Denoising Autoencoder を用いた多様な敵対的サンプルの生成

保田　和彦†　　吉永　直樹††　　豊田　正史††　　喜連川　優††

† 東京大学 大学院情報理工学系研究科
†† 東京大学 生産技術研究所 〒 153–8505 東京都目黒区駒場 4–6–1
E-mail: †{yasuda,ynaga,toyoda,kitsure}@tkl.iis.u-tokyo.ac.jp

あらまし　深層学習に基づくモデルは，入力に小さな変化を加えるだけでモデルの出力が（人間の感覚に反し）大きく変化することがあり，この不安定な挙動がモデルの解釈性と頑健性を確保する上で大きな問題となる．そこで，正しく扱える入力の近傍でモデルの出力が大きく変化する入力（敵対的サンプル）を生成し，この敵対的サンプルを深層学習モデルの解釈や頑健性の改善に用いる手法が近年研究されている．入力が離散かつ可変長系列となる自然言語処理タスクでは，近傍の入力（文）を見つけることが困難であるため，単語レベルで編集する単純な手法が主流である．結果，これらの手法で生成される敵対的サンプルは多様性に乏しく，特にモデルの解釈に用いるには十分ではない．本研究では Denoising Autoencoder を用いて文生成モデルを訓練し，文生成時にモデルの中間状態に摂動を加えることで多様な敵対的サンプルを生成する手法を提案する．この時，中間状態に文表現学習を加えることにより，摂動を加えた時の意味の変化を抑制した．極性分類問題において，敵対的サンプルを生成できた割合，追加学習時の精度で評価および人手によるアノテーションを行い，生成したサンプルに関して主観的・客観的分析を行った．

キーワード　自然言語処理，敵対的サンプル，生成モデル

1 はじめに

深層学習モデルは画像処理や自然言語処理をはじめとする様々な分野のタスクで目覚ましい性能向上を達成しているが，実応用に用いる上ではモデルの入力データに対する敏感性が大きな障害となる [1]．そのため学習したモデルの不安定性の理解が深層学習モデルを実応用で運用する際に重要になっている．

モデルの不安定性の理解のために，モデルが予測を失敗する例を探索することが重要であるが，従来行われているテストセットを用いた評価では十分ではない．テストセットとして必要となる要件は，定量的に評価可能で，タスクで起きうる事象に対して網羅的でありかつ誤った正解が付与されたサンプルがないことである．しかし，広大な入力空間に対して網羅的なテストセットを効率的に作ることは難しい．大規模なデータセットを機械的にもしくはクラウドワーカを使い作成することにより量によって網羅性を解決しようとしたとしても，注釈者が同じ傾向の問題を作成するなどの問題が生じる [2]．

そこで，自然言語処理分野においてモデルの精度低下の原因を理解するために，画像処理分野で発案された敵対的サンプルを用いることが提案されている [3]．敵対的サンプルとはモデルが正しく処理できるサンプルに対して誤答させるように変更を加えたサンプルのことで，自動的に生成する手法が研究されている．しかし，自然言語処理タスクは離散的な入力をとるため，画像処理における敵対的サンプル生成手法と同じ手法を使うことができず，単語の入替えを行う単純な手法が主流である [4]．そのため，限られた変更しか加えることができず，モデルの理解に使用する上であまり役に立たない．

本研究では，より柔軟な変更を加えた敵対的サンプルを生成する手法を提案する．具体的には，Denoising Autoencoder [5] を用いた文生成モデルを Quick Thoughts に基づく文表現学習 [6] とマルチタスク学習し，生成時にモデルの中間状態に摂動を加えることで敵対的サンプルを生成する手法を提案する．

実験では，極性分類タスクのデータセットを用いて，敵対的サンプルを生成し，客観的評価と主観的評価の両方を行うことにより，各手法の違いを明らかにする．客観的評価では実際に対象モデルを騙せた割合，文の自然さ，共通単語数による原文との類似度を測定し，主観的評価では生成した敵対的サンプルの流暢さ，原文とラベルが変化していないかの評価を行う．実験の結果，提案手法は原文と表層的な類似度が高く，流暢度の高い敵対的サンプルの生成が可能であることを確認した．

2 関連研究

深層学習モデルは人間では区別できない摂動によって大きく予測結果が変化するとの報告 [7] があり，この挙動が深層学習の評価を適切に行う際に大きな問題となる．このような人間に区別できない摂動によってモデルの出力を大きく変えてしまうものを敵対的サンプルという．

自然言語処理のタスクは入力が離散な単語系列であるため，数単語変更するだけで意味が大きく意味が変化する可能性があり（e.g. “good” → “bad”），そのタスクにおける入力の近傍の表現を得ることが難しい．この問題に対して，意味が変化しないような編集方法に制限したり [8]，離散な入力を連続値に変換するアプローチが取られている [9]．離散的な入力を連続な値にするにあたって，単語レベルと文レベルの 2 つの手法が

