Tell Me When Users Leave: Predicting Users' Abandonment of A Task-Oriented Chatbot Service using Explainable Deep Learning

Yu-Wei Yang
ray17606@gmail.com
National Yang Ming Chiao Tung University
Taiwan R.O.C

Chieh Hsu
hsu0120.cs09g@nctu.edu.tw
National Yang Ming Chiao Tung University
Taiwan R.O.C

Hsin-Chien Tung
achai0729.cs09g@nctu.edu.tw
National Yang Ming Chiao Tung University
Taiwan R.O.C

Hong-Han Shuai
hhshuai@nctu.edu.tw
National Yang Ming Chiao Tung University
Taiwan R.O.C

Yung-Ju Chang
armuro@cs.nctu.edu.tw
National Yang Ming Chiao Tung University
Taiwan R.O.C

ABSTRACT

Task-oriented chatbots have been widely used by businesses to support users in accomplishing predefined tasks. Yet, conversation breakdowns could result in users abandoning the chatbot service. Detecting or early predicting signals of users' chatbot abandonment could help businesses know when to provide assistance. Based on an annotated conversation log involving 1,837 users, we built two models, one end-to-end model built on top of pre-trained BERT models, and the other being an attention-based deep learning model trained from 102 different handcrafted features derived from annotated messages. The former achieved an AUROC of 90%. The latter explainable model, despite the extra effort of adding annotations, achieved a higher AUROC of 95.7% and provided additional insights into important features indicative of service abandonment, such as input types, error types, and presence of users' information-request within recently exchanged messages.

CCS CONCEPTS

• **Human-centered computing** → **Text input**; • **Computing methodologies** → *Machine learning*.

KEYWORDS

Responsiveness; instant messaging; computer-mediated communication; mixed-effect logistic regression; qualitative analysis

ACM Reference Format:
Yu-Wei Yang, Chieh Hsu, Hsin-Chien Tung, Hong-Han Shuai, and Yung-Ju Chang. 2021. Tell Me When Users Leave: Predicting Users' Abandonment of A Task-Oriented Chatbot Service using Explainable Deep Learning. In *3rd Conference on Conversational User Interfaces (CUI '21), July 27–29, 2021, Bilbao (online), Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3469595.3469630

CUI '21, July 27–29, 2021, Bilbao (online), Spain

ACM ISBN 978-1-4503-8998-3/21/07…$15.00
https://doi.org/10.1145/3469595.3469630

1 INTRODUCTION

With the rapid development of Natural Language Processing (NLP) with deep learning over the past few years [5, 11, 16–18], task-oriented chatbots are increasingly common tools for users to inquire about information and perform information tasks. Nevertheless, conversation breakdowns or users' perception that no progress is being made during the conversation are still found to be pervasive in task-oriented chatbot conversations. This is likely to cause not only user frustration but also user abandonment of the chatbot service due to frustration [2, 6]. Early prediction of users being likely to abandon task-oriented chatbot services could potentially allow chatbot service runners to provide users with assistance before they did so. However, little research attempt has been made to detect users abandoning the chatbot service, making the timing as to when to provide assistance remained unclear. Prior research also lacks insights into the signs indicative of users abandoning task-oriented chatbots. To fill the aforementioned gaps, we built the first explainable deep learning model with an attention mechanism that could detect in-the-moment users' abandonment of a banking chatbot on a real dataset provided by a banking institution. Using the attention-based learning model, we also have successfully identified features indicative of the occurrence of users abandoning the task-oriented chatbot. In evaluating the proposed explainable model, we compared it with SVM and Random Forest, as well as a pre-trained end-to-end model built on top of BERT. We showed that the model's performance in in-the-moment abandonment detection could achieve an AUROC of 95.7% and an F1-score of 83.2%. However, in early prediction of abandonment, i.e., predicting future abandonment by not considering the last message exchanged before abandonment, the BERT end-to-end model achieved better performance.

2 METHODOLOGY

2.1 Dataset and Data Preprocessing

The dataset we used contained users' conversations with a Facebook Messenger chatbot built by a banking institution, recorded from May 1, 2017, to July 31, 2017. The service that the chatbot provided included currency-exchange converting information, credit card introduction, housing-loan evaluation, and investment information.

