修　士　学　位　論　文

画像キャプション生成における
複数形表現の統一とその効果

平成 30 年度

茨城大学大学院理工学研究科

情報工学専攻

西　友佑

平成 30 年度茨城大学大学院理工学研究科情報工学専攻　修士学位論文

画像キャプション生成における複数形表現の統一とその効果

著者： 西　友佑(17NM717L)
指導教員：新納 浩幸 教授

論文要旨

本論文では，画像キャプション生成における生成文の品質向上を目的とし，その実現のために，訓練データ中の複数形表現を統一することを試みる．具体的には， two dogs や three cars と言った基数を用いた複数形表現に対して，基数を用いないより単純な表現への統一を行う．これにより，生成文における基数を使用していた部分の誤りが訂正され，品質の向上が期待できる．

入力された画像のキャプション（説明文）を生成する「画像キャプション生成」の研究は従来より活発に行われてきた．この画像キャプション生成に関する問題の一つとして，生成したキャプションの定量的な評価が難しい点が挙げられる．これは，正しいキャプションが一つに定まらないために起こる問題である．例えば，表現の言い換えや説明の詳しさによって，正しい説明文は無数に存在しうる．現在まで行われてきた多くの研究では，複数の自動評価尺度を利用することで，なるべく定量的に評価ができるように工夫されてきたが，自動評価尺度による評価では限界があり，人間の評価を完全になぞることは出来ない．そのため，その文章が正しいかどうかは突き詰めると主観的にしか判断できない．本論文では，人間が理解できる文章の詳しさの程度という点に着目し，生成されるキャプションをより単純にすることによって誤りを減らし，生成文の品質を向上させる．具体的な手法としては，基数などを用いた複数形表現をより単純な複数形表現へ統一する．これによって，生成されるキャプションの誤りを減らすことを目指している．

先に述べたような誤りの解消は，生成された後のキャプションを直接修正することでも解決可能に見える．しかし，深層学習を用いた画像キャプション生成では，深層学習の内部を詳しく検証することが難しいためにどの部分で誤りが発生しているかを特定することが困難である．また，時系列的な影響を受ける文章生成において，生成された後の一部分を変更しただけでは，文中の変更した箇所以降の部分において更なる誤りが発生する可能性がある．そのため，本研究では学習に使用する訓練データにおける複数形表現を予め書き換えておき，書き換えたデータを用いて学習を行う．

実験では，英語キャプション訓練データにMSCOCOのデータセット，日本語キャプション訓練データにSTAIR Captionを用いた．テストデータにはネットなどから収集した100枚の画像データを使用した．そして，書き換えられた訓練データで学習したシステムによるキャプションと，書き換え以前の訓練データで学習したシステムによるキャプションとを比較し，主観的に評価を行った．結果，キャプションの大きく変化した画像数は100枚中39枚となり，そのうち改悪された画像数が14枚であるのに対し，25枚の画像においてキャプションの改善が認められた．また，BLEUやMETEORなどの機械翻訳やキャプション生成における自動評価尺度を利用し，客観的な評価も同時に行った．結果として，英語キャプションでは精度の向上が認められた．

Master' s Thesis in Scholastic 2018, Major in Computer and Information Sciences,
Granduate School of Science and Engineering, Ibaraki University

Unification of Plural Form Expressions and Influence in Image Caption Generation.

Author : Yu Nishi (17NM717L)
Adviser : Prof. Hiroyuki Shinnou

ABSTRACT

In this study, we propose to unify plural form expressions in training data to improve the quality of sentences generated by an image caption generator. In particular, we convert the plural form expressions, such as "two dogs" and "three cars", using a cardinal number to the simpler expression not using such radix. Errors based on plural form expressions can be reduced and therefore, the quality of the generated sentence can be improved using proposed method.

Recently, image caption generation has received considerable research interest [5]. It is research of automatically generating captions (descriptions) of input images using computer. One of the problems of image caption generation is that it is difficult to quantitatively evaluate generated captions. This problem is occurred by the correct caption cannot define to one caption. For example, there are countless correct captions depending on the paraphrase of the expression and the details of the explanation. Many studies that have been done so far have been devised so that quantitative evaluation can be made as much as possible by using multiple automatic evaluation method, but in the evaluation by the automatic evaluation method can not trace completely evaluation by human . Therefore, it can be only judged subjectively whether the sentence is really correct or not. In this paper, we focus on it the degree of details of sentences that humans can understand. Improve the quality of the generated captions because the error is reduced by simplifying the generated captions to the extent that humans can understand. As a concrete method, the plural form expression using the radix is unified to a simpler plural form expression. This method aims to reduce the error of caption generated.

It must note that the above mentioned translation from complex expressions to the simple expressions cannot be achieved by rewriting the generated caption. This is because errors in the generated caption are related to each other. The partial rewriting may cause other errors. Therefore, in this study, we convert complex plural expressions in training data to simple plural expressions; we train a model using the rewritten data.

In the experiments, MSCOCO dataset for generating English captions and SATIR Caption dataset for generating Japanese captions was used as the training data. The test data comprised of 100 images that were not included in the training data. Then, the caption generated by a caption generator using unified data was compared with the caption generated by a caption generator using "not" unified data and subjectively evaluated. As a result, the number of images in which the caption changed drastically became 39 out of 100, out of which the number of images with deterioration caption generated was 14, whereas the plural form expressions was improved for 25 images. In addition, we conducted objective evaluation by an automatic evaluation method used in machine translation and caption generation such as BLEU. As a result, improvement in accuracy was recognized in generated caption in English.

目次

第1章

序論

1.1 概要

本論文では，画像キャプション生成における生成文の品質向上を目的とし，その実現のためにデータセット中の複数形表現を統一する．その際には日本語によるキャプションと英語によるの 2 種類のデータセットを利用した．またその効果を評価するために，人間による主観的な評価と機械翻訳においてよく使われている評価手法を用いる客観的な評価の2種類の評価手法により評価を行った．今回使用した画像キャプション生成のモデルは，畳み込みニューラルネットワークと再帰的ニューラルネットワークを組み合わせた深層学習モデルである．この深層学習モデルを学習させる際に使用する訓練データに改良を加えることで最終的な生成結果の品質を向上させることが目的である．英語キャプションでは two dogs や three cars といった基数を用いた複数形表現，日本語キャプションでは「2本」や「3枚」などの基数を用いた複数形表現に対して改良を加えた．これらの表現を基数を用いず，より単純な表現へと統一することで，訓練データを改良し，改良された訓練データを用いて学習を行うことにより，生成文の品質の向上が期待できる．また本研究は，2017年3月に言語処理学会年次大会にて発表した論文，画像キャプション生成における複数形表現の統一[18]について，追加の実験及び検証を行ったものである．

画像キャプション生成についての研究は従来より活発に行われてきた[5]．これは，入力された画像のキャプション（説明文）を生成する研究である．画像キャプション生成は多くの問題を抱えており，そのうちの一つに生成したキャプションを定量的に評価することが難しいという点が挙げられる．すでに幾つかの研究で定量的に評価するための手法として提案されているものでは，BLUE[10]やMETEOR[3]などが有名であるが，これらの方法には，言い換えに対応することが難しい，評価を行う言語によってはうまく評価できないなど様々な問題があり，決定的な評価方法はまだ存在しない．そのため，現在の画像キャプション生成の優劣を計るためには複数の自動評価尺度を利用し，それぞれの手法の特徴などからどの部分で優れているかを総合的に判断するのが主流の評価方法である．また，自動評価尺度によって定量的に評価をすることが難しい要因として，画像に対する正しい説明文が一意に決定しないという点が挙げられる．例えば，犬が1匹写っている画像に対して，「犬がいる」と説明した文章も「茶色の犬が芝生の上に寝転んでいる」と説明した文章も説明の詳しさ，表現の粒度の細かさの違いはあれど，どちらも正しい説

明文となり得るのである．また，複数の牛が写っている画像などでは，「いくらかの牛がいる」や「牛の群れがいる」，「牛が13匹いる」など，複数形の表現を変えただけでも様々な説明文が考えられる．本論文では，この点に着目し，生成されるキャプションをより単純なものにすることによって誤りを減らし，生成文の品質を向上させることを考えた．具体的な方法としては，基数などを用いた複数形表現を統一，単純化することで生成文の誤りを減らすことを目指した．

先に述べたような生成されたキャプションの誤りの解消は，生成された後のキャプションを直接修正することによっても実現可能に思える．しかし，深層学習を用いた画像キャプション生成において，深層学習のモデルの内部を詳しく検証することは難しく，どの部分で誤りが発生したのかを特定するのは困難である．また，時系列的な影響（次の語が前の語に左右される）を受ける文章生成において，生成後の文章の一部分を変更しただけでは文中の他の箇所において誤りが発生する可能性がある．加えて，訓練段階で誤りを減らした場合と比べて生成する文章が多くなればなるほど訂正する文章数が増えるため，計算量が増えてしまう問題も発生する．そのため，本研究では学習に使用する訓練データを予め書き換えておき，そのデータを用いて学習を行う．

実験では，英語キャプション生成の訓練データとして MSCOCO2014 のデータセットを使用し，日本語キャプション生成の訓練データには， MSCOCOの画像を利用したSTAIRCaption を利用した．テストデータにはネットなどから収集した100枚の画像データを使用した．そして，書き換えられた訓練データで学習したシステムによるキャプションと，書き換え前のデータで学習したシステムによるキャプションとを比較し，主観的に評価付けをした．結果として，英語キャプション生成実験では，主観的な評価において，キャプションが大きく変化した画像数は100枚中39枚であり，そのうち改悪された画像数が14枚，改善された画像が25枚となった．日本語キャプション生成実験においては，キャプションが大きく変化した枚数は33枚，そのうち改悪された画像数が13枚，改善された画像が25枚となった．また，英語キャプション生成実験では自動評価尺度を用いた客観的な評価においてもスコアの改善が見られた．日本語キャプション生成実験では，自動評価尺度を用いた客観的な評価において，一部のスコアの向上が見られたが，総じてスコアが低く，複数形表現の統一による効果をうまく測定することが出来なかった．日本語キャプション生成実験については，スコアが低い原因や複数形表現の統一による効果を今一度測定し直す必要がある．

