Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 653–661
Marseille, 11–16 May 2020
© European Language Resources Association (ELRA), licensed under CC-BY-NC

Understanding User Utterances in a Dialog System for Caregiving

Yoshihiko Asao[1], Julien Kloetzer[1], Junta Mizuno[1], Dai Saiki[2], Kazuma Kadowaki[1,2], Kentaro Torisawa[1,3]

[1]National Institute of Information and Communications Technology, Japan
[2]The Japan Research Institute, Limited, Japan
[3]Nara Institute of Science and Technology, Japan
[1]{asao,julien,junta-m,torisawa}@nict.go.jp
[2]{saiki.dai,kadowaki.kazuma}@jri.co.jp

Abstract

A dialog system that can monitor the health status of seniors has a huge potential for solving the labor force shortage in the caregiving industry in aging societies. As a part of efforts to create such a system, we are developing two modules that are aimed to correctly interpret user utterances: (i) a yes/no response classifier, which categorizes responses to health-related yes/no questions that the system asks; and (ii) an entailment recognizer, which detects users' voluntary mentions about their health status. To apply machine learning approaches to the development of the modules, we created large annotated datasets of 280,467 question-response pairs and 38,868 voluntary utterances. For question-response pairs, we asked annotators to avoid direct "yes" or "no" answers, so that our data could cover a wide range of possible natural language responses. The two modules were implemented by fine-tuning a BERT model, which is a recent successful neural network model. For the yes/no response classifier, the macro-average of the average precisions (APs) over all of our four categories (Yes/No/Unknown/Other) was 82.6% (96.3% for "yes" responses and 91.8% for "no" responses), while for the entailment recognizer it was 89.9%.

Keywords: dialog systems, caregiving, neural networks

1. Introduction

In aging societies, the caregiving industry is facing huge labor shortages.[1] Advanced natural language processing technologies could reduce the workload of case managers, who monitor the health of elderly people and plan appropriate caregiving services. We are participating in a project that is creating a dialog system that can not only ask seniors health-related questions, but also chit-chat with them to help prevent dementia. As a part of the project, we built two modules using BERT, a recent successful neural network model: (i) a **yes/no response classifier**, which is a module that recognizes user responses to health-related yes/no questions from the system, and (ii) an **entailment recognizer**, which is a module that detects health-related information voluntarily provided by a user.

We carefully designed a set of yes/no questions that are sufficient to monitor basic health statuses of seniors. However, natural languages allow many possible answers to yes/no questions. For example, in response to the question "Do you go out at least once a week?", a user might imply "yes" by saying "I see a doctor every Wednesday", instead of simply saying "yes". Moreover, a user may say "I don't know", or even ignore the question and bring up an irrelevant topic. Because our goal is to allow elderly people to have flexible conversations, our system must be able to interpret a wide range of possible natural language responses. This is why we need to develop a yes/no response classifier with advanced natural language processing technologies.

Successfully handled by our yes/no response classifier
System: お医者さんや薬局の人から薬を飲むときの注意点って説明してもらいましたか (Did your doctor or pharmacist explain how to take your medicine?) User: 必ずお水か白湯で飲んでくださいねって (They said I should take it with water.) Response category predicted by the system: YES
System: お住いの近辺で、何かの活動をしたり何かに参加することってありますか (Are you joining any neighborhood activities?) User: あまり興味がわかないんですけど (They don't really interest me.) Response category predicted by the system: NO
Successfully handled by our entailment recognizer
User: うがいなんて、気持ち悪い時にしかしませんよ (I gargle only when I feel sick.) The system detects that it entails: **The user does not regularly gargle.**
User: お薬の時間は家族が薬を持ってきて知らせてくれます (My family brings me my medicine when I need to take it.) The system detects that it entails: **The user communicates with his/her family; The user has family members who remind him/her to take his/her medicine;** etc.

Table 1: Examples of user utterances successfully recognized by our modules

We also need an entailment recognizer, our second module, because a user may voluntarily comment about her health status before the system asks the corresponding question. For example, in response to the question "Do you have a good appetite?", a user may say "Yes, I feel good these days. I have no problem with my blood pressure, either". In this example the user not only answers to the question but also provides information about her blood pressure, which the system has not asked yet. We expect that our system can detect such voluntarily provided information not only to record as much health information as possible, but also to appropriately manage the conversation flow. For example, it is inappropriate to ask "Is your blood pressure OK?" after the user says "My blood pressure is fine".

[1] https://www.mhlw.go.jp/stf/houdou/0000088998.html "On the estimated supply and demand of caregiving human resources towards the year 2025" (The Ministry of Health, Labour and Welfare, Japan) (in Japanese)

To apply machine learning approaches to the development of our modules, we manually constructed 280,467 question-response pairs and 38,868 voluntary utterances. We also propose a novel method to efficiently find entailment relations using *topic words*. With these datasets, we trained our yes/no response classifier and entailment recognizer using BERT. For the yes/no response classifier, we achieved an average precision (AP) of 96.3% for the "yes" responses and 91.8% for the "no" responses, even though we intentionally made the problem difficult by encouraging annotators to create responses that only indirectly said "yes" or "no". The macro-average of the APs over all of our four categories (Yes, No, Unknown and Other) was 82.6%. For the entailment recognizer, we achieved an AP of 89.9%. Table 1 shows a few illustrative examples that were successfully handled in our system. Although our data are in Japanese, we will translate them into English for presentation purposes.

