．本研究では，正しさの向上とは異なる目的で用いられてきたこれら損失関数が，言語生成における正しさの向上にどのように寄与するか検証する．

自動評価指標および人間によるランキング評価による実験より，1) 疑似負例を用いた学習により生成文の流暢性を損なうことなく正しさが向上する，2) 重視する性能指標により用いるべき損失関数が異なる，3) より近い意味の単語に置換する規則で生成した疑似負例が生成文の正しさをより向上させる，という 3 つの知見を得たことを報告する．

2. 疑似負例を用いたモデル学習

本節では，疑似負例を用いて data-to-text モデルを学習する枠組みについて述べる．以後，疑似負例の作成手法および 3 つの異なる損失関数による学習手法について順に議論する．

重要語	単語レベルの疑似負例リスト
続伸	続落, 反発, 反落
続落	続伸, 反発, 反落
反発	続伸, 続落, 反落
反落	続伸, 続落, 反発
上げ幅	下げ幅
下げ幅	上げ幅
高	安
安	高

表 1 重要語として定義した 8 単語と単語レベルの疑似負例の生成規則．文レベルの疑似疑似負例の生成にもこの規則を用いる．

2.1 規則による擬似負例生成

本稿における疑似負例は，単語レベルおよび文レベルに分かれる．これらの疑似負例は 2.2 節において説明する損失関数の計算に用いられる．

まず，単語レベルの疑似負例から説明する．我々の定義した単語レベルの疑似負例の作成規則を表 1 に示す．規則の作成においては，まず学習データを単語分割し頻度順に並べ替え，頻出語の上位から値動きについて言及する単語 (「反落」「反発」など) であるか否かを人手により判定した．その上で，値動きについて言及する単語のうち上位 8 語を重要語として定義した．これら 8 単語から 2 単語を選ぶすべての組み合わせを単語レベルの疑似負例作成規則の候補とした．表 2 に示すように，上位 8 語の重要語のいずれかが使用したデータセット中の文の 77%に含まれる．

文レベルの疑似負例は，正解文中の重要語を単語レベルの疑似負例によって置換することで作成する．ここで，単語レベルの疑似負例候補の中には「続伸」を「上げ幅」を置き換える規則のように，文レベルの疑似負例を作成する際に非文を生成する規則が存在する．例えば，「日経平均，続伸」という正しい文から「日経平均，上げ幅」という非文を生成する．このような生成規則は人手によりチェックし，単語レベルおよび文レベルの負例生成規則から取り除いた．表 1 に，最終的な規則定義を示す．

本稿では日本語データを対象とするが，上位語を重要語とし非文を生成する規則を除外するという単純な手法自体は言語依存しない．この手法の以外にも動詞を名詞に変換する規則は除外する人手によらない方策も考えうるが，市況テキストにおいては体言止めが多く，上位 8 語の組み合わせであればコストも小さく，安全性も高い．

2.2 学習法

提案する枠組みは Aoki ら [2] の Encoder-decoder モデルを基にする．このモデルにおいて，Encoder は日経平均，ダウ平均，為替などをそれぞれ多層パーセプトロンにより

読み込み，得られた各指標の表現を結合したベクトルを用いて LSTM によるデコーダを初期化する．デコーダは逐次単語を出力し最終的な市況コメントを得る．訓練時には学習データ中に含まれる正解文を用いて，交差エントロピーにより損失関数を計算する．我々の提案する枠組みでは訓練時の損失関数において正解文のみならず，前述した疑似負例も考慮する損失関数を用いる．以下に我々が生成文の正しさの向上のために用いる，3 つの損失関数について述べる．

2.2.1 尤度に基づく損失 (UNLIKE)

この損失関数は，Welleck ら [18] によって，言語生成モデルが同一の単語を連続して出力する問題に対処するために提案された．Welleck ら [18] らは直前に出力した単語を単語レベルの負例として扱い，次の時刻において負例の生起確率を下げるようモデル化した．我々の提案する枠組みにおいては，この損失関数を表 1 において定義する単語レベルの疑似負例の生起確率を下げ，生成文の正しさを向上させるよう以下のようにモデル化する:

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

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

ここで，$con(x_i)$ は表 1 の規則定義に基づき，単語レベルの疑似負例集合を返す．ハイパーパラメータ α によって，正解文を生成する言語モデルとしての良さを最適化する第 1 項と，単語レベルの疑似負例の生成を抑制する第 2 項の重要度を調整する．α は開発セットを用いて調整する．