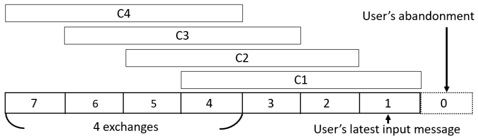


Figure 1: Explanation of C4 to C1

The data were stored in a spreadsheet, with each row contained one of 24,074 exchanges.[1] The research team verified all conversation exchanges by interacting with the chatbot designers, resulting in the final 19,451 exchanges being used for data labeling and model building. Users' conversation exchanges were assigned labels by two coders based on analyzing users' conversation intentions and systematic practices [1]. These data are examined on how these conversation exchanges were grouped, and assigned labels to each conversation exchange [12, 15]. Initially, the labels were iteratively generated and revised in the first 2% of the dataset until reaching the consensus regarding all labels' meanings, then continue this process for every 10% of the dataset. As every 2% of the dataset was labeled, the two coders check the reliability of the labels. In the end, 88 labels were assigned to every conversation exchange (with a Cohen Kappa of 0.802, indicating high inter-coder reliability [14]). Among the 88 labels, the label named "long-term service abandonment" (i.e. the user abandon chatbot for more than 10 days) was used as the prediction target for our model prediction (in total 908 instances). The other labels of all of the last four exchanges prior to chatbot abandonment were used for predicting in-the-moment chatbot abandonment (represented as the C1 condition in Figure 1). If a conversation session, a grouping of conversation messages, contained less than four exchanges, we dropped the sessions in order to unify the number of each condition. The final dataset consisted of a total of 1,686 sessions containing 17,266 sequences of four consecutive exchanges. In addition to in-the-moment abandonment prediction, we are also interested in how early a model could predict chatbot abandonment. The rationale is that users' intention of abandoning a chatbot may be traceable in earlier conversations (e.g. expressing anger in earlier messages). Therefore, we examined three additional conditions, C2, C3, C4, respectively, that consisted of earlier exchanges for predicting "future abandonment". As illustrated in Figure 1, for example, C2 used the features of the last second exchange through the last fifth; C3 used the features of the last third exchange through the last sixth, and so forth. Thus, the inclusion of the conditions C2, C3, and C4 models was to examine: how many messages earlier the model can look back to still achieve an acceptable prediction performance. As a result, all of these models represented attempts of early detection, i.e. early detecting that the user would leave the chatbot before the message that caused the abandonment. We expected that the prediction performance would drop at C2, C3, and C4, since the features of these exchanges were increasingly distant from abandonment; however, if the performance of the model for either C2, C3, C4 is acceptable, it means that early detection of chatbot abandonment may be promising.

2.2 Attention-based model

2.2.1 Feature Selection. In order to improve the interpretability of the attention-based model, it is necessary to simplify the model by reducing the number of labels. We first used feature selection to preprocess these labels. Specifically, we used the feature selection recursive feature elimination (RFE) with logistic regression (LR) to reduce the number of labels. The resulting labels are shown in Table 1.

2.2.2 Model Architecture. We introduced the attention mechanism for 1) performing the feature selection within the model and 2) providing the model explainability, i.e., visualizing the attention weight. Specifically, we built a deep learning model with an attention mechanism to observe what features were more likely to be associated with users abandoning the banking chatbot. We took the i-th sequence of four consecutive exchanges as the model input by concatenating twenty features of each timestamp, which is denoted by $\vec{X}_i = [\vec{X}_{i,1}, \vec{X}_{i,2}, \cdots, \vec{X}_{i,j}, \cdots, \vec{X}_{i,80}]$. To derive the attention weight for each input feature, we first transformed each $\vec{X}_{i,j}$ into a 32-dimensional continuous vector $\vec{H}_{i,j}$ by a dense layer followed by a sigmoid function, i.e., $\vec{H}_i = sigmoid(dense(\vec{X}_i))$, where $\vec{H}_i$ is a 2560-dim hidden vector (80×32). The attention weight vector, denoted by $\vec{W}_i = [\vec{W}_{i,1}, \vec{W}_{i,2}, \cdots, \vec{W}_{i,j}, \cdots, \vec{W}_{i,80}]$, is derived by summarizing the weight $\vec{H}_i$ for each feature, i.e.,

