2H2-2

シーングラフを用いた質問文生成によるデータ拡張の手法

横田 匡史　　中山 英樹
Masashi Yokota　　Hideki Nakayama

東京大学 大学院情報理工学系研究科
Graduate School of Information and Science and Technology, The University of Tokyo

Recently, neural networks have been successfully applied to computer vision and natural language processing. Image Question Answering is one of the most famous tasks which uses both visual and language information. While there are many researches which achieve state-of-the-art performance, they need a large amount of data. However, it is difficult to make dataset because of high cost to collect question answering data. Therefore, we propose the framework to augment dataset by generating QA pairs using scene graphs. We evaluate our method on Visual Genome dataset.

1. はじめに

近年，DeepLearning の研究が，自然言語処理や画像分類タスクなどの分野において目覚ましい成果を挙げており注目を浴びている．DeepLearning の発展により注目されている研究分野の一つに画像質問応答がある．これは，画像とそれに関した質問文を入力とし機械が与えられた質問文に対して正確に解答する事を目標とした研究分野である．しかし，画像質問応答を解くためのモデルの学習には，莫大な数のデータが必要となる．そのため，このデータの作成には時間面や費用面において大きなコストがかかり問題となっている．そこで，画像中のある性質をトップダウンに選び，それが答えとなるような質問文を逆生成する事ができれば，データの自動生成が可能であると期待される．

本研究ではデータの自動生成にシーングラフを用いた手法を提案する．シーングラフとは画像内の物体同士の関係をグラフ化したものであるため画像内にある物体の関係性を正確に表現している．そのため，物体同士の関係性を問うような質問文を生成する際に有効であると考えられる．そこで，本研究では Visual Genome[Krishna 16] の画像とシーングラフ，質問・解答ペアを用いて学習を行う．この時，図 1 のように画像とシーングラフ，解答を入力として質問文を生成するモデルを学習する．質問生成時においては，多様な質問文を生成するため，まず初めにシーングラフから解答抽出を行う．その後，画像とシーングラフ，抽出した解答から新しく質問文を生成することで，データ拡張を行っている．

提案手法の評価として，Visual Genome を用いて実験を行った．実験は，データ拡張したデータセットを用いて学習した質問応答モデルとオリジナルデータセットのみで学習した質問応答モデルで比較した．

2. 関連研究

2.1 画像質問応答

画像質問応答の代表的なデータセットに，Visual Question Answering (VQA) [Agrawal 15] がある．これは，画像と質問文とその答えのセットから構成されている．また，この VQA データセットには実際の写真を用いた実写画像データセットと自動でイラスト画像生成を行いデータセットを作成した抽象画像データセットの 2 種類がある．実写画像データセットにおいても抽象画像データセットにおいても質問文とその解答のペアの作成は，人力でされているためデータセット作成に要するコストが非常に高く問題となっている．

連絡先: 横田匡史, 東京大学大学院情報理工学系研究科

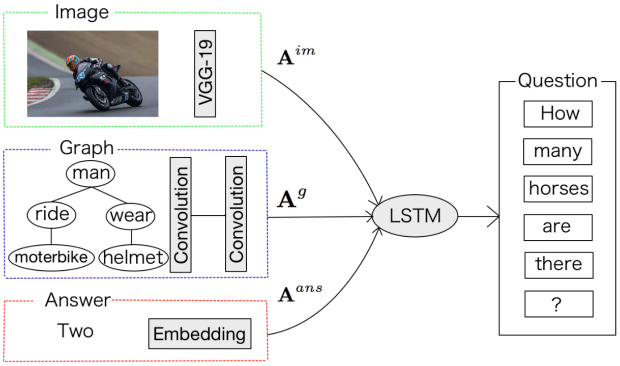

図 1: 質問生成モデルの構成．画像特徴量は VGG-19 における最終層の畳み込み層の出力を用いており，シーングラフは 2 層の畳み込み層によって特徴量を得る．また，入力の解答は Embedding によりベクトル化している．これらによって得られた特徴量を LSTM の初期値とし，質問文を出力している．

