AttnGAN における DAMSM の Attention に対する考察

1　はじめに

近年の人工知能の発達は目覚ましく，その研究分野は創作の領域にまで及んでいる．本稿では，創作能力を実現する人工知能手法として，テキストからの画像生成を目的とする AttnGAN[1] に着目する．

今回は AttnGAN における DAMSM のテキストエンコーダに関して考察し，大規模コーパスで事前学習済みの自然言語処理における Embedding 手法を導入することで， Attention の精度向上を目指す．

提案手法と従来手法で得られた Attention に対して，著者が作成した比較検証用データとの類似度を，SSIM に基づいて比較し，Attention の精度を評価する．

2　要素技術

2.1　AttnGAN

Generative Adversarial Networks (GAN) [2] とは，Generator と Discriminator という 2 つのネットワークを対立的に学習させる生成モデルである．

その派生である Attentional Generative Adversarial Networks (AttnGAN) は，段階的な GAN 構造と Attention 機構を導入し，テキストからの画像生成を目的としたモデルである．

AttnGAN は (1) 式による i 段階目の Discriminator での損失関数 L_{D_i} と (2) 式による Generator での損失関数 L_G で定義される．L_{DAMSM} は 2.2 章で後述する DAMSM モデルによる Attention に関する損失関数である．図 1 に AttnGAN のモデル概要図を示す．

$$
\begin{aligned}
L_{D_i} = &-\frac{1}{2}\mathbb{E}_{x_i\sim p_{\mathrm{data}}}[\log D_i(x_i)] \\
&-\frac{1}{2}\mathbb{E}_{\hat{x}_i\sim p_{G_i}}[\log\left(1-D_i(\hat{x}_i)\right)] \\
&-\frac{1}{2}\mathbb{E}_{x_i\sim p_{\mathrm{data}}}[\log D_i(x_i, c)] \\
&-\frac{1}{2}\mathbb{E}_{\hat{x}_i\sim p_{G_i}}[\log\left(1-D_i(\hat{x}_i, c)\right)]
\end{aligned}
\tag{1}
$$

$$
\begin{aligned}
L_G = \sum_{i=1}^{m}\langle &-\frac{1}{2}\mathbb{E}_{\hat{x}_i\sim p_{G_i}}[\log D_i(\hat{x}_i)] \\
&-\frac{1}{2}\mathbb{E}_{\hat{x}_i\sim p_{G_i}}[\log D_i(\hat{x}_i, c)]\rangle \\
&+\lambda L_{\mathrm{DAMSM}}
\end{aligned}
\tag{2}
$$

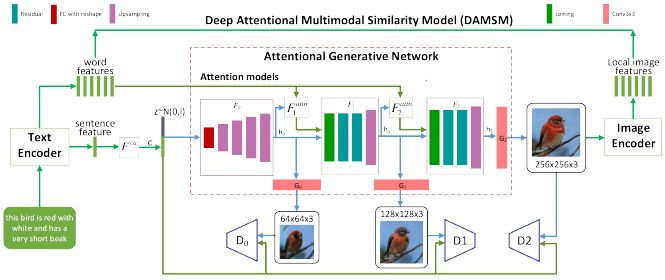


図 1: AttnGAN のモデル概要 [1]

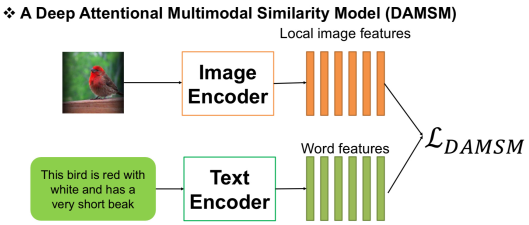


図 2: DAMSM の概要図 [1]

2.2　DAMSM

Deep Attentional Multimodal Similarity Model (DAMSM) は，画像の各部分領域とテキスト中の各単語をそれぞれ分散表現化し，得られた特徴量ベクトルによるマッチングスコアから L_{DAMSM} を算出するモデルである．図 2 に DAMSM のモデル概要図を示す．

2.2.1　テキストエンコーダ

単語 Embedding を入力として，Bi-direcrional Long Short Term Memory (双方向 LSTM) の中間層における出力を各単語の意味空間ベクトル $\boldsymbol{e}\in\mathbb{R}^{D\times T}$ として用いる．このとき， D は単語ベクトルの次元数， T は文章中の単語数である．このとき， $\boldsymbol{e}_i\in\mathbb{R}^{D}$ が i 番目の単語の特徴量ベクトルとなる．また，最終層における出力を文章全体の特徴量を表す $\bar{\boldsymbol{e}}\in\mathbb{R}^{D}$ とする．