$$\vec{W}_{i,j} = \sum_{k=1}^{32} \vec{H}_{i,32(j-1)+k} \tag{1}$$

The enhanced feature, $\vec{X}'_i$ is obtained by the element-wise multiplication of the input $\vec{X}_i$ and attention weight vector $\vec{W}_i$, i.e.,

$$\vec{X}'_i = \vec{W}_i \odot \vec{X}_i, \tag{2}$$

where $\odot$ is the element-wise multiplication. Finally, due to the imbalanced data, we use weighted binary cross-entropy loss to optimize the model. Let Y_i and $\hat{Y}_i = f(\vec{X}_i; \theta)$ denote the ground-truth label and the prediction of the i-th sample, respectively. The standard binary cross-entropy loss function is given by

$$-\frac{1}{M}\sum_{i=1}^{M}[Y_i \log f(\vec{X}_i;\theta) + (1-Y_i)\log(1-f(\vec{X}_i;\theta))], \tag{3}$$

where M is the number of training examples. To make the attention weights sparse for a better model interpretation, we add a weight regularization to constrain the attention weight. The final loss is obtained by

$$\mathcal{J} = -\frac{1}{M}\sum_{i=1}^{M}[Y_i \log f(\vec{X}_i;\theta) + (1-Y_i)\log(1-f(\vec{X}_i;\theta)) + \lambda||\vec{W}_i||_1], \tag{4}$$

where λ is a hyperparameter for controlling the sparsity of attention weights.[2]

2.3 End-to-end model built on pretrained BERT

We directly extracted features from raw conversation exchanges by transforming them into semantic embeddings using the state-of-the-art pre-trained contextual word representation model named BERT

[1] One exchange represents one user input, the response given by chatbot, and an intent recognized by the banking chatbot

[2] In the experiment, we empirically set $\lambda = 0.9$ by cross-validation.

Topic		User			
L0	Last message in a topic	L5	Requesting information	L10	Unclear meaning
L1	Topic continue	L6	Providing information	L11	Input by typing
L2	Topic retry (continuous)	L7	Chatting	L12	Input by button
L3	Topic retry (cross session)	L8	Complaining		
L4	Topic switch	L9	Looking for assistant		

Chatbot	
L13	Chatbot giving an unrelated response.
L14	Chatbot incapable of understanding the user's intent
L15	Chatbot giving a response(Dialog State Tracking)
L16	Chatbot misrecognizing the user's intent and providing an incorrect service at the beginning (Dialog State Tracking)
L17	Task Accomplished (Dialog State Tracking)
L18	Chatbot incapable of understanding the user's intent because the user stay in the topic that the chatbot has already left (Dialog State Tracking).
L19	Chatbot misrecognizing the user's intent and not accomplishing the user's request at the end (Dialog State Tracking).

Table 1: Features of the input data

[5], which has achieved an excellent performance in many topics in the NLP domain [5]. The input for the model was composed of three embeddings: token embeddings, segment embeddings, and position embedding. We organized the texts between users and the chatbot in each exchange into the format of [CLS] + user input text + [SEP] + chatbot response text + [SEP], and fed them into BERT to obtain the sentence embeddings of the exchange.

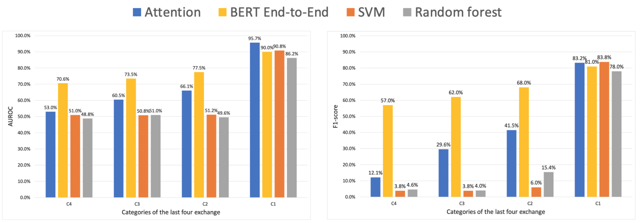
Then we used a long short-term memory (LSTM) network [8] as the downstream network for predicting user abandonment, of which the input was the sentence embeddings of the four exchanges of C1, C2, C3, and C4, respectively. Sigmoid function was used for generating the probability of user abandonment. Here, we used binary cross-entropy as loss function and Adam as the optimizer for training.

3 EXPERIMENTS

We evaluated the attention-based model by comparing it with two traditional classifiers, SVM [10] and Random Forest [4]. We compared the performance of the classifiers in the aforementioned four conditions, C1 through C4 in Fig 1, respectively, to investigate how early the classifiers can detect chatbot abandonment.

