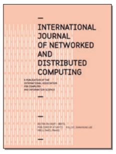
International Journal of Networked and Distributed Computing
Vol. 7(3); July (2019), pp. 107–112
DOI: https://doi.org/10.2991/ijndc.k.190710.002; ISSN 2211-7938; eISSN 2211-7946
https://www.atlantis-press.com/journals/ijndc



INTERNATIONAL
JOURNAL
OF NETWORKED
AND
DISTRIBUTED
COMPUTING

Research Article

"Never fry carrots without chopping" Generating Cooking Recipes from Cooking Videos Using Deep Learning Considering Previous Process

Tatsuki Fujii*, Yuichi Sei, Yasuyuki Tahara, Ryohei Orihara, Akihiko Ohsuga

Department of Informatics, The University of Electro-Communications, Chofugaoka 1-5-1, Chofu, Tokyo 182–8585, Japan

ARTICLE INFO

Article History

Received 25 March 2019
Accepted 10 May 2019

Keywords

Image captioning
deep learning
cooking
recipe
Doc2Vec

ABSTRACT

Research on deep-training captioning models that modify the natural-language contents of images and moving images has produced considerable results and attracted attention in recent years. In this research, we aim to generate recipe sentences from cooking videos acquired from YouTube. We treat this as an image-captioning task and propose two methods suitable for the work. We propose a method that adds a vector of a sentence already generated in the same recipe to the input of a captioning model. Then, we compare generated and correct sentences to calculate scores. We also propose a data-processing method to improve accuracy. We use several widely used metrics to evaluate image-captioning problems. We then train the same data with the simplest encoder–decoder model, compare it with correct recipe sentences, and calculate the metrics. The results indicate that our proposed methods help increase accuracy.

1. INTRODUCTION

Automatic captioning tasks that describe the content of images and moving images in natural language have important applications in areas such as search technology. In addition, captioning can assist with understanding content. Understanding of content can be deepened in a short time by reading captions. Among captioning models that use deep training, the encoder–decoder [1] model has generated considerable results and attracted attention, but many existing studies only consider the consistency of contiguous scenes over short periods. Considering the consistency of video segments as a matter of captioning has high importance. Generating cooking recipe sentences from cooking videos can be considered a captioning problem by treating recipes as captions. In addition, because the cooking video is constituted as a set of fragmentary tasks, a model that considers the consistency of the whole video is considered to be effective.

In the study of video captioning, few models focus on videos composed of fragmentary work sets, but we focus on such videos and the consistency of fragmented information. We propose a method to improve the accuracy of recipe-sentence generation from cooking videos. To determine whether the recipe sentence it generates is satisfactory, as shown in Figure 1, using a model that considers consistency within the recipe is important.

To ensure consistency of the generated caption with the same recipe, it is effective to provide captions already generated for the same recipe as the input for the decoder part of the encoder–decoder. When text information is inputted into a deep training model, vectorizing it is desirable, but in this research, we use Doc2Vec [2], which extends Word2Vec [3] from word to sentence vectorization. We also propose data-processing methods to make data set fit the task. One method is to remove scenes that are not suitable for captioning, such as when only people are on the screen, by using object detection. Another method is removing blurry scenes because such scenes might produce noise when training the model. In this paper, we propose a model that considers consistency in recipe-sentence generation from cooking videos by using deep training and data-processing methods. We also compare the accuracy of our proposed model and a simple encoder–decoder model trained in the same way. In addition, we indicate the effectiveness of the proposed data-processing methods. For evaluation, we use BLEU [4], METEOR [5], and CIDEr [6] scores, which are widely used to evaluate image-captioning problems.