1.2　構成

本論文では，画像キャプション生成における生成文の品質向上のため，データセット中の複数形表現を統一し，それに伴う生成文の変化を評価する．2章では画像キャプション生成についての説明を行う．3章では実験に用いたPython言語の説明及び，画像キャプション生成・評価に利用したフレームワーク「Chainer」について説明する．4章ではニューラルネットワークとはどういったものなのかを説明する．5章では本論文で提案する手法について具体的にどのような手順で行うのかについて説明を行う．6章では，実験に利用したデータセットについて，どのような特徴がありどういった分野で用いられて

いるかなどの説明を行う．7章では実験で生成したキャプションに対して評価を行うために利用した，自動評価尺度について説明を行う．

第2章

画像キャプション生成

画像キャプション生成とは，入力された画像からその画像についてのキャプション（説明文）を生成する研究である．画像キャプション生成を含む画像認識という分野では，多くの研究が行われてきた．最初は，画像に何が写っているのかを自動で認識するための研究があり，知識ベースやモデルベースと言った手法が提案され，その他様々な手法で画像に対する認識の研究が行われてきた．そうして，徐々に物体に写っている物体の認識精度は高まっていったが，テーブル，ボール，車といった単語がわかるだけで，それらの関係性（テーブルの下にボールがあるのか，テーブルの上にボールがあるのか）までは認識することができなかった．そこで，画像をより深く理解するために，画像キャプション生成という研究が行われるようになっていった．論文[5]では初めて画像のみの入力でキャプション生成を行っている．しかし，実際に文章自体を 1 から生成しているのではなく，データベース内の最適な1文を利用するという形である．

図2.1　画像キャプション生成の概念図

図2.1は画像キャプション生成について簡単に説明した図である．図にあるようなカモの写っている写真をキャプション生成器に入力すると，キャプション生成器がもっともらしいキャプションを生成・選択して出力する．このとき，キャプション生成器のモデルによっては，正解である確率が高い順に数パターンの候補を出力するモデルや，出力する文章の長さを設定することができるモデル, 人間の注視している部分に関して説明できるモデルなどがあり，モデルによって様々な特徴がある．

画像キャプション生成のアプローチとしては，大きく 2 パターンあり，それは既存の文章を利用する手法と新規で文章を生成する手法である．既存の文章を利用する手法は，ニュース記事についている画像について，画像を説明している適当な文章を利用するといった手法であったり，入力である新規の画像の特徴量と似た特徴量を持つ画像につい

ているキャプションを利用する手法などがあげられる．新規で文章を生成する手法としては，予め主語述語などというテンプレートを作っておき，単語を当てはめることで生成する手法や，完全に1から全ての文章を生成する手法などがある．ただし，どちらの手法を用いる場合においても，基となるデータセットにない組み合わせについては表現することが出来ないという大きな問題はついて回る．

そして，近年キャプションを新規で生成する手法として，深層学習を用いたモデルの提案[14]が発表された．これは，画像の物体認識などの分野で成果を挙げていたCNNと呼ばれるニューラルネットワークと，翻訳などに使用されていたRNNと呼ばれるニューラルネットワークを組み合わせることで，CNNから得られる画像の特徴量を入力としてRNNを使って文章を生成するというモデルである．このモデルでは，訓練データに画像と対になるキャプションのデータを人手により用意し，それを用いてニューラルネットワークのモデルを学習させておけば，以降は画像の入力のみで，学習した単語を組み合わせた新規のキャプションを生成することが出来るモデルである．今回の実験では，先に述べた論文[14]にて提案されている，畳み込みニューラルネットワーク（CNN）と再帰的ニューラルネットワーク（RNN）を組み合わせた深層学習モデルを用いて画像のキャプション生成を行っている．これらのニューラルネットワークについての詳しい説明は4章に記載されている．

第3章

Python

Pythonとは，プログラミング言語の1種である．Pythonは言語処理の分野で多く用いられているプログラミング言語で，機械学習やその他様々なパッケージを利用可能な，拡張性にとんだ言語である．また，Cなどの言語に比べて記述が簡単であるという利点もある．今回は，言語処理の分野で多く使われていることに加えて，後述するChainerと呼ばれる深層学習用のライブラリであるChainerがPythonに対応していたため，この言語を使用しての実験を行った．

実験に使用したPythonのバージョンは2.7.0であり，Anaconda2.4.0を使ってインストールを行った．Anacondaとは，Pythonの配布形態の1つで，numpyやscipyといったライブラリを一括でインストールすることが出来るものである．Python2.x，3.x の両バージョンに対応しており，LinuxやMac，Winなど様々なOSにも対応している．様々なOSにまたがって開発をする場合や，開発の環境を一定にしたい場合などに便利である．

3.1 Chainer

Chainerとは，ニューラルネットワークを記述するためのライブラリである．他のニューラルネットワークを記述するためのライブラリにはTensorFlow[*1]やCaffe[*2]，Torch[*3]などがある．Preferred Networksが開発したこのライブラリは，他のニューラルネットワーク用ライブラリよりも記述の仕方や動作についての習得が容易であり，実際に動かすまでの時間が短いのが特徴である．また，導入に際しても比較的簡単であるため，このライブラリを使用することとした．加えて，日本初のライブラリなので日本での利用が多く，トラブルの際や学習に利用できる日本語のドキュメントが多いことも利点である．ChainerのURLは`http://chainer.org/`である．

実験では，バージョン1.17.0を使用した．

[*1] `https://www.tensorflow.org/`
[*2] `http://caffe.berkeleyvision.org/`
[*3] `http://torch.ch/`

3.2　NLTK

NLTKは自然言語処理やテキストマイニングのために用いられるPythonのライブラリでNatural Language Tool Kitの略称である．これは分類，トークン化，語幹処理，タグ付け，構文解析，及び意味推論のためのライブラリで，50を超えるコーパス及びWordNetなどの語彙リソースに対して，簡易なインターフェースを提供する．

今回の実験では，7.7節で説明する，GLEUの実装のために利用した．NLTKにはgleu_score モジュールが存在し，そこには corpus_gleu や sentence_gleu といった関数が用意されており，簡単に利用することができる．今回使用した corpus_gleu 関数は，第1引数に参照文を分かち書きした単語のリストのリスト，第2引数に生成文を分かち書きした単語のリストを入力とすることで，GLEUスコアを計算することができる．

他にもBLEUの計算を関数1つで行うことができ，文章のトークナイズ（単語ごとに切り分けること）なども関数1つを呼び出すことで利用できる．統計やストップワード（多くの文章に一般的に含まれる特徴のない語）の除去なども可能であり，自然言語処理についての処理のほとんどが簡易に利用できる．しかし，基本的に英語に対しての処理を想定して作成されているため，日本語などの2バイト文字やマイナー言語に対してそのまま用いることは難しい．

第4章

ニューラルネットワーク

ニューラルネットワークとは線形識別モデルの 1 種で，動物の神経組織（ニューロン）を模して作られたモデルである．ニューラルネットワークの動作を説明するために以下の図4.1を用いる．

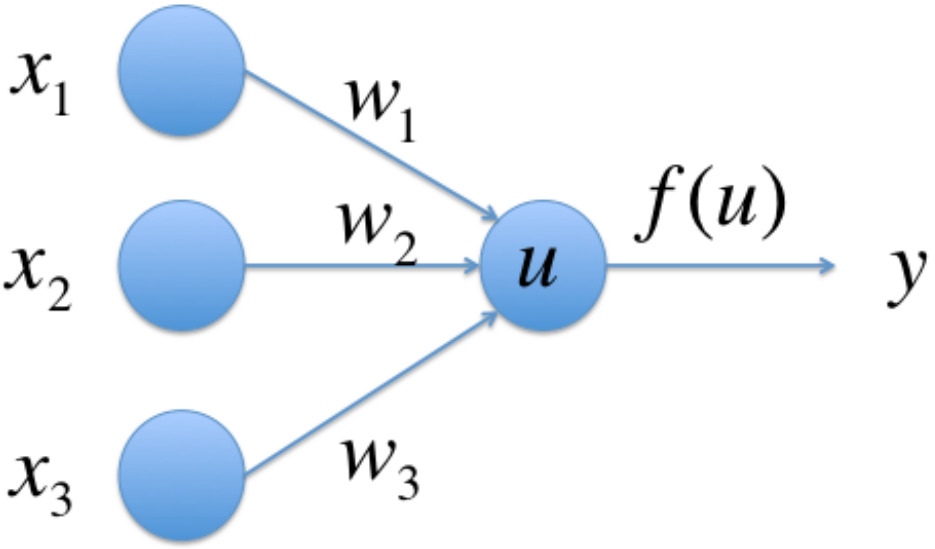

図4.1　単純パーセプトロン

この図は，ニューラルネットワークの中でもシンプルな単純パーセプトロンと呼ばれるモデルである．入力をx_1，x_2，x_3，出力を y とする単純パーセプトロンである．u は以下の式4.1で表される．

$$u = x_1 w_1 + x_2 w_2 + x_3 w_3 \tag{4.1}$$

この時，$f(u)$は活性化関数と呼ばれる関数である．これは，入力の和uから何かしらの出力を出す関数で，後で説明する誤差逆伝播法を用いるために微分可能な関数を設定する．

ここで，x_1からx_3の特徴量からそのデータがスパムメールであるかを判定することを

考える．この時，スパムメールであるかどうかの条件は以下の通りとする．

$$\begin{cases} u > 0 : (\text{スパムメールである}) \\ u < 0 : (\text{スパムメールでない}) \end{cases}$$

ここでuの数式を見ると，w_1からw_3がそれぞれの特徴量の影響力を調整していると考えられる．例えば，スパムメールである場合にx_1が深く関係しているのであれば，それに合わせてx_1の影響がuに出やすいようにw_1を大きくする．逆に関係がないようであれば影響を小さくするためにw_1で制限をかけるというように操作を行う．このw_1からw_3を重みとよび，これらをまとめたWを重みベクトルと呼ぶ．この重みベクトルを調整することで学習を行う．