3.1 Quantitative Classification Results

As shown in Figure 2, in the C1 condition, i.e. taking the user's last conversation exchange with the chatbot into account, the attention-based model performed the best in all performance metrics among all classifiers, with an AUROC being up to 95.7% and an F1-score of 83.2%. This result suggests that in the in-the-moment detection of user abandonment, using the attention-based model we designed can detect these instances with high accuracy and reliability.



(a) Performance of AUROC

(b) Performance of F1-score

Figure 2: Performance comparison on four models across four conditions of messages.

We further examine if it is possible to use earlier message exchanges to predict future chatbot abandonment, i.e. whether the performances of the classifiers in conditions C2, C3, C4, respectively, could still achieve acceptable performance.

However, in early prediction of chatbot abandonment, the results (Figure 2) show a dramatic drop of F1-score in all models except the BERT End-to-End model. This means that without considering the users' latest message with the chatbot makes all the supervised learning classifiers that rely on the labels not able to successfully early predict the occurrence of chatbot abandonment. We found that it was because the latest message was the most indicative of chatbot abandonment; according to our observations on the data, which is also suggested in prior research [3], we suspect that it was because abandonment attempts were ad hoc and are immediate responses to error messages [3]. Another main contributing factor for this performance drop was the focus of the hand-crafted coding; that is, most assigned labels to a conversation exchange were mainly related to that message. Consequently, when the features of that message were not taken into account in C2 through C4, the model missed vital information of the chatbot abandonment related to that message, including the error messages that the user lastly saw before leaving. On the other hand, the BERT end-to-end model considers the lexical feature, the semantic meaning and the context of the messages, making it possible to capture users' negative emotional responses toward the chatbot before abandonment. This may explain why it could more likely to predict future abandonment instances as early as in the C4 condition even if it did not consider the last error message.

3.2 Qualitative Insights

Despite the performance drop in early prediction, the attention-based model allows us to shed lights on the important features that contributed to the detection of in-the-moment chatbot abandonment. We used t-Distributed Stochastic Neighbor Embedding (t-SNE) [13] to reduce our data into 2 dimension and Density-based spatial clustering of applications with noise(DBSCAN) [7] to divide our 2-dimensional data into classes.

In Class 0, as shown in Figure 3; the top ten attention weights were M1-L11, M1-L0, M1-L5, M2-L11, M3-L11, M2-L5, M2-L4, M1-L14, M2-L0, M1-L4, where the feature list can be found in Table 1. *M* indicates at which conversation exchange the feature was given a high attention weight (1: latest; 4: the fourth latest). The highest weight in M1-L11 suggests that the presence of users manually typing in the last message was a strong indicator of chatbot abandonment (as opposed to selecting a button). In fact, this feature appeared from M1 through M3. L5 represents users requesting information from chatbot; this feature had a higher weight in the last two messages. M1-L14 represents the chatbot indicating that it does not understand the users' meaning (or cannot understand their intents) in the latest message. Taking these features together, this suggests that a significant portion of chatbot abandonment was correlated with users finding that the chatbot could not understand the requests they typed on their own. L0 represents "the last message", which should presumably appear whenever the user leaves a topic. Finally, L4 represents topic switch, which was another important feature in M1 and M2. The presence of this feature means that user feeling unsatisfied with the chatbot's responses after switching to a different topic was also indicative of their later chatbot abandonment. On the other hand, the fact that this feature appeared in M1 and M2 but not in more distant messages implies that the users in our dataset typically did not try more than two times.