2. RELATED WORK

In the subject of recent visual captioning via machine learning, many considerable results regarding image and video clip captioning have been reported. Guo et al. [7] and Li et al. [8] improved the accuracy of the task with a sequence-to-sequence model that uses a Convolutional Neural Network (CNN) [9] and Recurrent Neural Network (RNN). A CNN represents image features, and an RNN represents sentence features. LSTM [10] is widely used as an RNN, and the model is called an encoder–decoder, which is a type of end-to-end model that does not need to determine content words such as subjects. Therefore, it makes it possible to generate sentences

*Corresponding author. Email: fujii.tatsuki@ohsuga.lab.uec.ac.jp

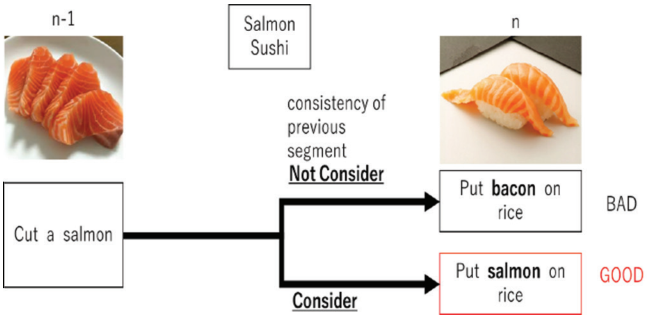


Figure 1 | Importance of considering consistency of previous captions.

directly from videos or images. For instance, Mao et al. [11] proposed a multimodal RNN to predict the probability of next words and previous words and given image features that are extracted by the CNN. Chen et al. [12] proposed a bidirectional mapping model between images and sentence descriptions that can reconstruct an image feature to generate captions.

Before the encoder–decoder model was proposed, many studies aimed to generate the sentence description of an image from visual information by focusing on intermediate states. Guadarrama et al.'s [13] model, for instance, considers each layer of neural net.

Many encoder–decoder captioning models have been studied this year. Oh et al. [14] used a data set annotated in Japanese for a cooking video; the data set is called Kyoto University Smart Kitchen (KUSK) [15]. Because this data set is compiled from a cooking video, the quality of the information is high, but the data amount is small and presented as dozens of videos. The amount of data was thought to be suppressed because it takes time to shoot and annotate the videos. Ushiku et al. adopted a technique called template training to compensate for this small amount of data. This estimates the subject, predicate, object, and other parts of speech based on a corpus consisting of recipe sentences; however, it is difficult to apply to other languages in its current state. Ushiku et al. [16] proposed a method that generates an interactive cooking video system using semantic annotation. To realize this, they used the ontology of the cooking recipe.

A captioning study using the latest encoder–decoder proposed a model that combines reinforcement learning and an encoder–decoder [17]. On the other hand, evaluating the captioning model is also a problem because many current captioning studies evaluate their methods with metrics using machine translation. Theses metrics could evaluate some digression of the methods, but the metrics are still not correlative enough. Some studies have aimed to resolve the issue, and Cui et al. [18] proposed a metric that has a high correlation with human definitions. The metric is trained like a discriminator.

Many encoder–decoders based on captioning models have been proposed, but few models aim for semantic consistency of a specific domain. Our paper proposes a model for cooking that considers the context of a recipe.

3. BASELINE MODEL

In this research, we propose a method based on the basic encoder–decoder model; therefore, we will describe the basic encoder–decoder model to clarify the differences. The encoder–decoder model was devised for machine translation, but after having been actively studied in recent years, it has been successfully adapted to captioning problems. The encoder is a CNN that encodes input information. The decoder is an RNN, and it receives input from the encoder and generates sentences by estimating words. The RNN, or the decoder, mainly uses Long Short-Term Memory (LSTM). The following describes the encoder and decoder when used for image captioning. The model is shown in Figure 2. The encoder's function when generating image captions is to encode the images to a fixed-length vector. In many cases, a trained CNN is used to encode the images by making them fixed-length vectors. Resnet [19], VGG [20], and others are used as trained CNN models. A schematic diagram is shown in Figure 3. The decoder's function is to generate sentences from linearized fixed-length vectors, which have been generated by the encoder. The vectors become the RNN's first hidden layer. The decoder's output is a collection of probabilities using softmax as an activation function. The words that the decoder predicts are not natural language, but a word map is created in advance and an ID is attached to each word token and encoded. The beginning of the sentence is the START token, and the end of the sentence is the END token. The number of occurrences at the time of word map creation includes the next three words and the UNK token. The word map is created from all captions included in a data set. An overview is shown in Figure 4.