2.2.2　画像エンコーダ

画像を入力として，Google による画像識別モデルである Inception-v3[3] 内の model_6e における出力を，各部分領域の意味ベクトル $\boldsymbol{f}\in\mathbb{R}^{768\times 289}$ として用いる．ここで，768 は部分領域の特徴量ベクトルの次元数，289($=17\times 17$) は部分領域の総数である．また，最終

層における出力を画像全体の特徴量を表す $\bar{\boldsymbol{f}} \in \mathbb{R}^{2048}$ として用いる．

全結合層 $\boldsymbol{W}$ により，画像の特徴量ベクトル $\boldsymbol{f}$ とテキストの特徴量ベクトルとの次元数を等しくし，画像の各部分領域の特徴量ベクトル $\boldsymbol{v} = \boldsymbol{W} \cdot \boldsymbol{f}$ および画像全体の特徴量ベクトル $\bar{\boldsymbol{v}} = \bar{\boldsymbol{W}} \cdot \bar{\boldsymbol{f}}$ を得る．ここで，$\boldsymbol{v} \in \mathbb{R}^{D \times 289}$ ， $\bar{\boldsymbol{v}} \in \mathbb{R}^{D}$ である．このとき， $\boldsymbol{v}_i \in \mathbb{R}^{D}$ を i 番目の部分領域の特徴量ベクトルとし， $\bar{\boldsymbol{v}} \in \mathbb{R}^{D}$ を画像全体の特徴量ベクトルとする．

2.2.3 マッチングスコア

画像内の各部分領域の特徴量とテキスト内の各単語の特徴量の対応を，(3) 式のようにマッチング $\boldsymbol{s}$ として

$$\boldsymbol{s} = \boldsymbol{e}^{\top} \boldsymbol{v} \tag{3}$$

と定義する．ここで，$\boldsymbol{s} \in \mathbb{R}^{T \times 289}$ であり，$\boldsymbol{s}_{i,\,j} \in \mathbb{R}^{D}$ は i 番目の単語の特徴量と j 番目の部分領域の特徴量のマッチングである．(4) 式のように，マッチング $\boldsymbol{s}$ を正規化することによって，$\bar{\boldsymbol{s}}_{i,\,j}$ を得る．

$$\bar{\boldsymbol{s}}_{i,\,j} = \frac{\exp(\boldsymbol{s}_{i,\,j})}{\sum_{k=0}^{T-1} \exp(\boldsymbol{s}_{k,\,j})} \tag{4}$$

次に，i 番目の単語と画像の各部分領域の関係性を動的に表す領域コンテクストベクトル $\boldsymbol{c}_i$ を (5) 式として定義する．ここで γ_1 は Attention のかかり方の配分を決定する温度パラメータの逆数である．

$$\boldsymbol{c}_i = \sum_{j=0}^{288} \frac{\gamma_1 \exp(\bar{s}_{i,\,j})}{\sum_{k=0}^{288} \gamma_1 \exp(\bar{s}_{k,\,j})} \boldsymbol{v}_j \tag{5}$$

$\boldsymbol{c}_i$ と $\boldsymbol{e}_i$ のコサイン類似度から， i 番目の単語と画像全体の関係を $R_w(\boldsymbol{c}_i, \boldsymbol{e}_i) = \frac{\boldsymbol{c}_i^{\top} \boldsymbol{e}_i}{\|\boldsymbol{c}_i\| \|\boldsymbol{e}_i\|}$ と表せる．これを用いて，画像 Q とキャプション D の単語レベルでのマッチングスコア $R_w(Q, D)$ を (6) 式と定義する．

$$R_w(Q, D) = \log \left(\sum_{i=1}^{T-1} \exp(\gamma_2 R(\boldsymbol{c}_i, \boldsymbol{e}_i)) \right)^{\frac{1}{\gamma_2}} \tag{6}$$

ここで， γ_2 は $R_w(\boldsymbol{c}_i, \boldsymbol{e}_i) = \frac{\boldsymbol{c}_i^{\top} \boldsymbol{e}_i}{\|\boldsymbol{c}_i\| \|\boldsymbol{e}_i\|}$ の重要度の拡張性を決定づける温度パラメータの逆数である．

