You May Like This Hotel Because ...: Identifying Evidence for Explainable Recommendations

Shin Kanouchi[1], **Masato Neishi**[2], **Yuta Hayashibe**[1], **Hiroki Ouchi**[3], **Naoaki Okazaki**[4]
[1]Megagon Labs, Tokyo, [2]The University of Tokyo,
[3]RIKEN, [4]Tokyo Institute of Technology
{shin187nlp,hayashibe}@megagon.ai,
neishi@tkl.iis.u-tokyo.ac.jp,
hiroki.ouchi@riken.jp, okazaki@c.titech.ac.jp

Abstract

Explainable recommendation is a good way to improve user satisfaction. However, explainable recommendation in dialogue is challenging since it has to handle natural language as both input and output. To tackle the challenge, this paper proposes a novel and practical task to explain evidences in recommending hotels given vague requests expressed freely in natural language. We decompose the process into two subtasks on hotel reviews: *evidence identification* and *evidence explanation*. The former predicts whether or not a sentence contains evidence that expresses why a given request is satisfied. The latter generates a recommendation sentence given a request and an evidence sentence. In order to address these subtasks, we build an Evidence-based Explanation dataset, which is the largest dataset for explaining evidences in recommending hotels for vague requests. The experimental results demonstrate that the BERT model can find evidence sentences with respect to various vague requests and that the LSTM-based model can generate recommendation sentences.

1 Introduction

Recently, dialog systems using Natural Language Processing technology have been adopted in interactive services such as call centers (Zumstein and Hundertmark, 2017). One challenging issue in a real-world scenario is vague requests[1] from users. For example, in a hotel booking service, users often ask operators for "a child-friendly hotel" or "a convenient inn." To respond to such vague requests, human operators need to explain the reason why the given request is satisfied. An example response would be, "This hotel has a large kids' space, so I recommend it for families with children like you." Responding to vague requests with evidences is effective because it not only strengthens the recommendation, but also urges users to make more concrete requests such as "I don't need a kids' space but want a baby stroller rental service."

Several studies have addressed explainable recommendations that produce natural language sentences (Zhao et al., 2014; Zhang et al., 2014; Wang et al., 2018; Zhao et al., 2019). One major approach is feature-based explanations. Zhang et al. (2014) generated explanation sentences using templates with slots, for example, "You might be interested in `[feature]`, on which this product performs well." However, by handling only predefined and limited features, this study cannot explain detailed evidences for each hotel such as "a view of Mount Fuji and Lake Kawaguchi." Furthermore, this study does not accept natural language requests as inputs, which is a major bottleneck for building dialog-based interactive systems.

In this study, we propose a novel and practical task to identify and explain evidences that satisfy a given vague request expressed freely in natural language. Specifically, assuming a practical situation of recommendation, we address a hotel booking service. When choosing a hotel on an interactive service, users make a wide range of vague requests, which differ from predefined aspects (Wang et al., 2010), emotional expressions (Chen et al., 2010) and questions (Rajani et al., 2019). In order to satisfy vague requests by recommending hotels with evidences, the system must understand a given request, associate the request to a hotel with spe-

[1]In this study, a vague request means one that does not specify a specific product, experience or service.

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 890–899

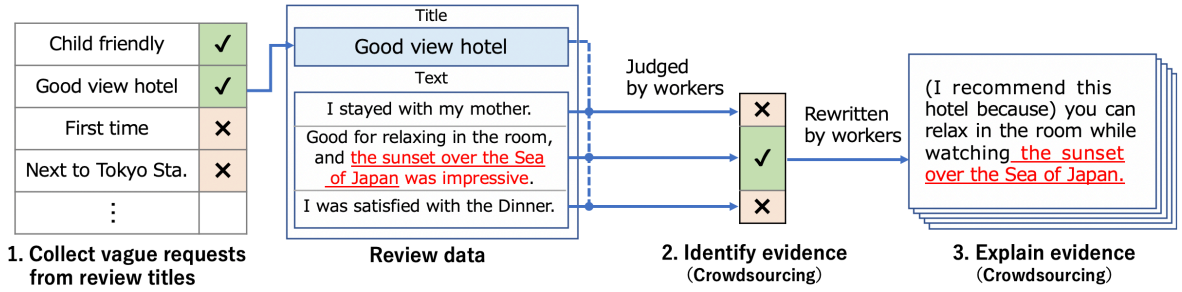

Figure 1: Pipeline for building the Evidence-based Explanation dataset

cific evidence, and generate an explanation (recommendation sentence) for the evidence.