2.2.2 文単位のマージン損失 (SENT)

この損失関数は正解文 $\mathbf{x}$ の尤度と文レベルの疑似負例 $\mathbf{x}^*$ の尤度の差が大きくなるほどに，損失が小さくなる．すなわち，正解と疑似負例のマージン最大化を目指す:

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

ここで，δ は $\mathbf{x}$ と $\mathbf{x}^*$ のマージンを制御する．この損失関数は，Noji と Takamura [11] によって，言語モデルが捉えることのできる文法構造をを分析する目的で提案された．一方，本研究ではこの損失関数を生成文の正しさの向上のために用いる．この損失関数は文全体を考慮しながら，正解文と文レベルの疑似負例の差に着目しながら言語モデルを学習する．しかしながら，文全体の中でどの箇所が値動きを表す重要語であるか，といった単語レベルの教師は明示的に用いない．

訓練時には交差エントロピーによる損失関数も用いる．各バッチに対し，まずはじめに正解文のみを用いて交差エントロピーを最小化するよう学習する．その後，文レベルの疑似負例を表 1 の規則を用いて生成し，文単位のマージン損失による追加学習を行う．マージン損失による学習時には，重要語を含まない文は学習データから取り除き，重要語を複数含む文の場合はランダムに重要語を 1 つ選択し文レベルの疑似負例を生成する．作成した疑似負例からバッチ数の 2 分の 1 の事例を抽出し文単位のマージン損失をそれぞれの事例に対し計算する．バッチ内でのマージン損失の平均値を最終的な損失とする．

2.2.3 単語単位のマージン損失 (TOKEN)

Noji と Takamura[11] は前述した 2 つの損失関数を組み合わせる手法も提案している．この損失関数では 式 (1) 内の $g(x_i^*)$ を以下のように置き換える:

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

この損失関数は文レベルのマージン損失 (SENT) が捉えようとする言語モデルとして正解文らしさに加え，尤度に基づく損失 (UNLIKE) が抑制しようとする単語レベルの疑似負例双方の利点を取り入れようとするものである．このような損失関数の組み合わせ手法により，言語モデルの分析を行う研究 [11] においては文法的な知識をより正確に捉えた言語モデルを学習できることが報告されている．本稿では，このような損失関数が data-to-text モデルにおいて生成文の正しさの向上に，どのように寄与するか議論する．

3. 実験

本節では，実験に用いるデータセット，ハイパーパラメタの調整方法，自動評価および人間による評価手法について説明する．

3.1 データセット

実験には Aoki ら [2] による前処理済みデータを用いる．このデータには時系列数値データとして，テキストの生成対象指標である日経平均株価に加え，9 つの補助的な指標データ[*1]を含む．これらの指標は ThomsonReutersDataScope-Select[*2]から入手した．各指標に対する時系列数値データには 5 分おきにトラッキングされた価格が格納されている．この価格列から短期の値動きを表す時系列数値データ，長期の値動きを表す時系列数値データの 2 つを抽出する．直近の 6 時間分の 5 分ごとの値動きを短期の値動きを表す時系列数値データとし，直近 7 日分の 1 日の終値の系列を長期の値動きを表す時系列数値データとした．Aoki らのベースライン実装と同様に，日経平均も含め 10 の指標データの短期および長期の値動き系列を多層パーセプトロンで読み込み，合計 20 の結合ベクトルを用いてデコーダが文生成する．なお，正解文は，日経 QUICK ニュースから入手した．データセットの統計を表 2 に示す．

[*1] 具体的には，日経 225 先物 (N225), 東証株価指数 (TOPX), S&P 500 (SPX), ダウ平均 (DJI), FTSE100 種総合株価指数 (FTSE), 香港株価指数 (HSI), 円ドル為替 (JPYUSD), 円ユーロ為替 (EURJPY) および JNIc1 である．

[*2] https://hosted.datascope.reuters.com/DataScope/

	Train	Valid	Test
文数	16,276	1,866	1,951
- うち重要語を含む文数	12,589	1,583	1,615
重要語数	19,005	2,592	2,634
1 文あたりの平均語数	13.17	12.77	12.69

表 2 The statistics of the dataset.