This paper is organized as follows. In Section 2., we review related work. In Section 3., we explain our proposed data construction processes in detail. Section 4. evaluates our datasets with neural network experiments. Section 5. concludes our paper.

2. Related work

There have been a variety of proposals of dialog systems for health care (Laranjo et al., 2018), as well as those designed for elderly people speaking Japanese (Takahashi et al., 2002; Kobayashi et al., 2010). To the best of our knowledge, however, our work is the first to construct large-scale datasets of question-answer pairs and entailment relations dedicated to caregiving.

A number of dialog act annotation schemes that have tags for "yes" and "no" responses have been proposed (Core and Allen, 1997; Walker and Passonneau, 2001; Bunt et al., 2010). There are also human-machine dialog corpora annotated for dialog acts (Georgila et al., 2010; El Asri et al., 2017).

Existing textual entailment datasets include RTE-3 (Giampiccolo et al., 2007), SICK (Marelli et al., 2014) and SNLI (Bowman et al., 2015) for English, and the RITE-2 binary classification dataset (Watanabe et al., 2013) for Japanese.

We did not rely on these existing resources in this study, because in preliminary studies we found that models trained with the data dedicated for our system performed better than domain-general datasets.

3. Data Construction

This section explains how we constructed our datasets in detail.

3.1. Data Creation for the Yes/No Response Classifier

For the yes/no response classifier, our annotators created 280,467 question-response pairs based on 1,901 carefully designed *seed questions*. Our data creation process consists of the following three steps, which is also illustrated in Fig. 1.

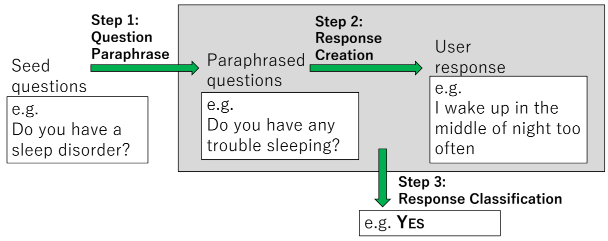
Step 1: Question Paraphrase
Seed questions
e.g. Do you have a sleep disorder?
Paraphrased questions
e.g. Do you have any trouble sleeping?
Step 2: Response Creation
User response
e.g. I wake up in the middle of night too often
Step 3: Response Classification
e.g. Yes

Figure 1: Data creation process for our yes/no response classifier. In Step 1, annotators paraphrase seed questions. In Step 2, they create responses to the paraphrased questions. In Step 3, they classify question-response pairs into four categories: Yes/No/Unk/Other.

Step 1: Question paraphrase. Annotators create paraphrases for the seed questions.

Step 2: Response creation. They create possible user responses to each paraphrased question.

Step 3: Response classification. They classify responses created in Step 2 into the following four categories: Yes, No, Unk and Other.

Each step is explained in more detail in the following subsections.

3.1.1. Step 1: Question Paraphrase

Before the annotation, we prepared a set of possible dialog scenarios from discussions with professional case managers based on the Care-Management Standard developed by the Japan Research Institute, Limited (JRI). The questions in the scenarios, called *seed questions* in this paper, were used for our current study. Seed questions were designed to cover basic health statuses, as well as statuses related to a few specific illness/disabilities. There are three different types of questions: (i) *basic* questions, which ask about the current status of the user; (ii) *change* questions, which ask about any changes in a user status from the last time the system was used; and (iii) *future* questions, which ask whether a status might change in the future. Examples of seed questions are shown in Table 2.

Type	Example
Basic	Do you go shopping by yourself?
Change	Has your shopping practice changed since the last time?
Future	Do you expect any changes in your future shopping practices?

Table 2: Seed questions

In Step 1, the annotators paraphrase the seed questions. We needed this step for a number of reasons. First, some seed questions are too formal or too technical and must be paraphrased into easier and more colloquial expressions for the dialogs. Second, by creating variations in the ways the system asks questions, it will become more interesting for users. Third, by preparing multiple paraphrases for each seed question, the system can try a different paraphrase when a user fails to understand a question.

While *basic* questions are mostly simple, *change/future* questions tend to be too complicated to ask out of the blue. Therefore, we asked the annotators to paraphrase them based on assumed contexts. For example, instead of paraphrasing "Do you expect any changes in your future shopping practices?" in isolation, annotators paraphrased it by assuming that the user has just answered "Yes" to the question "Do you go shopping by yourself?". In this context, the future question can be paraphrased as such sentences as "Would you like to keep going shopping by yourself?". This is more comprehensible than the abstract seed question. We specified contexts along with seed questions that were paraphrased when we asked the annotators to paraphrase them.

For the *basic* questions, nine annotators worked on each seed question, and each annotator created at most three paraphrases for each seed question. Thus, we created a maximum of 27 paraphrases for each. For the *change/future* questions, six annotators worked on each combination of a seed question and a context, and each annotator created only one paraphrase for each combination.

3.1.2. Step 2: Response Creation

In Step 2, the annotators created possible responses for each paraphrased question obtained in Step 1. We built five different datasets, which are summarized in Table 3.