To address these challenges, we decompose the process into two subtasks: *Evidence Identification* and *Evidence Explanation*. The former predicts whether a sentence contains evidence that expresses why a given request is satisfied. The latter generates a recommendation sentence given the evidence sentence. In order to focus on evidence explanations for requests, we assume that recommending hotels are given in advance in this study.

For these subtasks, we present an Evidence-based Explanation dataset, which is the largest dataset for explaining evidences in recommending hotels for vague requests. Assuming that titles of hotel reviews often correspond to vague requests, the dataset includes 37,280 hotel reviews with annotations for vague requests, evidence sentences for the requests, recommendation sentences based on the evidence sentences. The key feature of the dataset is the variety of requests: it includes 15,767 unique types of requests written in natural language. This dataset is publicly available[2].

We report experiments for the two subtasks in Section 3. We build a BERT (Devlin et al., 2019) model for the first subtask, which predicts whether a sentence contains evidence for a request. Experimental results show that the model can detect evidence sentence for various requests with a high (79.94) F1-score, and that the score does not drop so much even for requests unseen in the training data. We present encoder-decoder models for the second subtask, which rewrite an evidence sentence into a recommendation sentence. The experiments demonstrate that an LSTM (Luong et al., 2015) based model achieves the BLEU score (Papineni et al., 2002) of 56.09 with a gold evidence sentence given and that of 45.38 without a gold sentence (only a review and a request is given). We also report experiments when the two subtasks are combined to generate a recommendation sentence for a given review.

The contributions of this paper are as follows:

1. We propose a novel and practical task to explain evidences given vague requests expressed freely in natural language.
2. We create a new dataset by annotating review sentences with evidences and rewriting each evidence into a recommendation sentence. This is the largest dataset for explaining evidences in recommending hotels for vague requests.
3. Experiments show that our dataset enables to train models that can effectively find evidences to various vague requests and generate recommendation sentences.

2 Dataset Creation

In this section, we describe the procedure to create the Evidence-based Explanation dataset. The dataset is expected to include (i) vague requests from users, (ii) items (in this study, hotel candidates), (iii) evidence where an item satisfies an request, (iv) and a recommendation sentence based on each evidence. As a corpus that meets these requirements, we use review data on Jalan[3], which is a major hotel booking service in Japan.

On jalan, users can enter reviews after their stay at the hotel. In addition to review texts, Jalan accepts ratings for some specific aspects (e.g., 'Service' and 'Cleanliness'), similarly to other booking services (Wang et al., 2010). Although some aspects are similar to vague requests(e.g., "good service" or "cheap hotel"), the number of such pre-

[2] https://github.com/megagonlabs/ebe-dataset [3] https://www.jalan.net/

Category	Examples of requests		# of collection		# of annotations	
	With "inn" or "hotel"	Additional titles	"inn"	Additional	(Types)	
Clean	Clean hotel	Clean	15k	71k	3.6k	(0.8k)
Relax	Relaxing inn	Grate place to relax	8k	80k	3.6k	(1.1k)
Service	Helpful hotel	staff were very helpful	10k	143k	3.4k	(2.0k)
Useful	Useful inn	Useful for sightseeing	4k	113k	2.7k	(1.1k)
Child friendly	Child friendly hotel	Child friendly	3k	81k	2.6k	(1.3k)
Good view	Good view hotel	Good view	1k	34k	2.3k	(0.7k)
Delicious	Hotel with delicious food	Delicious dinner	1k	145k	2.3k	(1.0k)
Cost	Good low cost hotel	Low cost but very good	5k	89k	2.2k	(1.3k)
Good	Perfect hotel	Perfect	33k	278k	2.6k	(1.4k)
Others	Historic hotel	Historic atmosphere	19k	297k	11.8k	(5.2k)
Total	—	—	99k	1.3M	37.3k	(15.8k)

Table 1: Examples of collected vague requests and the number of collections, uses, and types

defined aspects is very limited and cannot cover diverse requests, such as "dog-friendly hotel."