2.2 Visual Genome

Visual Genome [Krishna 16] は，画像に対してシーングラフと質問文とその解答が紐付いたデータセットである．この Visual Genome は画像を 108,000 枚持つ．また，各画像に対し平均して 21 個の物体情報，18 個の物体に紐付いた色・形などを示す属性情報，18 個の'in' や'by' などの物体間の関係情報を持ち，さらに 16 個の質問文とその解答のペア持つ．また，シーングラフは，物体情報，物体に紐付いた属性情報，物体間の関係情報をグラフの頂点とし，辺情報は何も持たないグラフとして扱っている．本研究では，このシーングラフを用いる事で画像内の物体同士の関係情報を利用し，より良い質問文を生成できることを期待している．

2.3 質問生成

質問生成に関する研究として，人間が画像を見た時に関心を持ちそうな事柄に関する質問文を生成するタスク [Mostafazadeh 16] がある．例えば，マラソンで人が走っている写真を入力として与えられた時に'How many runners are there?' という質問ではなく，'Who won the race?' と言ったような人間が関心を持ちそうな質問を生成することを目標としている．しかし，このタスクはデータ拡張などを考慮したタスクではなく，画像から質問文を生成するのみであるためデータ拡張などに応用することは難しい．一方で，自然言語処理にお

いては質問生成モデルを用いて質問応答タスクのデータ拡張を行っている研究 [Yang 17] がある．この研究では，質問生成モデルと質問応答モデルを強化学習を用いて学習させることで，質問応答モデルの性能改善を達成している．

3. 提案手法

3.1 質問生成モデル

本節では，質問生成モデルの構成について説明する．このモデルでは，画像，シーングラフ，質問文とその解答の 4 種類のデータを用いて学習を行う．この時，画像とシーングラフ，解答の 3 つを入力とし，出力を質問文とする．ここで，画像は 224×224 のカラー画像，質問文は一文とする．また，解答は'apple' などの単語もしくは'playing tennis' などの小さなフレーズでクラスを分け，入力時はワンホットベクトルとして扱う．また，シーングラフは PATCHY-SAN [Niepert 16] を用いて，テンソルに変換し，それをモデルの入力としている．PATCHY-SAN を用いる事で性質の近い頂点同士が近くに配置されたテンソルを作ることが可能になる．ここで，画像とシーングラフ，解答はそれぞれ $\mathbf{d}^{im} \in \mathbb{R}^{224\times224\times3}, \mathbf{d}^{g} \in \mathbb{R}^{w\times k\times D}, \mathbf{d}^{ans} \in \mathbb{R}^{V}$ とする．この時，w,k は PATCHY-SAN におけるテンソルを作る際の頂点数，取得する隣接頂点数を表しており，さらに D, V は，頂点属性のサイズと解答候補数をそれぞれ示している．

次に各入力のエンコーダについて説明を行う．各入力は，以下のように処理を行う．

$$\mathbf{A}^{im} = F_{im}(\mathbf{d}^{im}), \tag{1}$$

$$\mathbf{A}^{g} = F_{g}(\mathbf{d}^{g}), \tag{2}$$

$$\mathbf{A}^{ans} = \mathbf{W}_{ans}\mathbf{d}^{ans}. \tag{3}$$

このとき，$\mathbf{A}^{im} = \{\mathbf{a}^{im}_0, ..., \mathbf{a}^{im}_{S_{im}}\}, \mathbf{a}^{im}_i \in \mathbb{R}^{E_{im}}$，$\mathbf{A}^{g} = \{\mathbf{a}^{g}_0, ..., \mathbf{a}^{g}_{S_g}\}, \mathbf{a}^{g}_i \in \mathbb{R}^{E_g}, \mathbf{A}^{ans} \in \mathbb{R}^{E_{ans}}, \mathbf{W}_{ans} \in R^{E_{ans}\times\mathbf{V}}$ である．S_{im} と S_g はそれぞれ画像またはグラフから抽出した特徴量テンソルの位置数または頂点数を表し，E_{im} と E_g は画像またはグラフから抽出した特徴量のチャネル数である．さらに，E_{ans} は，解答を Embedding した後の特徴量ベクトルの次元数を示す．また，F_{im} と F_g は畳み込み層である．