学習には損失関数と呼ばれるものを用いる．これは，出力された結果がどれだけ教師データとかけ離れているのかを表す関数で，基本的に誤差が大きければ大きいほど大きな値を返すものである．そのため，損失関数が返す値が最小になれば学習が終了したものと考えられる．損失関数には0-1損失，絶対誤差損失，二乗誤差損失，対数損失など様々な種類があり，問題にあった損失関数を選択することでより良い結果が得られる．また，学習には学習率と呼ばれるものが用いられる．これは1度の修正でどの程度重みの修正を行うかという割合である．大きすぎる場合は最適解を見逃してしまう場合があるので，ある程度小さい値を用いる．ただし，あまりにも小さすぎると1度の修正で変化する量が小さくなり，計算回数が増えてしまうため，注意が必要である．

活性化関数とはパーセプトロンの出力を決定する関数である．上で説明したような例ではあまり効果が無いが，多層パーセプトロンの場合には微分可能な連続した関数が用いられる．活性化関数として使用されているものには，ステップ関数，シグモイド関数，ReLU，双曲線正接関数などが挙げられる．これも問題によって適切な活性化関数を用いることでより良い結果が得られる．例では二値分類を行っているので，uから教師データのラベルに合わせた数値を出力するステップ関数を用いるのが良いと考えられる（教師データにおいてスパムメールであれば1，スパムメールでなければ-1のラベルがついている場合は，$u > 0$で1を出力，$u < 0$で-1を出力する）．この活性化関数を通して出てきたものが単純パーセプトロンの出力yとなる．

単純パーセプトロンを組み合わせて層を増やしたものが多層パーセプトロンと呼ばれるニューラルネットワークである．これは，単純パーセプトロンからの出力を別の単純パーセプトロンへと入力することで多層としているニューラルネットワークである．単純パーセプトロンでは線形識別可能な問題しか解けなかったが，多層パーセプトロンを用いることで，線形識別不可能な問題を解決することが出来るようになった．これにより，ニューラルネットワークの活躍の幅が増えることとなった．

誤差逆伝播法について説明を行う．これは，多層パーセプトロンのように複数の層を持つニューラルネットワークに用いられる，重み更新（学習）のための手法である．ここでは，誤差逆伝播法の概要を説明するために，ニューラルネットワークを活性化関数$f(x)$とし，層1つでの計算を$g(x)$とする．この場合，1層のニューラルネットワークは入力xが層$g(x)$を通り，活性化関数$f(x)$で出力が決定されるので$f(g(x))$という合成関数の形で表せる．逆に，出力から入力を導出するには微分をしていけば良いことになる．具体的には，

活性化関数の出力をfについて偏微分して$\frac{\partial f}{\partial g}$，さらに$g$について偏微分して$\frac{\partial f}{\partial g}\frac{\partial g}{\partial x}$ と表せる．これが誤差逆伝播法の基礎となる．さらに，これを3層のNNとすると，$f(g(g(g(x))))$となる．この時，$g(x) = Wx$，$\frac{\partial g}{\partial x} = W$とすると，誤差逆伝播法を適用した場合以下の式4.2のとおりとなる．

$$\frac{f(g(g(g(x))))}{\partial x} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial g}\frac{\partial g}{\partial g}\frac{\partial g}{\partial x} = \frac{\partial f}{\partial g}WWW = \frac{\partial f}{\partial g}W^3 \tag{4.2}$$

この式より層が増えるに連れて重みベクトルWの乗数が増えていくことがわかる．このことから，層数が増えていった場合に，重みベクトルが1以下である場合には重みベクトルが極端に小さくなってしまう問題が発生し（勾配消失問題），重みベクトルが1以上であった場合には極端に巨大になってしまうという問題が発生した（勾配爆発問題）．そのため，深い層（多層）のニューラルネットワークは実現が難しかった．この問題について解決方法が模索されていたが，ReLUと呼ばれる活性化関数を用いることで勾配消失問題については解決することができている．

4.1　CNN

CNNとは畳み込みニューラルネットワーク（Convolutional Neural Network）の略称である．このニューラルネットワークは主に画像の解析に用いられる．

CNNは名前の通り，入力された情報を畳み込み，つまり圧縮して学習を行う．CNNという技術は生物工学における，視覚が情報をどのようにして処理しているのかという観点から開発されている．脳は，物体を見る際にまず光を網膜に投影する．それに反応した細胞が視神経に情報を伝えるのだが，その際に光を受け取る細胞と視神経とは疎に結合している．つまり，複数の細胞の情報を視神経でまとめているのである．これが畳み込みの発想のもととなっている．具体的には，50×50のサイズの白黒の画像があった場合に，5×5の範囲ごとにフィルタを掛けて，フィルタ内において，白である領域が過半数なら白を出力，逆に過半数が黒ならば黒を出力すると言った作業を行う．この作業を，フィルタを少しずつずらしながら行うことによって，元の画像よりも小さな白黒画像のデータが出力される．これが畳み込みである．この時，白黒の画像が光を受けた細胞，フィルタを通して出てきた出力が視神経が受け取る情報と対応付けることが出来る．この畳込みを行うことによって，画像認識において大きな問題であった，画像の歪みや位置のズレなどによる誤差を吸収することが出来る．こうして出来たものを画像の特徴量と言い，この特徴量を利用することで，様々な応用が可能になる．

この図では，手書き文字認識におけるCNNのモデルが示されている．入力として，32×32のサイズの手書き文字画像を入力し，画像を6枚28×28の特徴量マップに畳み込んでいる．この時，特徴量のサイズが元の画像よりも小さくなるのは，畳み込みによって複数ドットの情報が1つのドットとして表現されるためである．この図のモデルでは，畳み込みが2回とサブサンプリング（プーリング）を2回それぞれ交互に行っている．これにより，徐々に特徴量を絞り込んでいくのである．これらの工程の後に，それぞれの特徴量を結合していき，最終的に10次元の情報として出力している．出力されるのは，その文字

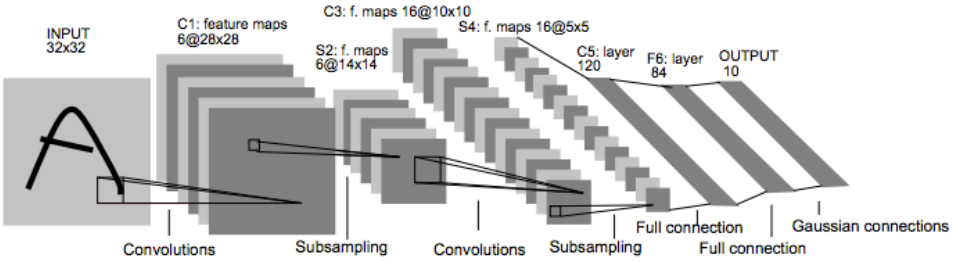

図4.2　手書き文字認識におけるCNN（図は論文[7]よりの引用）

がどの文字である可能性が高いのかという確率になる．この図では，文字がAである確率が一番高ければ正解，そうでなければ間違いというように結果を評価することが出来る．つまり，画像の解像度を徐々に落としていくことで，文字の特徴をわかりやすくしていく作業であると言える．ただし，解像度を落とす頻度が多すぎたり，1度に落とす解像度が大きすぎる場合には，うまく特徴を取ることが出来ない可能性もあるため，モデルを構築する際には慎重に行う必要がある．

CNNに限らず，ニューラルネットワークとは，学習にコストがかかる技術である．ニューラルネットワークの学習とは，訓練データ（特徴量など）の入力で訓練データの回答と同じ出力が出るように各層での重み付けを調整することを言い，そのための方法として前節で触れた誤差逆伝播法が多く用いられている．ニューラルネットワークを学習させ，重みを調整することで入力と同じものが出力される，つまり回答が未知のデータであっても判別ができるニューラルネットワークになるのである．学習には，純粋な計算能力はもとより，多数のニューラルネットワークの層を記憶・管理する必要があるため，一定以上のメモリが必要となる．また，画像処理のニューラルネットワークでは，一般的に次元数や層の数が多くなりがちであり，他のニューラルネットワークの学習に比べてメモリの使用量が多くなりやすい．最近ではサイズの大きな画像や画素数が大きい画像に対しての学習も多く行われており，訓練データの肥大化に合わせて，ニューラルネットワーク自体も巨大な物を使用する傾向があり，学習のためのコストは徐々に増大している．そのため，画像の特徴量のみを必要とする，画像を認識して何か別のことに使用するといった研究（画像キャプション生成など）では，すでに学習済みのニューラルネットワークのモデルを使用することがしばしばある．

今回の画像キャプション生成の深層学習モデルでは，画像をCNNで畳み込んで特徴量として捉えることで，RNNを使った文章生成のための入力としている．実験ではCNNから出力される特徴量の次元は4096次元であった．また，使用したCNNは学習のためのコストを抑えるために，論文[13]で発表されているものを用いた．このモデルは以下のURLにて公開されている．

`https://gist.github.com/ksimonyan/3785162f95cd2d5fee77#file-readme-md`

4.2　RNN

RNNとは再帰型ニューラルネットワーク(Recurrent Neural Network)の略称である．このニューラルネットワークは主に時系列的なデータの学習に用いられる．時系列的なデータとは，前のデータに次のデータの値が左右される文章や音声，動画といったデータのことである．

RNNとは再帰的（並列）にニューラルネットワークを構築することにより，時系列的なデータに対しての学習を可能にしたモデルである．シンプルな形のRNNでは，再帰を重ね続けることによって，学習がうまく行えなくなる問題が発生していたが，LSTM（Long Short Time Memory）とよばれる手法を組み込んだLSTMRNNという型を用いることによって，その問題を克服した．これによって，長い時系列データに対する学習が可能になったことで注目を集めている．今回の文章生成に用いたのもLSTMを利用したRNNである．また，多くのニューラルネットワークでは，入力や出力が固定次元であり，系列の量がそれぞれ違う場合（文章ごとに長さが違うなど）において適用することは難しかったが，RNNではそのようなデータに対しても適用することが可能である．