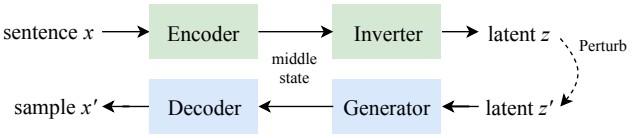

図 1　Zhao らの手法 [10] による敵対的サンプルの生成フロー

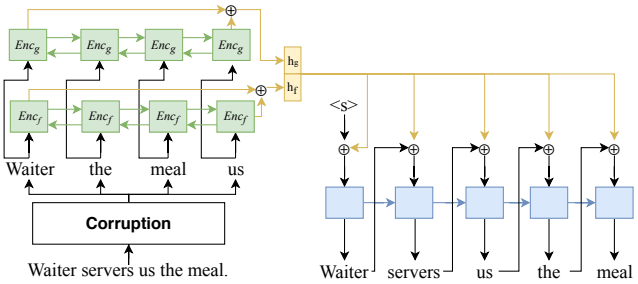

図 2　提案手法の概念図

考えられる．

Alzantot らは単語を実ベクトルに変換する単語分散表現を用いて類義語を自動的に求め，置き換える手法を提案している [4]．類義語に置き換えているため意味的な変化が小さいという利点がある．しかし，この手法では対象のモデルにおける未知語への置き換えにより敵対的サンプルの生成が行われていることが予備実験により分かった．未知語の対処は重要な課題ではあるが，精度低下の原因が明らかであるためモデルの理解という点においてこのような敵対的サンプルが多い単語置き換え手法は有用とは言い難い．また，置き換え可能な単語の数が少ない短文では敵対的サンプルの生成が難しいということも本研究で明らかとなった．

次に，文を実ベクトルに変換してから生成モデルを用いて敵対的サンプルを生成する手法について説明する [10]．Zhao らは ARAE という文生成モデルを用いて，元の文を連続な潜在空間に写像し摂動を加えてから生成を行っている（図 1）．生成モデルを使うことにより，単語置き換えと比べて削除，追加，並び替えなど多様な出力が可能であるため短文への適用も可能となる．しかし，Zhao らは中間表現を連続な潜在空間に写像し連続的な空間から妥当な中間表現に戻す作業を行っているが，潜在空間に意味的な尺度を明示的に組み込んでいないため，復元した時に似た意味の文が復元される保証がない．また，摂動を加えて復元するまでに非線形変換を複数回行うため，長い複雑な文を生成する場合，元の文から意味・表層ともに大きく変わってしまうことがある．

そこで，多様な変更を加えることができる生成モデルを使用しつつ，Zhao らの手法とは違い中間表現に直接摂動を加えることで情報の損失を抑え長文の生成を可能としたい．また，摂動を加えた時に意味の変化を抑制するために，生成モデルに意味を考慮させる必要がある．

3　提 案 手 法

本研究では Denoising Autoencoder に Quick Thoughts による文表現学習を加えた生成モデルを用いて敵対的サンプルを生成する手法を提案する（図 2）．これにより意味を考慮した中間表現に直接摂動を加えることが可能となり，Zhao らの手法の課題であった，長文の対応と文の意味を考慮して生成を行うことができる．

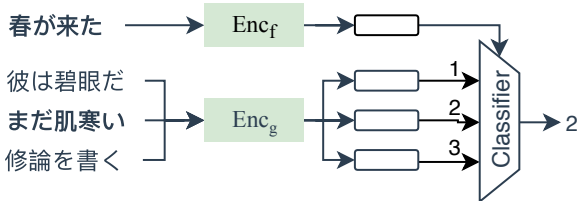

図 3　Quick Thoughts が行うコンテキスト予測タスク．

3.1　Denoising Autoencoder を用いた文生成モデル

中間表現での情報の損失を軽減するために直接摂動を加える必要があるが，通常の Autoencoder で摂動を加えて生成を行うと文法的に破綻した文ができてしまう．これは Autoencoder のデコーダが摂動を加えた時の学習をしておらず，摂動を与えることにより学習された分布から中間表現が外れた時に対応できないためである．そのため，Denoising Autoencoder（DAE）を用いることでノイズに対する頑健性を上げ，摂動を加えても文法的に正しい文が復元できるようにした．DAE は破壊的な変更を加えた入力から元のサンプルへの復元を学習する．入力に破壊的な変更を加えることでサンプルはデータの分布から外れるので，DAE はノイズを取り除き適切な分布上に戻すように学習を行っている．入力が文の場合，不完全な文から正しい文に直す学習を行うため，文法などの統語情報が学習されると考えられる．DAE により，中間表現に摂動を加えても，文法的に正しい文の生成が可能となる．

文に破壊的変更を加える方法は，Lample ら [5] にならい，文中の単語の順序の入替えと単語の削除を行った．まず，単語の順序を効率良く少量の変更を加えるために，次の操作を各文に行う．文を単語に分割し，各単語に文頭から順に連番をつける．その番号に一様分布 $\mathcal{U}(0, k+1)$ から抽出した乱数を足し，各単語を番号が昇順になるように並び変えることで，単語の順序を入れ替える．次に各単語を確率 p で取り除いたものを入力とする．

中間表現に摂動を加えた時に自然な文が生成されるために，中間表現をノイズに対して頑強にする必要がある．DAE の訓練を行う際に，中間表現にノイズを加えて学習を行った．正規分布からサンプリングしたノイズに倍率 r をかけたものを中間表現に加えた．しかし，ノイズを加えた状態では自己符号化の訓練自体が難しいので，一定ステップごとに 1 以下の実数倍することで学習とともに r を小さくした（焼きなまし）．

3.2　Quick Thoughts と DAE のマルチタスク学習

DAE だけでは文の意味について考慮できていない可能性がある．そこで，前節の文生成モデルを Quick Thoughts（QT）[6] に基づく文表現学習とのマルチタスク学習を行う．文表現とは