また文章レベルでのマッチングスコア $R_s(Q, D)$ は (7) 式として定義する．

$$R_s(Q, D) = \frac{\bar{\boldsymbol{v}}^{\top} \bar{\boldsymbol{e}}}{\|\bar{\boldsymbol{v}}\| \|\bar{\boldsymbol{e}}\|} \tag{7}$$

2.2.4 DAMSM loss

バッチサイズ が M の時， $R_w(Q, D)$ と $R_s(Q, D)$ を用いて M 組の画像と単語のペア $\{(Q_i, D_i)\}_{i=1}^{M}$ において，キャプション D_i が画像 Q_i と正解マッチングとなる条件付き確率はそれぞれ (8) 式，(9) 式として求められる．ここで，γ_3 はスムーズ化係数である．

$$P_w(D_i|Q_i) = \frac{\exp(\gamma_3 R_w(Q_i, D_i))}{\sum_{j=1}^{M} \gamma_3 R_w(Q_j, D_j)} \tag{8}$$

$$P_s(D_i|Q_i) = \frac{\exp(\gamma_3 R_s(Q_i, D_i))}{\sum_{j=1}^{M} \gamma_3 R_s(Q_j, D_j)} \tag{9}$$

これらの条件付き確率の負の対数尤度を取ることによって，(10) 式のように損失関数 L_1^w ， L_1^s を得る．

$$L_1^w = -\sum_{i=1}^{M} \log P_w(D_i|Q_i),\ L_1^s = -\sum_{i=1}^{M} \log P_s(D_i|Q_i) \tag{10}$$

また，ベイズ推定により，画像 Q_i がキャプション D_i と正解マッチングとなる条件付き確率は，それぞれ (11) 式，(12) 式として求められる．

$$P_w(Q_i|D_i) = \frac{\exp(\gamma_3 R_w(D_i, Q_i))}{\sum_{j=1}^{M} \gamma_3 R_w(D_i, Q_i)} \tag{11}$$

$$P_s(Q_i|D_i) = \frac{\exp(\gamma_3 R_s(D_i, Q_i))}{\sum_{j=1}^{M} \gamma_3 R_s(D_i, Q_i)} \tag{12}$$

これらの条件付き確率の負の対数尤度を取ることによって，(13) 式のように損失関数 L_2^w ， L_2^s を得る．

$$L_2^w = -\sum_{i=1}^{M} \log P_w(Q_i|D_i),\ L_2^s = -\sum_{i=1}^{M} \log P_s(Q_i|D_i) \tag{13}$$

これらの損失関数の総和をもって，DAMSM loss を $L_{\mathrm{DAMSM}} = L_1^w + L_2^w + L_1^s + L_2^s$ として定義する．

2.3 自然言語処理における単語の Embedding モデルと手法

2.3.1 Word2Vec

predictive な手法で実用上現在最も広く知られている Word2Vec[4] は，大量のテキストデータを解析し，各単語の意味をベクトル表現化する手法である．周辺単語から対象後を推測する Continuous Bag of Words (CBOW) と，ある語から周辺単語を対象として推測する skip-gram の 2 つの方法が存在する．

Google により，Google News (約 1 千億語) をコーパスとして，300 万語の単語 Embedding を収録した事前学習済みモデルが公開されている．

2.3.2 GloVe

Global Vectors for Word Representation (GloVe)[5] とは、コーパスを集約して得られたグローバルな単語の共起行列から統計的に学習し、単語の分散表現を獲得するという教師なし学習モデルである．

スタンフォード大学により，Wikipedia2014 (約 60 億トークン) をコーパスとして，40 万語の単語 Embedding を収録した事前学習済みモデルが公開されている．

2.3.3 fastText

fastText[6] とは， Word2Vec の延長線上にあるモデルであり，単語以外にその構成要素を subword として分解していることが特徴である．単語を subword に分解し，活用による変化のない基幹部分のベクトルを扱うため，使用メモリを削減でき，学習速度も早く，未知語にも強いという長所があるとされる．

Wikipedia2017，UMBC ウェブベースコーパス，statmt.org ニュースデータセット (計約 160 億トークン) をコーパスとして，約 100 万語の単語 Embedding を収録した事前学習済みモデルが公開されている．

2.3.4 ConceptNet Numberbatch

ConceptNet Numberbatch[7] とは，機械学習で不足しがちな常識的知識の補填のために基礎的な知識をまとめたデータベースである ConceptNet と複数のモデル (GloVe，Word2Vec 等) とを組み合わせた手法である．