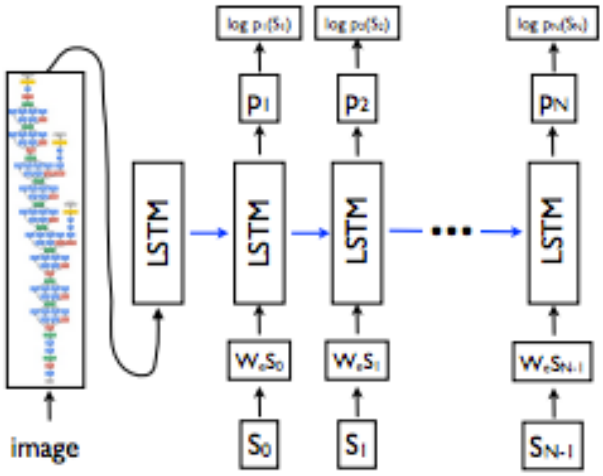

図4.3　実験で使用したCNNとRNNによる画像キャプション生成モデル（図は論文[14]より引用）

図4.3は実験で実際に使用したCNNとRNNによる画像キャプション生成モデルである．一番左の画像が入力されてLSTMと書かれたマスに入力が繋がっている．この部分がCNNによる画像の特徴量抽出を表している．LSTMの詳しい説明は後述するが，RNNの種類の1つである．RNNではニューラルネットワークを再帰的にして時系列的な処理を行う．この図においてRNNの部分はCNNからの出力を受け取った部分から右のまとまりを指す．これは，再帰的に定義されたRNNをわかりやすくするために並列のものとして展開した図である．

この図における学習の流れを辿りながらRNNによる文章生成を説明する．この時，学

習用の訓練データとして，画像とそれに付随する正解のN-1単語からなるキャプション($S_1 \sim S_{N-1}$)が与えられている．また，入力の際にこの文章には始端記号S_0と終端記号S_Nが付与される．まず，文章生成に先立って，前情報である画像の特徴量をLSTMへ入力する．そして，並列化されたニューラルネット（RNN）に文章の始端記号となるS_0を入力する．これに重みW_eをかけたものをベクトルとしてニューラルネットの層へと入力する．（このときLSTMについてはひとまず入力に対して出力を返す装置と考えておく）LSTMを通って出てきたp_1は入力S_0と事前に入力された画像の特徴量に対する出力であり，意味としては次に生成される単語の確率の集合となる．最後に，出力された確率のコストを最小化($\log p_1 S_1$)することで，最適な単語を出力する．これによって文章の先頭の単語が出力される．
次に，先頭の単語を出力する際にLSTMで得られた情報を隣のS_1が入力となるLSTM（第2単語を生成するLSTM）へと入力する．S_1はS_0の時同様重みW_eをかけてベクトル化された後に，前のRNN（S_0が入力されたニューラルネットワーク）の情報を持つLSTMへと入力される．そして，S_0のLSTMから渡された情報と$W_e S_1$からp_2が生成され，最終出力$\log p_2 S_2$となる．これを繰り返していき，文章の最後の単語S_{N-1}までをニューラルネットワークに入力し，終端記号S_Nが出力される．こうして文章の生成を行った後，出力と入力データの間違いからRNN用の誤差逆伝播法でニューラルネットワークを学習させていく．

具体例を出して説明する．”This is a pen.” という文章と，ペンが1本写った画像がある場合に，まずは画像をCNNに通してペンが写っているという特徴量を抽出し，始端記号S_0が入力されるニューラルネットワークへ入力する．この時出力として確率の集合p_1が出力される．この確率とS_1である単語 This とのコストを計算し一番小さなものが出力となる(This がp_1に含まれていればそれが最小)．次に，S_0を入力したニューラルネットワークのLSTMの情報と，単語 This (S_1)を入力し，p_2を生成し単語 is (S_2)とのコストを計算する．この作業を繰り返していき，入力が最終単語 pen で出力が終端記号S_Nとなったところで生成を終了する．その後，出てきた単語列と訓練データとの差から重みの調整（学習）をおこなう．実際に生成を行う場合は，訓練データのときのように元になるデータがないため画像の特徴量と始端記号を入力した後，前のニューラルネットワークで生成された出力（単語）を次のニューラルネットワークの入力として，終端記号が出力されるまで続けていく．

RNNの学習には通常のニューラルネットワークのような誤差逆伝播法ではなく，BPTT（Back-Propagation Through Time）とよばれる時系列的な構造も考慮した逆伝播の学習方法が採られている．この方法は，RNNを展開した際に，1つのニューラルネットワークの出力が次のニューラルネットワークの入力になっていることから，層の深いニューラルネットワークとみなして誤差逆伝播法を適用する方法である．RNNでは通常のニューラルネットワークとは違い，中間層のデータも次のニューラルネットワークの入力として使用しているため，ニューラルネットワークが出力した系列の微分と，中間層のデータの系列の微分とを用いて誤差逆伝播法を適用する．しかし，この方法は通常の誤差逆伝播法と同様に勾配消失問題が発生する場合がある，そのために提案されたのがLSTMで

ある．このBPTTにも年々改良が加えられている．また，BPTT以外の学習方法もいくつかあるため，データの特性や用途に応じて使い分けることが必要である．

LSTMとは，RNNの中間層に用いられるユニットの種類である．このユニットは記憶の保持を目的としたユニットである．RNNでは時系列的なデータを扱うことが出来るが，シンプルなタイプのRNNでは，時系列が長くなるに連れて勾配消失問題と呼ばれる問題が発生し，誤差逆伝播法が出来ず，上手く学習を行う事ができなかった（数ステップ以上の長さの学習は難しかった）．LSTMはその問題を解決するために，メモリセルと呼ばれる機構を持ち，これにより長期的な記憶の保持が出来るようになっている．しかし，保持し続けるだけでは，誤ったデータも保持し続けてしまう事があるため，それをリセットする忘却機構などが備え付けられている．これらの機構により，LSTMを用いたタイプのRNNは時系列の長い情報に対しても学習を行うことが出来るようになった．説明したのはLSTMの中でも比較的シンプルなタイプであるが，現在でもLSTMには様々な改良が加えられており，様々な種類のLSTMが提案されている．

今回の画像キャプション生成の深層学習モデルでは，CNNによって出された特徴量と，訓練データの文章を文章ごとに単語に分解したものを，RNNの入力とし，中間層（隠れ層）で512次元に変換し，出力層で入力された文章の単語数と同じ次元で出力している．

第5章

複数形表現の統一

5.1 英語における複数形表現の統一

複数形における問題点とは，複数の対象が写っている画像（犬や人間などが複数写り込んでいるもの）に対して生成されたキャプションにおいて，主語が単数になっていたり，複数形になっていても数が間違っていたりする[*1]ことである．これら間違いの多くは基数を含んだ複数形表現を生成した場合に現れる．

この問題を解決するために，学習に用いられるデータセットにおける複数形の表現をより単純な形に書き換え，複数形の表現方法を統一する．

実験では，MSCOCOのAnnotationつき画像データセットを用いた．ただし，MSCOCOで直接配布されているデータセットではなく論文[6]で使用され，以下のURL`http://cs.stanford.edu/people/karpathy/deepimagesent`で公開されているデータセットを用いた．

訓練データにおける複数形表現を統一した場合の，生成文の品質の変化を調べるために，変更を加えないデータをデフォルトデータセットとして，それを含め5パターンの訓練データを用意した．その内訳は以下のとおりである．

1. デフォルトのデータセット
2. 1について，頭文字の表記ゆれを取り除くために全ての文章を小文字にしたもの
3. 2について，複数形の文章について，基数の部分を some に置換したもの
4. 3について，some となっている部分を a group of にしたもの
5. 4について，a couple of となっている表現を a group of に統一したもの

これら，2から5のパターンのデータセットを用いて学習を行った深層学習モデルによって生成されたキャプションと，デフォルトデータセットを用いて学習を行ったモデルによって生成されたキャプションと主観的に比べることで品質が向上したかどうかを調べる．

[*1] 画像には3つ写っているのに2つと書かれるなど

5.2　日本語における複数形表現の統一

英語における複数形表現における問題と同様に，日本語に対しても複数形表現における問題点が存在する．画像キャプション生成モデルは，複数の対象が1枚の画像に写っているものに対して，しばしば数を間違えたキャプションを生成する．そのため，英語における複数形表現の統一と同様に，基数を用いた複数形表現に対する統一を行う．

訓練データに対して複数形表現の統一をするために，基数を用いた表現をより単純な形に置き換える，具体的には「2本」や「3枚」といった表現を「数本」,「数枚」といった表現に置き換えるという動作を行った．この際，文章中の数詞や助数詞を判別するためにオープンソースの形態素解析エンジンMeCabを用いた．MeCabについては5.4節にて説明を行っている．

訓練データには6.2節で説明している STAIRCaption とよばれる，MSCOCO の画像データに対して人手で日本語キャプションをつけたものを利用した．

訓練データに対する複数形表現の統一を行う前処理として，表記揺れを取り除くために正規化を行う必要がある．日本語には，英語における頭文字の大文字小文字の表記ゆれのようなものはないが，数字を表現する際に，漢数字を用いるかアラビア数字を用いるかのような揺れが存在する．今回の実験では，このような表記揺れを正規化するために，漢数字をアラビア数字へ変換した．また，「ひとつ」や「みっつ」のような和語による複数形表現については，すでに漢数字やアラビア数字として表記されているものは正規化・統一をおこなったが，ひらがなで書かれているものに対しては，数が少なかったために統一を行わなかった．

日本語における複数形表現の統一では，英語とは違い助数詞の問題が発生する．そのため，置換を行う際に数詞の次に来る助数詞によっては基数を「数」へ置換した際に文章に間違いが生じてしまうような助数詞が用いられている場合には，置換を行わないように実装した．置換を行わず無視をするような助数詞には「つ」や「番」などが多く存在した．