Name	Description
Main	Main dataset for which annotators freely created possible responses, except that they were instructed to avoid "easy" responses.
Easy	Smaller set of such "easy" responses as "Yes", "I do", and "Of course".
NegP	Responses that negate a presupposition of the question
Sys	Comments about the system
Irrelev	Irrelevant utterances

Table 3: Datasets for the yes/no response classifier

For the **Main** dataset, annotators freely created possible responses to each paraphrased question without any restrictions, except the following rule. Because we expected that simple answers such as "Yes, I do" could be easily handled without constructing a large dataset, we asked annotators to avoid creating two kinds of simple answers in **Main**: (i) responses that mean "yes" or "no" regardless of the question's content (roughly corresponding to such English responses as "yes", "no", "of course" and "never"), and (ii) responses that repeat the question's predicate (roughly corresponds to English responses such as "I do" and "It is"). For the **Main** dataset, no other restrictions were placed on the response's content. A response can mean "yes" or "no" (e.g., "I see a doctor every Wednesday" in response to "Do you go out at least once a week?"), but it can also be an utterance that reflects a confusion (e.g., "I have no idea"), hesitation (e.g., "Well, let's see.."), a clarification question (e.g., "What do you mean by *on a regular basis*?"), and any other types of responses that do not provide an answer (e.g., "I guess my dentures must be replaced" in response to "Do you clean your dentures by yourself?").

We created a smaller, second dataset to cover simple answers excluded from **Main** and called it **Easy**.

We created the following three additional datasets to cover instances that are underrepresented in **Main** or **Easy** and/or of particular interest:[2]

1. **NegP**: responses that negate a presupposition of the question (e.g., "I never see a doctor" in response to "Are you following the suggestions of your doctor about taking your medicine?"). Since such responses may mean that the system has made an incorrect assumption on the user's background and needs to resart the conversation from scratch, they require special care.
2. **Sys**: comments about the system rather than answers to the questions (e.g., "You asked the same question before" and "Can you speak louder please?"). Although such responses are likely to occur when the system is operating, our **Main** dataset lacks them because we simply asked our annotators to create responses to questions without providing more details about what the system being developed would be like.
3. **Irrelev**: irrelevant utterances (e.g., "Oops, I forgot to turn off the air conditioner" in response to "Do you eat three meals a day?"). A user may ignore the question and say something irrelevant. Such utterances should not be misidentified as answers. Nonsense inputs can also result from erroneous voice recognition.

For the **Main** dataset, we used all the paraphrased questions; six annotators created responses to each of the *basic* questions, and three annotators created responses to each of the *change* and *future* questions. For the other datasets, we only used a subset of the paraphrased *basic* questions with 5,824 questions. For the **Easy** dataset, six annotators created one response to each question in it. For the remaining three datasets (**NegP/Sys/Irrelev**), only one annotator created a single response to each question.

3.1.3. Step 3: Response Classification

In Step 3, the annotators classified the question-response pairs obtained in Steps 1 and 2 into the following four categories: YES, NO, UNK, and OTHER. YES refers to cases where the response means "yes" to the question, regardless of whether the user directly responded with "yes", or only implied it. For example, "I see a doctor every Wednesday" in response to the question "Do you go out at least once a week?" is classified as YES. NO refers to cases where the response means "no". Again, the response can be both a direct "no" and responses that imply "no". For example, "I used to, but now that's tough" in response to the question "Do you go out at least once a week?" is classified as NO. UNK refers to cases where the response means "I don't know". More accurately, UNK refers to responses from which we can learn that the user does not know the answer to the question; responses indicating that the user fails to understand, ignores, or refuses to answer the question belong to OTHER rather than UNK. OTHER covers all

[2] Because we did not place any restrictions on responses' contents when we built **Main** and **Easy** (except that "easy" responses were prohibited in **Main**), they may contain a small number of instances that could belong to **NegP**, **Sys** or **Irrelev**.

responses that do not belong to Yes, No or Unk. Table 4 shows examples with different category labels.

Question	Response	Label
Do you go out at least once a week?	Of course.	Yes
Do you go out at least once a week?	I see a dentist every Wednesday.	Yes
Do you go out at least once a week?	Nope.	No
Do you go out at least once a week?	I used to, but now that's tough.	No
Do you go out at least once a week?	I'm not sure.	Unk
Do you go out at least once a week?	Does it matter?	Other

Table 4: Examples of answer-response pairs and their labels

For each question-response pair, the annotator who created the response in Step 2 and the annotator who classified the pair in Step 3 were always different. Three annotators independently annotated each question-response pair, and the final decision was made by majority vote. We discarded any question-response pairs for which all three annotators disagreed about the labels.

For the three additional datasets, **NegP**, **Sys** and **Irrelev**, we skipped Step 3, and automatically labeled them as Other because the response types were specified for them in advance.

3.1.4. Data Statistics

Table 5 summarizes the numbers of the questions as well as the question-response pairs we constructed.

# of questions

Type	#seed questions	#paraphrased questions
Basic	644	10,517
Change	623	7,616
Future	634	7,516
Total	1,901	25,649

# of question-response pairs with category labels

	Yes	No	Unk	Other	Total
Main	132,628	76,516	2,736	20,368	232,248
Easy	19,038	14,012	55	170	33,275
NegP	-	-	-	2,600	2,600
Sys	-	-	-	6,172	6,172
Irrelev	-	-	-	6,172	6,172
Total	151,666	90,528	2,791	35,482	280,467

Table 5: Summary of annotated data sizes for the yes/no response classifier

Fleiss' κ (Fleiss, 1971) calculated over all the instances of our response classification task was 0.742, which indicates substantial inter-annotator agreement (Landis and Koch, 1977).[3]

3.2. Data Creation for the Entailment Recognizer

The goal of our entailment recognizer is to detect a piece of information voluntarily provided by a user that is equivalent to a "yes" or "no" response to one of our yes/no questions. We refer to such a piece of information as a *statement*. For example, when a user says "I go out every Wednesday", it entails the statement "the user goes out every week", which corresponds to a "yes" response to the question "Do you go out at least once a week?". Similarly, when a user says "I go out only once a month", it entails the statement "the user goes out less than once a week", which corresponds to a "no" response to the same question.