Consequently, we created a new dataset using review titles and review texts. In the review texts, users describe their impressions on the service of the hotel based on their real experiences. Additionally, the review titles often summarize the most salient point of the experiences and often include similar expressions to vague requests such as "dog-friendly hotel." Hence, assuming that some review titles express vague requests and that the corresponding review texts contain evidence, we extracted vague requests from review titles and annotated evidence sentences for requests in review texts. Finally, we rewrote the evidence sentences into recommendation sentences.

Figure 1 illustrates the overall pipeline to construct the dataset. It consists of three steps.

1. **Collect vague requests from review titles:** Use rules to find review titles that correspond to vague requests.
2. **Identify evidence:** Ask crowdworkers to identify whether each review sentence contains evidence for the request corresponding to the review title.
3. **Explain evidence:** Ask crowdworkers to write recommendation sentences based on the evidence sentences.

2.1 Collecting Vague Requests

Based on the fact that some titles have similar expressions to vague requests, we collected vague requests by selecting review titles. Some review titles are inappropriate as requests, for example, "Thanks" or "Stayed for the first time." Therefore, to comprehensively collect vague requests for hotels with less noise, we first extracted review titles that included words representing accommodations such as "inn" or "hotel." In addition, we applied filtering rules to remove other unuseful titles[4].

Considering the possibility of data imbalance, we performed a categorical analysis. First, we applied morphological analysis of the collected requests using SudachiPy (Takaoka et al., 2018) to normalize surface variations in the requests. We manually checked and categorized all filtered titles appearing more than twenty times in the corpus, which resulted in ten categories of vague requests.

The distribution of categories in the dataset was skewed; the numbers of instances for some categories were small. For example, "Good hotel" is common but not "Hotel with delicious food." This is because a small percentage of requests appear with the expression "inn" or "hotel." Titles such as "Delicious dinner" are more frequent than "Hotel with delicious food." Therefore, we extracted additional titles that contained the same content words as the extracted titles, excluding the accommodation expressions such as "inn" or "hotel." For example, "Hotel with delicious food" → "delicious" (excluding hotel and extracting a content word) → "Delicious dinner" (additional titles).

Table 1 shows examples of vague requests collected from review titles. We extracted about 1.4 million reviews (99k + 1.3M) that have the collected requests in titles (# of collection). For annotation in the next subsection, we selected 37,280 reviews (# of annotations). By expanding the collection rules, the number of requests increased greatly, and the data imbalance problem reduced. Furthermore, it also increased the variation of the request expressions. Overall, we collected 15,767 unique kinds of titles in 37,280 reviews.

[4]The rules include, for example, titles must not contain proper nouns and must contain one or more content words.

	Clean	Relax	Service	Useful	Child	View	Delicious	Cost	Good	Others	All
Ratio of Relevant [%]	82.1	69.7	85.3	83.8	71.9	82.6	91.6	69.6	72.9	68.6	75.6
Ratio of Evidence [%]	48.3	55.4	71.4	74.1	58.8	60.1	68.6	44.3	56.0	46.1	55.3

Table 2: Ratio of relevant and evidence sentences included in the review text for a request

Amount of evidence sentences	# of reviews	
No evidence	16,654	(44.7%)
1 evidence sentence	16,456	(44.1%)
2 evidence sentences	3,382	(9.1%)
$\geq$ 3 evidence sentences	788	(2.1%)

Table 3: Amount of evidence sentences in each review

2.2 Evidence Identification Dataset

We used Yahoo Crowdsourcing[5] to annotate review data with evidence for requests. Workers were shown a review title and a single sentence of the review text. Then they were asked, "Is the following sentence relevant to the title, and does it contain evidence for the title?" There were three options for the answer: Evidence, Relevant (not as Evidence), and Irrelevant. Relevant (not as Evidence) means that the sentence contains the same expression as the request or its synonymous expression, but it does not present an evidence to support the request (title). Although the evidence may make sense by combining two or more sentences, we annotated each sentence of the review independently to simplify the annotation work. We annotated 37,280 reviews in total ("# of annotations" in Table 1). For a higher quality, each task was annotated by five people. We also prepared check questions for each task.

Table 2 reports the ratios of the Evidence and Relevant instances by category. In the 'Useful' category, 74% of the reviews contained evidence in the text, while only 44% of the reviews in the 'Cost' category did. This is because users apt to explain the reason for an 'useful' hotel in a review, but because the necessity of explaining the reason for 'cheap' hotel is relatively low.

