Chat Detection in an Intelligent Assistant: Combining Task-oriented and Non-task-oriented Spoken Dialogue Systems

Satoshi Akasaki[*]
The University of Tokyo
akasaki@tkl.iis.u-tokyo.ac.jp

Nobuhiro Kaji
Yahoo Japan Corporation
nkaji@yahoo-corp.jp

Abstract

Recently emerged intelligent assistants on smartphones and home electronics (*e.g.*, Siri and Alexa) can be seen as novel hybrids of domain-specific task-oriented spoken dialogue systems and open-domain non-task-oriented ones. To realize such hybrid dialogue systems, this paper investigates determining whether or not a user is going to have a chat with the system. To address the lack of benchmark datasets for this task, we construct a new dataset consisting of $15,160$ utterances collected from the real log data of a commercial intelligent assistant (and will release the dataset to facilitate future research activity). In addition, we investigate using tweets and Web search queries for handling open-domain user utterances, which characterize the task of chat detection. Experiments demonstrated that, while simple supervised methods are effective, the use of the tweets and search queries further improves the F_1-score from 86.21 to 87.53.

1 Introduction

1.1 Chat detection

Conventional studies on spoken dialogue systems (SDS) have investigated either domain-specific task-oriented SDS[1] (Williams and Young, 2007) or open-domain non-task-oriented SDS (*a.k.a.*, chatbots or chat-oriented SDS) (Wallace, 2009). The former offers convenience by helping users complete tasks in specific domains, while the latter offers entertainment through open-ended chatting (or smalltalk) with users. Although the functionalities offered by the two types of SDS are complementary to each other, little practical effort has been made to combine them. This unfortunately has limited the potential of SDS.

This situation is now being changed by the emergence of voice-activated intelligent assistants on smartphones and home electronics (*e.g.*, Siri[2] and Alexa[3]). These intelligent assistants typically perform various tasks (*e.g.*, Web search, weather checking, and alarm setting) while being able to have chats with users. They can be seen as a novel hybrid of multi-domain task-oriented SDS and open-domain non-task-oriented SDS.

To realize such hybrid SDS, we have to determine whether or not a user is going to have a chat with the system. For example, if a user says "*What is your hobby?*" it is considered that she is going to have a chat with the system. On the other hand, if she says "*Set an alarm at 8 o'clock*," she is probably trying to operate her smartphone. We refer to this task as *chat detection* and treat it as a binary classification problem.

Chat detection has not been explored enough in past studies. This is primarily because little attempts have been made to develop hybrids of task-oriented and non-task-oriented SDS (see Section 2 for related work). Although task-oriented and non-task-oriented SDS have long research histories, both of them do not require chat detection. Typically, users of task-oriented SDS do not have chats with the systems and users of non-task-oriented SDS always have chats with the systems.

1.2 Summary of this paper

In this work, we construct a new dataset for chat detection. As we already discussed, chat detection

[*] Work done during internship at Yahoo Japan Corporation.

[1] They can be classified as single-domain or multi-domain task-oriented SDS.

[2] http://www.apple.com/ios/siri

[3] https://developer.amazon.com/alexa

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 1308–1319
Vancouver, Canada, July 30 - August 4, 2017. ©2017 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17-1120

has not been explored enough, and thus there exist no benchmark datasets available. To address this situation, we collected $15,160$ user utterances from real log data of a commercial intelligent assistant, and recruited crowd workers to annotate those utterances with whether or not the users are going to have chats with the intelligent assistant. The resulting dataset will be released to facilitate future studies.

The technical challenge in chat detection is that we have to handle open-ended utterances of intelligent assistant users. Commercial intelligent assistants have a vast amount of users and they talk about a wide variety of topics especially when chatting with the assistants. It consequently becomes labor-intensive to collect a sufficiently large amount of annotated data for training accurate chat detectors.

We develop supervised binary classifiers to perform chat detection. We address the open-ended user utterances, which characterize chat detection, by using unlabeled external resources. We specifically utilize tweets (*i.e.*, Twitter posts) and Web search queries to enhance the supervised classifiers.

Experimental results demonstrated that, while simple supervised methods are effective, the external resources are able to further improve them. The results demonstrated that the use of the external resources increases over 1 point of F_1-score (from 86.21 to 87.53).

2 Related Work

2.1 Previous studies on combining task-oriented and non-task-oriented SDS

Task-oriented and non-task-oriented SDS have long been investigated independently, and little attempts have been made to develop hybrids of the two types of SDS. As a consequence, previous studies have not investigated chat detection without only a few exceptions.[4]

Niculescu and Banchs (2015) explored using non-task-oriented SDS as a back-off mechanism for task-oriented SDS. They, however, did not propose any concrete methods of automatically determining when to switch to non-task-oriented SDS.

Lee et al. (2007) proposed an example-based dialogue manager to combine task-oriented and non-task-oriented SDS. In such a framework, however, it is difficult to flexibly utilize state-of-the-art supervised classifiers as a component.

Other studies proposed machine-learning-based frameworks for combining multi-domain task-oriented SDS and non-task-oriented SDS (Wang et al., 2014; Sarikaya, 2017). These assume that several components including a chat detector are already available, and explore integrating those components. They discuss little on how to develop each of the components. On the other hand, the focus of this work is to develop one of those components, a chat detector. Although it lies outside the scope of this paper to explore how to exploit chat detection method in a full dialogue system, the chat detection method is considered to serve, for example, as one component within those frameworks.