We first created 1,206 statements from our *basic* seed questions.[4] Based on this set, we built our dataset in the following two steps:

Step 1: Utterance creation. For each statement, annotators create possible utterances that entail the statement.

Step 2: Entailment classification. An utterance may entail more than one statement. To find such cases, we sampled pairs of statements and the annotators judged whether they have an entailment relation.

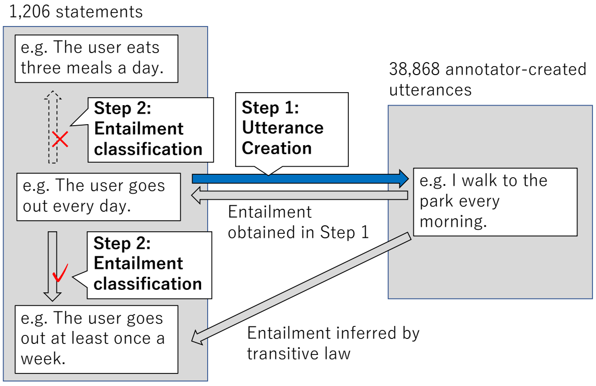
1,206 statements
e.g. The user eats three meals a day.
Step 2: Entailment classification
Step 1: Utterance Creation
38,868 annotator-created utterances
e.g. The user goes out every day.
e.g. I walk to the park every morning.
Entailment obtained in Step 1
Step 2: Entailment classification
e.g. The user goes out at least once a week.
Entailment inferred by transitive law

Figure 2: Data creation process for our entailment recognizer. In Step 1, annotators create utterances that entail statements. In Step 2, they judge whether statement pairs have an entailment relation. Results are used to expand positive utterance-statement entailment relations with the transitive law.

3.2.1. Step 1: Utterance Creation

The annotators created utterances that entail each statement. Examples are shown in Table 6. Twelve annotators worked on this task; each annotator created three utterances for each of the 1,206 statements. After cleaning duplicate utterances, we obtained 38,868 utterances and 38,999 entailment pairs.

[3] After the initial annotation was completed, another group of annotators re-examined a part of our data and we updated the labels of 8,086 items (3.0%). Our calculation of κ is based on the initial annotation results.

[4] We only deal with *basic* questions for the entailment recognizer and leave *change/future* questions for future work. The number of the statements does not simply equal twice the number of the *basic* questions, because our data contains statements that correspond to the result of a sequence of multiple questions.

statement	utterance
The user eats with his/her family every day.	I have dinner with my daughter every day.
The user does not eat with his/her family every day.	I only eat meals with my family on the weekends.

Table 6: Examples of statement-utterance pairs

3.2.2. Step 2: Entailment Classification

In the dataset that we created in Step 1, each utterance is associated with a single statement. However, an utterance may entail more than one statement. For example, “I go to the park every morning” entails both “the user goes out every day” and “the user goes out at least once a week”. It is, however, impractical to manually check every utterance-statement pair because we have more than 10 million such pairs. Instead, the annotators judged whether an entailment relation existed *between statements*, and then we expanded utterance-statement entailment pairs using the transitive law. The idea is illustrated in Fig. 2. If we know that the utterance “I walk to the park every morning” entails the statement “the user goes out every day” and that the statement “the user goes out every day” entails another statement, “the user goes out at least once a week”, we can infer that the utterance “I walk to the park every morning” entails “the user goes out at least once a week”.

Because the number of statement pairs is also very large, we created two subsets for annotation using different approaches: (i) a *word-overlap* subset and (ii) a *topic-word* subset. The former consists of statement pairs that share at least one content word (noun, verb or adjective). The latter is sampled from statement pairs that show high similarity in a sentence similarity measure, which we describe below. Both approaches are based on the idea that the more similar a pair of sentences is, the more likely they are to have an entailment relation.

We used the following similarity measure for creating the topic-word subset. For any pair of sentences S_1 and S_2, the similarity is measured by the probability that S_2 occurs in the same context as S_1. We used a topic word model that predicts $(t|S)$, the probability that a topic word t occurs in the context of sentence S, to estimate the similarity between sentences S_1 and S_2:

$$\begin{aligned} P(S_1|S_2) &\approx \sum_t P(S_1|t)P(t|S_2) \\ &= \sum_t \frac{P(t|S_1)P(S_1)P(t|S_2)}{P(t)} \end{aligned}$$

Assuming a uniform distribution for $P(S)$, we obtain:

$$P(S_1|S_2) \propto \sum_t \frac{P(t|S_1)P(t|S_2)}{P(t)}$$

This probability was used as the similarity measure between S_1 and S_2.

As a set of topic words, we used the 10,000 most frequent nouns in a database that we use for WISDOM X (Mizuno et al., 2016), which is a domain-general question-answering system being developed by our team. The input to our topic word model is a sentence and the outputs are a probability distribution over 10,001 items, where the first 10,000 items correspond to the 10,000 nouns, and the last item is a special output that represents “no noun”. Our topic word model is based on BERT[5] (Devlin et al., 2019), where a training instance is a sentence with a list of the nouns that appear either in it, in the previous sentence or in the next sentence. We trained it with 20 million training instances extracted from the same corpus as we used to pretrain the BERT model.