Table 3 shows the number of evidence sentences for each review request. Approximately half of the reviews contained evidence. Requests that have a lot of evidence per review were an unique feature of this dataset. For example, requests that express general goodness such as "good hotel" have lots of evidence. In this case, the task of labeling evidence sentences was similar to annotation efforts for sentiment analysis.

2.3 Evidence Explanation Dataset

Using crowdsourcing, we rewrote evidence sentences into recommendation sentences. First, we showed workers a review title and an evidence sentence. Then we asked them to write a recommendation sentence so that the sentence can be used to explain the evidence in recommending the hotel to a user. We annotated 25,804 sentences that at least three of the five workers judged to contain evidence in Section 2.2. We asked workers to report the following two cases. (1) The request is a negative expression such as "bad view." (2) There is no evidence in a given sentence[6]. To ensure the quality of the annotation, each sentence was annotated by five workers, and we prepared check questions for each task. In the check questions, we prepared negative expressions for requests, and confirmed that the workers followed the instructions properly.

Table 4 shows the number of the exact matches of five workers for the created recommendation sentence. When only extracting a phrase from a review is sufficient as a recommendation sentence, the five workers tended to produce an identical result. On the other hand, when a certain part in a review had to be rewritten, recommendation sentences from the five workers tended to differ.

3 Experiments

Using the annotated dataset, we conducted two experiments. (1) Evidence Identification and (2) Evidence Explanation. The former predicts whether a sentence contains evidence for a request, whereas the latter generates a recommendation sentence.

[5]It is a microtask crowdsourcing service in Japan. We mixed some check questions in the tasks and receive annotated data from only workers who answered the check questions correctly. We did not set gender or attribute limits of workers in all our tasks. https://crowdsourcing.yahoo.co.jp/

[6]We targeted sentences where at least three people judged to contain evidence. However, it was sometimes difficult to write recommendation sentences when two out of five workers judged that the sentence has no evidence.

# of same answers	# of sent.	Examples		
		Title	Evidence sentence	Recommendation sentence
$\geq$ 2 matches	13,100	Pet-friendly	This hotel is tolerant of dog lovers because you can sleep in a bed with your dog.	(We recommend this) because you can sleep in a bed with your dog.
All different	9,889	Nice open-air bath	The temperature of the bath was just right, and we spent a long time in the open-air bath watching the stars.	(We recommend this) because you can take a long open-air bath while gazing at the stars.
Negative req. ($\geq$ 3)	1,651	The scenery ...	We booked a Bay Bridge view, but it was only visible from the edge of the window.	–
No Evidence	1,164	Mountain side view	It was an ocean view hotel but we stayed on the mountain side.	–

Table 4: Examples of recommendation sentences rewritten by workers and matching rate of rewriting

	Reviews	Sentences	Positive (%)
Train	29,826	148,671	20,709 (13.9)
Dev	3,726	18,549	2,606 (14.0)
Test	3,728	18,823	2,489 (13.2)
Total	37,280	186,043	25,804 (13.9)

Table 5: Evaluation data for evidence identification

3.1 Evidence Identification Task

Task Description The task is to predict whether or not a sentence contains an evidence for a request. This is a binary classification problem. A positive example is a sentence to which at least three out of the five workers labeled evidence. All other sentences are treated as negative examples.

We randomly divided the data by review into training, development, and test set (see Table 5). We used the same data split in all experiments.

Experimental Settings We explored logistic regression[7] and BERT (Devlin et al., 2019) as classification models. For the tokenization, we used juman++[8] (Tolmachev et al., 2018) and Byte pair encoding (BPE) (Sennrich et al., 2016) with the vocabulary size of 8k. We pre-trained word2vec (Mikolov et al., 2013) CBOW model, and the BERT model on two million review sentences in Jalan. For the logistic regression, we calculated the TF-IDF[9] (Jones, 1972) vector and the average vector of word2vec for requests and sentences respectively. We used the request vector, the sentence vector, and the difference between the two vectors as features. The input to the BERT model was in the following order: request sentence, `[SEP]`, and evidence sentence. Hyperparameters of each model were tuned by the F1-score on the development set.

[7]Implemented in: https://scikit-learn.org

[8]https://github.com/ku-nlp/jumanpp

[9]We used the word frequency in the sentence as TF, and the word frequency of the review text as DF.