4. PROPOSED METHOD

We propose two methods: a captioning model and data-processing method. The overall model construction is shown in Figure 5.

4.1. Proposed Model

To ensure consistency between a recipe and its cooking video, we propose a method to improve the accuracy of recipe-sentence generation from cooking videos. For consistency, it was effective to give information about other captions in the same recipe to the decoder when generating captions. The last action in cooking was also most important. Based on the above, we propose a method to input

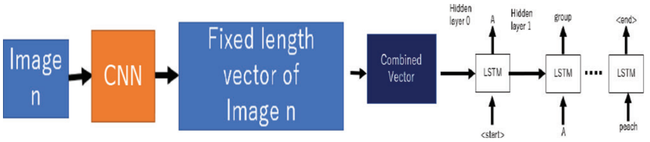


Figure 2 | Baseline encoder–decoder model.

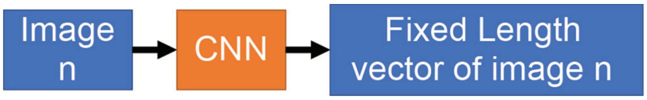


Figure 3 | Baseline model encoder.

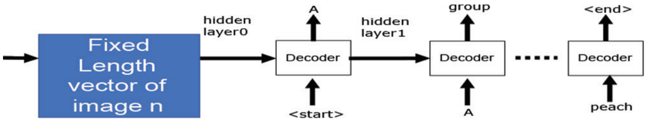


Figure 4 | Baseline model decoder.

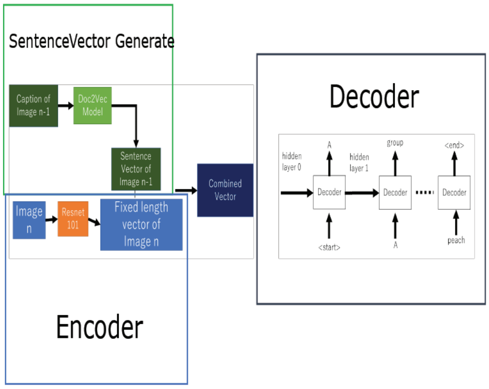


Figure 5 | Our proposed model.

captions generated for the image of the previous action into the decoder. Because sentence-based information cannot be passed to the decoder's LSTM in a natural language state, we perform a vectorization of the sentences. In this case, we convert them to sentence vectors using Doc2Vec. Although it is possible to create a word map for each word and to encode the word into a workable form by assigning an ID to each word token, we use Doc2Vec to consider more semantic information about sentences. We create a word map from all words from all recipes in a data set; therefore, frequently found words in recipes become tokens. The reason why we exclude words that are not found in the data set sometimes is because such words are too unique and become noise to training models.

4.1.1. Doc2Vec model

Doc2Vec is a model that converts variable length sentences (paragraphs) into fixed-length feature vectors and trains sentence vectors by training to predict words that follow a certain paragraph. For vectorized training and to vectorize sentences, we choose Doc2Vec instead of Sentence2Vec because Doc2Vec does not necessarily require a single sentence and can handle target sentences, regardless of their length. In this research, we prepare a sentence list of data sets to lay the groundwork for captioning and to train the Doc2Vec against it. It is possible to train the Doc2Vec model with data using other large-scale and diverse information, but according to Amanda et al., it is better to vectorize text with cooking sentences when dealing with the cuisine field. Because a report claimed that it goes up, we train the Doc2Vec model using the data that will be handled. When sentences are given to the trained model, a fixed-length sentence vector is obtained.

4.1.2. Input generator

The image vectors and sentence vectors are combined to create a vector to treat as the first hidden layer of the decoder. An overview is shown in Figure 6. Considering natural number n, n is larger than 1. For the nth image n in a recipe, the encoder obtains a fixed-length vector of image n. It obtains the sentence vector of image $n - 1$ with the caption of the generated $(n - 1)$th image and the trained Doc2Vec model. The fixed-length vector of the obtained image n and the sentence vector of image $n - 1$ are concatenated. When $n = 1$, an empty matrix is concatenated.