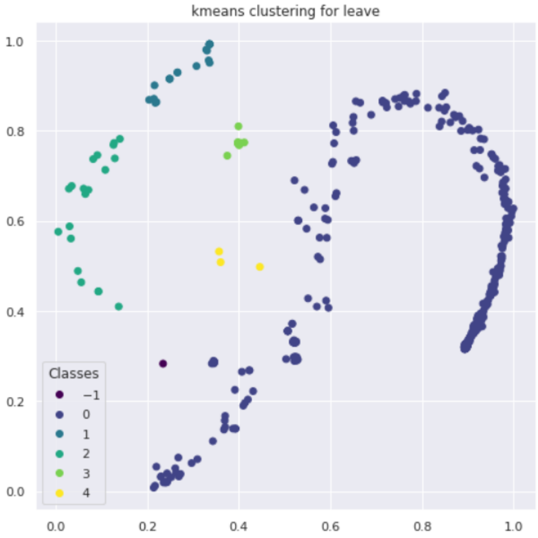
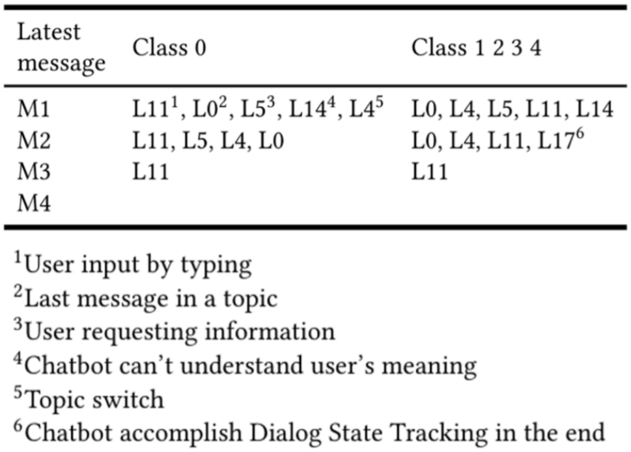


Figure 3: Classes of attention weight of user's abandonment



Latest message	Class 0	Class 1 2 3 4
M1	L11[1], L0[2], L5[3], L14[4], L4[5]	L0, L4, L5, L11, L14
M2	L11, L5, L4, L0	L0, L4, L11, L17[6]
M3	L11	L11
M4		

[1] User input by typing
[2] Last message in a topic
[3] User requesting information
[4] Chatbot can't understand user's meaning
[5] Topic switch
[6] Chatbot accomplish Dialog State Tracking in the end

Table 2: Top 10 features in different classes

As in other classes (see Table 2), although they all shared the same set of features, they were separated into different classes because of the differences in the weight of each feature, or in which messages at which the occurrence of the features were predictive. For further discussion, we firstly explain what is "entering dialog state tracking". As we mentioned earlier, the service that the chatbot provided included currency-exchange converting information, credit card introduction, housing-loan evaluation, and investment information. We define a conversation that entered dialog state tracking since the chatbot determined whether the user was attempting to one of the services or not. The feature that appeared in Classes 1 through 4 but was absent in Class 0 was L17, which stood for the chatbot having accomplished dialog state tracking in the end. The separation between Class 0 and other classes means that the way users left the chatbot, or the factors that caused the user to leave the chatbot, differed between whether they enter dialog state tracking or not. On the other hand, the similarity of the features between these classes also indicate that multiple signs of chatbot abandonment were pervasive.

4 DISCUSSION

4.1 Features Predictive of Chatbot Abandonment

We found that two sets of features identified by the model are predictive of users abandoning the banking chatbot. The features shared by both sets are particular essential signs for these abandonment instances: the chatbot being unable to understand the request users manually type on their own. This results resonates with several studies that have suggested that providing options for users can reduce the occurrence of conversation breakdowns [2, 3, 9]. It is perhaps when users selected to enter their own requests instead of following the options offered by the chatbot, the free-form requests were likely to cause the messages to be not appropriately understood (e.g. containing misspelling, unfinished sentence, or certain word) or handled by the chatbot. The requests were also likely to be out of the scope of the chatbot's intended services, causing that when perceiving the chabot unable to handle such specific requests, they were likely to just abandon the service. The other set of features indicated a scenario of users entering the dialog state tracking, which shared the core features but included additional features, suggesting a different path of users abandonment when the dialog state tracking had been activated.

4.2 Implications for Chatbot Service Owners

We have shown that building a supervised learning model on annotated conversation logs of nearly twenty thousand message exchanges can achieve great performances in detecting in-the-moment chatbot abandonment. If adding annotations is too prohibitive to conduct, building a BERT end-to-end model can also achieve a fairly good performance. The next critical question is: how early the business wants to be informed of these "red-flag" conversations. This question involves a choice of early prediction of chat abandonment versus an in-the-moment chatbot abandonment detection, where the latter considers the latest message the user enters and the response the chatbot is going to deliver to the user. If the business deems the timing of the latter as too late, it should be bear in mind that a BERT end-to-end model may perform better than other supervised learning models. Alternatively, annotators should consider wider context instead of focusing on the current message when annotating. Finally, the business should consider whether it is important to explain the prediction result or to get insights into the features of why users leave the chatbot the business runs. While a pre-trained end-to-end model can achieve better results in early prediction of chatbot abandonment, it is challenging to get insights into the features that cause the abandonment. On the other hand, although an explainable attention-based model can offer insights, the cost of building an attention-based model would entail tremendous effort, especially when the annotations embed larger contextual information.