Results and Analysis Table 6 reports F1-score of both models for each category and all categories. The F1-score of BERT for all the data was 79.94, which is 33.15 points higher than the logistic regression. Results in each category show that BERT had the highest F1-score for 'Useful' and the lowest for 'Good'. We analyze the results of the evidence identification by the BERT model from different perspectives in the following paragraphs.

Evidence Identification without Requests The F1-score of the BERT model was relatively high, considering the nature of this task, i.e., associating evidences to requests. However, we need to make sure whether the BERT model considers a request when identifying an evidence. Thus, we trained another BERT model without a request (only a sentence is given) as an input. The model trained without a request resulted in the F1-score of 43.22, which is 37 points lower than that with a request. This huge gap indicates that evidences in our dataset depend on requests and that the BERT model pays attention to requests properly.

For example, a model trained with a request predicts that the sentence, "It was pleasant in the room with a view of the sea" is evidence for a request "good view" but not for "good food". In contrast, a model trained without a request predicts that the both are evidence sentences.

Unseen Requests Since the dataset contains a wide range of requests, 30% of the requests in the test set are unseen, not appearing in the training set. Thus, we divided the test set in terms whether a request is unseen or not, and computed the F1-score in Table 7. Although the F1-score for unseen requests drops by 6.44 points, it is still high compared to the score trained without a request (described in the previous paragraph). This indicates that the model makes a successful prediction for

Model	Clean	Relax	Service	Useful	Child	View	Delicious	Cost	Good	Others	All
Logistic regression	44.39	46.03	51.21	61.30	49.39	61.02	54.34	30.24	34.76	37.15	46.79
BERT	79.52	82.89	84.98	89.48	85.04	81.23	82.54	73.59	68.85	73.89	79.94

Table 6: F1-score for evidence identification for each category

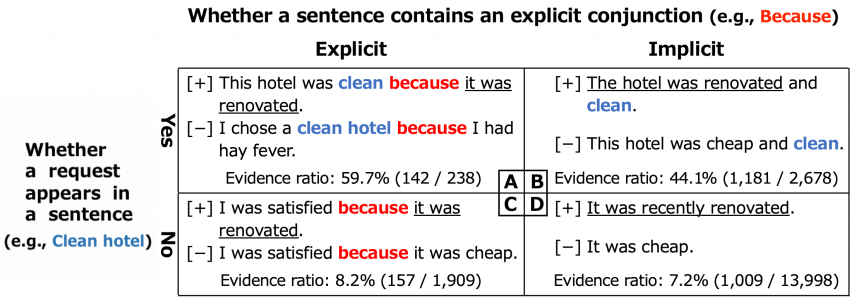

Figure 2: Characteristics of evidence sentences

Quadrant	F1
A	87.11
B	82.64
C	79.01
D	75.87
A+B	83.12
C+D	76.30
A+C	82.82
B+D	79.54
All	79.94

Table 8: F1-score for each quadrant

	F1	# of instances
Unseen requests	75.40	5,857
Seen requests	81.84	12,966

Table 7: F1-score for unseen/seen requests

majority of the unknown requests.

Examining successful predictions for unseen requests, we found that the same expression to the request often appears in the evidence sentence, For example, in response to a request for "*a good location to watch* a football game," the evidence sentence includes, "It's located in front of Tosu Station in Saga, and it's *a good location to watch* the Tosu football game." The expression in italic is considered to be a clue for predicting the evidence label for the sentence. The analysis of whether the request is included in the evidence sentence is discussed in detail in the next paragraph.

In contrast, we observed difficult instances as well. For example, the request (review title) is, "You can fully enjoy an *extraordinary* experience," and the evidence sentence is, "I was refreshed by soaking in a hot spring while listening to the chirping of birds and the sound of insects." The BERT model could not infer that the experience (hot spring, chirping birds) is *extraordinary* and that the sentence is an evidence for the request.

Characteristics of Evidence Sentences There are various ways to express an evidence sentence, for example, with and without a use of conjunctions. Figure 2 illustrates four categories (decomposed into two axes) of how a sentence presents an evidence for a request. The y-axis is whether a request expression appears in an evidence sentence. The x-axis is whether there is an explicit conjunction (e.g., 'because') expressing the discourse relation between a request and evidence. We have automatically divided these categories by rules.