4.1.3. Encoder

The construction of the encoder in our proposed model is shown in Figure 7. The CNN used as the encoder is Resnet-101, which has been trained to mitigate ImageNet's image classification problem. We convert the input image to a fixed-length vector of 2048 × 14 × 14. We combine image vectors and sentence vectors to create the first hidden layer of the decoder. Considering natural number n, for the nth image n in a recipe, the encoder obtains a fixed-length vector of image n. We obtain the sentence vector of image $n - 1$ with the caption of the generated $(n - 1)$th image and the trained Doc2Vec model. The fixed-length vector of the obtained image n and the sentence vector of image $n - 1$ are concatenated. When $n = 1$, an empty matrix is concatenated.

4.1.4. Sentence vector generator

We obtain the sentence vector from the most recently generated recipe in the same video (see Figure 8). The Doc2Vec model has been pretrained with all recipes in the data set. Salvador et al. [21] adopted skip-thoughts to obtain recipe representation in the research. They mentioned that training skip-thought with recipe sentences produces better scores than training with a general

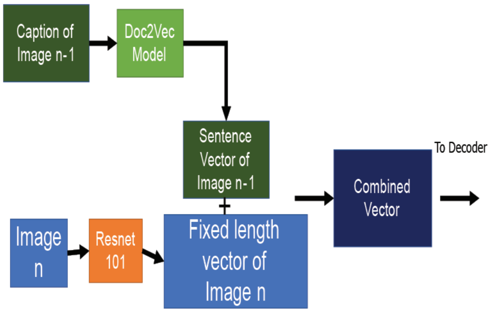


Figure 6 | Generator of input matrix.

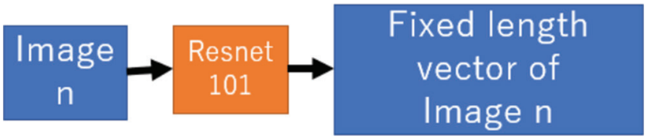


Figure 7 | Encoder of our proposed model.

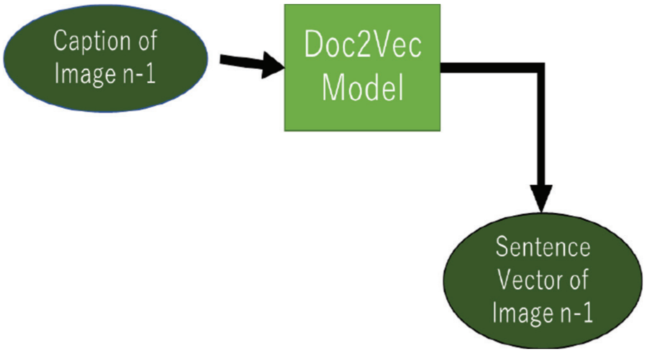


Figure 8 | We used Doc2Vec as sentence vector generator.

purpose data set. Doc2Vec extracts 1-dimensional fixed-length vector from one sentence.

4.1.5. Decoder

Figure 9 shows the construction of our proposed model's encoder, as well as the process of the decoder. We input a vector that combines the fixed-length vectors and sentence vectors of the decoder's image as the decoder's first hidden layer to estimate words and generate sentences. The loss function is a cross-entropy error. We use LSTM as part of the RNN of our model's decoder.

4.2. Data Processing

Our proposed data processing method consists of two parts. We believe choosing appropriate frames from video construct better data set for the task. The first method is choosing suitable scenes depends on object detection result. The other is excluding blurred frames.

4.2.1. Choice suitable scenes

Cooking videos can have several scenes that do not show the cooking process. These can be considered as noise when training, so we believe such scenes should be removed from the data set. To accomplish this, we adopt YOLO-v3, a state-of-the-art object-detection method. We use a model that is pretrained with the MSCOCO data set, which has 90 classes. When only the human class is detected in a scene, we do not include the scene in the data set. Figure 10 shows examples of the object detection results. In the left image, sandwich and person classes are detected, so the image is regarded as a suitable scene. The right one is excluded from the data set because only the person class is detected.