次にデコーダに関して説明を行う．デコーダには Long Short Term Memory (LSTM) [Hochreiter 97] を用いて質問文の生成を行っている．この時，LSTM におけるメモリセルの状態の初期値 c_0 と隠れ層の状態の初期値 h_0 は，それぞれ以下のように設定する．

$$\mathbf{c}^{im}_0 = \frac{1}{S_{im}}\sum_{k=1}^{S_{im}}\mathbf{a}^{im}_k, \tag{4}$$

$$\mathbf{c}^{g}_0 = \frac{1}{S_{g}}\sum_{k=1}^{S_{g}}\mathbf{a}^{g}_k, \tag{5}$$

$$\mathbf{c}_0 = F_{init,c}([\mathbf{c}^{im}_0; \mathbf{c}^{g}_0; \mathbf{A}^{ans}]) \in \mathbb{R}^m, \tag{6}$$

$$\mathbf{h}_0 = F_{init,h}([\mathbf{c}^{im}_0; \mathbf{c}^{g}_0; \mathbf{A}^{ans}]) \in \mathbb{R}^m. \tag{7}$$

m は LSTM の次元数を表している．また，$F_{init,c}$ と $F_{init,h}$ は Multi Layer Perceptron (MLP) であり，; はベクトルの結合を表している．

次にデコーダのアテンションに関して説明する．タイムステップ t における，画像またはグラフの特徴量に対して，アテンションを行う際の位置ごとの重み $\alpha^{im}_t, \alpha^{g}_t$ は以下のように求める．

$$e^{im}_{ti} = (\mathbf{a}^{im}_{ti})^T\mathbf{W}_{im}\mathbf{h}^{im}_{t-1}, \tag{8}$$

$$e^{g}_{ti} = (\mathbf{a}^{g}_{ti})^T\mathbf{W}_{g}\mathbf{h}^{g}_{t-1}, \tag{9}$$

$$\alpha^{im}_t = \frac{e^{im}_{ti}}{\sum_{k=1}^{S_{im}} e^{im}_{tk}}, \tag{10}$$

$$\alpha^{g}_t = \frac{e^{g}_{ti}}{\sum_{k=1}^{S_{g}} e^{g}_{tk}}. \tag{11}$$

ここで，$\mathbf{W}_{im} \in \mathbb{R}^{E_{im}\times E_{ans}}$ であり，T は転置を示している．

さらに式 (10) と式 (11) を用いて，画像とグラフの文脈ベクトル $\mathbf{z}^{im}_t, \mathbf{z}^{g}_t$ を求める．

$$\mathbf{z}^{im}_t = \sum_{j}^{S_{im}} \alpha^{im}_j \mathbf{a}^{im}_j, \tag{12}$$

$$\mathbf{z}^{g}_t = \sum_{j}^{S_{g}} \alpha^{g}_j \mathbf{a}^{g}_j. \tag{13}$$

そして，先の文脈ベクトルを用いて LSTM の入力とする．

$$\begin{pmatrix}\mathbf{i}_t\\ \mathbf{f}_t\\ \mathbf{o}_t\\ \mathbf{g}_t\end{pmatrix} = \begin{pmatrix}\sigma\\ \sigma\\ \sigma\\ \tanh\end{pmatrix}\mathbf{B}_{V+m+E_{im}+E_g,m}\begin{pmatrix}\mathbf{E}\mathbf{y}_{t-1}\\ \mathbf{h}_{t-1}\\ \mathbf{z}^{im}_t\\ \mathbf{z}^{g}_t\end{pmatrix}, \tag{14}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, \tag{15}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t). \tag{16}$$

$\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t, \mathbf{g}_t, \mathbf{h}_t$ は，それぞれ入力ゲート，忘却ゲート，出力ゲート，メモリセル，隠れ層の状態を示す．また，σ はシグモイド関数を示し，$\mathbf{B}_{x,y}$ は $\mathbb{R}^x \rightarrow \mathbb{R}^y$ と変換するようなパラメータを持った変数であり，学習により最適化される．