3.2 ハイパーパラメータ

SENT および TOKEN に含まれるマージンを制御するハイパーパラメータ δ と，UNLIKE の項の重みを制御する α は開発セットを用いて調整する．調整時には $\{0.01, 0.1, 1.0, 10, 100\}$ の中から開発セットでの性能を最大化する値を選択する．後述するように性能評価する指標が複数あるため，それぞれの指標において開発セットでハイパーパラメータを調整し，評価セットでの性能を報告する．

学習時のバッチサイズは 50 に設定した．100 イテレーションごとに学習されたパラーメタを保存し，開発セットでの性能がもっとも高くなる保存モデルでの性能を評価セットを用いて測った．最適化器には Adam [7] を初期学習率 0.001 で用いた．各指標は 32 次元の固定長ベクトルによって表現した．デコーダ側の LSTM の隠れ層次元は 256，単語は 128 次元の単語埋め込みで表現した．実験では異なるシードによる乱数を用いた初期化を行い，3 回の実験の平均値を報告する．

3.3 自動評価指標による定量評価

本研究の目的は生成文の正しさの向上であり，BLEU [12] のみ用いる評価は不適切である．そこで，生成文の正しさを評価する 4 つの自動評価指標を提案する．これらの提案指標により，疑似負例の使用が生成文の正しさに与える影響を多角的に評価することを目指す．

3.3.1 言語モデルの出力する尤度による正解率

疑似負例も用いて学習されたデコーダ側の言語モデルは，正解文と疑似負例を正しく区別できることが望ましい．すなわち，学習された言語モデルが正解文に高い生起確率を与え，疑似負例には低い生起確率を与えるとすれば，言語モデルが正解文と疑似負例を正しく区別していると考えられる．Sennrich ら [16] の用いた評価手法に倣い，正解文とその疑似負例の尤度を比較し，以下の式で正解率を計算する:

$$\text{正解率} = \frac{|\,\text{正解文の尤度が高い事例}\,|}{|\,\text{評価事例}\,|}. \quad (4)$$

学習時とは異なり，評価時には表 1 の規則を用いて作成したすべての正例ー疑似負例ペアを用いる．正解に与えられる尤度が，表 1 により生成したすべての疑似負例よりも高い場合に式 (4) の分子としてカウントする．

3.4 重要語に関する適合率および再現率

2.1 節において説明したように，表 1 に示す重要語についてのエラーは，生成文の正しさに直接的に影響する．よって，重要語をどの程度正しく出力できているか自動評価する指標が必要である．そこで，重要語に関する適合率および再現率を以下の式により計算する:

$$\text{適合率} = \frac{|\,\text{正しく出力できた重要語}\,|}{|\,\text{システムが重要語を出力した重要語}\,|}, \quad (5)$$

$$\text{再現率} = \frac{|\,\text{正しく出力できた重要語}\,|}{|\,\text{正解文に含まれる重要語}\,|}, \quad (6)$$

3.5 エラー率

前述の評価指標はシステムがどの程度正しく重要語を出力するか図る指標であった．一方，どの程度重要語に関する明らかなエラーが含まれているかという観点からも評価を行いたい．そこで，明らかなエラーを定量的に測るための指標として以下のように計算されるエラー率を提案する:

$$\text{エラー率} = \frac{|\,\text{単語レベルの疑似負例が含まれる文}\,|}{|\,\text{重要語 1 つ以上を正解文に含む評価事例}\,|} \quad (7)$$

適合率，再現率，エラー率を計算する際には，評価事例の正解文に重要語とその単語レベルの疑似負例双方を含むものをあらかじめ除外する．

3.6 人間による評価

人間による評価は疑似負例の効果を検証するために不可欠である．2 つのデータセットを用いて人手評価を行う．WHOLE は評価セットからランダムにサンプルした 100 事例である．CRUCIAL は疑似負例を用いないベースライン BASE および疑似負例を用いる手法 TOKEN が異なる重要語を出力した事例からランダムに抽出した 40 事例である．後者のデータセットを用いることにより，疑似負事例を用いることでモデルの出力が変化した部分のみを検証できる．

