Visually Grounded Paraphrase Extraction

Chenhui Chu,[1] **Mayu Otani**[2] **and Yuta Nakashima**[1]
[1]Institute for Datability Science, Osaka University
[2]Nara Institute of Science and Technology
chu,n-yuta@ids.osaka-u.ac.jp, otani.mayu.ob9@is.naist.jp

1 Introduction

A paraphrase is a restatement of the meaning of a word, phrase, or sentence within the context of a specific language (e.g., "a red jersey" and "a red uniform shirt" in Figure 1 are paraphrases). Paraphrases have been exploited for natural language understanding, and shown to be very effective for various natural language processing (NLP) tasks, including question answering, summarization, machine translation, text normalization , textual entailment recognition, and semantic parsing (Ganitkevitch and Callison-Burch, 2014).

In this paper, we propose a novel task to extract *visually grounded paraphrases (VGPs)*. We define VGPs as different phrasal expressions that describe the same visual concept in an image. Nowadays, with the spread of the web and social media, it is easy to collect large amounts of images with their describing text. For example, different news sites release news with the same topic using the same image; photos with many comments are posted to social networking sites and blogs. As these describing texts are written by different people but about the same image, there are potentially large amounts of VGPs in the describing text (Figure 1). We aim to accurately extract these paraphrases using the image as a pivot to associate different phrases.

The extracted VGPs can be applied to various computer vision (CV) and NLP tasks, such as image captioning (Vinyals et al., 2015) and visual question answering (VQA) (Wu et al., 2017), for the better understanding of both images and languages. For example, a VQA system must understand queries of different expressions about the same visual concept (e.g., "a male" and "the pitcher" in Figure 1) in order to answer a question properly. VGPs can also be applied to the evaluation of image captioning systems in the similar way as paraphrases have been applied for machine translation evaluation (Snover et al., 2009).

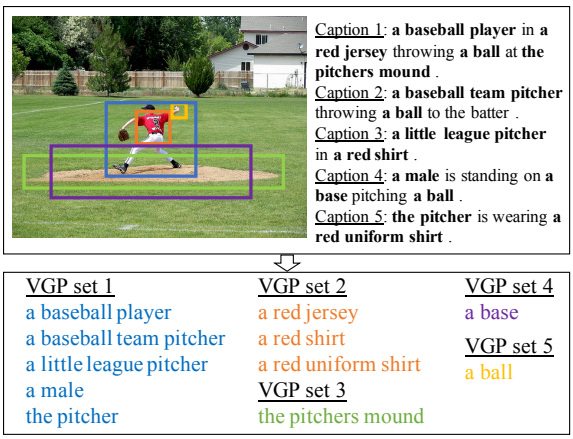

Figure 1: An example from the Flickr30k entities dataset, in which an image is described by five captions (entities in the captions are marked in bold). Our task is to extract the entities that describe the same visual concept (represented as an image region) in the image as VGPs. Note that the image regions are not given as input but are drawn here for comprehensibility.

As a pioneering study, we work on the Flickr30k entities dataset (Plummer et al., 2015). This dataset contains 30k images with 5 captions per image annotated via crowdsourcing, which can be seen as a very small subset of the data available in the web and social media. Figure 1 shows an example image together with its five captions taken from this dataset. In the Flickr30k entities dataset, entities (i.e., noun phrases) in the captions have been manually aligned to their corresponding image regions (Plummer et al., 2015). Therefore, we can obtain a set of phrases annotated with the same image region. This set of phrases are used as the ground truth VGPs in our study. The goal of this work is to extract these VGPs.

We formulate our task as a clustering task (Sec-

Copyright(C) 2018 The Association for Natural Language Processing.
All Rights Reserved.

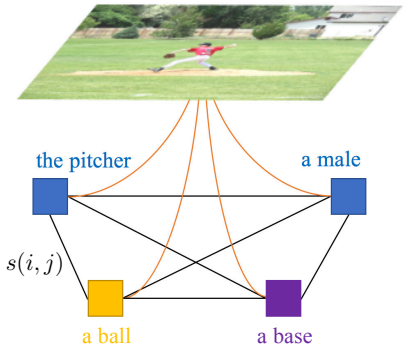

Figure 2: An overview of our VGP extraction formulation. We extract VGP via clustering, where the entity-entity similarity $s(i,j)$ is the key.

tion 2), where the similarity between each entity pair is crucial for the performance. We propose a supervised neural network (NN)-based method using both textual and visual features to explicitly model the similarity of an entity pair as VGPs (Section 3). Experiments show that our proposed NN-based method shows a good performance for VGP extraction.

