Vol.2019-NL-240 No.10
2019/6/13

Captioning Events in Tourist Spots by Neural Language Generation

Mai Nguyen[1,a)]　吉野 幸一郎[1,2,b)]　鈴木 優[1,3]　中村 哲[1]

Abstract: We present an application that captions events in tourist attractions by summarizing various information sources in natural language descriptions. The system is divided into two parts: what-to-say, which summaries information into structured data, and how-to-say, which produces natural language captions from input meaning representation. In what-to-say, information from several information sources, such as infrared sensors and social media, are extracted into a semantic frame. In how-to-say, we utilized semantically-conditioned long short-term memory neural networks to generate natural language captions for giving information to users in an understandable way. An empirical evaluation of the system shows the quality of generated text across five automated metrics. The generated sentences are used in the application system for helping visitors. The subjective evaluation shows the usefulness of the proposed system.

1. Introduction

Tourism has become a promising sector in the world. These days, tourists decide their travel plan based on many factors of tourist spots such as other user reviews, their congestion, and popularity. However, many information sources are changing every moment; it makes the decision making difficult. Generally, the data available for tourist consists of (i) continuous variables (e.g., congestion, etc.) and (ii) discrete variables (e.g., events, the information in social media, etc.). Hence, efficiently presenting these types of data is crucial for informed travel decision making. In some circumstances, giving a textual caption about a tourist spot is more likely to help travelers for making better decisions since the text summary is one of the easiest ways to understand information.

In this study, we address the task of generating textual descriptions about tourist spots to support user decision making. The data of each Point of Interest (POI) such as congestion or events are observed from multiple data sources such as official website, infrared sensors. Generally, we propose to divide the generation process into two parts: what-to-say, how-to-say. What-to-say extracts contextual information from various information sources into a meaning representation (MR). How-to-say generates a textual output from the MR based on a neural language generation (NLG) model, particularly semantically-conditioned Long Short-term Memory (SC-LSTM) [1]. An example is shown in Figure 1. We also combine the NLG with beam search and a n-best list reranker to suppress irrelevant information in the output.

MR	name[Fushimi Inari], crowded[high], time[festival days] recommended[no]
Reference	It is a good idea to avoid Fushimi Inari during festival days as it is extraordinarily crowded.

図 1 MR とそれに付与されたレファレンスの例．MR は key-value のペアの組で表現され，レファレンスは MR に記載された内容を説明する観光スポットに関する説明文．

Fig. 1 An example MR and reference. MR is a set of key value pairs, a reference is a caption in tourist spot describes the MR.

We have constructed an application for helping travelers with information of POIs, which changes moment by

[1] 奈良先端科学技術大学院大学 先端科学技術研究科
生駒市高山町 8916-5
[2] 科学技術振興機構, さきがけ
[3] 現在、岐阜大学 工学部 所属
[a)] nguyen.quynh_mai.nm9 at is.naist.jp
[b)] koichiro at is.naist.jp

moment, and integrated the NLG system as an interface to access information. We build a personalized recommendation feed in pushing function to notify the latest information of POIs, places that travels are interested in, with generated descriptions.

We conducted two experiments: the quality checking of generated sentences and human subjective evaluation about pushing notification. The first experiment is run on a crowd-sourced dataset in the context of an application describing sightseeing spots in Japan. We show that the proposed generation system increases the quality of generated text across five different automated metrics (BLEU, NIST, METEOR, ROUGE, and CIDEr) over rule-based baseline models. In order to assess the subjective performance of our system, an experiment with users using a mobile application integrated our NLG system was conducted. The result shows the benefits of our proposed system toward users during their trip.

We introduce the overall architecture of the proposed system and its components, such as pushing notifications that use the generated outputs and neutral language generation in Section 3. Section 4 details our experiments of the generation generation system with automatic evaluation, Section 5 describes experiments in real fields. We summarize related works by using smartphone application in Section 2 and offer conclusions in Section 6.

2. Related works

The task of generating text from has structured data been investigated in many domains such as weather forecast [2] [3], navigation assistance [4], sports [5]. Traditionally, approaches to such studies used handcrafted rules [2] or are templates. Despite its robustness and adequacy, the approach does not easily scale to large other domains and remains repetition issue.

Systems based on neural network (neural-based) attempt to learn generation directly from data, which enables the system for generating descriptions more naturally, removes the dependency on the predefined rules. Neural-based models have also been successfully applied to various data-to-text generation tasks. More recently, there has been some work on neural-based models for generating a description such as reviews from product attributes [6] and comments on markets [7]. However, the model of Wen et al. [1] takes only a short time to produce text and easier for building and extending to other domains.