ConceptNet と Word2Vec，GloVe をもとに，OpenSubtitles 2016 (65 言語，約 170 億トークン) をコーパスとして，約 192 万語の単語 Embedding を収録した事前学習済みモデルが公開されている．

2.4 SSIM

Structural similarity (SSIM)[8] とは，2 枚の画像間の形状に関する類似指標の一種である．構造的類似性とも呼ばれ，輝度・コントラスト・構造の 3 要素をもとに類似性を測定する．各ピクセル単体間のみの輝度やコントラストを参照するだけでなく，周辺のピクセルとの平均，分散，共分散をとることにより，周囲との空間的な相関を取り込んで比較するため，人間の感性による類似度と近い値を算出できるとされている．

表 1: 実験で用いたパラメータ

train data	826
eval data	551
test data	25
batch size	25
max epoch	600
snapshot_interval	50
learning rate	0.0002
max_words_num	20
embedding_dim	300
γ_1	4.0
γ_2	5.0
γ_3	10.0

表 2: 実験結果

SSIM	平均値		最大値	
手法	LSTM	GRU	LSTM	GRU
従来手法	0.613	0.615	0.630	0.635
Word2Vec	0.599	0.590	0.631	0.620
GloVe	0.569	0.568	0.595	0.593
fastText	0.622	0.619	0.642	0.638
ConceptNet	0.619	0.627	0.632	0.643

(14) 式に SSIM の算出式を示す．ここで，x, y はウィンドウ，c_1, c_2 はアルゴリズムパラメータ k_1, k_2 とダイナミックレンジ L からなり， $c_1, c_2 = (k_1 L)^2, (k_2 L)^2$ である．本実験では推奨値のウィンドウサイズ 11×11，$k_1 = 0.01, k_2 = 0.03, L = 255$ を用いる．

$$\mathrm{SSIM} = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \qquad (14)$$

3 提案手法

本研究では DAMSM におけるテキストエンコーダに対し，大規模コーパスで事前学習済みの自然言語処理に基づく Embedding 手法の導入を提案する．

また，Attention に対する比較検証用データを人手により作成し，DAMSM による Attention との SSIM を測定することによって，Attention を定量的に評価し比較する．

表 3: テストデータのキャプション例

図 3a	this bird is black with red and has a very short beak.
図 4a	this bird is almost completely gray with a dark gray crown.
図 5a	the full brown and gold tail of the bird with a large crown.

表 4: 図 3a での “black” に対する Attention の評価

手法	LSTM	GRU
従来手法	0.047	0.071
Word2Vec	0.236	0.345
GloVe	0.257	0.362
fastText	0.228	0.278
ConceptNet	0.328	0.280

表 5: 図 4a での “gray” に対する Attention の評価

手法	LSTM	GRU
従来手法	0.261	0.208
Word2Vec	0.268	0.103
GloVe	0.196	0.187
fastText	0.375	0.364
ConceptNet	0.331	0.316

表 6: 図 5a での “brown” に対する Attention の評価

手法	LSTM	GRU
従来手法	0.263	0.370
Word2Vec	0.402	0.368
GloVe	0.380	0.380
fastText	0.382	0.436
ConceptNet	0.434	0.434

4 実験

4.1 実験

CUB 鳥画像データセット [9] から，25 種類の鳥の画像とキャプション群を使用する．今回使用した画像の総枚数は 1402 枚であり，1 枚の画像につき 10 個のキャプションがつけられている．

テスト用に test データとして各種類の鳥から 1 枚ずつ選抜し，比較検証のため test データのキャプションは 1 種類に固定した．図 3a，図 4a，図 5a にテスト用に CUB 鳥画像データセットから選抜した 25 枚の鳥画像から 3 例を示す．表 3 にそれぞれのキャプションを示す．図 3b，図 4b，図 5b にキャプション中の “black”，“gray” ， “brown” に対して人手で作成した Attention の比較検証用データセットを示す．

表 1 に本実験で用いたパラメータを示す．各パラメータは先行研究と同一の値を用いた．

4.2 結果と考察

表 2 に全体に対する Attention と比較検証用データセットとの SSIM の epoch600 までの平均と最大値の実験結果を示す．図 3 から図 5 に各手法によって得られた Attention と人手により作成した Attention を並べて示す．また，表 4 から表 6 に人手により作成した Attention の比較検証用データセットとの SSIM による各手法の評価を示す．