2.2 Intent and domain determination

Chat detection is related to, but different from, intent and domain determination that have been studied in the field of SDS (Guo et al., 2014; Xu and Sarikaya, 2014; Ravuri and Stolcke, 2015; Kim et al., 2016; Zhang and Wang, 2016).

Both intent and domain determination have been investigated in domain-specific task-oriented SDS. Intent determination aims to determine the type of information a user is seeking in single-domain task-oriented SDS. For example, in the ATIS dataset, which is collected from an airline travel information service system, the information type includes flight, city, and so on (Tur et al., 2010). On the other hand, domain determination aims to determine which domain is relevant to a given user utterance in multi-domain task-oriented SDS (Xu and Sarikaya, 2014). Note that it is possible that domain determination is followed by intent determination.

Unlike intent and domain determination, chat detection targets hybrid systems of multi-domain task-oriented SDS and open-domain non-task-oriented SDS, and aims to determine whether the non-task-oriented component is responsible to a given user utterance or not (*i.e.*, the user is going to have a chat or not). Therefore, the objective of chat detection is different from intent and domain determination.

It may be possible to see chat detection as a spe-

[4] Unfortunately, we cannot discuss little about chat detection in existing commercial intelligent assistants since most of their technical details have not been disclosed. We make the best effort to compensate for it by comparing the proposed methods with our in-house intelligent assistant in the experiment.

cific problem of domain determination (Sarikaya, 2017). We, nevertheless, discuss it as a different problem because of the uniqueness of the "chat domain." It greatly differs from ordinary domains in that it plays a role of combining the two different types of SDS that have long been studied independently, rather than combining multiple SDS of the same types. In addition, we discuss the use of external resources, especially tweets, for chat detection. This approach is unique to chat detection and is not considered effective for ordinary domain determination.

It is interesting to note that chat detection is not followed by slot-filling unlike intent and domain determination, as far as we use a popular response generator such as seq2seq model (Sutskever et al., 2014) or an information retrieval based approach (Yan et al., 2016). Although joint intent (or domain) determination and slot-filling has been widely studied to improve accuracy (Guo et al., 2014; Zhang and Wang, 2016), the same approach is not feasible in chat detection.

2.3 Intelligent assistant

Previous studies on intelligent assistants have not investigated chat detection. Their research topics are centered around those on user behaviors including the prediction of user satisfaction and engagement (Jiang et al., 2015; Kobayashi et al., 2015; Sano et al., 2016; Kiseleva et al., 2016a,b) and gamification (Otani et al., 2016). For example, Jiang et al. (2015) investigated predicting whether users are satisfied with the responses of intelligent assistants by combining diverse features including clicks and utterances. Sano et al. (2016) explored predicting whether users will keep using the intelligent assistants in the future by using long-term usage histories.

Some earlier works used the Cortana dataset as a benchmark of domain determination (Guo et al., 2014; Xu and Sarikaya, 2014; Kim et al., 2016) or proposed a development framework for Cortana (Crook et al., 2016). Those studies, however, regarded the intelligent assistant as merely one example of multi-domain task-oriented SDS and did not explore chat detection.

2.4 Non-task-oriented SDS

Non-task-oriented SDS have long been studied in the research community. While early studies adopted rule-based methods (Weizenbaum, 1966; Wallace, 2009), statistical approaches have recently gained much popularity (Ritter et al., 2011; Vinyals and Le, 2015). This research direction was pioneered by Ritter et al. (2011), who applied a phrase-based SMT model to the response generation. Later, Vinyals and Le (2015) used the seq2seq model (Sutskever et al., 2014). To date, a number follow-up studies have been made to improve on the response quality (Hasegawa et al., 2013; Shang et al., 2015; Sordoni et al., 2015; Li et al., 2016a,b; Gu et al., 2016; Yan et al., 2016). Those studies assume that users always want to have chats with systems and investigate only methods of generating appropriate responses to given utterances. Chat detection is required for integrating those response generators into intelligent assistants.

2.5 Use of conversational data

The recent explosion of conversational data on the Web, especially tweets, have triggered a variety of dialogue studies. Those typically used tweets either for training response generators (*c.f.*, Section 2.4) or for discovering dialogue acts in an unsupervised fashion (Ritter et al., 2010; Higashinaka et al., 2011). This treatment of tweets differs from that in our work.

3 Chat Detection Dataset

In this section we explain how we constructed the new benchmark dataset for chat detection. We then analyze the data to provide insights into the actual user behavior.

3.1 Construction procedure

We sampled $15,160$ unique utterances[5] (*i.e.*, automatic speech recognition results) from the real log data of a commercial intelligent assistant, Yahoo! Voice Assist.[6] The log data were collected between Jan. and Aug. 2016. In the log data, some utterances such as "Hello" appear frequently. To construct a dataset containing both high and low frequency utterances, we set frequency thresholds[7] to divide the utterances into three groups (high, middle, and low frequency) and then randomly sampled the same number of utterances

[5] The utterances are all in Japanese. Example utterances given in this paper are English translations.