評価者として金融分野の専門家 1 名に評価を依頼した．この評価においては，専門的な 10 の指標を目視し生成文を評価する必要があり，近年頻繁に用いられるクラウドソーシングを用いた大人数の素人による評価の使用は適切でない．Aoki ら [2] の評価指標に合わせ，**正しさ**と**流暢性**という 2 つの観点から，正解文，BASE による生成文および TOKEN による生成文を良い順に並べ替えるよう指示した．並べ替え時には同順位の判定も許容した．評価時には日経平均に加え，9 つの指標データを折れ線グラフとして提示し，生成文の正しさを判定させた．指標データから判定できない事象を含む事例は，評価データから取り除いた．例えば，日経平均，急落 日銀総裁の発言受けといった生成文は数値データから正しさを判定できない上に，実際に日銀総裁の発言により日経平均が下落したかという因果関係について記者の主観が入っており，正しさを確認できない．

4. 結果

本節では，各性能評価指標におけるモデル性能について

Vol.2020-NL-246 No.30
Vol.2020-SLP-134 No.30
2020/12/3

	BLEU	正解率	再現率	適合率	エラー率
BASE	26.01	90.04	74.78	62.27	7.69
BLEU を用いてハイパーパラメータを調整					
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
正解率を用いてハイパーパラメータを調整					
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
再現率を用いてハイパーパラメータを調整					
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
適合率を用いてハイパーパラメータを調整					
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
エラー率を用いてハイパーパラメータを調整					
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***

表 3 自動評価の結果．* は BASE とのエラー率の差が統計的有意であった提案手法を示す ($p < 0.05$)．太字の数値はハイパーパラメータの調整に用いた指標において，BASE よりも良い性能でったことを示す．なお，エラー率以外については値が高いほど良い性能を示す．

述べ，擬似負例を用いることの効果について議論する．また，正しさの向上のために，より効果的な疑似負例の生成規則についても検証する．

4.1 疑似負例の効果

表 3 に自動評価指標による結果を示す．表は上から順に 6 つに分かれている．もっとも上の区分が擬似負例を用いないベースライン手法 BASE の性能である．表内のこれより下の部分では，本稿で提案する損失関数を用いた手法 (UNLIKE，SENT，TOKEN) を BLEU，正解率，再現率，適合率がそれぞれ最も良くなるよう開発セットでハイパーパラメータを調整し，評価セットで性能評価した結果を示す．太字の数値は，ハイパーパラメータの調整に用いた評価指標において BASE よりも良い性能を示し，疑似負例の効果が示された箇所である．

我々の提案する疑似負例を用いる手法は，BLEU で最適した UNLIKE および TOKEN を除いた，すべての評価指標において BASE よりも良い性能を示した．この結果より，疑似負例を用いた学習により生成文の正しさが向上することが示された．特に，エラー率の改善により，重要語に関する致命的なエラーが疑似負例の使用により軽減されたことがわかる．BLEU の悪化については，UNLIKE (25.54) と TOKEN (25.90) どちらも BASE (26.01) と比較し 0.47 および 0.11 と小さく，統計的有意差はない．よって，提案手法は BLEU を低下させず正しさを向上させることがわかる．

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

表 4 正しさについての人手評価．

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

表 5 流暢さについての人手評価．0-0, 1-0 や 0-1 といったスコアが存在するのはほぼすべての事例が同順位 (すべて流暢) と判定されたことによる，

4.2 損失関数の比較

次に，各提案手法 (UNLIKE，SENT，TOKEN) の違いについて述べる．TOKEN はもっとも良いエラー率 (6.08) を示した．一方，SENT は BLEU (26.56)，適合率 (63.62)，再現率 (78.86) において良い性能を示した．UNLIKE は正解率 (93.02) においてもっとも優位であった．以上の結果から，用いる損失関数によって効果的に性能を向上させることのできる指標が異なることがわかる．すなわち，損失関数は重視する評価指標によって適宜使い分けることが望ましい．SENT は一見多くの指標に良い性能を示し使いやすいように思えるが，例えば再現率で最適化した場合 (78.86) に適合率 (55.39) とエラー率 (8.69) が大きく犠牲になる．このように SENT が不安定な性能を示す中，TOKEN と UNLIKE の性能は安定する利点がある．

単語レベルの疑似負例を用いる手法 (UNLIKE および TOKEN) は用いない手法 (SENT) よりも，正解率，適合率およびエラー率の指標において良い性能を示した．これより，正しさの向上には単語レベルの疑似負例の活用が重要であることがわかる．

4.3 人間による評価

表 4 および表 5 に，人間による順位付け評価の結果を示す．表 4 は正しさに関する評価，表 5 は流暢さに関する評価である．数値は対象手法が他の手法よりもより良いと判定された回数である．結果，CRUCIAL データを用いた正しさの評価において TOKEN が BASE よりも 32 回良いと判定されており，BASE が良いと判定されたのはわずか 5 回であった．よって，人手評価においても擬似負例を用いた提案手法の効果が示された．一方，WHOLE データセットにおいては TOKEN と BASE は 19 対 18 と差がない．これより，疑似負例を用いた提案手法 TOKEN は重要語以外の箇所の正しさを損なうことなく，重要語部分の正しさを向上させる

E_{fluc}	全規則	相反する値動き	類似する値動き
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

表 6 疑似負例の種類によるエラー率の変化.