2 Paraphrase Extraction via Clustering

We formulate the paraphrase extraction from the Flickr30k entities dataset as a clustering task. Given an image and all the entities in the corresponding captions, the task is to cluster the entities to its corresponding visual concepts represented as image regions. The number of clusters (i.e., the number of paraphrase sets in a set of an image and captions) is not explicitly given in our task. Therefore, we apply the affinity propagation algorithm (Frey and Dueck, 2007) to cluster entities, which can estimate the number of clusters as well.

Affinity propagation creates clusters by iteratively sending two types of messages between pairs of entities until convergence. The first type is the responsibility $r(i,j)$ sent from entity i to candidate representative entity j, indicating the strength that entity j should be the representative entity for entity i, which is defined as:

$$r(i,j) \leftarrow s(i,j) - \max_{\forall j' \neq j} \{a(i,j') + s(i,j')\} \quad (1)$$

where $s(i,j)$ is the similarity between entities i and j. The second type is the availability $a(i,j)$ sent from candidate representative entity j to entity i, indicating to what degree that candidate representative entity j is the cluster center for entity i, which is defined as:

$$a(i,j) \leftarrow \min\left\{0, r(j,j) + \sum_{\forall i' \notin \{i,j\}} \max\{0, r(i',j)\}\right\} \quad (2)$$

At the beginning, the values of $r(i,j)$ and $a(i,j)$ are set to zero, and they are updated in every iteration until convergence. We optimize the number of clusters on a validation split by adjusting the preference (i.e., self similarity $s(i,i)$) of affinity propagation.

Figure 2 shows an overview of our formulation, where the similarity between the entities is the key. We propose a supervised NN-based model for computing this similarity.

3 Supervised Similarity Model Based on Neural Network with Image Attention

We compute the similarities of entity pairs as VGPs by explicitly modeling the associations between them and an image. Figure 3 illustrates our proposed NN model. Given an entity pair and its corresponding image, we construct two separated *fusion nets* for each entity (Figure 3 (right)). A fusion net represents an entity with a concatenation of its entity feature vector and visual context vector. The visual context vector is computed with an attention mechanism, indicating to which part of the image should be paid attention, in order to judge whether the entity pair is VGP or not. The outputs of the two fusion nets are then fed into a multilayer perceptron (MLP) to compute the similarity of the two entities.

Formally, let X be a 196×512 feature map[1] extracted from the `conv5_3` layer in the VGG-16 network for an input image; $\mathbf{x}_n$ is a 512 dimensional vector at position n of X. Given an entity feature vector $\mathbf{v}_i$ and $\mathbf{x}_n$, we first transform them with fully connected (FC) layers whose unit sizes are 512:

$$\tilde{\mathbf{x}}_n = \mathrm{norm}_{\mathrm{L2}}(W_v \mathbf{x}_n + \mathbf{b}_v) \quad (3)$$

$$\tilde{\mathbf{v}}_i = \mathrm{norm}_{\mathrm{L2}}(W_p \mathbf{v}_i + \mathbf{b}_p) \quad (4)$$

where $\mathrm{norm}_{\mathrm{L2}}(\cdot)$ indicates L2 normalization to an input vector. We then compute an attention value a_n for $\mathbf{x}_n$ as:

$$\mathbf{h}_n = \mathrm{relu}(\tilde{\mathbf{x}}_n + \tilde{\mathbf{v}}_i) \quad (5)$$

$$e_n = \mathbf{w}^\top \mathbf{h}_n \quad (6)$$

$$a_n = \frac{\exp(e_n)}{\sum_{n=1}^{N} \exp(e_n)} \quad (7)$$

[1] An image is split into $14 \times 14 = 196$ sub-images, and represented as a 196×512 feature map.

Copyright(C) 2018 The Association for Natural Language Processing.
All Rights Reserved.