5.3　TreeTagger

パターン3を作成するにあたって，英文の品詞を形態素解析によって調べるためにTreeTagger[11, 12]を用いた．これは，シュトゥットガルト大学のHelmut Schmidが開発したツールで，英語やフランス語，ドイツ語など多様な言語に対しての形態素解析を行うことが出来るツールである．また，多種のプログラミング言語にも対応しており，PythonやJava，Rubyなどで使用できる．実験では置換を行うプログラムをPythonで書いていたため，Pythonを使用して実装を行った．具体的には，訓練データにおける全文章をTreeTaggerによって形態素解析し，1つの文章中に基数（Cardinal number : 品詞コードはCD）が存在し，名詞の複数形（noun plural ：品詞コードはNNS）が存在する文章[*2]に対して，品詞コードがCDである単語についての置換を行った．この際，単語が one の場合は，単純化のために置換可能であると想定した単語 some に置換することが出来ない

[*2] Two dogs are running in the yard といった文章

ため省いている．また，TreeTaggerを用いたのはパターン3をを作成する時のみであり，パターン4,5については，Pythonの正規表現を用いて， some や a couple of に合致した部分を置換している．

5.4 MeCab

日本語における複数形表現を判別するために，オープンソースの形態素解析エンジンMeCab [20]を用いた．

MeCabは京都大学情報学研究科ー日本電信電話株式会社コミュニケーション化学基礎研究所共同研究ユニットプロジェクトを通じて開発された形態素解析エンジンである．日本語に対する自然言語処理の研究によく利用されており，形態素解析や分かち書きなどの機能が利用できる．また，各種スクリプト言語バインディングが存在し，Perl,Ruby,Pythonなどの様々な言語において利用可能である．加えて，ユーザ辞書を追加することが出来るため，拡張性にも優れている．他の形態素解析システムにはChaSen[19]やJUMAN[21]などが存在する．

今回はPythonを用いて実装を行った．訓練データの日本語キャプションに対して形態素解析を行い，数字の部分だけを抽出し「数」という単語に置換を行った．ただしこの時，文章として誤りが発生しないように，時間を表す際に用いられている数字や，ひとつふたつなどの和語での数え方，数字の部分を「数」に置換した際に単語として不自然になるような場合[*3]の場合は置換を行わないよう例外として処理を行った．

[*3] 2番目など

第6章

データセット

6.1 MSCOCO

MSCOCO（Microsoft Common Objects in Context）データセットとは画像に対してキャプションが付与されているデータセットであり，Microsoftによって提供されている[9]．MSCOCOデータセットは画像キャプション生成・物体認識・画像分類などの様々な研究に用いられておりそういった研究におけるベンチマークデータセットとして扱われてきた．また，多くの研究で利用されているために，サポートも多く存在しており，生成された文章に対して，いくつかの機械翻訳の評価方法でスコア付してくれるAPIや人手でキャプションを生成した際に生じた表記揺れやミスなどを正規化してくれるものなどが存在する．特に表記揺れなどは生成文の精度に直接関わってくる部分のため，しっかり行っておくべきである．また，深層学習を利用した画像キャプション生成システムにおける画像分類の層は学習に時間がかかるが，MSCOCOに対しては学習済みのモデルが存在し，そのモデルを活用することで実験にかかる時間の短縮を行うことができる．

ただし，本論文でMSCOCOデータセットを用いる際には実験の性質上，キャプションを変更する必要があったため，正規化は自分で行った．本論文で使用したデータセットは2014年の画像データセットであり，画像枚数が123,287枚，それに付与されたキャプション総数は616,767文となっていた．変更されたキャプション数などのデータは8章で詳しく説明する．

6.2 STAIR caption

STAIR Capionは千葉工業大学人工知能・ソフトウェア技術研究センター及び奈良先端科学技術大学院大学が共同で開発したMSCOCOの画像データをもとに日本語のキャプションを付与したものである[1].

これまでの画像キャプション生成に用いられてきたデータセットのほとんどは英語によるキャプションが付与されたもので，日本語キャプションが付与されたデータセットは少なかった．この問題の解決策としてMSCOCO2014年版の訓練・開発・テスト用データのすべての画像164,062枚のそれぞれに対して人手により各５文のキャプションを付与した．したがって，STAIR caption には820,310文の日本語キャプションのデータが存在する．

STAIR Caption を利用する際の利点として，MSCOCOデータセットに合わせてデータセットが構築されているため，MSCOCOに対するサポートAPIが利用できると言った点や，他の日本語キャプションよりも語彙数やキャプション数が優れている点が挙げられる．また，MSCOCO のデータセット構造に準じる形で構築されているので，MSCOCOデータセットに触れたことがある場合に理解や把握が早く利用しやすい．

本論文ではこのSTAIR Captionを用いて日本語キャプション生成の実験を行った．実験では，複数形表現を統一するために，キャプションの形態素解析を行い，数字とそうでない単語を判別し，置換を行った．形態素解析にはMecab[20]を用いた．

STAIR Caption は以下Listing6.1のような構造を持つ JSON ファイルとして配布されている．

Listing 6.1　STAIR Caption の構造

```
annotation{
    "id'                    : int,
    "image_id"              : int,
    "caption"               : str,
    "tokenized_caption"     : str,
    }
```

第7章

評価手法

7.1 Microsoft COCO Caption Evaluation

Micrsoft COCO Caption Evaluation は github 上に公開されている，MSCOCOのキャプションに対して評価を行うためのプログラムである[16]．MSCOCOの論文[4]作成に携わった Tsung-Yi Lin や Xinlei Cheu らが開発した．

このプログラムを利用するにはJava1.8.0 及び Python2.7 が必要である．また，セットアップ時に Stanford CoreNLP 3.6.0 をダウンロードする必要がある．

このプログラムを利用することで，以下の評価手法を利用することができるようになる．各評価手法については7.2節から7.6節で詳しく説明している．

- METEOR
- CIDEr
- BLEU
- ROUGE-L
- SPICE

7.2 METEOR

METEORは機械翻訳に対する評価尺度の一つである[3]．参照訳と出力の1-gramの合致数をもとにした評価尺度で，1-gramの適合率と再現率及び，どれくらい並びが参照訳と似ているかを計算する文単位の評価手法である．METEORは他の自動評価尺度に対して，表現の微妙な違いや活用の仕方などを吸収して評価できる方法として設計された．そのため，様々な言語で類義語集を用意したり，語幹だけのマッチや単語が完全一致していなくてもマッチしていると判定する．これによって文章ごとの微妙な違いを吸収し，本質的な文章同士の一致を評価することができる．しかし，日本語などの言語に対応した類義語集が存在しないなどの問題も存在する．METEORは0から1の値をスコアとして出力し，スコアが高いほど評価が良い．

METEORは再現率Rと適合率Pに基づくF1スコアに対して単語の非連続性に対するペナルティ関数Penを利用した評価指標である．以下に式7.1,7.2,7.3としてMETEORの計算方法を示す．このとき，式7.3にあるmは機械翻訳の生成文と参照訳との間で一致した

単語数であり，cは一致した各単語を対象として語順が同じものを1つのまとまりとして統合した場合のまとまりの数である．仮に，機械翻訳の生成文と参照役が完全一致している場合には$c = 1$となり，語順が全て逆の場合には$c = m$となる．α,β,γはパラメータである．

$$METEOR = (1 - Pen) \times F \tag{7.1}$$

$$F = \frac{P \times R}{\alpha \times P + (1-\alpha) \times R} \tag{7.2}$$

$$Pen = \gamma \times \left(\frac{c}{m}\right)^{\beta} \tag{7.3}$$

7.3 CIDEr

CIDEr（Consensus-base Image Description Evaluation）は画像キャプション生成の評価に用いられる，n-gramの出現回数のTF-IDF重み付き評価尺度となる．このスコアは高いほど質の良いキャプションとされる．この評価尺度は複数の人間が正しいと判断した（consensus-based）キャプションに対して良いスコアをつけることを目標として，提案された．この手法では単語は語幹または語根の形式に正規化され，各文章はN-gramのセットとして表される．生成文のN-gramが参照訳に多く含まれれば（一致していれば）含まれるほど，その生成文のスコアを高くする．しかし，参照訳に含まれないN-gramがある場合は，間違いである可能性が高いので，ペナルティをかける．この動作をTF-IDF重み付けを利用することによって実装している．

以下にCIDErの計算式を式7.4,7.5,7.6として示す．これらの式によりN-gram:w_kの出現回数$h_k(s_{ij})$のTF-IDF重み付き評価尺度を計算する．式7.4ではの前半部分ではTF（リファレンスS_{ij}の中で出現回数の多いN-gramに大きい重みをつける）を計算し，後半部分ではIDF（全画像を通して出現回数の多いN-gramに小さな重みをつける）といった計算を行っている．この時，j =リファレンス番号であり，i =画像番号である．式7.5では仮説c_iと対応するすべてのリファレンスの集合$S_i = \{s_1, s_2, \cdots, s_m\}$の各文章についてコサイン類似度をとり，対応する全文章との平均をとる．式7.3では最終的なスコアを算出する．この時，w_nはN-gramを正規化する項であり，BLEUなどと同様に$\frac{1}{N}$を用いている．また，Nの値も$N = 4$を用いるのが望ましいとされる．

$$g_k(s_{ij}) = \frac{h_k(s_{ij})}{\sum_{\omega_l \in \Omega} h_l(s_{ij})} \log\left(\frac{|I|}{\sum_{I_p \in I} min\left(1, \sum_q h_k(s_{pq})\right)}\right) \tag{7.4}$$

$$CIDEr_n(c_i, S_i) = \frac{1}{m}\sum_j \frac{g^n(c_i) \cdot g^n(s_{ij})}{\|g^n(c_i)\| \, \|g^n(s_{ij})\|} \tag{7.5}$$

$$CIDEr(c_i, S_i) = \sum_j w_n CIDEr(c_i, S_i) \tag{7.6}$$

7.4　BLEU

BLEU(BiLingual Evaluation Understudy)スコア[10]とは，機械翻訳によって生成されたテキストを自動的に評価するための指標の一つである．評価の優劣は機械翻訳が出力した生成文と高品質な参照訳[*1]との類似度を0から1までの数字で表される．値が0の場合，生成文と参照訳の間に一致する部分がない（品質が低い）ことを表し，値が1の場合は，参照役と完全に一致していることを表す．一般的に0.2から0.4程度のスコアであれば適切もしくは主旨が理解できるレベル，それ以上であればより高品質な生成文であるとされる．