For both the word-overlap and topic-word subsets, three annotators independently judged whether each pair has an entailment relation, and the final decision was made by majority vote. For the word-overlap subset, the annotators annotated all 33,988 statement pairs. For the topic-word subset, we randomly sampled 47,374 instances from the top 500,000 statement pairs ranked by the similarity measure and excluded 2,626 instances that were already included in the word-overlap subset. The annotators annotated the remaining 47,374 instances.[6] After the annotation was completed, we extended our entailment pairs using the transitive law $(A \rightarrow B) \wedge (B \rightarrow C) \Rightarrow (A \rightarrow C)$. The transitive law was recursively applied until no new entailment relation was found.

An example of newly found entailment relations in the topic-word subset is “the user has an overdose” → “the user does not follow the rules when taking her medicine”. By the transitive law, we can infer, for example, that the utterance “I sometimes forget I’ve just taken my medicine and take it again”, which entails “the user has an overdose”, also entails the statement “the user does not follow the rules when taking her medicine”.

3.2.3. Data Statistics

In Step 1, we obtained 38,999 entailment pairs. In step 2, we first obtained 622 entailment relationships: 506 from the 33,988 statement pairs in the word-overlap subset and 116 from the 47,374 statement pairs in the topic-word subset. By recursively applying the transitive law, the number of entailment relations between statements increased from 622 to 891, and the number of positive utterance-statement pairs rose from 38,999 to 65,157.

Table 7 summarizes the numbers of utterance-statement pairs we constructed. We call the data created by Step 1 only *Original*, and the data extended by Step 2 *Extended*. For the experiments we assumed that every utterance-statement pair that was not known to be positive was nega-

[5] The model was pretrained with the same method as our modules in Section 4., except that we used a smaller corpus and a few different settings. The model was pretrained for five million steps with a maximum sequence length of 128 using an Adam optimizer with a batch size of 50, a learning rate of 1e-4 and a warmup rate of 1%. As a pretraining corpus, we used 195,674,025 sentences, which was about half the size of the pretraining corpus that we used for building our modules.

[6] A pair of statements corresponding to “yes” and “no” responses to the same question never has an entailment relation. For example, we know in advance that “the user has a sleep disorder” does not entail “the user does not have a sleep disorder”. Such pairs were excluded before creating the subsets.

tive.[7] Due to this assumption, we have many more negative than positive instances.[8]

	Positive	Negative	Total
Original (Step 1 only)	38,999	46,835,809	46,874,808
Extended (Steps 1 and 2)	65,157	46,809,651	46,874,808

Table 7: Summary of annotated data sizes for entailment recognizer

Fleiss' κ calculated over all the instances of our entailment classification task was 0.481, which indicates moderate inter-annotator agreement.

4. Neural Network Experiments

This section describes our neural network experiments based on the data explained in the previous section.

Both the yes/no response classifier and the entailment recognizer were created by fine-tuning the same pretrained BERT model. BERT (Devlin et al., 2019) is a pretrained language model that was developed using Transformer techniques (Vaswani et al., 2017), which is the best-performing model currently available to us in Japanese. BERT follows a recent trend; it is pretrained for language modeling first and then fine-tuned for a specific task.

The BERT model in our modules was pretrained from scratch, following Devlin et al.'s (2019) $\text{BERT}_{\text{BASE}}$ settings, except that we used a batch size of 1,024 and a vocabulary size of 100,000 (resulting in 163M parameters). The model has 12 layers, 768 hidden states and 12 heads. The model was first trained for one million steps with a maximum sequence length of 128 using an Adam optimizer with a learning rate of 1e-4 and a warmup rate of 1%, and trained for 100,000 additional steps with a maximum sequence length of 512 and a learning rate of 2e-5. We used two and five NVIDIA V100 GPUs for the first and second pretraining phases, both with mixed precision (Micikevicius et al., 2018).

As a pretraining corpus, we tried a variety of available corpora and found that the model pretrained for event causality recognition (Kadowaki et al., 2019) performed best for our task as well. Following their approaches, we used 400,765,020 sentences extracted from 46,564,280 passages, where each passage, consisting of seven sentences, contained at least one event causality detected by a CRF-based causality recognizer (Oh et al., 2013) from four billion web pages.

We used the morphological analyzer MeCab (Kudo et al., 2004)[9] and the dictionary JumanDic (Kurohashi et al., 1994) to tokenize Japanese sentences into words throughout this study.

4.1. The Yes/No Response Classifier

4.1.1. Procedure

For the yes/no response classifier, we used a BERT model with two input segments, where Segment 1 corresponds to a question and Segment 2 corresponds to a response. The two segments are concatenated with special token [SEP] and each segment is given distinct segment embeddings. The output is a category label, which is YES, NO, UNK or OTHER. We divided our datasets into training, validation, development and test bins such that the question-response pairs originating from the same seed question belong to the same bin. The numbers of instances for our experiments are summarized in Table 8.

	Train	Val	Dev	Test	Total
Main	171,860	20,625	19,774	19,989	232,248
Easy	24,256	3,155	2,816	3,048	33,275
NegP	1,900	245	207	248	2,600
Sys	4,497	584	520	571	6,172
Irrelev	4,497	584	520	571	6,172
Total	207,010	25,193	23,837	24,427	280,467

Table 8: Instances in our dataset for yes/no response classifier

In each experiment, we trained the model with an Adam optimizer with a batch size of 32. We searched for the best hyperparameters from all the combinations of the learning rates of 1e-5, 2e-5, 3e-5, 4e-5 and 5e-5, and epoch numbers of 1, 2, 3, 5, 10 and 20. We measured a model's performance with the macro-average of average precision (AP) over each category (YES/NO/UNK/OTHER). We selected the best model by the performance on the development data and reported its performance on the test data.