The top-left quadrant A includes a request expression and an explicit conjunction in the sentence. Although 60% contain evidence for a request, quadrant A has the smallest volume. On the other hand, the lower-right quadrant D has the largest volume, but has the smallest ratio of including evidence for the request (only 7%). The evidence for quadrant A can be collected by a simple rule, but it is comprised of only about 6% of the total evidence. Our dataset successfully extracts other evidence expressions using the relationship between the review title and the text.

Table 8 shows the F1-score of the BERT model for each quadrant. The F1-score of quadrant A, which contains an explicit conjunction and request words, was highest (87.11). It was 7.17 points higher than the average F1-score of all test data. On the other hand, the F1-score of quadrant D, which does not contain an explicit conjunction nor any request words, was lowest (75.87). It was 4.07 points lower than the average F1-score of all test data. In addition, the F1-score of quadrant A+B was 6.82 points higher than the F1-score of quadrant C+D, indicating that the presence of the request expression in the evidence sentence significantly impacts on the performance of predicting evidence.

We examined successful cases in quadrant D, which is the most difficult of all. In these cases, we found that expressions similar to the requests often appear in the evidence sentence. For exam-

ple, in response to the request "I am soothed by a *meal*," the evidence sentence is "I was impressed by the deliciousness of the freshly made egg rolls for *breakfast*." In the example, the word 'meal' in the request is related to the word 'breakfast.' However, the model could not recognize that the sentence, "We have a foot washing place next to the entrance, gum roller and wet tissue, it was very thorough," contains an evidence for the request, "An inn where I can stay with my pet dog." This may be due to the lack of similar expressions for the request in the sentence, and the failure to associate dog and dog amenities.

3.2 Evidence Explanation Task

Task Description The task generates a recommendation sentence given request and evidence sentences. We used only the data that three or more workers rewrote into recommendation sentences in Section 2.3. Each evidence sentence had multiple recommendation sentences rewritten by the workers, and we use all of them as training data. We use BLEU (Papineni et al., 2002) to evaluate generated sentences.

Experiment Settings We compared three models: a rule-based model and two neural network models. The rule-based model rewrites an evidence sentence into a recommendation sentence by focusing on the root node in the parse tree of the evidence sentence. The rules include: if the root node is a verb, adjective, or auxiliary verb, add "because" at the beginning; if the root node is a noun, add "because of" at the beginning; and if the root node in an adverb, add "because you can do" at the beginning.

For neural network models, we employed an LSTM model with attention (Luong et al., 2015) and a Transformer model (Vaswani et al., 2017), assuming that the task is translation from an evidence sentence into a recommendation sentence. We used the FAIRSEQ (Ott et al., 2019) to implement the models. We tokenized it using Juman++ and BPE. The input to the model was in the following order: request sentence, `[SEP]`, and evidence sentence. Hyper-parameters of the models were tuned by the BLEU score on the development set.

For the evaluation, we used the BLEU score on sentences tokenized by Juman++ (not by BPE). Since the number of references for each evidence sentence was not constant, we randomly selected one.

Method	BLEU
No-rewrite	47.17
Rule-based	50.26
LSTM	56.09
Transformer	55.79

Table 9: BLEU score to generate recommendation given evidence and a request

Method	BLEU	F1
Pipeline (BERT → LSTM)	45.38	63.30
End-to-end (LSTM)	16.27	49.13

Table 10: BLEU score to generate recommendation given review text and a request

Results and Analysis Table 9 shows BLEU scores of generated recommendation sentences. 'No-rewrite' is the baseline where the evidence sentence is treated as the recommendation sentence without a rewrite. Compared with this baseline (47.17 BLEU), all generation methods obtained higher BLEU scores. The score of the LSTM-based model (56.09) was 0.30 points higher than that of the Transformer-based model (55.79). However, the BLEU score of the rule-based model was only 5.83 point lower than the LSTM-based model. This implies that this task requires fewer rewrites than we expected.

There are some differences between the outputs of the rule-based model and the LSTM-based model. The rule-based model tends to produce longer sentences because it cannot generate a sentence from scratch. In addition, the rule-based model fails when an evidence sentence includes unnecessary information, for example, "it's close to the station and it's convenient, so we'd like to use it again." The LSTM-based model could successfully generate "(We recommend this hotel) Because it's also close to the station and it's convenient," although the rule-base model kept "so we'd like to use it again" and generated, "Because it's close to the station and it's convenient, so we'd like to use it again."