BLEUスコアは以下の式7.7によって表される．また，この数式7.7におけるBPは式7.8で計算される．BLEUスコアは短すぎる生成文に対するペナリティをかけるBP部分と，N-gram適合率を計算する部分の2つで構成されている．BP(Brevity Penalty)の計算では生成文の長さと最も近い参照訳の長さを比較して，生成文が短すぎる場合にスコアにペナルティをかける．N-gram適合率の計算では，1-gramでは単語の適切さを評価し，それ以上の長さのN-gramでは生成文の流暢さを評価する．

それぞれの変数の意味は次のとおりとなる．

- c：生成文の長さ
- r：参照訳の長さ
- N：N-gram数，一般的に4-gramが用いられることが多い
- W_n：N-gramの重みを表す．一般的に$\frac{1}{N}$で表される．

$$BLEU = BP \cdot exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{7.7}$$

$$BP = \begin{cases} 1 & if \quad c > r \\ e^{\left(1-\frac{r}{c}\right)} & if \quad c \leq r \end{cases} \tag{7.8}$$

この評価手法は文章生成の結果の指標にしばしば用いられるが，いくつか問題点を抱えている．まず1つに，内容語と機能語の区別がないため，文意が全く変わっているのにペナルティが軽い，もしくは文意はあっているのにスコアが低いといった現象が起こる場合がある．「a」「the」といった機能語が抜けている場合と，「dog」となるべき単語が誤って「cat」となっている場合で同じペナルティがかかるため，例えば，「a dog is running in the yard.」という文章に対して，「dog is running in the yard.」と「a cat is running in the yard」の2文に対するペナルティがおなじになってしまう．

2つ目に，先程の問題にも関わりのある部分で，分の意味や文法の正しさまでは十分に評価出来ないという問題がある．例えば，「not」のような否定を意味する単語があるかないかによって文の意味は正反対になることがあるが，BLEUスコアでは判断することが出来ない．また，しばしば用いられる4-gramでのBLEUスコアでは長い範囲の依存性が

[*1] 一般的に複数の人手によって与えられた文章

無視されるため，文法的に合わない文に小さなペナルティしか課さない場合がある．加えて，文章の字面しか見ることが出来ないために，同義語や活用などの変化に弱く，そういった文章を品質の悪いものとして評価しがちである．

3つ目の問題点としては，N-gramの特性上，参照訳と同じ単語が使われているだけで品質の悪い文章[*2]を持ってきた際に1-gramが$\frac{5}{5}$となってしまい，スコアが良くなってしまう問題がある．ただし，この問題に関しては一度使った単語は使えなくするといった制限を加えることで抑えるモデルが提案されている．

しかし，この手法が他の手法に比べて比較的容易かつ安定した指標であり，人手による評価の結果との相関が認められているため，多くの論文において機械翻訳などの生成文の指標として用いられているのが現状である．

7.5　ROUGE-L

ROUGEとはテキスト要約の評価によく使われる評価尺度の一つである[8]．ROUGEにはいくつかバリエーションが有り，本論文で用いたのはROUGE-Lと呼ばれる手法である．ROUGE-Lを選んだ理由としては，人手との相関が認められており，MSCOCOの評価APIで利用可能であるという点が挙げられる[*3]．これは，生成した要約と人手で作成した要約とで一致する最大のシーケンスを評価するという手法である．一致する最大のシーケンス（Longest Common Subsequene）の頭文字からLがついている．この評価手法も他の手法と同じように人手で作成された参照訳と生成された要約との類似度をもとに評価を行っている．

スコアを計算する際にはPrecisionとRecallという2つの概念がある．Precisionとは，生成した要約がどれだけ人手の要約に含まれているかを表し，この値が高ければ生成した要約に人手で生成された参照訳と一致した部分が多く存在し，似通っているということになる．Recallとは，人手で生成された参照訳中の単語をどれだけ当てられたかを表し，この値が高ければ，人手で生成された参照訳中の単語を多く利用できていることとなり，似通っているということになる．以下の式7.9,7.10によって評価される．そして，これらをバランスさせたF1スコアを算出し，それをもって要約の評価をおこなう．

$$Recall = \frac{LSC\left(summary_{words}, reference_{words}\right)}{reference_{words}} \tag{7.9}$$

$$Precision = \frac{LSC\left(summary_{words}, reference_{words}\right)}{summary_{words}} \tag{7.10}$$

ROUGE-Lでは一致する最大のシーケンスを評価するので，要約の連なりを評価する．また，この連なりの間には単語が挟まっていてもよい，としている[*4]．また，N-gramのように連なりの最大値の設定などもないため，最大限文脈を考慮することができる．た

[*2] 「the」という単語が使われている参照訳に対して「the the the the the」のような生成文のとき

[*3] バリエーションごとの人手評価との相関は論文[8]を参照

[*4] そのため，間に無駄な単語が挟まってしまっていてもスコアが高く計算されてしまう場合がある

だし，最大のシーケンスについて評価を行うため，それ以外の場所がどれだけ一致していても（またはしていなくても）考慮できないという弱みがある．

7.6 SPICE

SPICE（Semantic Propositional Image Caption Evaluation）とは，参照分と生成文の意味がどの程度一致しているのかを，深い解析に基づいて計算する画像キャプションの自動評価指標の1つである．

以下にSPICEの算出方法の概要を示す．詳しい計算方法などは，論文[2]に載っている．

まずはじめに，$G(c)$は候補キャプションcのシーングラフであり，参照キャプションのシーングラフは$G(S) = \{G(s_1), G(s_2), \cdots, G(s_i)\}$で表される．次に，キャプションからシーングラフを生成するために意味解析を行う．

意味解析は以下の式で定義される字幕解析のサブタスクによって行われる．この時，1組のオブジェクトクラスC，1組の関係型R，1組の属性型Aおよびキャプションcが与えられると，cをシーングラフに構文解析することができる．式7.11において，$O(c) \subset C$はcで表されるオブジェクトの集合$E(c) \subseteq O(c) \times R \times O(c)$はオブジェクト間の関係を表すハイパーエッジの集合，そして$K(c) \subset O(c) \times A$はオブジェクトに関連付けられている属性のセットである．実際の場合，C，RおよびAは新しいオブジェクト，関係，及び属性の種類が識別されるときに展開されるオープンワールドのセットであり，表現できるオブジェクト関係，及び属性の種類に制限はない．

$$G(c) \quad = \quad \langle O(c), E(c), K(c) \rangle \tag{7.11}$$

次にF値の計算を行う．候補シーングラフと参照シーングラフの類似性を評価するために，シーングラフ内の意味関係をタプルとしてみる．この時，シーングラフからタプルを返す関数Tを次の式7.12のように定義する．

$$T\left(G(c)\right) \quad \triangleq \quad O(c) \quad \cup \quad E(c) \quad \cup \quad K(c) \tag{7.12}$$

シーングラフの意味命題をタプルの集合としてみて，2つのシーングラフのマッチングタプルを返す関数を表す演算子として$\otimes$を定義する．次に式7.13で精度P，式7.14で再現率R，式7.15で$SPICE$を定義します．

$$P\left(c, S\right) \quad = \quad \frac{\left|T\left(G(c)\right) \otimes T\left(G(S)\right)\right|}{\left|T\left(G(c)\right)\right|} \tag{7.13}$$

$$R\left(c, S\right) \quad = \quad \frac{\left|T\left(G(c)\right) \otimes T\left(G(S)\right)\right|}{\left|T\left(G(S)\right)\right|} \tag{7.14}$$

$$SPICE\left(c, S\right) \quad = \quad F_1\left(c, S\right) \quad = \quad \frac{2 \quad \cdot \quad P\left(c, S\right) \quad \cdot \quad R\left(c, S\right)}{P\left(c, S\right) \quad + \quad R\left(c, S\right)} \tag{7.15}$$

意味論的な構造のマッチングを取るなどの手順がサブタスクとして定義されるが，最終的に計算しているのはF値であるため，SPICEは理解しやすく，0から1の間のスコアと

して算出されるため，良し悪しが簡単に判断できる．また，CIDErとは異なり，SPICEはコーパスに対する単語頻度などのデータセットの統計を利用しないため，小規模データセットでも大規模データセットでも用いることができる．

7.7　GLEU

GLEUとは機械翻訳によって生成されたテキストを自動的に評価するための手法の一つである[15]．機械翻訳の生成した文章を評価する手法の一つにBLEUがあるが，BLEUはコーパス単位での指標として設計されているため，単一の文章の評価に使用する際にいくつかの問題が発生する．その問題を解決するために，GLEUと呼ばれるスコアが用いられる．

GLEUスコアでは1から4-gramまでのトークンの全ての部分配列を記録する．次に再現率を計算する．これは，参照文中の全てのn-gramに対して生成文の全てのn-gramがどの程度合致しているかの割合で示される．さらに，精度は生成文中の全てのn-gramに対して，参照文中の全てのn-gramがいくつ合致しているかの割合で示される．そして，GLEUスコアは単純に再現率と精度の最小値で表される．

このGLEUスコアは常に0から1の間で計算され，全く合致しなければ0，全て合致すれば1となる．また，生成文と参照文を入れ替えた場合に対称性を持つ．

論文[15]の実験によると，GLEUスコアはBLEUスコアとコーパスレベルにおいてとても良く相関しているが，単一文章レベルでの欠点は解消されている，と報告されている．

第8章

実験

8.1 英語キャプションに対する複数形表現の統一

8.1.1 実験設定

今回の実験では，全ての訓練データのパターンに対して，RNNの学習の回数を100回とし，学習を行ったモデルから生成されたキャプションを結果として主観的に評価を行った．学習したモデルからキャプションを生成する際に使用したテスト用のデータには，学習時に使用されていない画像を100枚用意した．このうち，50枚を複数の対象が写っている画像，もう50枚を対象が1つであったり風景のみの画像とした．これは，複数形表現の品質が向上したかどうかだけでなく，それ以外の表現についても品質の向上，もしくは低下が見られるかどうかの検証を行うためである．また，BLEUなどの自動評価尺度を利用するために，検証用データから，複数形表現を用いているキャプションを持つ画像100枚，複数形表現を用いているキャプションを持たない画像100枚の計200枚を抜き出し，検証を行った．