[6] `https://v-assist.yahoo.co.jp`

[7] We cannot disclose the exact threshold values so as to keep the detailed statistics of the original log data confidential.

Label	Example	No. of votes
CHAT	Let's talk about something.	5
	What is your hobby?	7
	I don't have any holidays this month.	5
	I'm walking around now.	6
	Do you like cats?	5
	You are a serious geek.	7
NONCHAT	Show me a picture of Mt. Fuji.	6
	What's the highest building in the world?	5
	A nice restaurant near here.	7
	Wake me up at 9:10.	7
	Brighten the screen.	6
	Turn off the alarm.	7

Table 1: Example utterances and the numbers of votes. NONCHAT utterances are further divided into information seeking (top) and device control (bottom) to facilitate readers' understanding.

#Votes	No. of utterances
4	1701
5	2670
6	4978
7	5811

Table 2: Distribution of the numbers of votes.

from each of the three groups. During the data collection, we ensured privacy by manually removing utterances that included the full name of a person or detailed address information.

Next, we recruited crowd workers to annotate the $15,160$ utterances with two labels, CHAT and NONCHAT. The workers annotated the CHAT label when users were going to have chats with the intelligent assistant and annotated the NONCHAT label when users were seeking some information (*e.g.*, searching the Web or checking the weather) or were trying to operate the smartphones (*e.g.*, setting alarms or controlling volume). Note that our intelligent assistant works primarily on smartphones and thus the NONCHAT utterances include many operational instructions such as alarm setting. Example utterances are given in Table 1.

Seven workers were assigned to each utterance, and the final labels were obtained by majority vote to address the quality issue inherent in crowdsourcing. The last column in Table 1 shows the number of votes that the majority label obtained. For example, five workers provided the CHAT label (and the other two provided the NONCHAT label) to the first utterance "Let's talk about something."

3.2 Data analysis

The construction process described above yielded a dataset made up of $4,833$ CHAT and $10,327$ NONCHAT utterances.

We investigated the annotation agreement among the crowd workers. Table 2 shows the distribution of the numbers of votes that the majority labels obtained. The annotation given by the seven workers agreed perfectly in $5,811$ of the $15,160$ utterances (38%). Also, at least six workers agreed in the majority of cases, $10,789$ ($= 4,978 + 5,811$) utterances (71%). This indicates high agreement among the workers and the reliability of the annotation results.

During the data construction, we found that a typical confusing case arises when the utterance can be interpreted as an implicit information request. For example, the utterance "*I am hungry*" can be seen as the user trying to have a chat with the assistant, but it might be the case that she is looking for a local restaurant. Similar examples include "*I have a backache*" and so on. One solution in this case might be to ask the user a clarification question (Schlöder and Fernandez, 2015). Such an exploration is left for our future research.

Additionally, we manually classified the CHAT utterances according to their dialogue acts to figure out how real users have chats with the intelligent assistant (Table 3). The set of dialogue acts was designed by referring to (Meguro et al., 2010). As shown in Table 3, while some of the utterances are boilerplates (*e.g.*, those in the GREETING act) and thus have limited variety, the majority of the utterances exhibit tremendous diversity. We see

Dialogue act (No. of Utter.)	Example
GREETING (206)	Hello. Merry Christmas.
SELFDISCLOSURE (1164)	I am free today. I have a sore throat.
ORDER (716)	Please cheer me up. Give me a song!
QUESTION (1551)	Do you have emotions? Are you angry?
INVITATION (130)	Let's play with me! Let's go to karaoke next time.
INFORMATION (214)	My cat is acting strange. It snowed a lot.
THANKS (126)	Thank you. You are cool!
CURSE (172)	You're an idiot. You are useless.
APOLOGY (9)	I'm sorry. I mistook, sorry.
INTERJECTION (151)	Whoof. Yeah, yeah.
MISC (394)	May the force be with you. Cock-a-doodle-doo.

Table 3: Distribution over dialogue acts and example utterances.

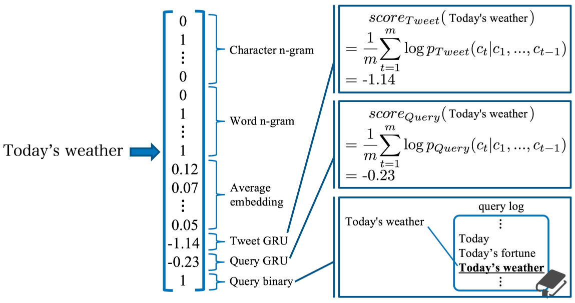
Today's weather
0
1
0
0
1
1
0.12
0.07
0.05
-1.14
-0.23
1
Character n-gram
Word n-gram
Average embedding
Tweet GRU
Query GRU
Query binary
$score_{Tweet}$(Today's weather)
$= \frac{1}{m}\sum_{t=1}^{m} \log p_{Tweet}(c_t|c_1, ..., c_{t-1})$
= -1.14
$score_{Query}$(Today's weather)
$= \frac{1}{m}\sum_{t=1}^{m} \log p_{Query}(c_t|c_1, ..., c_{t-1})$
= -0.23
Today's weather
query log
Today
Today's fortune
Today's weather

Figure 1: Feature vector representation of the example utterance "Today's weather." The upper three parts of the vector represent the features described in Section 4.1 (character n-gram, word n-gram, and average of the word embeddings). The three additional features explained in Section 4.2 are added as two real-valued features (Tweet GRU and Query GRU) and one binary feature (Query binary).

a wide variety of topics including private issues (*e.g.*, "*I am free today*") and questions to the assistant (*e.g.*, "*Are you angry?*"). Also, we even see a movie quote ("*May the force be with you*") and a rooster crow ("*Cock-a-doodle-doo*") in the MISC act. These clearly represent the open-domain nature of the user utterances in intelligent assistants.