3.3 End-to-end Experiment

In this section, we present an experiment to generate a recommendation sentence given review data (a request and review sentences) as an input. Combining the subtasks 1 and 2, this end-to-end experiment converts a hotel review into a recommendation sentence. When a given review does not have evidence, a system is expected to yield a special

Review title: The view was great! **Review text**: On the first day, I purchased a premium sightseeing ticket before check-in. Then I went to the Higashiyama Zoo to meet a handsome gorilla, Shabani. On the next day, I went to the Maglev and Railway Museum, and enjoyed it. In the room, where Nagoya Castle can be seen beautifully, the children were delighted. ... **Reference**: (I recommend this hotel) Because the room has a beautiful view of Nagoya Castle **Pipeline**: Because the children were overjoyed because they had a beautiful view of Nagoya Castle. **End-to-end**: Because you have a beautiful view of Nagoya Castle from your room.
Review title: Happy with my doggy **Review text**: The couple, father and two dogs used it. ... The rooms were spacious, and the large windows overlooked the sea. I was very satisfied. ... While many pet-friendly hotels are disappointing, this one is recommended. ... **Reference**: While many pet-friendly hotels are disappointing, this one is recommended **Pipeline**: Because pet-friendly hotels are very convenient **End-to-end**: Because the rooms were spacious the large windows overlooked the sea, and I was very satisfied
Review title: A hotel recommended for families with children **Review text**: Because I have a 16-month old child, I was drawn to the plan that included room service and a private hot spring... The mattress was thin because of its age, so it would have been better if it were thicker. The hot springs and customer service were good, and it was good that the staff treated my children kindly. **Reference**: Because the hot springs and customer service were good, and it was good for children. **Pipeline**: Because the hot springs and customer service were good, and it was good for children. **End-to-end**: Because the pool and customer service were good, and it was good for children.

Table 11: Examples of generating recommendation sentences given the review data

token `[no-evidence]`.

We explored two approaches, pipeline and end-to-end. The pipeline method is simply a combination of the models from Sections 3.1 and 3.2. The method first predicts whether a sentence in a review present an evidence for a request by using the BERT model. It then generates a recommendation sentence by using the LSTM-based model for the request and the predicted evidence sentence with the highest score assigned by the BERT model only when the review includes evidence sentences. If the BERT model predicts no sentence in the review as evidence, the method generates `[no-evidence]`.

The end-to-end method is an encoder-decoder LSTM model that directly generates a recommendation sentence given a review title and text. An input to the model is request and `[SEP]`, followed by multiple sentences of the review. When a review did not contain an evidence for the request, the model is trained to generate `[no-evidence]`.

Table 10 shows the BLEU scores and the macro-average F1-scores of the methods. The macro-average F1-score is defined similarly to the evaluation conducted by Rajpurkar et al. (2016)[10]. The pipeline method outperformed the end-to-end method, achiving a BLEU score of 45.38, 29.11 points higher than the end2end model. This is probably because the pipeline model could utilize the pre-trained BERT model and because training the end-to-end method was difficult with very long sequences of tokens given as inputs. In addition, the end-to-end method tends to output too many `[no-evidence]` and the total number of output words is low, so the BLEU score is also low due to brevity penalty.

Table 11 presents examples of the generated sentences. In the first example, the both models successfully generated appropriate recommendation sentences. Although the end-to-end method generated the natural sentence in the second example, the recommendation is nothing to do with the request, "happy with my doggy." In the third example, the end-to-end method generated the word "pool", which was actually false because the the review text only refers to "hot spring." We observed these incorrect generations from the end-to-end method more than from the pipeline method.

4 Related Work

Several studies addressed explainable recommendations (Sarwar et al., 2001; Diao et al., 2014; Zhao et al., 2014; Zhang et al., 2014; Wang et al., 2018; Zhang et al., 2020; Zhao et al., 2019). In feature-based explanations, Zhang et al. (2014) generated textual sentences as explanations using templates such as "You might be interested in [feature], on which this product performs well." In aspect-based explanations, Wang et al. (2010) discovered latent ratings on each aspect, and selected sentences related to each aspect to help users better understand the opinions given a set of review

[10]The metric measures matches of bag-of-tokens in the reference and generated sentences. For reviews without an evidence, we regard that the system output is correct if the generated output is no-evidence.

texts with the overall ratings. Zhao et al. (2019) formulated a problem called personalized reason generation and generated a recommendation sentence given a song name, author, and user tag as input. The inputs of those studies were user vectors created from the user's action history or limited aspects. However, our study deals with a wide range of natural language requests for a dialog system in the hotel booking domain.