表 2 から，全体に対する Attention では，fastText もしくは ConceptNet の学習済みモデルを用いたものの精度が高いことが見られた．一方で，表 4 のように，色やパターン模様を示す名詞に対して，提案手法では，従来手法と比較して全体的に精度の向上が見られたが，特に Word2Vec もしくは GloVe において，その傾向が顕著であった．これは各手法による Attention のかかり方の傾向の違いから生じていると考えられる．

Attention を SSIM に基づいて評価・比較することにより，Word2Vec や GloVe では局所的に極端な勾配の大きい Attention を付与する傾向があり，fastText や ConceptNet では全体的に緩やかに勾配の小さい Attention を付与する傾向があることを定量的に測定することができた．

5 まとめと今後の課題

本研究では AttnGAN における DAMSM の Attention に関して考察し，大規模コーパスで事前学習済みの自然言語処理における Embedding 手法を導入するモデルを提案した．また，Attention に対して SSIM に基づいて定量的に評価することで，提案手法の有効性を確認することができた．

今後の課題としては，Attention に対する定量的評価指標の改良と確立，試行回数の増加による有意性の確認，大規模データセットへの適用，評価手法やテキストエンコーダのさらなる改良，画像エンコーダへの考察，生成画像の評価などがあげられる．

参考文献

[1] Qiuyuan Huang Han Zhang Zhe Gan Xiaolei Huang Xiaodong He Tao Xu, Pengchuan Zhang. Attngan: Fine-grained text to image generation

with attentional generative adversarial networks. 2018.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014.

[3] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, Vol. abs/1512.00567, , 2015.

[4] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc., 2013.

[5] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *In EMNLP*, 2014.

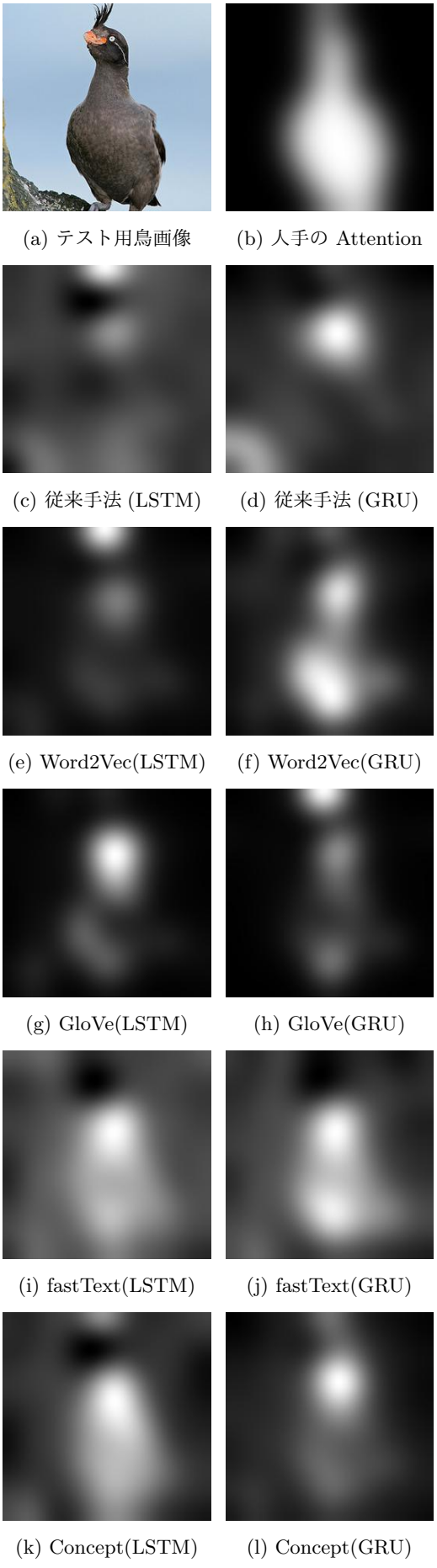
[6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135–146, 2017.

[7] Robyn Speer, Joshua Chin, and Catherine Havasi. ConceptNet 5.5: An open multilingual graph of general knowledge. pp. 4444–4451, 2017.

[8] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, Vol. 13, No. 4, pp. 600–612, April 2004.

[9] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. 2011.