Interestingly, some users curse at the intelligent assistant probably because it failed to make appropriate responses (see the CURSE act). Although such user behavior would not be observed from paid research participants, we observe a certain amount of curse utterances in the real data.

4 Detection Method

We formulate chat detection as a binary classification problem to train supervised classifiers. In this section, we first explain the two types of classifiers explored in this paper, and then investigate the use of external resources for enhancing those classifiers.

4.1 Base classifiers

The first classifier utilizes SVM for its popularity and efficiency. It uses character and word n-gram ($n = 1$ and 2) features. It also uses word embedding features (Turian et al., 2010). A skip-gram model (Mikolov et al., 2013) is trained on the entire intelligent assistant log[8] to learn word embeddings. The embeddings of the words in the utterance are then averaged to produce additional features.

The second classifier uses a convolutional neural network (CNN) because it has recently proven to perform well on text classification problems (Kim, 2014; Johnson and Zhang, 2015a,b). We follow (Kim, 2014) to develop a simple CNN that has a single convolution and max-pooling layer followed by the soft-max layer. We use a rectified linear unit (ReLU) as the non-linear activation function. The same word embeddings as SVM are used for the pre-training.

4.2 Using external resources

We next investigate using external resources for enhancing the base classifiers. Thanks to the rapid evolution of the Web in the past decade, a variety of textual data including not only conversational (*i.e.*, chat-like) but also non-conversational ones are abundantly available nowadays. These data offer an effective way of enhancing the base classifiers. We specifically use tweets and Web search queries as conversational and non-conversational text data, respectively.

We train character-based[9] language models on

[8] We used the same log data used in Section 3. The detailed statistics is confidential.

[9] We also trained word-based language models in prelim-

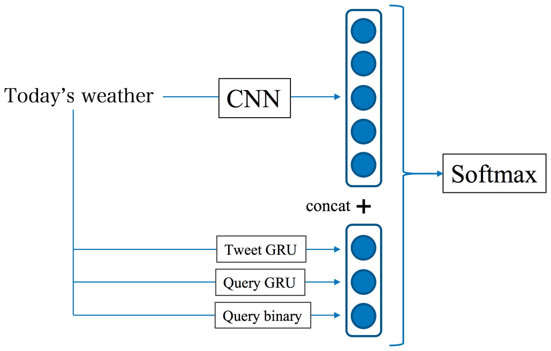

Figure 2: Architecture of our CNN-based classifier when the input utterance is "Today's weather." The output layer of CNN and the three additional features explained in Section 4.2 are concatenated. The resulting vector is fed to the soft-max function.

tweets and Web search queries, and use their scores (*i.e.*, the normalized log probabilities of the utterance) as two additional features. Let $u = c_1, c_2, \dots, c_m$ be an utterance made up of m characters. Then, the score $score_r(u)$ of the language model trained on the external resource $r \in \{\text{tweet}, \text{query}\}$ is defined as

$$score_r(u) = \frac{1}{m} \sum_{t=1}^{m} \log p_r(c_t \mid c_1, \dots, c_{t-1}).$$

The GRU language model is adopted for its superior performance (Cho et al., 2014; Chung et al., 2014). Let $\mathbf{x}_t$ be the embedding of t-th character and $\mathbf{h}_t$ be the t-th hidden state. GRU computes the hidden state as

$$\begin{aligned}
\mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \\
\mathbf{z}_t &= \sigma(\mathbf{W}^{(z)}\mathbf{z}_t + \mathbf{U}^{(z)}\mathbf{h}_{t-1}) \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}^{(h)}\mathbf{x}_t + \mathbf{U}^{(h)}(\mathbf{r}_t \odot \mathbf{h}_{t-1})) \\
\mathbf{r}_t &= \sigma(\mathbf{W}^{(r)}\mathbf{x}_t + \mathbf{U}^{(r)}\mathbf{h}_{t-1})
\end{aligned}$$

where $\odot$ is the element-wise multiplication, σ is the sigmoid and $\tanh$ is the hyperbolic tangent. $\mathbf{W}^{(z)}$, $\mathbf{U}^{(z)}$, $\mathbf{W}^{(h)}$, $\mathbf{U}^{(h)}$, $\mathbf{W}^{(r)}$, and $\mathbf{U}^{(r)}$ are weight matrices. The hidden states are fed to the soft-max to predict the next word.

We also use a binary feature indicating whether the utterance appears in the Web search query log or not. We observe that some NONCHAT utterances are made up of single entities such as location and product names. Such utterances are considered to be seeking information on those entities. We therefore use the query log as an entity dictionary to derive a feature indicating whether the utterance is likely to be a single entity.