There also been some work on personalized recommendation feed. TripAdvisor [*1] has developed a system consisting of reviews from users. The recommendations are generated based on the stored reviews; however, the real-time changes are not considered in the application. We realize a system that can describe real-time events in tourist spots by utilizing generation based captioning and a real-time data update system.

3. System Architecture

The goal of our system is building a system that can describe what happens in tourist spots to users in real-time. We built a system that receives a variety of information sources and then generates a description to notify users. Our system consists of three modules: back-end system including event extraction (what-to-say), natural language generation (how-to-say), and pushing notification server. An overview of the system was given in Figure 2 and its components are described as follows.

- **Back-end system** stores data observed from various data source such as infrared sensors, social media, user's application. The back-end system also extracts values from these data to fill states defined in a meaning representation. Once the back-end system receives a notification request, the system sends a pushing to specified mobile applications.
- **Pushing notification server** asks POIs update to the back-end system, summarizes updates into MR, sends a generation request to the neural language generation system, and sends a pushing request to the back-end system with the generated sentence.
- **Neural language generation system** generates a sentence according to the MR given by the pushing notification server.

Our proposed parts of the system include two main components: i) pushing notification server and ii) the natural language generator integrated into the server. We apply a character-based version of semantically-conditioned Long Short-term Memory proposed by Wen et al. [1] as our generator, which has a gate to control the generated semantics. Some details are updated to fit our domain. A pushing notification server is a bridge element to encapsulate various information into one MR once it received new information related to attractions. The MR is sent to the generator to produce a description in natural language. The generated caption is sent in the push request to the back-end server and then notify on a mobile application

[*1] www.tripadvisor.com

to corresponding users.

3.1 Pushing Notification

Pushing notifications are an important feature, retain and re-engage users and monetize on their attention. To enhance user's experience, we integrate pushing notification server with our natural language generator in our target application, which delivers comprehensive, up-to-date information about POIs. On the other hand, the server also takes responsibility for the what-to-say part, which summarizes information about POIs.

We designed the pushing notification server with two main functions. First, the server checks the back-end database with a specific period. If the system detects any newly updated information about POIs hold by users, the system will produce input MRs corresponding to new information, generate descriptions from the MRs by using the neural language generation system and send push request within the report to the back-end system.

3.2 Neural Language Generation

We utilized Semantically Conditional Long Short-term Memory Network (SC-LSTM) proposed by (Wen et al., 2015b) as our generator, which has gates to control the generated semantics.

We give a brief overview of SC-LSTM model and refer to the original paper [1] for an in-depth description. The model extends the original LSTM [8] cell by adding a control cell in charge of sentence planning as a frame that has some slot-values. The frame is the input of the SC-LSTM, converted to a one-hot encoded MR vector, which represents the value for each property. This cell manipulates the MR-vector during the generation process to ensure the information is fully rendered in the utterance. The cell acts as a forget gate keeping track of what information should be retained for future time steps and discards the others.

4. Experiments of Generation

The experiment was designed to investigate whether our neural language generation system can generate a high-quality description of certain spots in the tourist domain. To train the generator, we first collected data from a human via a crowd-sourcing platform. To assess the effectiveness of our generation system, we employ various reference-based metrics on the test set: BLEU [9], ROUGE-L [10], METEOR [11], NIST [12], CIDEr [13] score. Table 1 shows description of the metrics above.

Measure	Description
BLEU	Calculate precision as ratio of correctly generated n-grams. We used four-gram
NIST	Extension of BLEU score, weights the n-grams informativeness. We used four-gram
METEOR	Based on the harmonic mean and computes F-measure
ROUGE-L	Recall-based measure which compares the longest common subsequence
CIDEr	Based on TF-IDF scoring, compares similarity of n-grams to majority of references

表 1　各自動評価手法の説明.
Table 1　Short descriptions of automated metrics.

4.1 Data collection

We collected data that has pairs of a MR (frame) and its captioning sentence. Each MR includes an unordered list of slot (or attribute) and value pairs. The captioning sentence is a natural language description corresponding to the MR, possibly consisting of several sentences. An example of the corpus is shown in Figure 1

For data collection, we prepared a set of MRs containing seven attributes. The particular attributes used are provided in Table 2 with their data types. The order of attributes in MR was randomly selected and then fed to the crowd workers. Workers were shown each MR and asked to enter an appropriate utterance in natural English corresponding to the MR. The crowd workers are not primed by ordering used.