事がわかる.

流暢さの観点からは，ほとんどの手法が 0 対 0 や 0 対 1 などと判定されている．これは，ほとんどの事例が流暢であり，差が出なかったためである．よって，疑似負例を用いる手法は流暢さを低下させない事がわかる.

4.4 正しさの向上により寄与する疑似負例に関する分析

次に，疑似負例の生成手法の種類の影響について議論する．分析のために，表 1 において定義した重要語を 2 種類に分ける．1 つ目は最終的に値上がる語で，例えば反発や続伸である．2 つ目は最終的に値下がる語で，例えば反落や続落である．このように重要語を 2 種類に分けると，単語レベルの疑似負例の生成規則を「類似する値動きに置換する」規則と「相反する値動きに置換する」規則に分けることができる．例えば，反発から続伸に変換する単語レベルの疑似負例の生成規則は，どちらも最終的に値上がる語であるので「類似する値動きに置換する」規則である．一方，反発から続落に変換する規則は「相反する値動きに置換する」規則となる．表 6 にすべての規則を用いて疑似負例を作成した場合,「類似する値動きに置換する」規則もしくは「相反する値動きに置換する」規則を用いて疑似負例を作成した場合のエラー率を示す．結果,「相反する値動きに置換する」規則を用いた TOKEN 以外では BASE と比較しエラー率の軽減が見られた．さらに,「類似する値動きに置換する」規則の方がすべての提案手法において「相反する値動きに置換する」規則よりも大きくエラー率を軽減し，効果的であることがわかる.

この結果をさらに深堀りするために，ベースライン手法 BASE のエラーを分析した結果を表 7 に示す．この表は，BASE が重要語の出力を間違えた際に，どの重要語を間違えて出力したかを表している．例えば，反発と出力すべき事例に対し続伸と出力するエラーは 87 回観測された．表において，太字の値は「類似する値動き」を表す語を間違えて出力するエラーを示す．このように，太字の値が上位に出現していることから,「類似する値動き」を間違えて出力するエラーがより多く発生している．したがって，多く出現するエラーが解消されたと考えれば，類似する値動きに置換する規則が効果的に働いた表 6 の結果は納得できるものである．本研究では，人的なコストの削減や他のタスクへの応用可能性を考慮し，上位 8 語を抽出しすべての組み合わせから非文を生成する規則を取り除くという単純な

count	Gold	Generated crucial terms
87	**反発**	**続伸**
75	**続落**	**反落**
60	続落	反発
48	反落	続伸
38	**続伸**	**反発**
27	反落	反発
24	続伸	反落
16	反発	続落
15	**反落**	**続落**
13	続落	続伸
12	反発	反落
9	続伸	続落
6	安	高
4	高	安
3	上げ幅	下げ幅

表 7 BASE による生成文のエラーとその回数.

	例 1
Ref	東証 前引け 続伸 ， 急落 から の 買い戻し 続く
Base	日経平均，反発．前引けは *69* 円高の *15,751* 円
Ours	日経平均，続伸．前引けは *69* 円高の *15,751* 円
	例 2
Ref	日経平均，反発で始まる．*70* 円高の *16,100* 円
Base	日経平均，反落で始まる．米株安で利益確定売り先行.
Ours	日経平均，反落で始まる．米株安で利益確定売り先行.

表 8 出力例

負例生成手法を採用した．しかし，もしも開発セットの詳細なデータ分析などに基づき効率的な規則を事前に作ることができれば，さらに性能を向上させることが可能だと思われる．自動で効果的な負例を生成する手法の研究なども今後の方向として興味深い.

同様のエラーは，例えば Arthur ら [3] が *Tunisia* と *Nigeria* を機械翻訳器が間違えるエラーを例示していたりと，類似の問題は data-to-text 以外の生成タスクにおいても見られる．疑似負例を用いた枠組みは他の生成タスクにも幅広く適用できる可能性がある.