出力となる質問文の単語の確率分布は以下のように定める．

$$p(y_t; \mathbf{A}^{im}, \mathbf{A}^{g}) = \mathtt{softmax}(\mathbf{M}f_{out}(\mathbf{y}_{t-1}; \mathbf{h}_t; \mathbf{z}^{im}_t; \mathbf{z}^{g}_t)). \tag{17}$$

ここで，f_{out} は MLP であり，$\mathbf{M}$ は $\mathbb{R}^{4m\times V}$ のパラメータである．

3.2 質問生成時における解答抽出方法

この節では質問生成時の解答抽出方法に関して説明する．質問生成時は，3.1 で説明した質問生成モデルの学習済みモデルを用いる．また，画像とそれに紐づくシーングラフは，既存の VisualGenome の学習用データから選択したものを用いる．そして，質問生成時における解答抽出は，シーングラフを用いて行う．この時，なるべく多様な質問・解答ペアでデータ拡張を行ったデータを用いた方が質問応答モデルの性能を向上させられることが期待される．そこで，シーングラフからの解答抽出方法として，シーングラフから解答候補を全列挙し，そこからランダムに一つ解答候補を選ぶ方法をとった．具体的には，まず質問生成したい画像に対応するシーングラフから全ての頂点ラベルを抽出し，それらの集合を L_s とする．次に，頂点のラベルとそれに対して隣接している頂点のラベルを結合させた bi-gram の集合を L_c とする．さらに全ての頂点ラベルから同じラベルを持つものを全てカウントし，各ラベル毎にカウントした集合を L_{cnt} とする．このように用意された L_s，L_c，L_{cnt} で構成される解答候補の集合からランダムに解答を抽出

表 1: データ拡張の有無における質問タイプ別の質問分類モデルの精度の変化．ベースラインは DeeperLSTMQ を用いている．また，other はどの質問タイプにも当てはまらない質問を表し，all は全テストデータにおける accuracy である．

	accuracy[%]	
質問タイプ	ベースライン	ベースライン (+ 提案手法)
where	26.16	25.24
how many	39.13	37.88
what	41.58	40.04
when	87.83	87.25
other	43.95	39.26
all	43.88	41.45

表 2: 質問生成モデルへの入力の違いによる性能比較．G, I, A はそれぞれグラフと画像，解答を示している．例えば GIA はグラフと画像，解答を入力とした質問生成モデルを学習し，そのモデルを用いて学習用データをデータ拡張した時の質問応答モデルの精度を示している．

Model	accuracy[%]
GIA	41.45
IA	41.11
GA	39.71

しそれを入力の解答として，質問生成を行う．以上により，得られた画像とシーングラフ，解答からビームサーチを用いて質問文生成を行う．

3.3 質問応答モデルの学習に関する工夫

この節では，3.2 で説明した方法でデータ拡張したデータで学習する際の質問応答モデルの学習方法について説明する．質問応答モデルとは画像と質問文を入力として，質問文の解答を出力するモデルである．この時，生成された質問文は完全に自然な文章ではないため，上手く精度が向上しない場合がある．そこで [Yang 17] では，生成したデータと既存のデータをタグを導入することで別ドメインとして学習させ，性能向上をしている．本研究でも，[Yang 17] と同様に，生成したデータを別ドメインのデータとして学習させ，双方に共通の知識を学習させることで性能向上を図る．具体的なタグの導入方法として，質問文の入力時に，既存の質問文には，質問文の最後の単語に”true_tag”を挿入し，生成した質問文の最後尾には，”gen_tag”を挿入し学習を行っている．