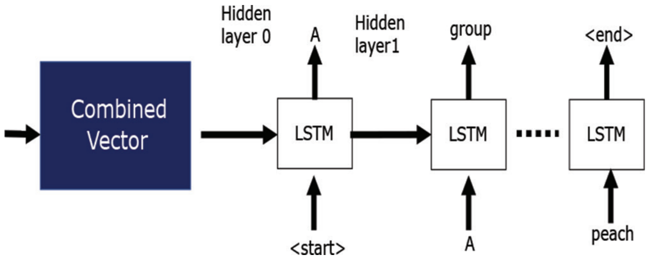


Figure 9 Decoder of our proposed model.

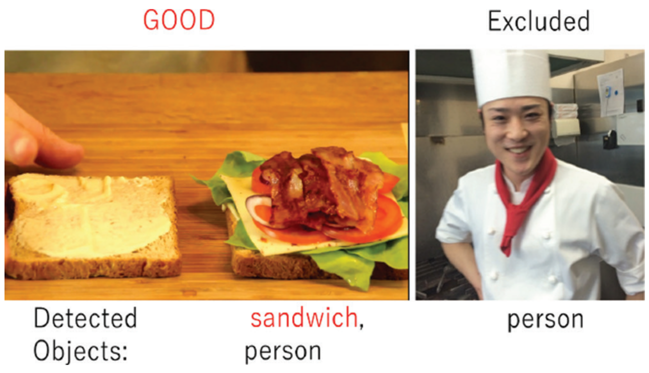


Figure 10 The results of YOLO object detection.

4.2.2. Blur detection

In a complete video, blurred scenes harm data set reliability. We introduce a fast Fourier transform to detect these scenes, which are then excluded from the data set. Figure 11 is an example of the images. The left one is a little blurry, but it is still acceptable. The right one is completely blurry, so it will be excluded from the data set. We adjusted a threshold via manpower.

5. EXPERIMENTS

We compare the accuracy of the basic encoder–decoder model with the proposed model. We also evaluate the effectiveness of our data processing methods.

5.1. Data Set

We use YouCookII [22], which is an English data set annotated with descriptive text for each cooking motion of a video acquired from YouTube. The animation time is about 5–10 min. We divide the data set into three subsets (training, validation, and test) and acquire one image at each in-motion time corresponding to each recipe annotated from each video. We assign 1333 videos to training, 229 videos to validation, and 228 videos to test. The videos are chosen randomly. The data amount is indicated in Table 1.

5.2. Preprocessing and Hyper Parameters

We perform training of the Doc2Vec model for vectorizing sentences for all sentences of the training, validation, and test subsets annotated in the YouCookFI animation. At this time, we set the epoch number to 100. In the training of the proposed method, we set the epoch number to 100, the batch size to 32, and the optimization function to Adam 13. For comparative experiments, the basic encoder–decoder model was also trained with similar parameters.

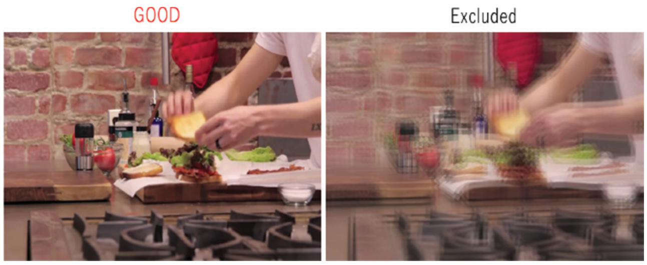


Figure 11 The image on right is excluded from the dataset.

Table 1 Amount of data contained in and obtained from YouCook

	Train	Validation	Test
Number of videos	1333	229	228
Number of obtained images	9705	1646	1646

Table 2 | Score comparison of each methods

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr
Baseline model	0.39	0.29	0.22	0.18	0.21	0.85
Proposed model	0.41	0.30	0.23	0.19	0.22	0.92
Baseline model + Proposed data processing	0.41	0.31	0.24	0.19	0.22	0.90
Proposed model + Proposed data processing	**0.43**	**0.32**	**0.24**	**0.20**	**0.23**	**0.94**

5.3. Captioning Target

In this research, we generate recipe sentences from cooking videos, but instead of captioning the entire video, we retrieve one image per range that involves explanatory text annotated using YouCookII. Thus, we treat recipe generation as a series of image-captioning problems rather than captioning problems. For the test, we also extract one image per range. We handle it as an image-captioning problem rather than a video-captioning problem. Because explanation part only human appearing is included in many cooking videos on YouTube and such kind of scenes could be noise. Therefore, we believe image captioning is better than video captioning for the research.