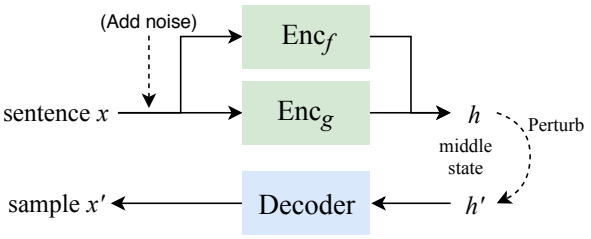

図 4　提案手法による敵対的サンプルの生成フロー

離散的な文を低次元で連続的なベクトルで表したものであり，学習した文表現は分類問題などに活用される．QT は候補文から同一コンテキストの文を推定するタスクの学習を行うことで文表現を獲得する（図 3）．これにより文の表層のみならず意味を考慮したベクトル表現が学習される．この学習を DAE に加える目的は文のベクトル表現に摂動を加えた時, 元の文から大きく意味が変化しないように，DAE の中間表現に意味を考慮させるためである．QT と DAE のマルチタスク学習を行う際は，f と g に同一の文を入力しそれぞれの中間表現を連結し線型変換して次元を落としてデコーダに入力する（図 2）．

提案手法の訓練は英語の生コーパスから抽出した文を用いて行う．各ステップごとに DAE の復元損失と QT による損失をそれぞれ足し合わせたものを損失として勾配の計算を行った．QT の学習の際に使用するコンテキストは入力文の前後一文を使用し，ミニバッチ内で同一レビューの文が重複してコンテキストの候補としてミニバッチ内の文のコンテキストを使用する．

3.3　敵対的サンプルの生成

前述の生成モデルを使って敵対的サンプルを生成する手法について説明する．まず生成モデルを使い候補となる類似文の生成を行う（図 4）．その後，候補文から敵対的な結果となる文を選択する．候補文の生成には以下の 2 つの手法を試す．

摂動の付加 (Perturbing) 中間表現に正規分布 $\mathcal{N}(0, \sigma)$ からサンプリングした摂動を加えて生成を行う．文表現の学習をしているので，多少の変更を加えても意味的な変化が少ないと考えられる．摂動は Zhao らの手法を参考に，まず確率分布から N 個の摂動ベクトルをサンプリングし，各摂動を中間表現に加えて生成し，対象のモデルを騙せるか（敵対的か）どうかを分類する．敵対的でなかった場合，その摂動を破棄し新しい摂動をサンプリングする．敵対的だった場合は，最小の摂動を求めるために 2 分木探索を行う．つまり，その摂動を半分にしたものを次の摂動として採用し，もし新しい摂動を加えて生成されたサンプルが敵対的でなかった場合は，その摂動と元の摂動の中間を次の摂動とする．これを 1 回の試行として，B 回試行を繰り返す．

破壊的変更からの復元 (Denoising) DAE の学習によって入力中の足りない情報を補完して生成を行うことができるが，その際元の文が完全に復元できるわけではない．例えば，“She serves us the meal.” という文から主語の “She” を取り除いて復元した場合，“Waiter serves us the meal.” のように動詞 “serves” にあう主語が学習によって生成される．このように破壊的変更によって違う似た文が生成されるので，これを候補文とした．破壊的変更は学習時と同じように，単語の順番の入替えと欠落によって行った．

また，Perturbing と Denoising それぞれに対して，デコーダで復元時に多様な文を生成するために生成時に確率的に単語選択する手法（**Sampling**）を試した．通常デコーダの出力するトークンは出力層の最も高い値から決定されるが，代わりに出力層の出力にソフトマックスをかけたものを確率分布として考え，そこからサンプリングを行った．これによって同じ中間表現であっても途中で分岐し多様な文の生成を行うことができる．つまり，Perturbing，Perturbing+Sampling，Denoising および Denoising+Sampling の 4 つの方法で敵対的サンプルの候補文生成を行った．また，生成された文に未知語トークンが含まれていた場合は候補文として採用しなかった．

候補文の生成後，対象のモデルに入力し敵対的な出力（異なる分類結果など）となった文を敵対的サンプルとして採用する．複数ある敵対的サンプルから最も流暢なものを選択するために，N-gram 言語モデルを用いて敵対的サンプルの perplexity と元の文の perplexity の比を計算して最も変化が小さいものを採用する．入力が複数の文からなる文章の場合，それぞれの文について敵対的サンプル候補を生成し，最も対象モデルの出力の尤度を下げた文を採用する．一文の変更で敵対的な出力にならなかった場合は，変更した文を固定して他の文の変更を行うことを繰り返した．

4　評 価 実 験

提案手法と既存手法の比較のために，生成した敵対的サンプルに関して客観的および主観的評価を行った．また，既存手法の課題である長文生成が改善されたのか評価した．

4.1　評 価 尺 度

敵対的サンプルの要件として以下のものが上げられる．

条件 1　　モデルの出力を大きく変化させること

条件 2　　入力として自然であること

条件 3　　人から見た変化が小さいこと

敵対的サンプル生成モデルは，これらの 3 条件について評価を行わなければならないが，人間により評価を行う場合コストがかかることから大規模な評価が難しい．そこで本研究では，人による小規模な主観評価に加え，自動評価尺度による大規模な客観的評価を行い，コストのかかる主観評価の代替が可能かどうか比較を行った．それぞれの評価項目と主観的および客観的評価の方法を表 1 にまとめる．条件 3 はラベルの変化と原文との類似度の 2 つの方法で評価した．