4.5 出力例と定性分析

表 8 に代表的な出力例を示す．最初の例において，正解は続伸であるものの BASE は反発を出力している．TOKEN は続伸を出力しエラーが解消されている．このように，疑似負例を用いることによりより良い重要語を選択している例を多く観測した.

2 つ目の例では，BASE も我々の提案手法も正しい重要語 (反落) を出力している．しかしながら，どちらの手法も実際の指標データ (ダウ平均) では米株高の状況であったにもかかわらず，米株安という事実とは異なる言及を出力した．日本の株式市場は米国の株式市場が閉じたあとに開くため，日経平均はダウ平均からの影響を大きく受ける．日

経平均が反落している場合，米株安であることが多い．このケースでは，日経平均が反落しているものの，実際には米株高という頻度が低い状況であった．入力のデータとして米国株式市場の指標データであるダウ平均 (DJI) も含まれてはいるが，低頻度の現象に対しては，正しいテキスト生成が難しい．

市況コメントは日経平均の大まかな値動きを表現する前半部分（日経平均，反発）と，その要因などを補足する後半部分（米株安で利益確定売り先行）に分かれる．本研究で着目した重要語はほとんどの場合，前半部分に記述される．したがって，実際の事例を分析すると後半部分のエラーが散見された．Aoki ら [2] は日経平均だけでなくダウ平均など 9 つの外部指標データを追加入力することで，このようなエラーが軽減することを報告している．このような外部データを注意機構の工夫や制約の導入なども含め，どのようにモデル化すると，さらにエラーを軽減できるかについての分析は，今後の研究として興味深い．また，後半部分の正しさを向上させる疑似負例を効率的に作成する手法についても今後の課題とする．

5. 関連研究

data-to-text 課題は人物の略歴 [8], [9]，スポーツ概況 [4], [6], [13], [14], [19] や市況コメント [1], [2], [10] などを対象に広く研究されている．市況コメントについては，Murakami ら [10] や Aoki ら [2] は本稿と同様に文レベルの課題を扱っているのに対し，Aoki ら [1] のような文書レベルの生成課題も存在する．各々のドメインが固有の特徴を有し，ドメイン特化した手法の工夫が見られるものの，Encoder-decoder を用いたニューラルネットワークによる言語生成モデルを活用し流暢なテキストを生成する手法を採用する点では共通している．入力データを正しく解釈する取り組みとして，言及すべき箇所に着目するエンコーダ [4], [13]，入力データに含まれるエンティティをモデル化する手法 [6], [14] など発展的な手法が多く登場している．

一方，出力文の正しさについての問題は言語生成課題において長らく指摘されている．この問題は data-to-text 課題にとどまらず、機械翻訳タスク [3], [16] や要約 [15] など入力がテキストである課題でも存在する．アラインメント辞書 [3] やコピー機構 [15] は、正しさに関するエラーを軽減するための手法として有望だが，入力がテキストである課題を想定しており，data-to-text 課題には適用できない．

損失関数を工夫することでモデルが疑似負例を考慮するよう拡張する我々の方針は，負例を用いた言語処理研究と関連する．Huang ら [5] はビーム探索で得られた生成文候補と参照文を比較するためのマージン損失を提案した．Noji と Takamura [11] は文法的に崩れた文を負例として活用し，マージン損失を用いて言語モデルを学習することでモデルが構文的な情報を認識する能力が向上することを報告している．言語生成課題においては，Welleck ら [18] が同じ単語が繰り返し出力される問題を軽減するために，直前に出力した語を単語レベルの負例として扱う手法を提案した．本稿では，異なる目的で提案されたこれらの損失関数を data-to-text 課題における生成文の正しさを向上させる目的に用いた．

6. おわりに

本稿では疑似負例を data-to-text モデルの学習時に活用するための枠組みを提案した．自動評価および人手評価により，疑似負例を用いることで生成文の正しさが向上することが確かめられた．疑似負例を学習時に用いるという手法は，多くの言語生成タスクに幅広く適用可能である．今後の研究の方針として，他のタスクでの性能評価や効果的な負例を自動構築する手法の探求などが挙げられる．

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（JPNP15009，JPNP20006）の結果得られたものである．実験には，産総研の AI 橋渡しクラウド (ABCI) を利用した。この原稿は COLING2020 に採択済み論文 [17] を基にした．

参考文献

[1] Aoki, K., Miyazawa, A., Ishigaki, T., Aoki, T., Noji, H., Goshima, K., Kobayashi, I., Takamura, H. and Miyao, Y.: Controlling Contents in Data-to-Document Generation with Human-Designed Topic Labels, *Proceedings of the 12th International Conference on Natural Language Generation (INLG2019)*, pp. 323–332 (2019).