Attribute	**Data Type**	**Example value**
name	verbatim string	Kyoto Tower
event	verbatim string	cherry blossom, gion festival
stateEvent	dictionary	happening, finished
crowded	dictionary	high, low, average
recommended	boolean	Yes / No
time	enumerable	now, holidays
popular	boolean	Yes/No

表 2　ドメインが持つスロットとその取り得る値.
Table 2　Domain attributes and attribute types.

We used Figure-eight [*2] platform to set up our annotation process and to access an online workforce. Around 3300 utterances were collected from crowd workers. After processing and grouping each reference according to its

[*2] https://www.figure-eight.com/

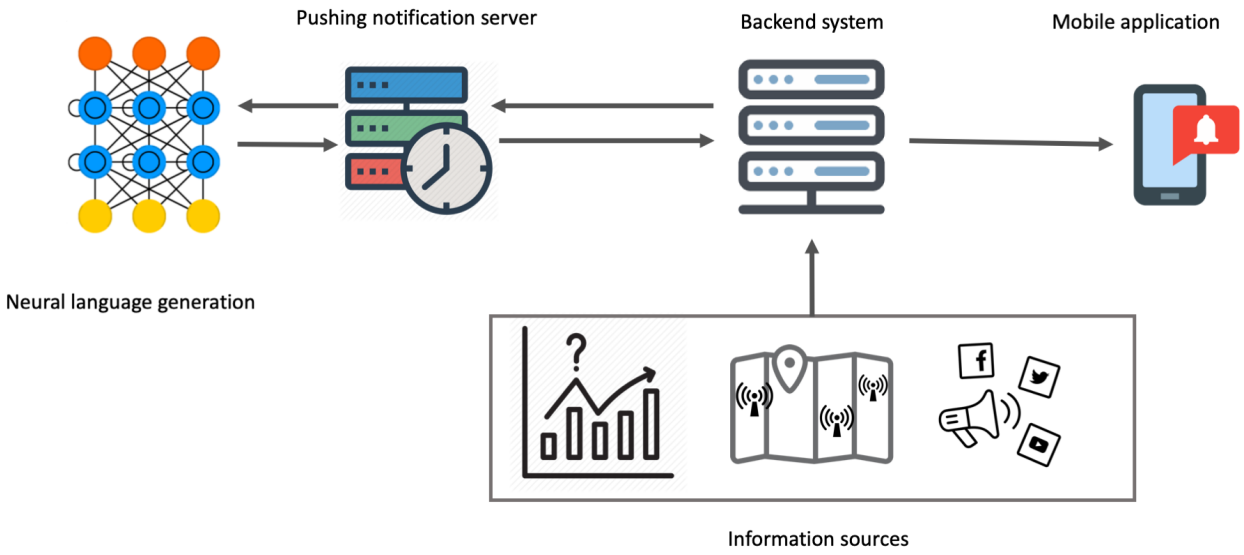

図 2 提案システムの概要．Pushing notification server は Backend system に保管された情報を MR へと変換し，Neural language generator へ送信する．Generator の出力はイベントの説明としてサーバに送られる．

Fig. 2 Architecture of the proposed system. Pushing notification server summarize information into MR vector as input to the generator. The output of generator is sent to main server to notify to users as notification of event description.

combined attributes, we obtained 283 distinct MRs. On average, each MR consists of 4 slots, and there are 11.6 different RF for each MR.

4.2 Experiment settings

Since our model is working on a character level, we treat each reference as a sequence of character, where each character is represented as a 1-hot content vector. To avoid data sparsity, we substituted some properties value such as name and event to named entity markers. For example, *name* and *event* values are replaced with tokens 'X-name' and 'X-event' respectively.

表 3 MR およびレファレンス文の総数．

Table 3 Data split by number of references and number of MRs.

	Training	Validation	Testing
REF	2800	200	296
MR	283	80	80

The collected data is split into a training, validation, and testing set in the ratio 85%:6%:9% (refer to Table 3 for the data-spit in details). The validation and test data are multi-references; the validation set has, on average, 8.1 references for each MR. A separate test set with previously unseen combinations of attributes contains 80 MRs and its references are unseen in the training dataset and used for automatic evaluation of the generator. We implement our character-based version of SC-LSTM in the TensorFlow frame-work [14]. During training, we train models with Adam optimizer [15] to minimize the cross-entropy as the loss function. A similar dropout method was adopted by [16] who use the same dropout mask for inputs, outputs, and hidden layers at each time step. Based on a few preliminary experiments, we use a hidden state of size 1024 for LSTM cell with a batch size of 256 and a maximum sequence length of 256.