The resulting three features are incorporated into the SVM-based classifier straightforwardly (Figure 1). For the CNN-based classifier, they are provided as additional inputs to the soft-max layer (Figure 2).

5 Experimental Results

We empirically evaluate the proposed methods on the chat detection dataset.

5.1 Experimental settings

We performed 10-fold cross validation on the chat detection dataset to train and evaluate the proposed classifiers. In each fold, we used 80%, 10%, and 10% of the data for the training, development, and evaluation, respectively.

We used `word2vec`[10] to learn 300 dimensional word embeddings. They were used to induce the additional 300 features for SVM. They were also used as the pre-trained word embeddings for CNN.

We used the `faster-rnn` toolkit[11] to train the GRU language models. The size of the embedding and hidden layer was set to 256. Noise contrastive estimation (Gutmann and Hyvärinen, 2010) was used to train the soft-max function and the number of noise samples was set to 50. Maximum entropy 4-gram models were also trained to yield a combined model (Mikolov et al., 2011).

The language models were trained on 100 millions tweets collected between Apr. and July 2016 and 100 million Web search queries issued between Mar. and Jun. 2016. The tweets were sampled from those received replies to collect only conversational tweets (Ritter et al., 2011). The same Web search queries were used to derive the binary feature. Although it is difficult to release those data, we plan to make the feature values available together with the benchmark dataset.

We used `liblinear`[12] to train L_2-regularized L_2-loss SVM. The hyperparameter c was tuned

inary experiments and found that character-based ones perform consistently better.

[10] https://code.google.com/archive/p/word2vec

[11] https://github.com/yandex/faster-rnnlm

[12] https://www.csie.ntu.edu.tw/~cjlin/liblinear

Model	Acc.	P	R	F_1
Majority	68.12	N/A	N/A	N/A
Tweet GRU	72.07	54.54	74.40	62.94
In-house IA	78.31	62.57	79.51	70.03
SVM	90.51	86.42	83.45	84.91
SVM+embed.	91.35	87.62	84.88	86.21
SVM+embed.+tweet-query	**92.15**	**88.61**	**86.50**	**87.53**
CNN	85.16	83.40	68.12	74.41
CNN+pre-train.	90.84	87.03	83.80	85.36
CNN+pre-train.+tweet-query	91.48	87.78	85.18	86.56

Table 4: Chat detection results.

over $\{2^{-10}, 2^{-9}, \ldots, 2^{10}\}$.

The CNN was implemented with `chainer`.[13] We tuned the number of feature maps over $\{100, 150\}$, and filter region sizes over $\{\{2\}, \{3\}, \{1, 2\}, \{2, 3\}, \{3, 4\}, \{1, 2, 3\}, \{2, 3, 4\}\}$. The mini-batch size was set to 32. The dropout rate was set to 0.5. We used Adam ($\alpha = 0.001$, $\beta1 = 0.9$, $\beta2 = 0.999$, and $\epsilon = 10^{-8}$) to perform stochastic gradient descent (Kingma and Ba, 2015).

5.2 Baselines

The following baseline methods were implemented for comparison:

Majority Utterances are always classified as the majority class, NONCHAT.

Tweet GRU Utterances are classified as CHAT if the score of the GRU language model trained on the tweets exceeds a threshold. We used exactly the same GRU language model as the one that was used for deriving the feature. The threshold was calibrated on the development data by maximizing the F_1-score of the CHAT class.

In-house IA Our in-house intelligent assistant system, which adopts a hybrid of rule-based and example-based approaches. Since we cannot disclose its technical details, the result is presented just for reference.

5.3 Result

Table 4 gives the precision, recall, F_1-score (for the CHAT class), and overall classification accuracy results. We report only accuracy for **Majority** baseline. **+embed.** and **+pre-train.** represent using the word embedding features for SVM and the pre-trained word embeddings for CNN, respectively. **+tweet-query** represents using the three features derived from the tweets and Web search query.

Table 4 represents that both of the classifiers, SVM and CNN, perform accurately. We see that both **+embed.** and **+pre-train.** improve the results. The best performing method, **SVM+embed.+tweet-query**, achieves 92% accuracy and 87% F_1-score, outperforming all of the baselines. CNN performed worse than SVM contrary to results reported by recent studies (Kim, 2014). We think this is because the architecture of our CNN is rather simplistic. It might be possible to improve the CNN-based classifier by adopting more complex network, although it is likely to come at the cost of extra training time. Another reason would be that our SVM classifier uses carefully designed features beyond word 1-grams.

Table 4 also represents that the external resources are effective, improving F_1-scores almost 1 points in both SVM and CNN. Table 5 illustrates example utterances and their language model scores. We see that the language models trained on the tweets and queries successfully provide the CHAT utterances with high and low scores, respectively. Table 6 shows chat detection results when each of the three features derived from the external resources is added to **SVM+embed.** The results represent that they are all worse than **SVM+embed.+tweet-query** and thus it is crucial to combine all of them for achieving the best performance.