[2] Aoki, T., Miyazawa, A., Ishigaki, T., Goshima, K., Aoki, K., Kobayashi, I., Takamura, H. and Miyao, Y.: Generating Market Comments Referring to External Resources, *Proceedings of the 11th International Conference on Natural Language Generation (INLG2018)*, pp. 135–139 (2018).

[3] Arthur, P., Neubig, G. and Nakamura, S.: Incorporating Discrete Translation Lexicons into Neural Machine Translation, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP2016)*, pp. 1557–1567 (2016).

[4] Gong, H., Feng, X., Qin, B. and Liu, T.: Table-to-Text Generation with Effective Hierarchical Encoder on Three Dimensions (Row, Column and Time), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP2019)*, pp. 3134–3143 (2019).

[5] Huang, J., Li, Y., Ping, W. and Huang, L.: Large Margin Neural Language Model, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP2018)*, pp. 1183–1191 (2018).

[6] Iso, H., Uehara, Y., Ishigaki, T., Noji, H., Aramaki, E., Kobayashi, I., Miyao, Y., Okazaki, N. and Takamura, H.: Learning to Select, Track, and Generate for Data-to-Text, *Proceedings of the 57th Annual Meeting of the*

Association for Computational Linguistics (ACL2019), pp. 1620–1629 (2019).

[7] Kingma, D. P. and Ba, J.: Adam: A method for stochastic optimization, *Proceedings of the International Conference on Learning Representations 2015 (ICLR2015)* (2015).

[8] Lebret, R., Grangier, D. and Auli, M.: Neural Text Generation from Structured Data with Application to the Biography Domain, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP2016)*, pp. 1203–1213 (2016).

[9] Liu, T., Wang, K., Sha, L., Chang, B. and Sui, Z.: Table-to-text generation by structure-aware seq2seq learning, *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI2018)*, pp. 4881–4888 (2018).

[10] Murakami, S., Watanabe, A., Miyazawa, A., Goshima, K., Yanase, T., Takamura, H. and Miyao, Y.: Learning to Generate Market Comments from Stock Prices, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL2017)*, pp. 1374–1384 (2017).

[11] Noji, H. and Takamura, H.: An Analysis of the Utility of Explicit Negative Examples to Improve the Syntactic Abilities of Neural Language Models, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL2020)*, pp. 3375–3385 (2020).

[12] Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J.: BLEU: a method for automatic evaluation of machine translation, *Proceedings of the 40th annual meeting on association for computational linguistics (ACL2002)*, pp. 311–318 (2002).

[13] Puduppully, R., Dong, L. and Lapata, M.: Data-to-text generation with content selection and planning, *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI2019)*, Vol. 33, pp. 6908–6915 (2019).

[14] Puduppully, R., Dong, L. and Mirella, L.: Data-to-text Generation with Entity Modeling, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL2019)*, pp. 2023–2035 (2019).

[15] See, A., Liu, P. J. and Manning, C. D.: Get To The Point: Summarization with Pointer-Generator Networks, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL2017)*, pp. 1073–1083 (2017).

[16] Sennrich, R.: How Grammatical is Character-level Neural Machine Translation? Assessing MT Quality with Contrastive Translation Pairs, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL2017)*, pp. 376–382 (2017).

[17] Uehara, Y., Ishigaki, T., Aoki, K., Noji, H., Keiichi, G., Kobayashi, I., Takamura, H. and Miyao, Y.: Learning with Contrastive Examples for Data-to-text Generation, *Proceedings of the 28th International Conference on Computational Linguistics (COLING2020)*, pp. xx–xx (to appear) (2020).

[18] Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K. and Weston, J.: Neural Text Generation with Unlikelihood Training, *2020 International Conference on Learning Representations (ICLR2020)* (2020).

[19] Wiseman, S., Shieber, S. M. and Rush, A. M.: Challenges in data-to-document generation, *arXiv preprint arXiv:1707.08052* (2017).