4.1.2. Results

Training set	Test set	AP (%)				
		YES	NO	UNK	OTHER	macro avg.
Main	Main	95.4	**89.6**	**64.6**	**77.3**	**81.7**
Easy	Main	85.3	69.5	21.7	54.9	57.5
Main+Easy	Main	**95.5**	89.5	64.3	77.0	81.5
Main	Easy	98.3	97.7	83.0	60.4	84.9
Easy	Easy	99.4	99.0	67.1	81.7	86.8
Main+Easy	Easy	**99.6**	**99.3**	**87.4**	**91.1**	**94.4**
Main	Main+Easy	96.1	**92.0**	64.4	75.9	82.1
Easy	Main+Easy	87.4	76.5	29.3	56.6	62.5
Main+Easy	Main+Easy	**96.3**	91.8	**65.1**	**77.0**	**82.6**

Table 9: Experimental results for yes/no response classifier

Table 9 shows the experimental results for our yes/no response classifier when we trained and tested models with **Main**, **Easy** or both. When both the training and test sets are from **Main**, the YES category achieved an AP score of 95.4%, and the NO category has an AP score of 89.6%. UNK and OTHER have lower AP scores, and the macro-average over the four categories was 81.7%.

When we built the **Main** dataset, we asked annotators not to include such simple answers as "yes" and "I do". Nonetheless, the model trained with **Main** performed better when it is tested with **Easy**, which consists of such simple answers, than when it was tested with **Main**. This is possibly because a small number of erroneous inclusion of simple answers, or the similarity between the simple answers and more complex answers were sufficient to predict **Easy** with a considerably high precision.

[7] To estimate the number of noises introduced by this assumption, annotators checked 5,000 instances that were randomly sampled from the negative instances in Original. We found that 15 (0.3%) out of the 5,000 utterance-statement pairs were positive, which indicates that around 140,000 instances are falsely labeled as negative in Original.

[8] In preliminary studies, we also created smaller datasets by randomly sampling negative instances, but this did not contribute to the overall performance in neural network experiments.

[9] https://taku910.github.io/mecab/

When we used only **Easy** as the training data, the model performed poorly when it was tested with **Main**, suggesting that training with **Easy** alone is not sufficient to correctly classify a wide range of possible responses. When we used both **Main** and **Easy** as the training data (**Main+Easy** in Table 9), it achieved the best result when tested with **Easy**; the model achieved an average precision over 99% for both Yes and No. It did not significantly affect the **Main** result. This suggests that our approach in which we created a large dataset of 'difficult' responses and a smaller dataset of 'easy' responses was effective. We will use the model trained with **Main+Easy** as our baseline in subsequent discussions.

	Train	Val	Total
Gen	33,933	1,054	34,987
Main+Easy.Small	33,933	1,054	34,987

Table 10: Sizes of **Gen** and **Main+Easy.Small**

Training set	AP (%)				
	Yes	No	Unk	Other	macro avg.
Main+Easy (baseline)	96.3	91.8	**65.1**	**77.0**	**82.6**
Main+Easy.Small	94.7	88.1	54.4	71.4	77.2
Gen	82.6	68.5	28.3	5.7	46.3
Main+Easy+Gen	**96.4**	**92.2**	62.4	76.6	81.9

Table 11: Results of models trained with **Gen** and tested with **Main+Easy**

To see whether our construction of domain-specific annotated data was crucial for our goal, we trained the models with a domain-general yes/no question-answer dataset called **Gen**. This dataset was created in the following manner. First, 9,986 domain-general yes/no questions were extracted from four billion web pages. Next, for each of them, annotators created the following four kinds of responses: Yes, No, Unk and Other.[10] The numbers of **Gen** instances are shown in Table 10.

Because **Gen** is smaller than **Main+Easy**, we could not easily judge whether the performance differences between them were due to the difference in quality or in size. To make them directly comparable, we also created **Main+Easy.Small**, for which instances were randomly sampled from **Main+Easy** such that its size matches that of **Gen**.

Our results are summarized in Table 11. The model trained with **Gen** has significantly lower AP scores than those of both **Main+Easy** and **Main+Easy.Small**, suggesting that building domain-specific data plays an essential role for our goal. The addition of **Gen** to **Main+Easy** did not improve the performance either.

Below we discuss the experimental results with the three additional datasets: **NegP**, **Sys** and **Irrelev**.

NegP is a set of responses that negate a question's presupposition. When a user's response belongs to this category, the system might have wrong assumptions about the user, and that it must start over with correct background information. Thus, instances of **NegP** should be distinguished from all other categories. To see whether it is possible, we temporarily made a fifth category, NegP, and gave this label to all instances of **NegP**, while the instances from the other datasets remained unchanged. Our results are summarized in Table 12. The model predicted the **NegP** category with an AP of 83.4%, which is slightly over the macro-average.