5.4. Metrics

We conduct a comparison of the basic encoder–decoder model and BLEU, METEOR, and CIDEr scores of the proposed method, as well as a comparison of the generated caption examples.

The BLEU score is a technique used for machine translation. In the case of captioning, it compares the generated caption with the correct sentence. The higher the numerical value, the better. BLEU-*n* (natural number) runs comparisons using *n*-gram language units. METEOR, which is also used for evaluating machine translation, produces good correlations with human judgement. CIDEr evaluates image description quality. It also calculates the similarity of automatically generated sentences to reference sentences. The higher all three metrics scores are, the better. In our research, we calculate the metrics for each image, not for each recipe.

5.5. Results

Table 2 shows scores for each metric for each method combination. As a result, our proposed methods increase each score. Our proposed model scored around 1.05 times higher on BLEU and METEOR and 1.08 times higher on CIDEr than the baseline model. As mentioned before, CIDEr is defined for image-captioning problems, and our proposed method is effective for other captioning problems. Regarding the data-processing method, our model increases the scores of the baseline and the proposed model. When compared with those not applied, they scored about 1.05 times higher.

Figure 12 shows a comparison of the generated recipes between the baseline and the proposed model. The baseline model generated sentences, including ones about sardines, without context, but our proposed method did not. We consider our proposed method to be effective when considering consistency with the recipe.

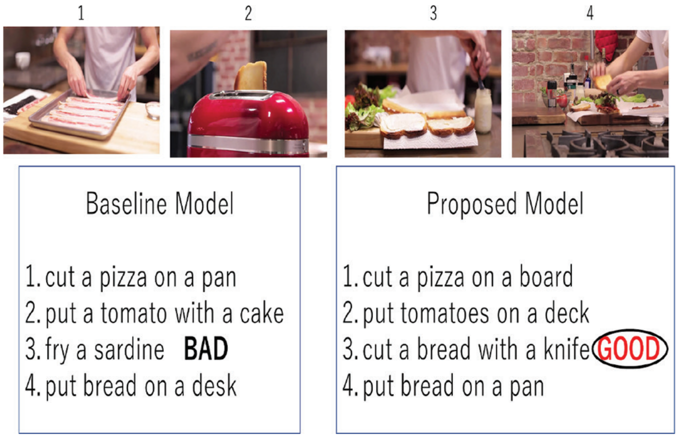


Figure 12 | Comparison of generated recipes.

6. CONCLUSION

In this paper, we prepared recipe sentences as captions for cooking videos composed of fragmented process sets by using deep training to determine the consistency of the entire moving picture and by comparing the captions generated to the recipe. We proposed a method that considers consistency within the recipes by giving the decoder the previous vectorized caption using Doc2Vec. We also proposed a data-processing method to improve accuracy. In comparable experiments, the difference in precision from the basic encoder–decoder model was not great, but some of the generated recipe statements showed a remarkable difference. The future prospects are described below. Although we evaluated scores and examples of generated recipes, we still have to introduce an evaluation method that can calculate the consistency of each recipe on our proposed model.

This research proposed using only information from previous captions generated within the recipe as the input of the decoder. However, to maintain consistency across the whole recipe, comparing the generated caption with all the captions in the recipe was conceivable to propose training for calculating loss. In our data-processing method, we removed images that had only the person class in their object detection results. We had to consider what was not suitable for the data set in this research and determine an accurate way to separate suitable images from unsuitable ones to increase scores. Considering the results of each score, the data-processing methods can be reinforced; therefore, we are going to add more methods to increase accuracy.