実験に用いた訓練データの調整は以下のような手順で行った．

1. 訓練データの全キャプションを小文字に変換（パターン2の作成）
2. TreeTaggerを用いての品詞の確認・基数の置換（パターン3の作成）
3. 単語someの置換（パターン4の作成）
4. 慣用句 a couple of の置換（パターン5の作成）

また，実行環境は以下の表8.1のとおりである．この環境下において，GPUを用いてRNNの学習100回にかかった時間は14時間程度であった．

表8.1 実行環境

OS	Ubuntu 16.04 LTS
GPU	GeForce GTX 750 Ti
Python	2.7.0 Anaconda 2.4.0
Chainer	1.17.0

8.1.2　データセット

実験で使用したデータセットの内容は，画像枚数が123,287枚，それらに付与された総キャプション数が616,767文となっている．TreeTaggerを用いて全文章についての形態素解析を行い，カウントを行った結果，デフォルトデータセットのうち複数形が含まれている文章は284,549文あり，さらにそのうちの65,643文が基数を含むキャプションであった．この基数を含むキャプションの基数部分を some に置き換えたものがパターン3のデータセットとなる．ただし，複数形が含まれており基数も存在するが，その基数が one のみである場合は some に置き換えることができないため，省いている．そのため，実際に置換を行ったキャプション数は63,794文であった．

次に，パターン4を作成するためにキャプション中に some を含むキャプションに対して置換を行った．some を a group of に置換したキャプション数は86,173文であった．この数は，デフォルトデータセットの時点で some が使用されていた文章も含むため，パターン3で置換した文章数よりも多くなっている．また，置換した後にカウントを行っているため，デフォルトデータセットの時点で some が含まれており，パターン3の作成時に some が増えた文章については1つとして数えられている．

そして，デフォルトデータセットのうち a couple of が使用されていたキャプションが9,433文，a group of が使用されていたキャプションが19,359文あった．パターン5を作成するためにこの9,433文に含まれている a couple of を全て a group of に置換した．

以上から，パターン5における a group of を含むキャプション数は，113,072文となった．置換したキャプションの総和よりも最終的な a group of を含むキャプション数が少ないのは，パターン2からパターン5まで順に変換していく間に一つのキャプションに変換すべき語がいくつかあるキャプション[*1]があったため，重複してカウントされたものと考えられる．

以下に各パターンを生成するために変更した文章数を単語ごとにまとめたものを表8.2として示す．

表8.2　単語ごとの変更文章数

変更した単語	変更した文章数
基数	65643
some	86173
a couple of	9433

8.1.3　結果

実験の結果として，キャプション生成に用いた画像と，それぞれの5.1節で説明した各パターンのモデルにおける生成文を5文ずつ以下に示す．

図8.1において，それぞれのパターンでの生成結果を以下に示す．

[*1] some が使用されているかつ a couple of が使用されているキャプションなど．

図8.1　生成文がより良くなった場合

パターン1を用いた場合

a bird that is standing in the water
a flock of birds standing on top of a beach
a couple of birds standing on top of a beach
a bird that is standing on a beach
a flock of birds sitting on top of a beach

パターン2を用いた場合

a bird that is standing in the water
a bird flying over a body of water
a bird standing on top of a sandy beach
a couple of birds that are standing in the water
a bird standing on top of a body of water

パターン3を用いた場合

a bird flying over a body of water
a bird that is standing in the water
a couple of birds that are standing in the water
a bird that is sitting on a rock
a bird that is standing on a beach

パターン4を用いた場合

a group of ducks swimming in a body of water
a group of birds sitting on top of a sandy beach
a group of birds sitting on a rock in the water
a group of birds are standing in the water
a black and white bird sitting on a rock

パターン5を用いた場合

a group of ducks swimming in a body of water
a group of ducks are swimming in the water
a group of birds that are standing in the water
a group of ducks that are standing in the water
a group of ducks are swimming in a pond

図8.2　生成文が悪くなった例

図8.2において，それぞれのパターンでの生成結果を以下に示す．

パターン1を用いた場合

a brown and white dog laying on a bed
a dog laying on top of a bed
a black and white dog laying on a bed
a brown and white dog laying on top of a bed
a close up of a dog laying on a bed

パターン2を用いた場合

a close up of a dog wearing a hat
a close up of a dog laying on a couch
a close up of a dog wearing a santa hat
a black and white cat sitting on a chair
a close up of a dog sitting on a chair

パターン3を用いた場合

a dog that is sitting in the grass
a dog that is laying down in the grass
a dog that is sitting on a couch
a dog that is laying down on a bench
a close up of a dog on a leash

パターン4を用いた場合

a dog that is sitting in the grass
a dog that is sitting on a couch
a dog that is sitting on a chair
a dog that is laying down on a couch
a close up of a dog laying on a couch

パターン5を用いた場合

a close up of a cat laying on a bed
a close up of a cat laying on a couch
a close up of a dog laying on a couch
a close up of a dog laying on a bed
a cat that is laying down on a couch

100枚の画像に対して，デフォルトデータセットでの生成文と各訓練データのパターンでの生成文を比較した結果を表8.3に示す．

表8.3　デフォルトデータセットとの比較

	良くなった生成文数	悪くなった生成文数
パターン2	13	11
パターン3	10	12
パターン4	15	13
パターン5	25	14

この結果より，置換したキャプション数が多くなっていくに連れて変化したキャプション数が増加していることがわかる．また，パターン4までは良くなったキャプション数と悪くなったキャプション数にほとんど差がないが，パターン5では良くなったキャプションがより多くなっている．パターン3では良くなったキャプション数よりも悪くなったキャプション数のほうが多くなっているが，生成されたキャプションを見ると，置換して増加したはずの単語some がほとんど使用されていなかった（500文中1,2文程度であった）．

パターン4とパターン5では，置換した a group of やa couple of の表現が多く出現していた．その影響でキャプションが良くなった場合が多く見られたが，副作用として主語となる対象の認識が誤っているものも見られた[*2]．

また，機械翻訳の評価指標であるBLEUやMETEORなどを用いて，人手による評価で一番良い結果を出したパターン５についてスコアを算出したところ，以下の表8.4のような結果になった．BLEUの後ろの数字はNgramの数字を表している[*3]．このとき，GLEUの値のみMSCOCO CaptionEvaluationで実装されていなかったため，NLTKを用いて実装を行い検証した．検証結果のスコアに対して，良い結果のほうには下線をふっている．

この結果より，BLEUスコアは全てのNgramに対して複数形表現を統一したデータセッ

[*2] 数匹の猫が写っている画像を羊と表現するなど．
[*3] BLEU4ならば4gramBLEUの結果を示す．

表8.4　評価スコア

	データセットパターン５	デフォルトデータセット
BLEU1	0.636	0.632
BLEU2	0.473	0.446
BLEU3	0.336	0.309
BLEU4	0.232	0.219
CIDEr	0.611	0.620
ROUGE-L	0.457	0.444
METEOR	0.196	0.193
SPICE	0.132	0.123
GLEU	0.463	0.465

トパターン5のほうが良い結果が出ている．また，ROUGE-L，METEOR，SPICEにおいても，複数形表現を統一したデータセットのほうが良いスコアを得ている．しかし，CIDEr・GLEUにおいてはデフォルトのデータセットのほうが良いスコアとなっている．

8.2　日本語キャプションに対する複数形表現の統一

8.2.1　実験設定

今回の実験では，MSCOCOをもとにした日本語キャプションデータセットであるSTAIR Captionに対して，複数形表現の統一を行い，複数形表現を統一したデータセットで学習したモデルが生成したキャプションとデフォルトのデータセットで学習したモデルが生成したキャプションの２種類のキャプションに対して評価を行う．STAIR Captionについては6.2節で，複数形表現の統一の手法については5.2節にて説明をする．また，本実験の学習にはgithubにて公開されている，論文[14]について日本語でも利用可能な形で実装したプログラムを利用した[17].

学習はバッチサイズを32，イテレーションを25000として行った．生成させるキャプションは，複数形表現を統一した結果，どのような影響が出るかを確かめるため，参照文において複数形表現を用いている文章を持つ画像100枚，複数形表現を用いている文章を持たない画像100枚の計200枚に対してデフォルトモデル・複数形表現統一モデルの両モデルで画像キャプション生成を行い，その結果について7章で説明した評価手法を用いて評価した．また，英語キャプションの実験と同様に，複数形表現を用いている画像50枚と用いていない画像50枚に対して生成されたキャプションについて，主観的な評価を行った．

8.2.2　データセット

実験には6.2節で説明したSTAIR Caption を用いた．このデータセットには164,063枚の画像と各画像に付き5文，計820,310文のキャプションが含まれる．ただし，公開されて

いるデータセットではテスト画像に付けられたキャプションは除かれており，訓練データのキャプション数は，413,915文で，確認用データのキャプション数は202,520，計616,435文のキャプションを用いて実験を行った．

実験ではまず，漢数字によって記述されている部分についてアラビア数字に直す正規化を行った．この時変更した文章数は63,014文であった．また，訓練データの総キャプションのうち複数形・単数形問わず数字を用いている文章は66,467文であり，複数形表現の統一を行ったのは42,637文で，これは訓練データの総キャプションの10.3% を占める．

表8.5　単語ごとの変更文章数

訓練データの文章数	数字を用いている文章数	複数形表現を統一した文章数
413,915	66,467	42,637

8.2.3　結果

実験の結果として，生成文の品質が向上した場合と劣化してしまった場合のキャプション生成に用いた画像と．その生成文を以下に示す．

図8.3　生成文の品質が向上した場合

図8.3に対して，デフォルトモデル及び複数形表現の統一を行ったモデルのそれぞれで生成されたキャプションを以下に示す．

図8.3に対して生成されたキャプション

デフォルトモデル：象 が 鼻 を 使っ て 水 を 飲ん で いる
統一したモデル：象 が 数 頭 、 鼻 を 丸め て 水 を 飲ん で いる

図8.4に対して，デフォルトモデル及び複数形表現の統一を行ったモデルのそれぞれで生成されたキャプションを以下に示す．

図8.4　生成文の品質が劣化してしまった場合

図8.4に対して生成されたキャプション

デフォルトモデル：草原 の 中 に キリン が 2 頭 立っ て いる
統一したモデル：キリン が 数 頭 、 柵 の 中 に いる

100枚の画像に対して生成された文章に対する主観での評価結果を表8.6として表す．

表8.6　デフォルトデータセットとの比較

	良くなった生成文数	悪くなった生成文数
複数形表現を用いている画像	12	6
複数形表現を用いていない画像	8	7
計	20	13

この結果より，参照文に複数形表現を用いている（複数の対象が画像中に存在する）画像に対しては複数形表現の統一の効果が表れているが，複数形表現を用いない画像に対しては，生成文の品質の向上は認められなかった．品質の向上した生成文の多くは複数形表現の誤りが訂正された文章であった．逆に品質が悪くなってしまった生成文には，似た文章や言い回しが多く，複数形表現の統一による副作用であると考えられる．