In the field of sentiment analysis, research that extracts evidence based on sentiment expressions has attracted attention (Chen et al., 2010; Gui et al., 2016; Kim and Klinger, 2018). Chen et al. (2010) extracted the cause of a target emotional expression based on a rule. Gui et al. (2016) annotated an emotional expression and its cause. These studies aimed to gather useful information to extract emotional expressions and provide evidence simultaneously by examining the reputations for specific products. Although our study also aims to collect useful information, the requests are not limited to emotional expressions. In addition, we generate recommendation sentences.

Our study can be viewed as a special application of argument mining in the domain of hotel review. Liu et al. (2017) used manually annotated arguments of evidence-conclusion discourse relations in 110 hotel reviews. The study showed the effectiveness of several combinations of argument-based features. In Japanese, Murakami et al. (2009) proposed a method to collect consents and dissents for queries that can be answered with Yes or No. As part of that, they extracted evidence using rules. Our dataset is useful as training data to extract evidence in argument mining.

5 Conclusion

We proposed a novel task of predicting an evidence to satisfy a request and generating a recommendation sentence. We built an Evidence-based Explanation dataset for the task. The experimental results demonstrated that the BERT model could find evidence sentences with respect to various vague requests and that the LSTM-based model could generate recommendation sentences.

Future directions of this study include choosing the best evidence sentence from multiple candidate sentences for a vague request from a user and developing a concierge service that can recommend a hotel with evidence.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable feedback.

References

Ying Chen, Sophia Yat Mei Lee, Shoushan Li, and Chu-Ren Huang. 2010. Emotion cause detection with linguistic constructions. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, pages 179–187.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 4171–4186.

Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 193–202.

Lin Gui, Dongyin Wu, Ruifeng Xu, Qin Lu, and Yu Zhou. 2016. Event-driven emotion cause extraction with corpus construction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 1639–1649.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Evgeny Kim and Roman Klinger. 2018. Who feels what and why? annotation of a literature corpus with semantic roles of emotions. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pages 1345–1359.

Haijing Liu, Yang Gao, Pin Lv, Mengxue Li, Shiqiang Geng, Minglan Li, and Hao Wang. 2017. Using argument-based features to predict and analyse review helpfulness. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 1358–1363.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 1412–1421.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations (ICLR 2013)*.

Koji Murakami, Eric Nichols, Suguru Matsuyoshi, Asuka Sumida, Shouko Masuda, Kentaro Inui, and Yuji Matumoto. 2009. Statement map: assisting information crediblity analysis by visualizing arguments. In *Proceedings of the 3rd Workshop on Information Credibility on the Web (WICOW 2009)*, pages 43–50.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations (NAACL-HLT 2019)*, pages 48–53.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 311–318.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 4932–4942.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 2383–2392.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW 2001)*, pages 285–295.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1715–1725.

Kazuma Takaoka, Sorami Hisamoto, Noriko Kawahara, Miho Sakamoto, Yoshitaka Uchida, and Yuji Matsumoto. 2018. Sudachi: a Japanese tokenizer for business. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 2246–2249.

Arseny Tolmachev, Daisuke Kawahara, and Sadao Kurohashi. 2018. Juman++: A morphological analysis toolkit for scriptio continua. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2018)*, pages 54–59.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS 2017)*, pages 5998–6008.

Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 783–792.

Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable recommendation via multi-task learning in opinionated text data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 165–174.

Yongfeng Zhang, Xu Chen, et al. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends in Information Retrieval*, 14(1):1–101.

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 83–92.

Guoshuai Zhao, Hao Fu, Ruihua Song, Tetsuya Sakai, Zhongxia Chen, Xing Xie, and Xueming Qian. 2019. Personalized reason generation for explainable song recommendation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(4):1–21.

Xin Wayne Zhao, Yanwei Guo, Yulan He, Han Jiang, Yuexin Wu, and Xiaoming Li. 2014. We know what you want to buy: a demographic-based system for product recommendation on microblogs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1935–1944.

Darius Zumstein and Sophie Hundertmark. 2017. Chatbots–an interactive technology for personalized communication, transactions and services. *IADIS International Journal on WWW/Internet*, 15(1).