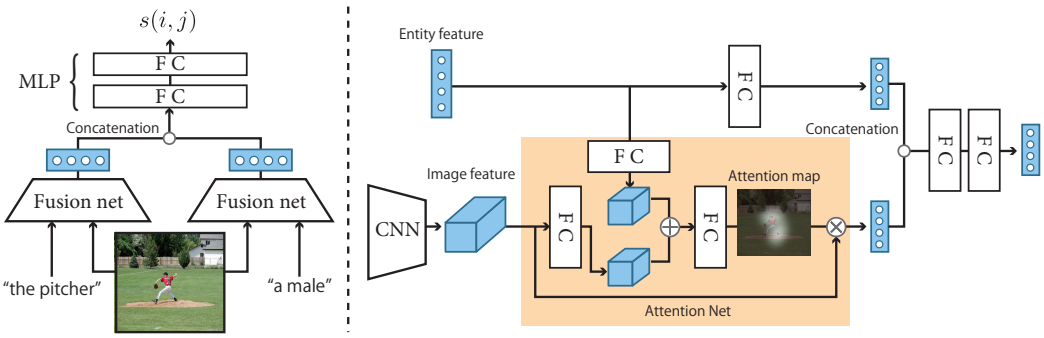

Figure 3: Supervised NN similarity model with image attention (left) and its fusion sub-network (right).

where $N = 196$. After obtaining a_n, we fuse a visual and an entity feature vector to $\mathbf{y}_i$ as:

$$\mathbf{c} = \sum_{n=1}^{N} a_n \mathbf{x}_n \quad (8)$$

$$\mathbf{y}_i = U[\mathrm{norm}_{\mathrm{L2}}(\mathbf{c}), \tilde{\mathbf{v}}_i] + \mathbf{d} \quad (9)$$

where $[\cdot, \cdot]$ indicates the concatenation of two vectors, $\mathbf{c}$ is a visual context vector. We compute fusion feature vectors $\mathbf{y}_i$ and $\mathbf{y}_j$ with the corresponding image. Finally, we feed them to a two-layer MLP network with ReLU non-linearities, whose unit sizes are 128 and 1, respectively, to produce the similarity of the entity pair.

4 Experiments

We conducted experiments on the Flickr30k entities dataset (Plummer et al., 2015). This dataset contains 31,837 images, which is described with 5 captions annotated via crowdsourcing. We followed the 29,873 training, 1,000 validation, and 1,000 test image splits used in the phrase localization task (Plummer et al., 2015). Our task is to automatically cluster the entities in the captions that describe the same visual concept (i.e., region in the dataset) in the image as VGPs. Entities that share the same ID and group type (e.g., "a red jersey," "a red shirt" and "a red uniform shirt" in Figure 1 share the same entity ID and group type "/EN#19026/clothing") are treated as the ground truth VGP clusters in our evaluation.

We evaluated both clustering and pairwise performance. The entity clustering performance for each image was measured with adjusted Rand index (ARI). We report the mean of ARI scores for all the images in the test split. The pairwise performance was evaluated with precision, recall, and F-score. We report the performance using the similarity threshold[2] tuned on the validation split that maximizes the F-score.

We used the affinity propagation implementation in Scikit-learn for clustering. We compared three types of entity feature vectors:

- Word embedding average (WEA): we represented each word with a 300 dimensional word2vec vector pre-trained on the Google News corpus. We removed stop words in each entity, and calculated the representation of each entity using the average of all word embeddings.
- Fisher vector (FV): Fisher vector is a pooling over word2vec vectors of individual words (Klein et al., 2014). Entity feature vectors were computed using the Fisher vector toolkit released by the authors.[3]
- Fisher vector w/ CCA (FV+CCA): image region feature vectors and entity feature vectors were projected into a 4,096 dimensional space CCA trained on the training split of the Flickr30k entity dataset.

We compared settings that directly use the cosine similarity between two entity feature vectors as entity-entity similarity $s(i, j)$. For our supervised NN method, we compared the following settings:

- Supervised NN (SNN): to show the effectiveness of the fusion net (Section 3), we compared a supervised NN-based setting that only feeding the entity feature vectors to the MLP (Figure 3 (left)) for paraphrase similarity prediction. This setting only uses entity feature vectors as input for the NN. It was trained on the training split of the Flickr30k entity dataset. We used all the ground truth VGP pairs in the training split as positive instances. During training, we constructed

[2] An entity pair with a similarity higher than a threshold is treated as VGPs.

[3] https://owncloud.cs.tau.ac.il/index.php/s/vb7ys8Xe8J8s8vo

Copyright(C) 2018 The Association for Natural Language Processing.
All Rights Reserved.