Table 7 shows examples of feature weights of **SVM+embed.+tweet-query**. Tweet GRU and query GRU denote the language model score features. The others are word n-gram features. We see that the language model scores have the large

[13] http://chainer.org

Score (tweet/query)		Label	Utterance
−0.964	−1.427	CHAT	Halloween has already finished.
−0.957	−1.610	CHAT	Let's sleep.
−1.233	−0.562	NONCHAT	Pokemon Go install.
−1.837	−0.682	NONCHAT	Weekly weather forecast.

Table 5: Examples of the language model scores. The first two columns represent the scores provided by the GRU language models trained on the tweets and Web search queries, respectively. The third and fourth columns represent the label and utterance.

Feature	Acc.	P	R	F_1
tweet GRU	91.53	87.62	85.49	86.53
query GRU	91.38	87.55	85.06	86.28
query binary	91.42	87.56	85.21	86.36

Table 6: Effect of the three features derived from the tweets and Web search queries.

Feature	Weight	Feature	Weight
tweet GRU	1.128	query GRU	−0.771
I	0.215	call to	−0.217
Sing	0.195	volume	−0.196

Table 7: Examples feature weights of **SVM+embed+tweet-query**.

#Votes	#Utter.	Acc.	P	R	F_1
4	1701	66.67	55.41	59.81	57.53
5	2670	87.72	80.46	83.01	81.72
6	4978	96.02	92.73	93.87	93.30
7	5811	98.33	96.73	97.68	97.20

Table 8: Chat detection results across the numbers of votes that the majority label obtained.

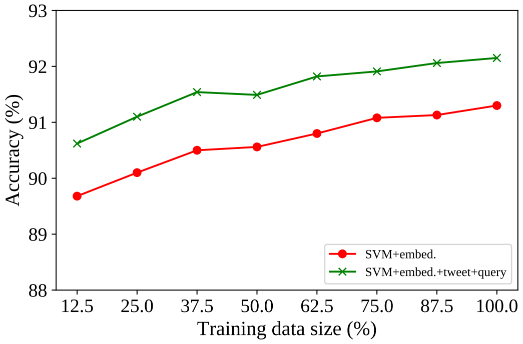

Figure 3: Learning curve of the proposed methods. The horizontal axis represents what percentage of the training portion is used in each fold of the cross validation. The vertical axis represents the classification accuracy.

positive and negative weights, respectively. This indicates that effectiveness of the language models. We also see that the first person has a large positive weight, while terms related to device controlling ("call to" and "volume") have large negative weights.

Table 8 represents chat detection results of **SVM+embd.+tweet-query** across the numbers of votes that the majority label obtained. As expected, we see that all metrics get higher as the number of agreement among the crowd workers becomes larger. In fact, we see as much as 98% accuracy when all seven workers agree. This implies that utterances easy for humans to classify are also easy for the classifiers.

5.4 Training data size

We next investigate the effect of the training data size on the classification accuracy.

Figure 3 illustrates the learning curve. It represents that the classification accuracy improves almost monotonically as the training data size increases. Although our training data is by no means small, the shape of the learning curve nevertheless suggests that further improvement would be achieved by adding more training data. This implies that a very large amount of training data are required for covering open-domain utterances in intelligent assistants.

The figure at the same time represents the usefulness of the external resources. We see that **SVM+embed.+tweet-query** trained on about 25% of the training data is able to achieve comparable accuracy with **SVM+embed.** trained on the entire training data. This result suggests that the external resources are able to compensate for the scarcity of annotated data.

5.5 Utterance length

We finally investigate how the utterance length correlates with the classification accuracy. Fig-

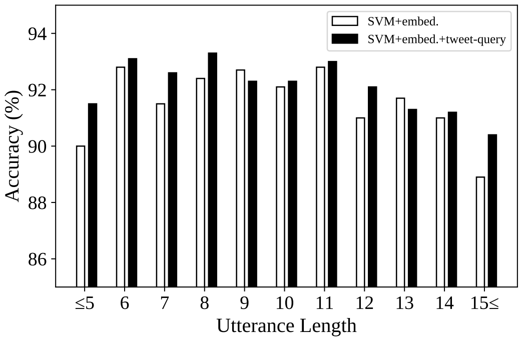

Figure 4: Classification accuracy across utterance lengths in the number of characters.

ure 4 illustrates the classification accuracies of **SVM+embed.** and **SVM+embed.+tweet-query** for each utterance length in the number of characters.

Figure 4 reveals that the difference between the two proposed methods is evident in short utterances (*i.e.*, ≤ 5). This is because those utterances are too short to contain sufficient information required for classification, and the additional features are helpful. We note that Japanese writing system uses ideograms and thus even five characters is enough to represent a simple sentence.

We also see a clear difference in longer utterances (*i.e.*, $15 \leq$) as well. We consider those long utterances are difficult to classify because some words in the utterances are irrelevant for the classification and the n-gram and embedding features include those irrelevant ones. On the other hand, we consider that the language model scores are good at capturing stylistic information irrespective of the utterance length.

6 Future Work

As discussed in Section 3.2, some user utterances such as "*I am hungry*" are ambiguous in nature and thus are difficult to handle in the current framework. An important future work is to develop a sophisticated dialogue manager to handle such utterances, for example, by making clarification questions (Schlöder and Fernandez, 2015).

We manually investigated the dialogue acts in the chat detection dataset (*c.f.*, Section 3.2). It is interesting to automatically determine the dialogue acts to help producing appropriate system responses. Some related studies exist in such a research direction (Meguro et al., 2010).