4.1.1　客観的評価

攻撃成功率（条件 1）　攻撃成功率は対象のモデル（分類器）

表 1　客観的評価と主観的評価の対応

評価項目	客観的評価	主観的評価
条件 1: 敵対性	攻撃成功率	—
条件 2: 言語としての自然さ	Perplexity	Fluency
条件 3: ラベル不変性	敵対的学習	Label
条件 3: 原文との類似度	共通単語数	Similarity

が正解したサンプルに対して敵対的サンプルを生成できた（攻撃が成功した）割合とした．攻撃成功率が高いほど，その生成手法は多くの入力に対して敵対的サンプルの生成が行えたということであり，生成手法の汎用性を確かめることができる．

Perplexity（条件 2）　Perplexity を用いて，生成した敵対的サンプルの言語としての自然さを評価した．Perplexity は文中の各単語の尤度の逆数を幾何平均した値であり，不自然な文を入力として与えると perplexity が高くなる．Perplexity によって文の自然さを数値化できるので，これを使って生成された敵対的サンプルの自然さの評価を行った．

敵対的学習（条件 3）　敵対的サンプルのラベルが元のサンプルと変わっていないか確認するために，訓練データに敵対的サンプルを追加して学習を行った．敵対的サンプルの真のラベルが変化していた場合，敵対的サンプルを加えて学習を行うと，学習データに間違ったデータが混在するため，加えなかった時と比べてモデルの精度が低下するはずである．

共通単語数（条件 3）　単語の入替えしか行わない手法と比べて生成モデルを使った手法は生成される敵対的サンプルの自由度が高いため，元の文と似た文となっている保証がない．そこで，共通する単語の数を数え，生成した敵対サンプルと元の文の類似度を測定する．文の意味に関わる内容語（名詞・形容詞・副詞・動詞・数詞・代名詞）のみ測定の対象とした．品詞タグ付けには StanfordCoreNLP [11] を使用した．この時，生成モデルの出力はすべて小文字になっているので，TrueCaseAnnotator を使い補正した．[1] 次に，単語の時制等による変化を吸収するために見出し語化を行った．元の文と敵対的サンプルそれぞれの内容語の見出しの集合に対して，共通する単語を数え，適合率と再現率を次式で計算した．

$$\text{適合率} = \frac{\{\text{共通単語}\}}{(\{\text{共通単語}\} + \{\text{生成文にのみ出現した単語}\})}$$

$$\text{再現率} = \frac{\{\text{共通単語}\}}{(\{\text{共通単語}\} + \{\text{元の文にのみ出現した単語}\})}$$

1 : https://stanfordnlp.github.io/CoreNLP/caseless.html

表 2　モデルのハイパーパラメータ

エンコーダ・デコーダ	GRU の層数	1
	GRU の隠れ層	500 次元
	単語埋め込み表現	500 次元
	語彙数	50,000
雑音	k	3
	p	0.1
	r	0.1
	焼きなまし比	0.999
	焼きなまし間隔	100 ステップ

4.1.2　主観的評価

条件 2，条件 3 の客観的評価が人から見た時に妥当か検証を行うために，人手による主観的評価を行った．本研究では，次の 3 つについて人による注釈を行った．各指標を 3 人で注釈を行い，過半数以上の合意が取れたものをそのサンプルに対するアノテーションとした．

Fluency（条件 2）　文の流暢さを評価した．一般的に 5 段階で評価するが，今回は文法的に破綻していないかを確かめることが目的であるため,「理解不能（0）」，「流暢ではない（1）」，「流暢（2）」の 3 段階に分類した．アノテータには，文法的に破綻していて文の意味が理解できないものを「0」，文意は分かるものの不自然な所があるものを「1」，人が書いたと思えるものは「2」にとするように説明した．この時，意見が割れたもの（3 人で評価し「理解不能」,「流暢ではない」,「流暢」にそれぞれ 1 票ずつ入った時）は合意が取れていないものとして扱う．サンプルのアノテーションを行った後，各手法の流暢度を次式で計算した．

$$\text{流暢度} = \frac{\text{「流暢」} + \text{「流暢ではない」}}{\text{全サンプル数} - \text{対立}}$$

Label（条件 3）　元の文と敵対的サンプルのラベルが変化していないか確かめるために，ラベルの注釈を行った．先入観を持たせないように，敵対的サンプルのみ与え元の文は与えずに注釈を行った．実験で使用した極性分類タスクは 2 値分類だが，生成文は極性がなくなる可能性があるため，「中立」を追加した．この注釈作業は Fluency が 1（「流暢でない」）以上になったものを対象に行った．

Similarity（条件 3）　元の文との類似度を測るために，既存手法並びに提案手法によって生成された敵対的サンプルを与え，元の文に意味的により近い敵対的サンプルを選択させた．元の文とラベルが変化している場合は，違いが大きいとした．

4.2　実 験 設 定

4.2.1　提案手法の訓練

提案手法の実装と訓練方法について説明する．エンコーダ，デコーダともに RNN の一種である GRU [12] を使用した．GRU は LSTM [13] と比べてパラメータが 25 %少ない上 NMT にお

表 3　極性分類タスクのデータセット

	SemEval	Yelp
訓練データ数	0	5.3M
評価データ数	1554	10k
一サンプルあたりの平均文数	1.0	8.29
一文あたりの平均単語数	13.8	22.50