4. 実験

4.1 実験設定と実験の流れ

本研究では，VisualGenome を用いて実験を行った．VisualGenome ではテスト用データが指定されていないためランダムにデータセットの 90%を学習用データとして選び，残りをテスト用データとして学習と評価を行った．また，質問文とシーングラフの語彙は出現数が学習用データにおいて 20 回を超えたもののみとした．その結果，それぞれの語彙数は 4117 個，8073 個となった．さらに，解答は頻出する上位 3000 種類を用いた．これはデータセットのおよそ 71.5%を占めている．

また，PATCHY-SAN を用いてグラフからテンソルを作成する際のパラメータ w と k の値をそれぞれ 90，5 としている．また，グラフにおける頂点属性は GloVe [Pennington 14] の値を用いており，さらにファインチューニングをしている．画像処理部分では，VGG[Simonyan 15] の畳み込み層の最終層を特徴量として用いている．また，VGG のパラメータは固定しファインチューニングは行っていない．

表 3: GIA のテストデータにおける Bleu 値．質問生成モデル GIA を用いてテスト用データから生成した質問文と元の質問文から BLEU 値を計算した結果を示している．

model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
GIA	0.0934	0.0707	0.0459	0.0328

本研究では以下の流れで実験を行った．

1. 学習用データで質問生成モデルを学習
2. 学習済み質問生成モデルを用いて学習用データのデータ拡張
3. データ拡張したデータセットを用いて質問応答モデルを学習

4.2 モデル

質問生成モデルでは，最適化アルゴリズムを Adam，学習率を 1.0×10^{-4}，バッチサイズを 200 として学習を行った．ここで，G, I, A はそれぞれグラフと画像，解答とする．本研究における質問生成モデルの比較として，GIA と IA，GA の計 3 つのモデルを用いて比較を行った．GIA，IA，GA はそれぞれ入力を (画像・グラフ・解答)，(画像・解答)，(グラフ・解答) として学習した質問生成モデルである．また，質問応答モデルは DeeperLSTMQ[Agrawal 15] を用いた．この時のハイパーパラメータは [Agrawal 15] と同様である．なお，本研究における全ての実験条件において質問応答モデルの条件は一定とした．

4.3 実験結果と考察

まず，生成した質問文をデータに加えたときと加えなかった時の質問応答モデルの比較を行った結果を表 1 に示す．表 1 では，質問タイプ別に accuracy を算出している．where, how many, what, when は，それぞれから始める質問での accuracy を示している．また，other はこれらのどれにも当てはまらない質問の accuracy であり，all は全テスト用データにおける accyracy を示している．これより，生成したデータを追加した際に精度が低くなってしまう事が確認できる．

次に，GIA と IA，GA の計 3 つのモデルからそれぞれ生成されたデータを追加して質問応答モデルを学習した時の結果を表 2 に示す．表 2 より，GIA において最も良い結果が得られた事が確認できる．また，生成された質問の質を確認するために GIA において，テスト用データを用いて生成した質問文と元々付与されている質問文から BLEU 値を計算し，比較した結果を表 3 に示す．この時の入力となる解答はテスト用データのものを用い，シーングラフから解答抽出は行っていない．加えて，表 3 を評価する際に実際に生成された質問文と元の質問文の例を図 2 に示す．以上より，一部では図 2a のように上手く質問文が生成されている場合もあるが表 3 から見ると，全体として生成された質問文の質が低い事が推測される．上手く質問文が生成できなかった例として，図 2b が挙げられる．図 2b において生成された’on the man’ という表現はデータセット中に頻出する表現であった．このように，人がいない写真に対して，’on the man’ という表現を出力している例は他にも多く見られた．その他にも，データセット中に頻出する’the pattern of’ や’the right to’ などの表現が入力情報に対して関係ない場合であっても出力されている場合が多く存在した．そ

Original: When was the picture taken?
Generated: What time of day was this photo taken?

(a) 質問文を上手く生成できた例

Original: What is gray and white?
Generated: What is on the man in front of the picture?