Dataset	AP (%)					
	Yes	No	Unk	Other	NegP	macro avg.
Main+Easy+NegP	95.9	91.2	69.6	76.4	83.4	83.1

Table 12: Results of models trained and tested with **NegP**

Training set	AP (%)				
	Yes	No	Unk	Other	macro avg.
Main+Easy	96.1	91.5	64.7	79.3	82.9
Main+Easy+Sys	96.3	92.0	64.2	83.3	**83.9**

Table 13: Results tested with **Main+Easy+Sys**

Training set	(incorrect)			(correct)	
	Yes	No	Unk	Other	Total
Main+Easy	148	77	2	344	571
Main+Easy+Sys	37	26	0	508	571

Table 14: Model predictions for the **Sys** test set

Tables 13 and 14 summarize the results of our experiments with **Sys**, which is the set of responses that are comments about the system rather than answers to questions. When a model is trained with **Main+Easy**, only 334 (58.5%) out of 571 **Sys** instances were correctly classified as Other. The result was significantly improved by adding the **Sys** training data: 508 (89.0%) out of 571 instances were correctly classified. This suggests that **Main** lacks responses like "You asked the same question before", but this issue was greatly improved by adding the **Sys** dataset.

Next we ran experiments with **Irrelev**, which is the set of responses that are irrelevant to the question. Our results with it are summarized in Tables 15 and 16. When trained with **Main+Easy**, only 317 (55.5%) out of 571 **Irrelev** instances were correctly classified as Other. When we added the **Irrelev** training data, 530 (92.8%) out of 571 instances were correctly classified, and the overall performance was improved as well.

One might think that instances of irrelevant responses could be inexpensively constructed using utterances from an unrelated dataset. Based on this idea, we created pseudo training data called **Rand** in the following manner. Instances of random utterances were taken from annotator-created utterances for a domain-general dialog system, WEKDA. We randomly selected three domain-general utterances for each of the 25,645 paraphrased questions, and labeled all of them as Other. We obtained 76,935 pseudo training data.

By using **Main+Easy+Rand** instead of **Main+Easy** as the training set, the correctly classified **Irrelev** test instances increased from 317 (55.5%) to 476 (83.4%), as in Table 16, which suggests that pseudo training data somewhat effectively handled the irrelevant utterances. However, the addition of **Rand** was not as effective as the addition of **Irrelev** in predicting the **Irrelev** test set, even though **Rand** is larger

[10] When we constructed **Gen**, the definitions of Unk and Other were slightly different; for example, Unk includes not only cases in which the user did not know the answer, but also cases in which the user did not understand the question. These differences might have negatively affected the performance of models trained with **Gen**.

Training set	AP (%)				
	Yes	No	Unk	Other	macro avg.
Main+Easy	96.1	91.6	**65.1**	78.8	82.9
Main+Easy+Irrelev	**96.3**	91.9	64.6	**84.0**	**84.2**
Main+Easy+Rand	**96.3**	**92.0**	61.3	74.4	81.0
Main+Easy+Irrelev+Rand	**96.3**	91.9	65.0	82.5	83.9

Table 15: Results tested with **Main+Easy+Irrelev**

Training set	(incorrect)			(correct)	
	Yes	No	Unk	Other	Total
Main+Easy	175	79	0	317	571
Main+Easy+Irrelev	28	13	0	530	571
Main+Easy+Rand	61	34	0	476	571
Main+Easy+Irrelev+Rand	23	14	0	534	571

Table 16: Model predictions for **Irrelev** test set

than **Irrelev**. It was also accompanied with slight overall performance degradation, as shown in Table 15. This suggests that while **Rand** is more inexpensive, the manually constructed **Irrelev** dataset serves our goal better.

4.2. The Entailment Recognizer

4.2.1. Procedure

For the entailment recognizer, we used the same pretrained BERT model as for the yes/no response classifier. We tried two designs in our experiments: (i) a two-segment design, where the input is a concatenated sequence of an utterance and a statement, and the output is a single binary classification label, and (ii) a multi-label design, where the input is an utterance only, and the output is 1,206 labels each of which corresponds to a statement. An advantage of the two-segment design is that we can flexibly add new statements when it is in use, while in the multi-label design the number of statements must be fixed before training; however, the multi-label design is faster both in training and in prediction because the 1,206 statements share computation, except for the final output layer.

In all the experiments, we divided our dataset into training, validation, development and test bins as shown in Table 17. The table shows the number of utterances in each bin; the number of data points is 1,206 statements multiplied by the number of utterances.

Train	Val	Dev	Test	Total
29,891	2,989	2,990	2,998	38,868

Table 17: Instances in our dataset for entailment recognizer

In our dataset, the proportion of positive instances is very small, as we showed in Table 7. For a model not to miss rare positive instances, we used a rescaled cross-entropy loss function, where false-negative errors are more heavily penalized than false-positive errors. More precisely, we multiplied the losses for false-negative and false-positive errors by p and $1 - p$ respectively ($0.5 \leq p < 1$). We tried different settings of p as a part of a hyperparameter search. For the two-segment design, we trained models with an Adam optimizer with a batch size of 256. We searched for the best hyperparameters from all the combinations of the learning rates of 1e-5, 2e-5, 3e-5, 4e-5 and 5e-5, epoch numbers of 1, 2, 3 and 5, and p, a loss multiplier for the false-negative errors, of 0.5, 0.9, 0.99 and 0.999. For the multi-label design, we trained the models with an Adam optimizer with a batch size of 32. We searched for the best hyperparameters from all the combinations of the learning rates of 1e-5, 2e-5, 3e-5, 4e-5 and 5e-5, epoch numbers of 20, 50, 100 and 200, and a p of 0.5, 0.9, 0.99 and 0.999. We measured the performance of a model with the macro-average of average precision (AP) over the 1,206 statements. We selected the best model by the performance on the development data and reported its performance on the test data.