5 FUTURE WORK

Given an increasing number of businesses attempting to leverage a task-oriented chatbot as another medium to provide services, we deem it important to anticipate when users would abandon a task-oriented chatbot service that is intended to help the business. Being able to detect these instances in-the-moment or to predict these instances early, businesses can know when to assist users to achieve users' requests accordingly. We have shown that when performing in-the-moment abandonment detection, an explainable attention-based model, despite the extra effort of adding annotations, not only achieved a good prediction outcome, but also provided insights into important features indicative of chatbot abandonment. However, if the business aims to achieve early prediction of chatbot abandonment, a BERT end-to-end model may be more favorable. Our future work includes comparing the false positive and false negative prediction results between the the BERT end-to-end model and the attention-based model, hoping to get insights into the differences in the signs they picked up in their prediction. Second, we may use different methods to analyze the attention layers of BERT end-to-end model and get what words in the conversation does BERT end-to-end model focus on. Third, we aim to automatically generate the annotations we used for the attention-based model using the BERT end-to-end model, which, if successfully, can reduce human effort in generating these labels. Then we will combine it with our attention-based-model to yield a complete user abandonment prediction. Finally, our model use the data with at least four exchanges, we want to build models with any number exchanges to detect whether user will leave chatbot, which can be more widely used in every users in the future.

REFERENCES

[1] Paul M. Aoki, Margaret H. Szymanski, Luke Plurkowski, James D. Thornton, Allison Woodruff, and Weilie Yi. 2006. Where's the "Party" in "Multi-Party"? Analyzing the Structure of Small-Group Sociable Talk *(CSCW '06)*. Association for Computing Machinery, New York, NY, USA, 393–402.

[2] Zahra Ashktorab, Mohit Jain, Q. Liao, and Justin D. Weisz. 2019. Resilient Chatbots: Repair Strategy Preferences for Conversational Breakdowns. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019).

[3] Anonymous author. [n.d.].

[4] L. Breiman. 2004. Random Forests. *Machine Learning* 45 (2004), 5–32.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] S. Engelhardt, Emmeli Hansson, and Iolanda Leite. 2017. Better Faulty than Sorry: Investigating Social Recovery Strategies to Minimize the Impact of Failure in Human-Robot Interaction. In *WCIHAI@IVA*.

[7] M. Ester, H. Kriegel, J. Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD*.

[8] S. Hochreiter and J. Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9 (1997), 1735–1780.

[9] Mohit Jain, Pratyush Kumar, R. Kota, and S. Patel. 2018. Evaluating and Informing the Design of Chatbots. *Proceedings of the 2018 Designing Interactive Systems Conference* (2018).

[10] R. Jones and K. Klinkner. 2008. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *CIKM '08*.

[11] Dongkeon Lee, KyoJoong Oh, and Ho-Jin Choi. 2017. The chatbot feels you - a counseling service using emotional response generation. *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)* (2017), 437–440.

[12] Jane Lockwood. 2017. An analysis of web-chat in an outsourced customer service account in the Philippines. *English for Specific Purposes* 47 (07 2017), 26–39.

[13] L. V. D. Maaten and Geoffrey E. Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605.

[14] Jennifer Snyder-Duch Matthew Lombard and Cheryl Bracken. 2005. Practical Resources for Assessing and Reporting Intercoder Reliability in Content Analysis Research Projects. Retrieved April 19, (January 2005).

[15] Wyke Stommel, Trena Paulus, and David Atkins. 2017. "Here's the link": Hyperlinking in service-focused chat interaction. *Journal of Pragmatics* (03 2017).

[16] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems (NIPS).*

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L. Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. *ArXiv* abs/1706.03762 (2017).

[18] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense Knowledge Aware Conversation Generation with Graph Attention. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*.