Method	ARI all / single / multi	Precision all / single / multi	Recall all / single / multi	F-score all / single / multi
WEA	49.55 / 48.48 / 49.31	62.95 / 46.15 / 62.77	69.67 / 67.04 / 79.23	66.14 / 54.66 / 70.05
FV	45.42 / 43.55 / 41.80	66.60 / 37.23 / 67.89	58.59 / 31.32 / 77.05	62.34 / 34.02 / 72.18
FV+CCA	54.97 / 51.84 / 50.76	64.79 / 55.79 / 68.24	82.20 / 75.83 / 84.98	72.46 / 64.28 / 75.69
SNN (WEA)	60.44 / 55.06 / 53.26	77.86 / 83.66 / 74.50	84.58 / 75.16 / **88.96**	81.08 / 79.18 / 81.09
SNN+image (WEA)	60.55 / 55.42 / **55.82**	79.47 / 81.01 / 77.26	84.56 / 79.35 / 87.06	81.94 / 80.17 / 81.86
Ensemble (WEA)	61.04 / 55.02 / 54.83	80.65 / 78.68 / 77.38	84.79 / 83.14 / 88.85	82.67 / 80.85 / 82.72
SNN (FV)	48.13 / 46.04 / 47.22	64.21 / 45.92 / 66.40	65.93 / 50.89 / 76.51	65.06 / 48.28 / 71.10
SNN+image (FV)	48.00 / 47.83 / 48.31	63.49 / 52.62 / 66.86	68.20 / 55.62 / 78.01	65.76 / 54.08 / 72.01
Ensemble (FV)	50.14 / 49.86 / 48.25	65.48 / 54.87 / 70.51	71.43 / 56.24 / 76.54	68.33 / 55.55 / 73.40
SNN (FV+CCA)	60.68 / 56.58 / 54.04	**83.11 / 85.19** / 77.44	82.13 / 79.30 / 87.69	82.62 / 82.14 / 82.25
SNN+image (FV+CCA)	61.56 / 54.86 / 54.14	82.51 / 84.52 / 80.28	84.19 / 81.85 / 86.82	83.34 / 83.16 / 83.43
Ensemble (FV+CCA)	**62.42 / 56.83** / 54.86	82.71 / 84.10 / 80.91	**85.67 / 83.50** / 87.06	**84.16 / 83.80 / 83.87**

Table 1: VGP extraction results ("all" evaluates on all entities, "single" and "multi" only evaluate on entities consist of one single token and multiple tokens after removing stop words, respectively).

mini-batches with 15% of positive instances and 85% of randomly sampled negative instances.

- SNN+image: this setting is for our proposed supervised NN-based method described in Section 3. We again compared the three different entity feature vectors. We used VGG-16 for the image features. The model was trained with the same configuration as the SNN setting.
- Ensemble: the ensemble of the SNN and SNN+image models that takes the average similarity given by both models. The motivation of this setting is to complement these two models to each other.

Table 1 shows the results of all the different methods. We can see that FV+CCA significantly outperforms WEA and FV. This is because it uses visual information in the training split that transforms the entity vectors and visual vectors into the semantic space that is helpful for detecting VGPs. NN-based methods using any entity feature vectors outperforms the methods that uses them directly. The reason for this is that it directly uses the paraphrase supervision in the training split. Using entity representation with better ARI and F-score for the SNN method can achieve better results. The performance improvement by SNN on FV is not as large as WEA and FV+CCA, and we suspect the reason for this is the sparseness of the Fisher vectors. Our proposed method (SNN+image) that uses both textual and visual features shows better performance compared to SNN that uses textual features only, indicating that the usage of visual features is helpful for our VGP extraction task. The ensemble of SNN and SNN+image further improves the performance, which means that these two models complement each other.

5 Conclusion

In this paper, we proposed a novel task to extract VGPs describing the same visual concept in an image. We proposed a NN-based method that uses both the textual and visual information to model the similarity between the VGPs. Experiments on the Flickr30k entities dataset showed that we achieved a good performance.

Acknowledgments

This work was supported by ACT-I, JST.

References

B. J Frey and D. Dueck. 2007. Clustering by passing messages between data points. *Science* 315(5814):972–976.

J. Ganitkevitch and C. Callison-Burch. 2014. The multilingual paraphrase database. In *LREC*. pages 4276–4283.

B. Klein, G. Lev, G. Sadeh, and L. Wolf. 2014. Fisher vectors derived from hybrid gaussian-laplacian mixture models for image annotation. *CoRR* abs/1411.7399.

B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*. pages 2641–2649.

M. G. Snover, N. Madnani, B. Dorr, and R. Schwartz. 2009. Ter-plus: Paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation* 23(2-3):117–127.

O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. 2015. Show and tell: A neural image caption generator. In *CVPR*. pages 3156–3164.

Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. Hengel. 2017. Visual question answering: A survey of methods and datasets. *CVIU* pages 1–20.

Copyright(C) 2018 The Association for Natural Language Processing.
All Rights Reserved.