(b) 質問文を上手く生成できなかった例

図 2: テスト用データにおける元の質問文と生成した質問文の例．質問生成モデル *GI* でテスト用データの画像とグラフ，解答のペアから質問文を生成した時の質問文とテスト用データにあるオリジナルの質問文である．

のため，エンコーダが写真やグラフから情報を上手く抽出できていない可能性もあるが，デコーダがこれらの表現を過学習してしまっている可能性もある．以上の実験結果より，質問文の質の低さが表 1 のような結果になった原因とて考えられる．また，質問文の質が低い原因として以下のことが考察できる．

(a) 質問文を生成するのに必要な情報がエンコーダから得られていない

(b) データセット中に頻出する表現をデコーダが過学習してしまっている

これらの対策として，(a) に関しては，現状は PATCHY-SAN を用いてシーングラフからテンソルを取得しているが，グラフが複雑な時には上手く特徴を取得できていない可能性がある．そこで，Visual Genome のシーングラフから SVO などの関係を持ったトリプレットを取得し，独立した部分シーングラフから特徴量を得ることで，質問生成しやすい特徴を得やすいのではないかと考えられる．(b) に関しては，損失関数に適切な正則化項を導入することで，過学習をしにくくさせる事が考えられる．また，(a) と (b) の問題を対処した後，さらに質問文の質を向上させるために [Yang 17] などのように強化学習を用いて生成する質問文の質を向上させていく事が考えられる．[Yang 17] では，生成した質問文を質問応答モデルに入力した際にその正解率を報酬として，強化学習させることで，質問応答モデルが答えられるような質問文を生成するように質問生成モデルが学習される．本研究においても同じような手法で質問生成モデルの性能改善は可能であると考えられる．

5. おわりに

Visual Genome を用いて，質問応答モデルのためのデータ拡張に関する研究を行ってきたが，好ましい結果を得る事ができなかった．その問題点として，生成した質問文の質が低いためだと考えられる．この問題に対する今後の改善案として，トリプレットの部分グラフから学習する，または損失関数に適切な正則化項を導入するなどが考えられる．また，さらなる質問文の質の向上のために強化学習を用いて質問生成モデルの改善を行うことが挙げられる．

一方で，本研究で提案したシーングラフからの質問生成と解答抽出のフレームワークは，有益だと考えられる．なぜならば，任意の質問文を生成させることが可能であるからである．今回は生成した質問文を更新せずに質問応答モデルの学習を行った．しかし，質問応答モデルの精度に応じて，生成する質問文を変化させ，データ拡張により追加したデータを更新していく事が理想である．本研究での解答抽出方法で，求める質問文になるような解答を優先して選択させることで，質問文を生成させることが可能である．それにより，汎用的な質問応答モデルを学習させることが可能であると考えられる．そのため，今後は質問生成モデルの性能を改善するのに加え，質問応答モデルの正解率に応じて生成する質問文を変化させる仕組みの開発もしていきたい．

参考文献

[Agrawal 15] Agrawal, A., Lu, J., Antol, S., Mitchell, M., Zitnick, C. L., Batra, D., and Parikh, D.: VQA: Visual Question Answering, *In Proc. of ICCV* (2015)

[Hochreiter 97] Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural Computation, 9(8):17351780* (1997)

[Krishna 16] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., Bernstein, M. S., and Fei-Fei, L.: Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations, *arXiv preprint arXiv:1602.07332* (2016)

[Mostafazadeh 16] Mostafazadeh, N., Misra, I., Devlin, J., Mitchell, M., He, X., and Vanderwende, L.: Generating Natural Questions About an Image, *Annual Meeting of the Association for Computational Linguistics* (2016)

[Niepert 16] Niepert, M., Ahmed, M., and Kutzkov, K.: Learning Convolutional Neural Networks for Graphs, *In Proc. of ICML* (2016)

[Pennington 14] Pennington, J., Socher, R., and Manning, C. D.: GloVe: Global Vectors for Word Representation, *In Proc. of EMNLP* (2014)

[Simonyan 15] Simonyan, K. and Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, *In Proc. of ICLR* (2015)

[Yang 17] Yang, Z., Hu, J., Salakhutdinov, R., and Cohen, W. W.: Semi-Supervised QA with Generative Domain-Adaptive Nets, *arXiv preprint arXiv:1702.02206* (2017)