いて同等の性能を示すことが分かっており，NMT と同じエンコーダ・デコーダモデルである提案モデルでも有用だと考え採用した．また，長文に置いて入力情報を忘れてしまう問題を緩和するために双方向 RNN と同様に入力文を順方向と逆方向それぞれ違う GRU を用いてエンコードして最終的な出力を連結した．Zhao らと同じ実装に倣い，エンコーダの出力を連結し，線型変換をかけてデコーダの単語埋め込み表現と同じ次元数にしてから，埋め込み表現と連結してデコーダに入力を行った．エンコーダの出力を隠れ状態の初期値として入力した場合と比べて学習が早くなる効果があった．確率的勾配降下法の一種である，モーメンタム付き SGD，Adam [14], AMSGrad [15] を試し，AMSGrad が最も収束が早くかつ開発データにおける DAE の BLEU スコアが高かったので採用した．学習率は 0.0005，他はデフォルトの設定で訓練を行った．実装はすべて Pytorch (version: 0.4.1) で行った．学習時のハイパーパラメータを表 2 に示す．学習時間と GPU のメモリ上限の問題から一文あたりの最大単語数を 50，バッチサイズを 128 とした．BLEU スコアが開発データで最も高くなったモデルを敵対的サンプルの生成に使用した．

4.2.2　極性分類タスク

本研究では，攻撃対象に極性分類タスクを解く分類器を使用した．分類器の評価には SemEval2016 の Task5 のレビューデータ [16] のデータセットを使用した．SemEval2016 の Task5 にはいくつかのトピックに分かれたデータが存在し各サンプルは negative，positive，neutral の 3 値分類されているが，今回はレストランレビューデータから negative もしくは positivie なものだけ抜き出し使用した．SemEval は訓練データが存在しないため，モデルの訓練には同じレストランレビューの Yelp レビューデータ [17] を使用した．Yelp レビューデータは極性情報を持たず，5 段階評価の数値のみ存在するので，1〜2 を negative，4〜5 を positive として 3 は使用しなかった．各データセットの文章数, 一文章あたりの文数，一文あたりの単語数を表 3 に示す．前処理としてテキストを小文字化し，単語分割した．訓練データから語彙を構築し，出現頻度上位 5 万単語を使用した．

分類器には TextCNN [18] を使用した．TextCNN のフィルタの幅を 3,4,5 とし，各フィルタのチャネル数は 128，単語埋め込み表現の次元 d は 300 とし，fasttext [19] の公開されている埋め込み表現を初期値に設定した．fasttext に存在しなかった語彙は一様分布 $\mathcal{U}(-1/\sqrt{d}, 1/\sqrt{d})$ で初期化した．最適化手法は AMSGrad [15] で， 学習率は 0.001, β_1 および β_2 はそれぞれ 0.9, 0.999 とした．学習したモデルは Yelp のテストデータで F 値 0.97，SemEval で F 値 0.89 となった．

4.2.3　敵対的サンプルの生成

テストデータのうち訓練したモデルが正しく分類できたサンプルからランダムに 500 件抽出し，それぞれの手法で敵対的サンプルの生成を行った．極性分類タスクでは，モデルの出力が真逆の極性（元がネガティブならポジティブ）に分類された時攻撃成功とした．提案手法および，ARAE で敵対的サンプルの生成を行う時，各試行で加える摂動の数 $N = 200$，最大試行回数 $B = 15$ の制限をかけた．摂動を加える方法では摂動の大きさ σ として $0.2, 0.5$ の 2 通りを試した．Alzantot らの手法は彼らの論文に従い，単語入替えの最大試行回数 $G = 20$，各試行における個体数 $S = 60$，入替え単語候補数 $N = 8$ として生成を行った．

4.2.4　N-gram 言語モデル

候補文の順位付と実験で perplexity を測定するために N-gram 言語モデルを使用した．yelp のレビューデータおよび wikipedia データから抽出した文を用いて学習した．実装は KenLM [20] を使用し，$N = 5$ で学習した．SemEval の文の Perplexity を測定した結果を 60.15 となった．同じレストランレビューのデータであるため，低い Perplexity となっていることが分かる．

4.2.5　敵対的学習

訓練データからランダムに抽出した 2000 件に対して各手法で敵対的サンプルの生成し，生成した敵対的サンプルを訓練データに加えて学習を行い，テストセットでモデルの精度（F 値）を測った．訓練データ全体に敵対的サンプル 2000 件加えて訓練を行っても影響が少ないため，敵対的サンプルの元となった 2000 件と敵対的サンプルを加えた 4000 件で訓練する場合と敵対的サンプルのみの 2000 件で訓練する場合の 2 通りで実験を行った．その他のパラメータは元の分類器と同じとした．

4.2.6　主観的評価設定

提案手法（perturbing, perturbing+sampling, denoising, denoising+sampling）と ARAE，単語置き換えの全ての手法で敵対的サンプルの生成に成功したサンプルからランダムに 50 件選択し元の文と合わせて計 350 件のアノテーションを行った．1 サンプルあたり 3 人アノテーションを行った．Fluency と Label のアノテーションを行う時は，各サンプルを無作為な順に並び替えてアノテートを行った．アノテーションは Fluency, Label，Similarity の順で行った．