In this paper, because we obtained only one image for each process, the amount of data is not large compared with other deep learning research studies. To improve accuracy, we are going to experiment so that we can determine how many images per process is best. As another solution to increase accuracy, we aim to introduce video-captioning models, not image-captioning models. Furthermore, we have to find a way to merge the proposed method in this paper with other video-captioning models For the evaluation index and the evaluation method, we only compared the basic encoder–decoder model and the proposed method, but by comparing the latest captioning technique and the proposed method and by using other indicators, improved outcomes may emerge. In this paper, we used an annotated data set for training and evaluation, but we aim to generate recipes from non-annotated videos for evaluation in our future work. This means that automating separate video processes must be realized.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Numbers: JP17H04705, JP18H03229, JP18H03340, 18K19835, JP19H04113, JP19K12107.

REFERENCES

[1] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, et al., Learning phrase representations using RNN encoder–decoder for statistical machine translation, 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1724–1734.

[2] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.

[3] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv: 1301.3781, 2013.

[4] K. Papineni, S. Roukos, T. Ward, W-J. Zhu, BLEU: a method for automatic evaluation of machine translation, 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002, pp. 311–318.

[5] S. Banerjee, A. Lavie, METEOR: an automatic metric for MT evaluation with improved correlation with human judgments, ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Association for Computational Linguistics, Ann Arbor, Michigan, 2005, pp. 65–72.

[6] R. Vedantam, C. Lawrence Zitnick, D. Parikh, CIDEr: consensus-based image description evaluation, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Boston, MA, USA, 2015, pp. 4566–4575.

[7] Z. Guo, L. Gao, J. Song, X. Xu, J. Shao, H.T. Shen, Attention-based LSTM with semantic consistency for videos captioning, 24th ACM International Conference on Multimedia, ACM, Amsterdam, The Netherlands, 2016, pp. 357–361.

[8] G. Li, S. Ma, Y. Han, Summarization-based video caption via deep neural networks, 23rd ACM International Conference on Multimedia, ACM, Brisbane, Australia, 2015, pp. 1191–1194.

[9] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE, IEEE, 1998, pp. 2278–2324.

[10] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997), 1735–1780.

[11] J. Mao, W. Xu, Y. Yang, J. Wang, Z-H. Huang, A. Yuille, Deep captioning with multimodal recurrent neural networks (m-RNN), arXiv preprint arXiv:1412.6632, 2014.

[12] X. Chen, C. Lawrence Zitnick, Mind's eye: a recurrent visual representation for image caption generation, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Boston, MA, USA, 2015, pp. 2422–2431.

[13] J. Donahue, L.A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, et al., Long-term recurrent convolutional networks for visual recognition and description, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Boston, MA, USA, 2015, pp. 2625–2634.

[14] K-J. Oh, M-D. Hong, U-N. Yoon, G-S. Jo, Automatic generation of interactive cooking video with semantic annotation, J. Univ. Comput. Sci. 22 (2016), 742–759.

[15] A. Hashimoto, T. Sasada, Y. Yamakata, S. Mori, M. Minoh, KUSK dataset: toward a direct understanding of recipe text and human cooking activity, 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, ACM, 2014, pp. 583–588.

[16] A. Ushiku, H. Hashimoto, A. Hashimoto, S. Mori, Procedural text generation from an execution video, 8th International Joint Conference on Natural Language Processing, volume 1: Long Papers, Asian Federation of Natural Language Processing, Taipei, Taiwan, 2017, pp. 326–335.

[17] X. Wang, W. Chen, J. Wu, Y-F. Wang, W.Y. Wang, Video captioning via hierarchical reinforcement learning, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT, USA, 2018, pp. 4213–4222.

[18] Y. Cui, G. Yang, A. Veit, X. Huang, S. Belongie, Learning to evaluate image captioning, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT, USA, 2018, pp. 5804–5812.

[19] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, 2016, pp. 770–778.

[20] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, 2014.

[21] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, et al., Learning cross-modal embeddings for cooking recipes and food images, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, USA, 2017, pp. 3020–3028.

[22] L. Zhou, C. Xu, J.J. Corso, Towards automatic learning of procedures from web instructional videos, 32nd AAAI Conference on Artificial Intelligence, 2018, pp. 7590–7598.