4.2.2. Results

Dataset	Design	AP (macro avg.) (%)
Original	Multi-label	84.4
	Two-segment	**86.2**
Extended	Multi-label	88.3
	Two-segment	**89.9**

Table 18: Experimental results for the entailment recognizer

We conducted experiments with Original, which is the result of Step 1 in Section 3.2.1., and Extended, which has more positive entailment pairs discovered by the entailment classification annotation and the transitive law (Section 3.2.2.). Our experimental results are summarized in Table 18. The two-segment model slightly outperformed the multi-label model in both datasets. We achieved an AP of 86.2% with Original, and 89.9% with Extended.

5. Conclusion

We created large annotated datasets to help develop a dialog system that can monitor the health statuses of seniors. Our datasets consist of 280,467 question-response pairs and 38,868 voluntary utterances. We evaluated them with BERT-based models; Our yes/no response classifier correctly classifies a user's response to a yes/no question with an average precision of 82.6%, despite the fact that our **Main** dataset, which accounts for over 80% of our data, was intentionally complicated by prohibiting annotators from creating such simple answers as "yes" or "I do". We also built a module that detects a user's voluntary mentions about their health statuses and classify them to 1,206 categories, with an average precision of 89.9%.

So far we relied on annotator-created examples, but we expect that annotators and real senior users are different in many aspects (Georgila et al., 2010). As the system continues to develop, we expect that actual usage data can be employed to improve the system further.

6. Acknowledgements

This study was supported by the Cross-ministerial Strategic Innovation Promotion Program (SIP) of the Council of Science, Technology and Innovation (CSTI), Japan: "A Hybrid Multimodal Dialogue System Using Big Data on the Internet and Application-Specific Dialogue Scenarios".

7. Bibliographical References

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Bunt, H., Alexandersson, J., Carletta, J., Choe, J.-W., Fang, A. C., Hasida, K., Lee, K., Petukhova, V., Popescu-Belis, A., Romary, L., Soria, C., and Traum, D. (2010). Towards an ISO standard for dialogue act annotation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA).

Core, M. G. and Allen, J. (1997). Coding dialogs with the DAMSL annotation scheme. In *AAAI Fall Symposium on Communicative Action in Humans and Machines*, volume 56, pages 28–35. Boston, MA.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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, Volume 1 (Long and Short Papers)*, pages 4171–4186.

El Asri, L., Schulz, H., Sharma, S., Zumer, J., Harris, J., Fine, E., Mehrotra, R., and Suleman, K. (2017). Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219, Saarbrücken, Germany, August. Association for Computational Linguistics.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Georgila, K., Wolters, M., Moore, J. D., and Logie, R. H. (2010). The MATCH corpus: a corpus of older and younger users' interactions with spoken dialogue systems. *Language Resources and Evaluation*, 44(3):221–261.

Giampiccolo, D., Magnini, B., Dagan, I., and Dolan, B. (2007). The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, June. Association for Computational Linguistics.

Kadowaki, K., Iida, R., Torisawa, K., Oh, J.-H., and Kloetzer, J. (2019). Event causality recognition exploiting multiple annotators' judgments and background knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5815–5821, Hong Kong, China. Association for Computational Linguistics.

Kobayashi, Y., Yamamoto, D., Koga, T., Yokoyama, S., and Doi, M. (2010). Design targeting voice interface robot capable of active listening. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 161–162. IEEE.

Kudo, T., Yamamoto, K., and Matsumoto, Y. (2004). Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, pages 230–237.

Kurohashi, S., Nakamura, T., Matsumoto, Y., and Nagao, M. (1994). Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of The International Workshop on Sharable Natural Language*, pages 22–28.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Laranjo, L., Dunn, A. G., Tong, H. L., Kocaballi, A. B., Chen, J., Bashir, R., Surian, D., Gallego, B., Magrabi, F., Lau, A. Y. S., and Coiera, E. (2018). Conversational agents in healthcare: a systematic review. *Journal of the American Medical Informatics Association*, 25(9):1248–1258, 07.

Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S., and Zamparelli, R. (2014). SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8, Dublin, Ireland. Association for Computational Linguistics.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., and Wu, H. (2018). Mixed precision training. In *International Conference on Learning Representations*.

Mizuno, J., Tanaka, M., Ohtake, K., Oh, J.-H., Kloetzer, J., Hashimoto, C., and Torisawa, K. (2016). WISDOM X, DISAANA and D-SUMM: Large-scale NLP systems for analyzing textual big data. In *Proceedings of the 26th International Conference on Computational Linguistics*.

Oh, J.-H., Torisawa, K., Hashimoto, C., Sano, M., De Saeger, S., and Ohtake, K. (2013). Why-question answering using intra- and inter-sentential causal relations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1733–1743.

Takahashi, S., Morimoto, T., Maeda, S., and Tsuruta, N. (2002). Spoken dialogue system for home health care. In *Seventh International Conference on Spoken Language Processing*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems 30*, pages 6000–6010.

Walker, M. and Passonneau, R. (2001). DATE: A dialogue act tagging scheme for evaluation of spoken dialogue systems. In *Proceedings of the First International Conference on Human Language Technology Research*.

Watanabe, Y., Miyao, Y., Mizuno, J., Shibata, T., Kanayama, H., Lee, C.-W., Lin, C.-J., Shi, S., Mitamura, T., Kando, N., Shima, H., and Takeda, K. (2013). Overview of the recognizing inference in text (RITE-2) at NTCIR-10. In *Proceedings of the 10th NTCIR Conference*, pages 385–404.