図 3: “black” に対する Attention



(a) テスト用鳥画像 (b) 人手の Attention

(c) 従来手法 (LSTM) (d) 従来手法 (GRU)

(e) Word2Vec(LSTM) (f) Word2Vec(GRU)

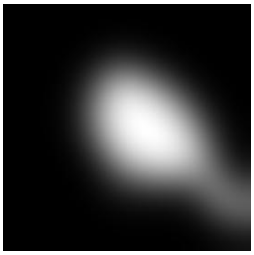
(g) GloVe(LSTM) (h) GloVe(GRU)

(i) fastText(LSTM) (j) fastText(GRU)

(k) Concept(LSTM) (l) Concept(GRU)

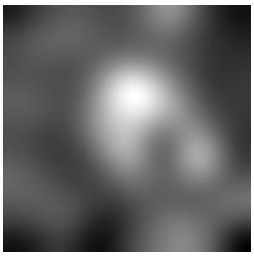
図 4: “gray” に対する Attention

(a) テスト用鳥画像



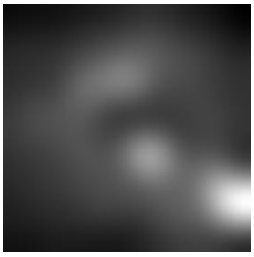
(b) 人手の Attention

(c) 従来手法 (LSTM)



(d) 従来手法 (GRU)

(e) Word2Vec(LSTM)



(f) Word2Vec(GRU)

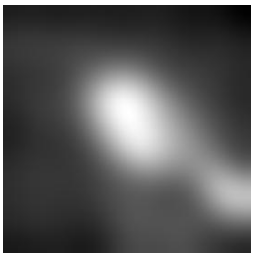
(g) GloVe(LSTM)

(h) GloVe(GRU)

(i) fastText(LSTM)

(j) fastText(GRU)

(k) Concept(LSTM)

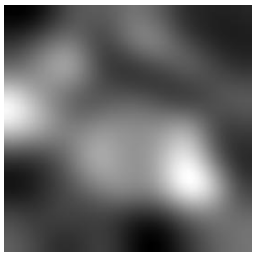


(l) Concept(GRU)

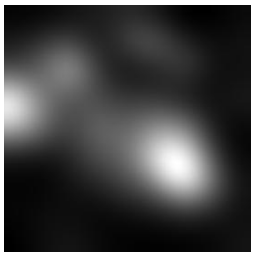
図 5: “brown” に対する Attention

(a) テスト用鳥画像

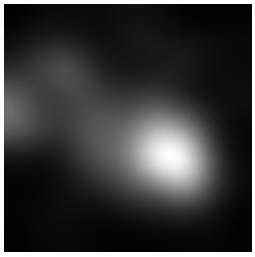
(b) 人手の Attention



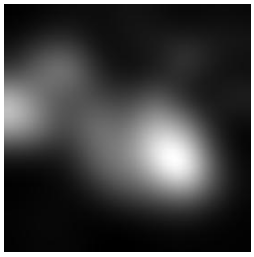
(c) 従来手法 (LSTM)



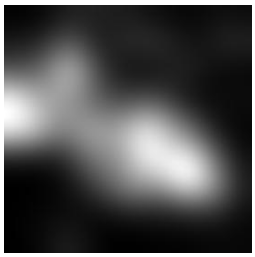
(d) 従来手法 (GRU)



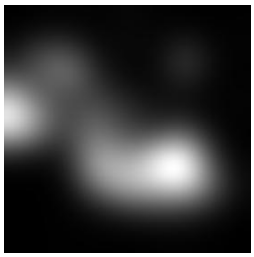
(e) Word2Vec(LSTM)



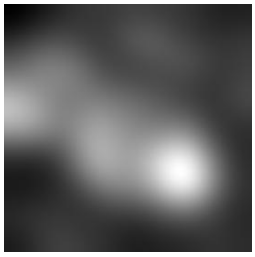
(f) Word2Vec(GRU)



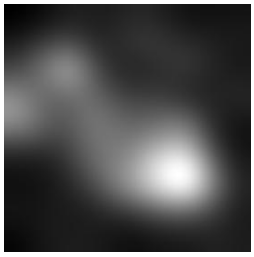
(g) GloVe(LSTM)



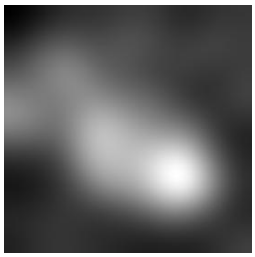
(h) GloVe(GRU)



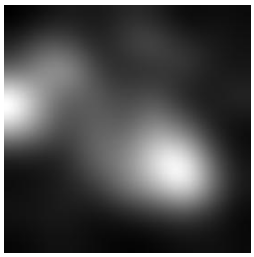
(i) fastText(LSTM)



(j) fastText(GRU)



(k) Concept(LSTM)



(l) Concept(GRU)