4.3 実験結果

4.3.1　長文の生成

提案手法で ARAE の課題であった長文生成が改善されたかを確認するために，Autoencoder としての精度を測定した．図 5 から全ての文長において提案手法の方が高い BLEU となった．ARAE の精度が文長とともに大きく低下する理由は，潜在空間へ変換し元の中間表現への復元を行っているため，情報の保存が難しく復元時に大きく変わってしまうからだと考えられる．提案手法では，中間表現をそのままデコーダへ入力を行うので高い復元性能を誇ることができた．

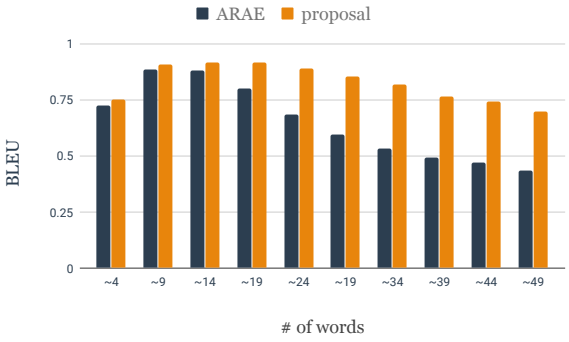

図 5　入力文長と提案手法と ARAE の Autoencoder としての精度．

表 4　各敵対的サンプル生成モデルの攻撃成功率と N-gram 言語モデルによる Perplexity．

Model	攻撃成功率	Perplexity
Perturbing ($\sigma = 0.2$)	0.348	64.56
Perturbing ($\sigma = 0.5$)	0.934	76.96
Perturbing ($\sigma = 0.2$)+Sampling	0.765	74.29
Perturbing ($\sigma = 0.5$)+Sampling	0.982	62.14
Denoising	0.835	97.64
Denoising+Sampling	0.991	177.33
ARAE [10]	0.998	151.34
単語置き換え [4]	0.683	814.13

4.3.2　客観的評価

生成した 500 件の敵対的サンプルに対して 4 つの自動評価可能な尺度で評価を行った．

攻撃成功率（条件 1）

各手法で敵対的サンプルの生成を行った時の攻撃成功率を表 4 に示す．Perturbing では σ を大きくすることで，成功率が高くなった．また，Denoising でも高い割合で攻撃が成功している．どちらの方法でもデコード時に Sampling を行うことで，攻撃成功率が高くなった．同様に生成モデルベースである ARAE も高いスコアとなった．一方で単語置き換えによる手法は他の手法と比べ，低いスコアとなった．これは単文では置き換え可能な単語が少なく，候補となる文の生成が難しいためだと考えられる．

敵対的サンプルの Perplexity（条件 2）

次に，N-gram 言語モデルで敵対的サンプルの perplexity を測った（表 4）．Perturbing が最も perplexity が低くなり，摂動の大きさによる perplexity の変化は攻撃成功率の上昇幅に比べると大きくなかった．このため，これ以降の実験では攻撃成功率の高い $\sigma = 0.5$ でのみ行った．Sampling を行った時，σ が大きい場合は perplexity が下がるが，小さい場合は逆に高くなった．デコード時に出力単語を確率的に選択しているため，広い範囲の perplexity の候補文が作成されたため，もともとの攻撃成功率が高い $\sigma = 0.5$ では perplexity が低いものが選ばれたのだと考えられる．“perturbing + sampling” と “denoising + sampling” で生成された文を見比べてみると，後者の方が同じ入力文でも非文になっているものが多く存在した．これは “denoising + sampling” では破壊的変更が加えられた不完全な文が入力されるため，復元時に候補となりうる単語が多く存在し，確率分布に従って選択をした時のばらつきが大きくなったと予想される．ARAE も生成ベースだが，Perplexity が高くなった．これは，ARAE では生成した敵対的サンプルから最も良いものを選択する時に一番摂動が小さいものを残すからである．

単語置き換えによる手法が一番高い Perplexity となった．類義語を外部リソース（Glove）を用いて選択しているため，データセットにない単語が選択され，言語モデルに未知語や低頻出語が入力されてしまい perplexity 高くなった．表 5 に生成された敵対的サンプル中で，対象のモデルでは未知語となった単語の割合を示す．単語置き換えによって生成された文は元の文と比べて多くの未知語を含んでいることが分かった．単語置き換え手法では対象のモデルで未知語になるように単語の入替えを行うことで，敵対的サンプルの生成を行っていると考えられる．

表 5　各サンプル中の未知語の割合

	単語数	未知語の割合（%）
Perturbing ($\sigma = 0.5$)	6,754	0.22
ARAE	5,831	0.27
単語置き換え	3,271	7.06
SemEval データセット	19,287	2.86

表 6　敵対的学習によるモデル精度の変化．“No Adversarial” は敵対的サンプルを加えなかった場合の精度で，この精度との差を Δ とした．

Model	With original		Only adversarial	
	F-score	Δ	F-score	Δ
No Adversarial	0.831		0.831	
Perturbing	0.811	-0.021	0.703	-0.129
Perturbing+Sampling	0.798	-0.034	0.705	-0.127
Denoising	0.795	-0.037	0.690	-0.142
Denoising+Sampling	0.797	-0.035	0.685	-0.147
ARAE	0.801	-0.031	0.685	-0.147
単語置き換え	0.822	-0.009	0.746	-0.085

敵対的学習（条件 3）

すべての手法で，敵対的サンプルを加えた場合精度が悪化した（表 6）．元のサンプルを加えた場合は大きく悪化しなかったが，元のサンプルを加えなかった場合は単語置き換え以外は大きくスコアを落とした．特に denoising により生成された敵対的サンプルはほぼ学習ができていないため，生成されたサンプルの真のラベルが変化していると考えられる．一方で単語置き換えでは精度低下が少なかった．

共通単語数（条件 3）

元の文と敵対的サンプルの類似度を測るために，共通する内容語の数を測定した結果を表 7 に示す．提案手法，特に perturbing は ARAE と比べて高いスコアとなった．これは ARAE

表 7　敵対サンプルと元の文間で共通する内容語の割合.