Although we used only text data to perform chat detection, we can also utilize contextual information such as the previous utterances (Xu and Sarikaya, 2014), the acoustic information (Jiang et al., 2015), and the user profile (Sano et al., 2016). It is an interesting research topic to use such contextual information beyond text. It is considered promising to make use of a neural network for integrating such heterogeneous information.

An automatic speech recognition (ASR) error is a popular problem in SDS, and previous studies have proposed sophisticated techniques, including re-ranking (Morbini et al., 2012) and POMDP (Williams and Young, 2007), for addressing the ASR errors. Incorporating these techniques into our methods is also an important future work.

Although the studies on non-task-oriented SDS have made substantial progress in the past few years, it unfortunately remains difficult for the systems to fluently chat with users (Higashinaka et al., 2015). Further efforts on improving non-task-oriented dialogue systems is an important future work.

7 Conclusion

This paper investigated chat detection for combining domain-specific task-oriented SDS and open-domain non-task-oriented SDS. To address the scarcity of benchmark datasets for this task, we constructed a new benchmark dataset from the real log data of a commercial intelligent assistant. In addition, we investigated using the external resources, tweets and Web search queries, to handle open-domain user utterances, which characterize the task of chat detection. The empirical experiment demonstrated that the off-the-shelf supervised methods augmented with the external resources perform accurately, outperforming the baseline approaches. We hope that this study contributes to remove the long-standing boundary between task-oriented and non-task-oriented SDS.

To facilitate future research, we are going to release the dataset together with the feature values derived from the tweets and Web search queries.[14]

Acknowledgments

We thank Manabu Sassano, Chikara Hashimoto, Naoki Yoshinaga, and Masashi Toyoda for fruitful discussions and comments. We also thank the anonymous reviewers.

[14] `https://research-lab.yahoo.co.jp/en/software`

References

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*. pages 1724–1734. http://www.aclweb.org/anthology/D14-1179.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555.

Paul Crook, Alex Marin, Vipul Agarwal, Khushboo Aggarwal, Tasos Anastasakos, Ravi Bikkula, Daniel Boies, Asli Celikyilmaz, Senthilkumar Chandramohan, Zhaleh Feizollahi, Roman Holenstein, Minwoo Jeong, Omar Khan, Young-Bum Kim, Elizabeth Krawczyk, Xiaohu Liu, Danko Panic, Vasiliy Radostev, Nikhil Ramesh, Jean-Phillipe Robichaud, Alexandre Rochette, Logan Stromberg, and Ruhi Sarikaya. 2016. Task completion platform: A self-serve multi-domain goal oriented dialogue platform. In *Proceedings of NAACL (Demonstrations)*. pages 47–51. http://www.aclweb.org/anthology/N16-3010.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of ACL*. pages 1631–1640. http://www.aclweb.org/anthology/P16-1154.

Daniel (Zhaohan) Guo, Gokhan Tur, Scott Wen tau Yih, and Geoffrey Zweig. 2014. Joint semantic utterance classification and slot filling with recursive neural networks. In *Proceedings of IEEE SLT Workshop*.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of AISTATS*. pages 297–304.

Takayuki Hasegawa, Nobuhiro Kaji, Naoki Yoshinaga, and Masashi Toyoda. 2013. Predicting and eliciting addressee's emotion in online dialogue. In *Proceedings of ACL*. pages 964–972. http://www.aclweb.org/anthology/P13-1095.

Ryuichiro Higashinaka, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, Yuka Kobayashi, and Masahiro Mizukami. 2015. Towards taxonomy of errors in chat-oriented dialogue systems. In *Proceedings of SIGDIAL*. pages 87–95. http://aclweb.org/anthology/W15-4611.

Ryuichiro Higashinaka, Noriaki Kawamae, Kugatsu Sadamitsu, Yasuhiro Minami, Toyomi Meguro, Kohji Dohsaka, and Hirohito Inagaki. 2011. Building a conversational model from two-tweets. In *Proceedings of ASRU*. pages 330–335.

Jiepu Jiang, Ahmed Hassan Awadallah, Rosie Jones, Umut Ozertem, Imed Zitouni, Ranjitha Gurunath Kulkarni, and Omar Zia Khan. 2015. Automatic online evaluation of intelligent assistants. In *Proceedings of WWW*. pages 506–516.

Rie Johnson and Tong Zhang. 2015a. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of NAACL*. pages 103–112. http://www.aclweb.org/anthology/N15-1011.

Rie Johnson and Tong Zhang. 2015b. Semi-supervised convolutional neural networks for text categorization via region embedding. In *Advances in NIPS*, pages 919–927.

Joo-Kyung Kim, Gokhan Tur, Asli Celikyilmaz, Bin Cao, and Ye-Yi Wang. 2016. Intent detection using semantically enriched word embeddings. In *Proceedings of IEEE SLT Workshop*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*. pages 1746–1751. http://www.aclweb.org/anthology/D14-1181.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Julia Kiseleva, Kyle Williams, Ahmed Hassan Awadallah, Aidan Crook, Imed Zitouni, and Tasos Anastasakos. 2016a. Predicting user satisfaction with intelligent assistants. In *Proceedings of SIGIR*. pages 45–54.