4.3 Results

We compared our generator with a rule-based generator and reported the scores for the automatic evaluation, including the metrics BLEU, ROUGE-L, METEOR, NIST, and CIDEr scores by comparing with references in the testing set. Table 4 shows that our model performs better than the rule-based method; the neural-based generator archives almost the best scores. In particular, our model outperformed the baseline model with an improvement of 0.2 points in BLEU, 0.36 in NIST, 0.47 in CIDEr. In terms of fluency, most of the captions generated by our generators show that the models learn to produce fluent sentences in domain style. In terms of informativeness,

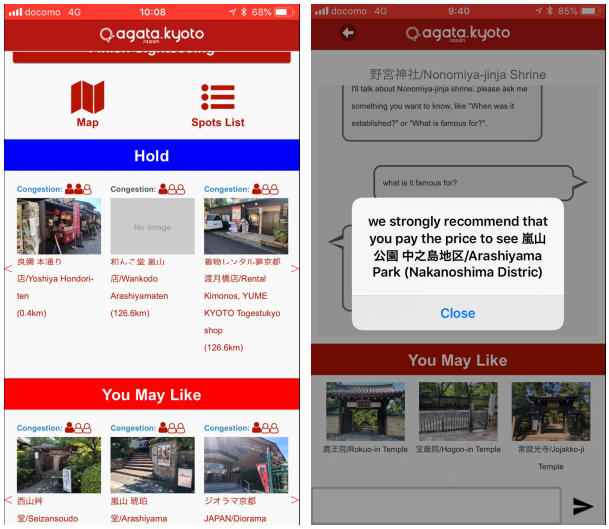

図 3　提案システムの画面.
Fig. 3　Screen shots of the proposed application system.

while some outputs show enough information once the input is short, some outputs are not entirely informative: missing or incorrect information (see Table 5).

5. System Evaluation in Real Field

We conducted experiments to evaluate the overall system in the real field with users by informing information updates. Once the system received new information related to the user interests, the system notifies them to users as pushing of description in natural language. This section describes the details of the experimental settings and results.

5.1 Experiment Setting

As described in Figure 2, information of any POIs and users were written on the backend system by their recognition system asynchronously. Our pushing notification server checked the backend database in every five minutes. If the system found any updates about POIs held by the user or POIs recommended by the system, the system generated sentences and sent a push request to the backend system. We did not control the time of pushing in our experiment; the pushing notification was sent immediately when the backend system received the pushing request from the pushing notification server.

We conducted the experiment at Arashiyama area in Kyoto. 100 POIs were selected preliminary and registered on our application system, as shown in Figure 3. Congestion of each POI is recognized by infrared sensors or human observers in three degrees (not crowded, moderately crowded and highly crowded).

We collected two groups of experiment participants: 12 students and 10 ordinary people who have never been to Arashiyama area. We conducted two experiments for each group. Any participants are requested to use the application during their travels and visit at least three POIs by considering application suggestions and their preference, from 10 AM to 3 PM.

5.2 Experimental Results

After the experiment, we requested them to evaluate the usefulness of the pushing function of the application in 5 degree as follows.

- 5: Very Satisfied
- 4: Rather Satisfied
- 3: Satisfied
- 2: Not Satisfied
- 1: Not Satisfied at All

The number of participants selected each score is shown in Table 6. From the result, about 60% of participants are satisfied by the application. However, the remaining people are not satisfied. In questionnaires from users, some users said that the number of pushing was too much, and the timing was not satisfactory. We need to improve this point in future systems.

6. Conclusion and Future Work

We presented a real-time system for generating a description of tourist attractions that refers to multiple resources. Our automatic showed that our generation model produces fluent text overall, enhances the quality of textual description in the tourist domain. In term of human evaluation, our system could support decision markers during their trip. Finally, a number of potential limitations need to be considered. First, the neural-based language generators have difficulty capturing long-term structure. Second, the user's personalized travel feed notify them too often, we need to consider the pushing timing. In the future, we would like to investigate these latter aspects in this field.

参考文献

[1] Tsung-Hsien Wen, Milica Gasic, N. M. P. S. D. V. and Young, S.: Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems, *Proceedings of 2015 Conference on Empirical Methods in Natural Language Processing(EMNLP)*, pp. 1711–1721 (2015).
[2] Sripada, S. G., Reiter, E. and Davy, I.: SUMTIME-MOUSAM: Configurable Marine Weather Forecast Generator (2003).

表 4　各自動評価手法の結果.
Table 4　Results of the automatic metrics.