Model	再現率	適合率
Perturbing	0.765	0.660
Perturbing+Sampling	0.641	0.511
Denoising	0.546	0.753
Denoising+Sampling	0.527	0.566
ARAE	0.520	0.503
単語置き換え	0.688	0.671

表 8　各敵対的サンプルの Fluency アノテーション.

	理解不能	流暢でない	流暢	対立	流暢度
Perturbing	11	19	16	4	0.76
Perturbing+Sampling	14	11	23	2	0.71
Denoising	15	14	13	8	0.64
Denoising+Sampling	22	11	11	6	0.50
ARAE	20	18	10	2	0.58
単語置き換え	13	16	8	7	0.65
元サンプル	2	8	35	5	0.96

が摂動を加えるときに，エンコーダとデコーダの間で 2 回多層パーセプトロンによる変換が行うのに比べ，提案手法では余分な変換をせずに生成しているためである．

Perturbing は再現率が適合率より高くなっており，逆に denoising では適合率が再現率より高くなった．このことから，perturbing では元の文に単語を追加し，denoising では元の文から単語を削除することによって敵対的サンプルを生成していると考えられる．1 単語消す操作と 1 単語を加える操作を行っていることに等しいため，単語置き換えは適合率と再現率がほぼ同じ値になっており，perturbing とほぼ同等のスコアとなった．

4.3.3　主観的評価結果

次に，客観的評価が人の評価と整合性があるのか確認するために 3 つの主観的評価を行った．結果を示す前に，アノテータ間の合意度を計算しておく．合意度には Fleiss の κ [21] を使用した．350 件を 2 つのグループ（3 名ずつ）で注釈したため，それぞれのグループ内で Fluency アノテーション結果から κ を算出したところ，0.589, 0.659 となった．これは十分高い合意関係が得られていると考えられる [22]．κ を計算する際，注釈者が非英語圏であるため「流暢」な英文の判定に個人差が生じると考え「流暢ではない」，「流暢」を区別せずに計算した．

Fluency（条件 2）

サンプルの Fluency についてアノテータ間で合意がとれた数を集計した結果を表 8 に示す．変更を加えていない元サンプルの評価を見ると，約 8 割以上の文が「流暢」もしくは「流暢でない」に正しく分類されている．合意が取れなかったサンプルは複数の文がつながってしまっているものや，文の途中で切り出したようなものであった．

生成された敵対的サンプルの中では perturbing が最も流暢度が高く，次に denoising を加えたものが高かった．どちらも sampling によるデコードを行うと，不自然な文が増えてしまっているが，denoising+sampling の方が顕著に増えている．既存手法である ARAE に関しては,「理解不能」に分類されているものが多く，また多くが「流暢でない」と判定された．どちらも Perplexity での評価と一致する結果である．

表 9　各敵対的サンプルのラベルアノテーション.

	# of samples	same	oppsite	neutral
Perturbing	39	0.49	0.28	0.18
Perturbing+Sampling	36	0.33	0.44	0.17
Denoising	35	0.20	0.40	0.31
Denoising+Sampling	28	0.11	0.68	0.18
ARAE	30	0.43	0.20	0.33
単語置き換え	31	0.84	0.07	0.03
元サンプル	48	0.92	0.04	0.04

Label（条件 3）

次に,「流暢でない」もしくは「流暢」で合意が取れた文について，極性ラベルをアノテートし，元のラベルと変化しているかを集計した（表 9）．変更を加えていない元サンプルでは 92 % でアノテーションと元ラベルが一致しアノテータは正しく分類できた．Perturbing が最もラベルの変化が小さかった．しかし，すべての手法において生成された敵対的サンプルが同一のラベルと判定された割合は 5 割を切っていた．特に denoising は逆のラベルに変化しているものが多かった．また，sampling を行うことで，逆の極性になる確率が高くなっている．Perturbing はエンコード時に全ての入力情報を持っているため，denoising と比べラベルの変化が抑制されているが，ARAE より逆のラベルになった割合が高かった．しかし，これは perturbing が ARAE より劣っているというわけではなく，表 8 の結果から分かるように提案手法の方が流暢な文が生成でき，50 サンプル中同一ラベルのサンプルを生成できた数は ARAE より多く，単語置き換えと同程度であった．

Similarity（条件 3）

最後に，perturbing と ARAE による敵対的サンプルのどちらがより元の文に近いか測定した．前述の Label アノテーションの結果，両手法とも真のラベルと同一もしくは中立となったサンプルのみ評価に使用した．3 名によりアノテーションを行い，過半数を超えた方をより近いとした結果，73 %のサンプルで perturbing による敵対的サンプルがより元の文に近いと判断された．この時 $\kappa = 0.70$ と高い値になっており，アノテータ間で十分に合意が取れていると考えられる．ARAE で生成された敵対的サンプルと比べてより原文に近い文が生成できていることが分かった．これは，表 7 の結果からも分かるように共通単語が多いためだと考えられる．