Julia Kiseleva, Kyle Williams, Ahmed Hassan Awadallah, Aidan C. Crook, Imed Zitouni, and Tasos Anastasakos. 2016b. Understanding user satisfaction with intelligent assistants. In *Proceedings of SIGCHIIR*. pages 121–130.

Hayato Kobayashi, Kaori Tanio, and Manabu Sassano. 2015. Effects of game on user engagement with spoken dialogue system. In *Proceedings of SIGDIAL*. pages 422–426. http://aclweb.org/anthology/W15-4656.

Cheongjae Lee, Sangkeun Jung, Seokhwan Kim, and Gary Geunbae Lee. 2007. Example-based dialog modeling for practical multi-domain dialog system. *Speech Communication* 51(5):466–484.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of NAACL*. pages 110–119. http://www.aclweb.org/anthology/N16-1014.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016b. A persona-based neural conversation model. In *Proceedings of ACL*. pages 994–1003. http://www.aclweb.org/anthology/P16-1094.

Toyomi Meguro, Ryuichiro Higashinaka, Yasuhiro Minami, and Kohji Dohsaka. 2010. Controlling listening-oriented dialogue using partially observable markov decision processes. In *Proceedings of Coling*. pages 761–769. http://www.aclweb.org/anthology/C10-1086.

Tomas Mikolov, Anoop Deoras, Daniel Povey, Lukas Burget, and Jan Cernocky. 2011. Strategies for training large scale neural network language models. In *Proceedings of ASRU*. pages 196–201.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in NIPS*. pages 3111–3119.

Fabrizio Morbini, Kartik Audhkhasi, Ron Artstein, Maarten Van Segbroeck, Kenji Sagae, Panayiotis Georgiou, David R. Traum, and Shri Narayanan. 2012. A reranking approach for recognition and classification of speech input in conversational dialogue systems. In *Proceedings of SLT*. pages 49–54.

Andreea I. Niculescu and Rafael E. Banchs. 2015. Strategies to cope with errors in human-machine speech interactions: using chatbots as back-off mechanism for task-oriented dialogues. In *Proceedings of ERRARE*.

Naoki Otani, Daisuke Kawahara, Sadao Kurohashi, Nobuhiro Kaji, and Manabu Sassano. 2016. Large-scale acquisition of commonsense knowledge via a quiz game on a dialogue system. In *Proceedings of OKBQA*. pages 11–20. http://aclweb.org/anthology/W16-4402.

Suman Ravuri and Andreas Stolcke. 2015. A comparative study of neural network models for lexical intent classification. In *In Proceedings of ASRU*. pages 368–374.

Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of twitter conversations. In *In Proceedings of NAACL*. pages 172–180. http://www.aclweb.org/anthology/N10-1020.

Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In *Proceedings of EMNLP*. pages 583–593. http://www.aclweb.org/anthology/D11-1054.

Shumpei Sano, Nobuhiro Kaji, and Manabu Sassano. 2016. Prediction of prospective user engagement with intelligent assistants. In *Proceedings of ACL*. pages 1203–1212. http://www.aclweb.org/anthology/P16-1114.

Ruhi Sarikaya. 2017. The technology behind personal digital assistants: An overview of the system architecture and key components. *IEEE Signal Processing Magazine* 34(1):67–81.

Julian J. Schlöder and Raquel Fernandez. 2015. Clarifying intentions in dialogue: A corpus study. In *Proceedings of the 11th International Conference on Computational Semantics*. pages 46–51. http://www.aclweb.org/anthology/W15-0106.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of ACL*. pages 1577–1586. http://www.aclweb.org/anthology/P15-1152.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of NAACL*. pages 196–205. http://www.aclweb.org/anthology/N15-1020.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in NIPS*, pages 3104–3112.

Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. 2010. What is left to be understood in atis? In *Proceedings of IEEE SLT Workshop*. pages 19–24.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of ACL*. pages 384–394. http://www.aclweb.org/anthology/P10-1040.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In *Proceedings of Deep Learning Workshop*.

Richard S. Wallace. 2009. *The Anatomy of A.L.I.C.E.*, Springer, pages 181–210.

Zhuoran Wang, Hongliang Chen, Guanchun Wang, Hao Tian, Hua Wu, and Haifeng Wang. 2014. Policy learning for domain selection in an extensible multi-domain spoken dialogue system. In *Proceedings of EMNLP*. pages 57–67. http://www.aclweb.org/anthology/D14-1007.

Joseph Weizenbaum. 1966. Eliza–a computer program for the study of natural language communication between man and machine. *Communications of the ACM* 9(1):36–45.

Jason D. Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language* 21(2):393–422.

Puyang Xu and Ruhi Sarikaya. 2014. Contextual domain classification in spoken language understanding systems using recurrent neural network. In *Proceedings of ICASSP*. pages 136–140.

Zhao Yan, Nan Duan, Junwei Bao, Peng Chen, Ming Zhou, Zhoujun Li, and Jianshe Zhou. 2016. Docchat: An information retrieval approach for chatbot engines using unstructured documents. In *Proceedings of ACL*. pages 516–525. http://www.aclweb.org/anthology/P16-1049.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *Proceedings of IJCAI*. pages 2993–2999.