次に，自動評価尺度を用いた評価を以下の表8.7に示す．評価方法はBLEU・ROUGE-L．SPICE・GLEUを用いた．同値の結果を除いて，数値上良い結果となっている方のスコアには下線をふっている．

この結果から，SPICEによる評価では複数形表現を統一した結果，スコアの向上が認められた．しかし，他の評価手法においては，スコアの向上が認められず，複数形表現を統一しないほうがスコアが高いものもあった．特にBLEUやGLEUのスコアが複数形表現を統一したデータセットとデフォルトデータセットともにとても低くなっている．

表8.7　評価スコア

	複数形表現の統一を行ったデータセット	デフォルトデータセット
BLEU1	0.198	0.192
BLEU2	0.131	0.131
BLEU3	0.073	0.079
BLEU4	0.043	0.052
ROUGE_L	0.210	0.208
SPICE	0.224	0.215
GLEU	0.085	0.088

第9章

考察

深層学習モデルを用いた画像キャプション生成において，間違ったキャプションが生成された場合，その原因がどこにあるのかを特定することは容易ではない．そこで，本実験では訓練データの書き換えを行い，書き換え後の訓練データを用いて学習させたモデルとデフォルトデータセットを用いて学習させたモデルのそれぞれからキャプション生成し，生成結果の検証を行った．

結果として，英語キャプションでは，書き換えた部分はもちろん，書き換えられていない部分においても誤りの改善が見られた．これは，深層学習における文章生成の部分において，前の語を踏まえた上で次の語の確率を求めるという仕組みに深く関係があると考えられる．例えば，深層学習での文章生成では，訓練データにおいて two という単語の次に来る単語が dogs と cars しかなかった場合には， two のあとには，dogs か cars が続く傾向があるのだとモデルは学習してしまう．そのため，2匹の羊が写っている画像を入力として与えた場合に，two という単語が生成されたとしても，その次に sheep という単語が出現しづらくなる．それにより，2匹の羊が写っている画像に対して「2匹の白い犬がいる」などの誤ったキャプションが発生しやすくなる．今回の実験では two に当たる部分，つまり複数形の表現を統一したことによって，次に来る語の選択肢が大幅に増えた．それにより，書き換えを行った部分以外でも誤りが発生しにくくなったと考えられる．

また，英語キャプションに対する複数形表現の統一に対する自動評価尺度では，CIDErとGLEUのスコアにおいて複数形表現の統一後のスコアが下がっていた．この理由としては，CIDErではTF-IDFによる重み付けやペナルティが発生するため，複数形表現を統一することで数詞が頻出し，一般的な単語扱い（is や the など）となってしまい，そのペナルティによってスコアが下がったのだと考えられる．GLEUに関してはほぼ同値のため，サンプルのとり方による誤差ではないかと考えられる．

一方で，日本語キャプションにおける複数形表現の統一による効果は，あまり芳しいものとは言えない．主観的評価では対象が複数写っている画像に対しての生成文では品質の向上が認められたが，それ以外の画像に対しての生成文では品質の向上は認められなかった．また，複数形表現の統一による副作用として，表現の偏りが多く見られた．自動評価手法による客観的な評価においては，SPICEを除く，ほぼすべての自動評価尺度において同値もしくは複数形表現を統一した場合のスコアのほうが低くなっており，特にBLEUやGELUなどのN-gramに基づくスコア付をする評価手法に関して非常に低い値が

出ている．またSPICE以外のスコアはデータセットの違いにかかわらず総じて低い値になっており，そもそもの生成文のクオリティが低いことがわかる．実際に，生成されたキャプションの一部を人手により確認してみたところ，ほぼ一致していない文章を生成している例が多く見られた．しかし，SPICEの結果に関しては，僅かだが誤差ではない向上が見られている．SPICEは文の意味構造を基準として評価を行っているため，BLEUなどの字面やN-gramを元にしたスコアと違い，妥当なスコアが出ていると考えられる．そして，このスコアが高いということは，複数形表現を統一したことによって，字面に左右されず，意味的に参照文と似通った文章が生成出来ているという意味であり，複数形表現を統一した場合のほうがスコアが高いため，「人間には正しいと感じられる品質の高い文章」をより多く生成することが出来たのだと捉えられる．

日本語キャプション生成実験において，このように総じて低いスコアが算出されてしまった原因としては，学習が足りない事が考えられる．今回の実験では，マシンの性能や時間などの制約から，バッチサイズを32，イテレーションを25,000として実験を行ったが，データセット全体に対するエポックとしては2週することが出来ていなかった．そのため，学習が足りずスコアが落ちてしまったのではないかということが考えられる．また，STAIR Caption の論文[17]では，BLEU-1で0.763，BLEU-4で0.385のスコアが得られたという結果があるので，デフォルト・統一後ともにこれほどスコアが低くなるということは，学習，キャプションの生成，検証などのどこかにミスが有った可能性も考えなければならない．スコアの計算にはMSCOCO Caption Evaluation を利用した他，NLTKを用いてGLEUのスコアの実装を行っている．GLEUの実装に対するテストとして行った，1文単位でのスコアの計算ではSTAIR Caption の論文に見られるようなスコアが得られていたため，検証の時点では問題は無いと考えられるが，学習，キャプションの生成，検証それぞれもう一度確認する必要がある．

第10章

結論

本論文では，画像キャプション生成の複数形表現に注目することで，生成されるキャプションの品質の向上を行った．具体的には，訓練データのキャプションの複数形表現を統一した．英語キャプションの生成において，MSCOCOデータセットを訓練データに用いた実験では，人手による主観的評価および自動評価手法を用いた客観的な評価のどちらでも生成されるキャプションの品質の向上が確認できた．しかし，日本語キャプションの生成において，STAIRCaption データセットを訓練データに用いた実験では，人手による主観的な評価では複数形が用いられるような画像に対しては品質の向上が認められたものの，それ以外ではあまり効果が認められず，自動評価手法を用いた客観的な評価では，一部の手法を除き，品質の向上が認められなかった．

英語キャプションに対する複数形表現の統一の効果が認められた一方で，日本語キャプションに対する複数形表現の統一の効果は一部を除き確認することが出来なかった．しかし，日本語キャプションに対する複数形表現の統一は効果がないと断定することは難しい．なぜなら，画像キャプション生成において，間違ったキャプションを生成した場合，その原因がどこにあるのかを特定するのは容易ではない．そのため，日本語キャプション生成における複数形表現の統一について効果がないと判断するには，複数形表現の統一に問題があったのか，それとも学習や生成の段階でどこかに問題が有ったのかをきちんと検証しなければならない．

改めて，本論文では複数形表現の統一手法の提起及び実装と文章生成において一般的な自動評価手法の利用と人手による評価を行った．英語における複数形表現の統一では想定していたとおりの文章の品質の向上が認められたが，日本語における複数形表現の統一では，一部を除き，想定通りの結果が得られなかったため，今後は日本語キャプション生成の実験において統一手法に問題はなかったか，学習は十分だったかなど，原因について調査していく必要がある．

謝辞

本研究を進めるにあたり，熱心にご指導いただいた指導教員の新納教授に感謝いたします．また，多くのご指摘をいただきました自然言語処理研究室の皆様に感謝します．

参考文献

[1] Stair captions: Constructing a large-scale japanese image caption dataset. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 417–421, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016.

[3] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72. Association for Computational Linguistics, 2005.

[4] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, Vol. abs/1504.00325, , 2015.

[5] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *European Conference on Computer Vision*, pp. 15–29. Springer, 2010.

[6] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3128–3137, 2015.

[7] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324, 1998.

[8] C.-Y. LIN. Rouge : A package for automatic evaluation of summaries. *Proc. Workshop on Text Summarization Branches Out, Post Conference Workshop of ACL 2004*, 2004.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755. Springer, 2014.

[10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method

for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

[11] Helmut Schmid. Improvements in part-of-speech tagging with an application to german. In *In proceedings of the acl sigdat-workshop*. Citeseer, 1995.

[12] Helmut Schmid. Probabilistic part-ofispeech tagging using decision trees. In *New methods in language processing*, p. 154. Routledge, 2013.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[14] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164, 2015.

[15] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, Vol. abs/1609.08144, , 2016.

[16] Tsung-Yi Lin Ramakrishna Vedantam Xinlei Chen, Hao Fang. Microsoft coco caption evaluation. In *http://github.com/tylin/coco-caption*, 2015.

[17] Yuya Yoshikawa. Image captioning with convolutional neural networks trained on mscoco and stair captions (japanese). In *https://github.com/yuyay/chainer_nic*, 2018.

[18] 西友佑, 新納浩幸, 古宮嘉那子, 佐々木稔. 画像キャプション生成における複数形表現の統一. 言語処理学会 第23回年次大会 発表論文集(2017年3月), pp. 194–197, 2017.

[19] 奈良先端科学技術大学院大学情報科学研究科. Chasen. *http://chasen.naist.jp/hiki/ChaSen/*, 2007.

[20] 拓工藤, 薫山本, 裕治松本. Conditional random fieldsを用いた日本語形態素解析. 情報処理学会研究報告自然言語処理（NL）, Vol. 2004, No. 47, pp. 89–96, may 2004.

[21] 松本裕治. 日本語形態素解析システムJUMAN使用説明書, 1994.