System	BLEU	NIST	METEOR	ROUGE-L	CIDEr
rule-based	0.41	5.67	0.36	**0.67**	2.23
our generator	**0.43**	**6.03**	**0.37**	0.64	**2.70**

表 5　テストセットから提案法により生成されたキャプションの例（赤字は誤った情報の出力）.
Table 5　Sample of event caption generated by our generator system from the test set. Errors are marked in color(incorrect information)

#	Example of descriptions about tourist spots in Japan
1	*name[yodo river park], event[cherry blossoms], state[happening], recommended[no]* cherry blossoms is upcoming in yamaga castle park , you should not go there .
2	*name[Yoyogi Park], crowded[no], time[now], recommended[no]* Yoyogi Park is not crowded right now, but it is not recommended to visit.
3	*name[Akihabara], crowded[low], time[festival days], recommended[no]* Akihabara is slightly crowded during festival days, so it is not recommended to visit.
4	*name[kyoto imperial palace], event[aoi festival], state[happening], crowded[low], recommended[yes]* Aoi festival is happening in kyoto imperial palace, it is medium crowded, you should go there.
5	*name[Shibuya], crowded[high], time[morning], recommended[yes]* Shibuya is extremely crowded in the morning, but it is recommended to visit.

表 6　主観評価の結果．質問の解答番号に回答した人数を示している.
Table 6　Results of subjective evaluation. Numbers of answers associated to the question for pushing function are shown.

Experiment	Answers					Ratio of satisfied
	5	4	3	2	1	
Students	1	1	5	1	4	58%
Ordinary people	2	1	3	3	1	60%

[3] Belz, A.: Automatic Generation of Weather Forecast Texts Using Comprehensive Probabilistic Generation-space Models, *Nat. Lang. Eng.*, Vol. 14, No. 4, pp. 431–455 (online), DOI: 10.1017/S1351324907004664 (2008).
[4] Dale, R., Geldof, S. and Prost, J.-P.: CORAL: Using Natural Language Generation for Navigational Assistance, *Proceedings of the 26th Australasian Computer Science Conference - Volume 16*, ACSC '03, Darlinghurst, Australia, Australia, Australian Computer Society, Inc., pp. 35–44 (online), available from ⟨http://dl.acm.org/citation.cfm?id=783106.783111⟩ (2003).
[5] Liang, P., Jordan, M. and Klein, D.: Learning Semantic Correspondences with Less Supervision, *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Suntec, Singapore, Association for Computational Linguistics, pp. 91–99 (online), available from ⟨https://www.aclweb.org/anthology/P09-1011⟩ (2009).
[6] Dong, L., Huang, S., Wei, F., Lapata, M., Zhou, M. and Xu, K.: Learning to Generate Product Reviews from Attributes, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Valencia, Spain, Association for Computational Linguistics, pp. 623–632 (online), available from ⟨https://www.aclweb.org/anthology/E17-1059⟩ (2017).
[7] Murakami, S., Watanabe, A., Miyazawa, A., Goshima, K., Yanase, T., Takamura, H. and Miyao, Y.: Learning to Generate Market Comments from Stock Prices, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, Association for Computational Linguistics, pp. 1374–1384 (online), DOI: 10.18653/v1/P17-1126 (2017).
[8] Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, *Neural Comput.*, Vol. 9, No. 8, pp. 1735–1780 (online), DOI: 10.1162/neco.1997.9.8.1735 (1997).
[9] Kishore Papineni, Salim Roukos, T. W. and Zhu, W.: Bleu: a method for automatic evaluation of machine translation, *Proceedings of the 40th annual meeting on association for computational linguistics*, p. 311318.
[10] Lin, C.-Y.: Rouge: A package for automatic evaluation of summaries, *Text summarization branches out: Proceedings of the ACL-04 workshop* (2004).
[11] Denkowski, M. and Lavie, A.: Meteor universal: Language specific translation evaluation for any target language, *Proceedings of the ninth workshop on statistical machine translation.*, pp. 376–380 (2014).
[12] Doddington, G.: Automatic evaluation of machine translation quality using n-gram coccurrence statistics.
[13] Ramakrishna Vedantam, C. L. Z. and Parikh, D.: Cider: Consensus-based image description evaluation, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575 (2015).
[14] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. and Zheng, X.: TensorFlow: A system for large-scale machine learning, *12th USENIX Symposium on Operating Systems Design and*

Implementation (OSDI 16), pp. 265–283 (2016).
[15] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *arXiv e-prints* (2014).
[16] Gal, Y. and Ghahramani, Z.: A Theoretically Grounded Application of Dropout in Recurrent Neural Networks, *arXiv e-prints*, p. arXiv:1512.05287 (2015).