4.4　ま と め

本研究で評価項目とした敵対性，言語としての自然さ，ラベル不変性，原文との類似度について客観的評価および主観的評価を行った結果を各手法で比較し表 10 にまとめた．Perplexity で測定した言語としての自然さは，生成モデルベースの手法で

表 10　各評価項目における各手法の比較

モデル	敵対性	自然さ		ラベル不変性		類似度	
		客観	主観	客観	主観	客観	主観
Perturbing	○	○	○	×	△	○	○
ARAE	○	△	×	×	×	×	△
単語置き換え	△	×	△	△	○	○	—

ある提案手応と ARAE においては主観的評価と似た傾向を示したが，単語入替えを行う手法ではまったく一致しなかった．これは未知語により，言語モデルが正しく評価できなかったためであると考えられる．そのため，未知語に対処するためにより大規模なコーパスを使うか，外部リソースの単語埋め込み表現を扱える言語モデルを使うことで解決が可能だと考えている．敵対的学習の実験から生成ベースの手法は単語置き換えの手法と比べて精度が下がることが分かった．主観的評価により，提案手法で生成モデルではサンプルのラベルの変化率を見ると半分以上のサンプルが元と違うラベルになっており，一方で単語置き換えでは 8 割が元のラベルのままであった．敵対的学習の実験を行うことでラベル不変性の評価が可能であると考えられる．しかし，同じラベルのサンプルが約 3 割の perturbing+sampling と，約 1 割の denoising+sampling で敵対的学習の性能差がなかったことから，変化したラベルが一定以上を超えると差は明確でなくなると考えられる．ラベルノイズの量とモデルの精度の関係について調査する必要がある．また，これらの比較を正確に行うにあたって統計的検定を行う必要があるが，これは今後の課題とする．

5 おわりに

本研究では，自然言語分野における敵対的サンプルの生成に取り組んだ．単語の入替えによる手法ではなく，生成モデルを使用することにより，多様な入力テキストに対して敵対的サンプルの生成が可能になると考え，また従来の生成ベースではできなかった長文生成と文の意味を考慮するために，Denoising Autoencoder に文表現学習を加えたモデルによるた敵対的サンプル生成手法を提案した．生成した敵対的サンプルの評価を行うにあたり，広く使われている方法がなかったため，評価項目の整理を行い，客観的・主観的の両方で評価を行った．提案手法は原文と表層的な類似度が高く，流暢度の高い敵対的サンプルの生成が可能であることを確認したが，主観的評価によってラベルが元の文から変化している例が多く存在した．

今後の課題は，客観的評価と主観的評価の関係性が解析し，敵対的サンプルの評価を定量的に行えるようにすることである．

謝　辞

本研究の一部は，情報通信研究機構の委託研究の成果である．

文　献

[1] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust Physical-World Attacks on Deep Learning Visual Classification. In *CVPR*, pp. 1625–1634, 2018.

[2] M. Tsuchiya. Performance Impact Caused by Hidden Bias of Training Data for Recognizing Textual Entailment. In *LREC*, 2018.

[3] Y. Belinkov and J. Glass. Analysis methods in neural language processing: A survey. *arXiv:1812.08951*, 2018.

[4] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang. Generating natural language adversarial examples. In *EMNLP*, pp. 2890–2896. Association for Computational Linguistics, 2018.

[5] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato. Unsupervised Machine Translation Using Monolingual Corpora Only. In *ICLR*, 2018.

[6] L. Logeswaran and H. Lee. An efficient framework for learning sentence representations. In *ICLR*, 2018.

[7] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[8] M. T. Ribeiro, S. Singh, and C. Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *ACL*, pp. 856–865. Association for Computational Linguistics, 2018.

[9] M. Sato, J. Suzuki, H. Shindo, and Y. Matsumoto. Interpretable adversarial perturbation in input embedding space for text. In *IJCAI*, pp. 4323–4330, 2018.

[10] Z. Zhao, D. Dua, and S. Singh. Generating natural adversarial examples. In *ICLR*, 2018.

[11] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *ACL*, pp. 55–60, 2014.

[12] K. Cho, B. v. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, pp. 1724–1734. Association for Computational Linguistics, 2014.

[13] F. A. Gers, J. A. Schmidhuber, and F. A. Cummins. Learning to forget: Continual prediction with lstm. *Neural Comput.*, Vol. 12, No. 10, pp. 2451–2471, October 2000.

[14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, 2014.

[15] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. In *ICLR*, 2018.

[16] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra, and G. Eryiğit. Semeval-2016 task 5 : aspect based sentiment analysis. In *SemEval*, pp. 19–30. Association for Computational Linguistics, 2016.

[17] Y. Dataset, 2014. https://www.yelp.com/dataset/challenge.

[18] Y. Kim. Convolutional neural networks for sentence classification. In *EMNLP*, pp. 1746–1751. Association for Computational Linguistics, 2014.

[19] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *TACL*, pp. 135–146, 2017.

[20] K. Heafield. KenLM: faster and smaller language model queries. In *EMNLP*, pp. 187–197, 2011.

[21] J. Fleiss and M. Davies. Jackknifing functions of multinomial frequencies, with an application to a measure of concordance. *Am J Epidemiol*, 1982.

[22] J. Landis and G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, Vol. 33, No. 1, pp. 159–174, 1977.