Defining And Detecting Inconsistent System Behavior in Task-oriented Dialogues

Léon-Paul Schaub, Vojtech Hudecek, Daniel Stancl, Ondrej Dusek, Patrick Paroubek

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Defining And Detecting Inconsistent System Behavior in Task-oriented Dialogues

Léon-Paul Schaub[1, 2] Vojtěch Hudeček[3] Daniel Štancl[3]
Ondřej Dušek[3] Patrick Paroubek[1]
[1]Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, 91400 Orsay, France
[2]Akio, 43 rue de Dunkerque, 75010 Paris, France
[3]Charles University, Faculty of Mathematics and Physics, Malostranské náměstí 25, 118 00 Prague, Czechia
schaub@limsi.fr, hudecek@ufal.mff.cuni.cz, stancl@ufal.mff.cuni.cz,
odusek@ufal.mff.cuni.cz, pap@limsi.fr

RÉSUMÉ

Définition et détection des incohérences du système dans les dialogues orientés tâche.
Nous présentons des expériences sur la détection automatique des comportements incohérents des systèmes de dialogues orientés tâche à partir du contexte. Nous enrichissons les données bAbI/DSTC2 (Bordes *et al.*, 2017) avec une annotation automatique des incohérences de dialogue, et nous démontrons que les incohérences sont en corrélation avec les dialogues ratés. Nous supposons que l'utilisation d'un historique de dialogue limité et la prédiction du prochain tour de l'utilisateur peuvent améliorer la classification des incohérences. Si les deux hypothèses sont confirmées pour un modèle de dialogue basé sur les réseaux de mémoire, elles ne le sont pas pour un entraînement basé sur le modèle de langage GPT-2, qui bénéficie le plus de l'utilisation de l'historique complet du dialogue et obtient un score de précision de 0,99.

ABSTRACT

We present experiments on automatically detecting inconsistent behavior of task-oriented dialogue systems from the context. We enrich the bAbI/DSTC2 data (Bordes *et al.*, 2017) with automatic annotation of dialogue inconsistencies, and we demonstrate that inconsistencies correlate with failed dialogues. We hypothesize that using a limited dialogue history and predicting the next user turn can improve inconsistency classification. While both hypotheses are confirmed for a memory-networks-based dialogue model, it does not hold for a training based on the GPT-2 language model, which benefits most from using full dialogue history and achieves a 0.99 accuracy score.

MOTS-CLÉS : système de dialogue orienté-tâche, incohérences, modèle utilisateur, apprentissage automatique.

KEYWORDS: task-oriented dialogue systems, inconsistency, user model, machine learning.

1 Introduction

Compared to traditional pipeline architectures, the recent end-to-end neural-network-based dialogue systems have a simpler design and less space for error accumulation, but suffer from less control over the training, reduced explainability and a need for large amounts of training data and computing power, not to mention the difficulty to incorporate external knowledge bases. To address these problems, Madotto *et al.* (2018) developed Mem2Seq, an end-to-end architecture based on memory

Volume 1 : conférence principale, pages 142–152.

networks (Weston *et al.*, 2015; Sukhbaatar *et al.*, 2015) for learning from an external database with a relatively small model. Chen *et al.* (2019) then built WMM2Seq, a Mem2Seq-based dialogue system inspired by cognitive science research, whose architecture is composed of two memory networks, one learning from the dialogue history and the other from a knowledge base. Other neural-based works build on two-stage decoding within the same network (Hosseini-Asl *et al.*, 2020; Ham *et al.*, 2020). However, neither architecture solves the problem of system inconsistencies inherent in any dialogue generation task. Indeed, during a human-machine conversation, it is not uncommon to observe the machine saying something unexpected or inconsistent (Litman *et al.*, 2006; Engelbrecht & Möller, 2010). A detection and correction of these inconsistencies is difficult, but would constitute an important improvement since it would allow the system to correct itself (Zhang *et al.*, 2019), bringing us one step closer to a lifelong learning architecture (Veron, 2019; Hancock *et al.*, 2019). In this paper, we make the following contributions: (1) we enrich a task-oriented dialogue dataset with inconsistencies annotation, (2) we show that dialogue inconsistencies correlate with failures of the respective dialogues and (3) we perform a series of experiments to train and evaluate inconsistency classification models based on history and user modeling.

2 Related Works

In machine translation, Ma *et al.* (2019) showed that an incremental/simultaneous translation model can get faster by anticipating sequences, with results close to full sentence translation. In dialogue, Shang *et al.* (2020) reached state-of-the-art results for dialogue act classification by labeling speaker change in dialogue turns during learning, which shows the importance of speaker roles in the conversation. Auguste *et al.* (2019) take this idea one step further by learning to classify the dialogue act of the current and the next dialogue turn, with comparable results. Finally, Lin *et al.* (2020) create a system called "Imagine then Arbitrate" (ITA) to learn when to answer and when to listen, by imagining what the user will say to anticipate possible system errors.

Regarding system error analysis, Whitney *et al.* (2017) model with a POMDP (Sammut & Webb, 2010) the uncertainty of a dialogue agent when answering a user question to improve the answer accuracy. Welleck *et al.* (2019) use a natural language inference model to improve a system's consistency in a dialogue, Li *et al.* (2020) then integrate consistency into the system training signal. Dziri *et al.* (2019) apply a similar inference-based approach for dialogue system evaluation. During the DSTC6 shared task (Hori *et al.*, 2019), inconsistency detection for non-task-oriented dialogues was one of the problems investigated; however, the inconsistencies found remain quite specific to this type of dialogue. Gao *et al.* (2019) show that when a conversation exceeds a certain number of dialogue turns, end-to-end systems see their performance decrease, which they attribute to the the conversation history becoming noisy if it is too large. To our knowledge, there has not been a system that predicts the next user's turn and filters dialogue history to anticipate system inconsistencies.

3 Inconsistency Classification

We need to distinguish between understanding or decision errors in human-human dialogues, and bot-specific inconsistencies in a human-machine dialogue. Indeed, during a task-oriented conversation between two humans, errors or problems lead to almost systematic co-corrections between the two

interactors. Self-initiated and self-repaired or hetero-initiated co-corrections are applied whenever an error occurs (Chernyshova, 2018). However, these errors are corrected thanks to inference, i.e. the explicit errors of a speaker cause the listener to interpret and infer what the first speaker really meant. In the same way, the speaker who made the error infers what the listener could have understood and inferred, in order to self-correct (Deppermann, 2018; Fernandez *et al.*, 2006). In human-machine interaction, this inference is impeded by human knowing that they are talking to a machine with an a priori limited capacity for inference and correction. Therefore, when the user or the machine commits human-like errors, this limitation inevitably lowers the confidence and patience on the user part. The main focus in this paper is, however, on *inconsistencies* – system errors that are not human-like but endemic to the system's learning process. Inconsistencies may often cause the user to end the dialogue unsuccessfully (cf. our experiment in Section 4.2). For the machine to acquire some co-correction capability, i.e. to be able to detect inconsistencies in its own responses, it must have some representation of the user interpreting them. The DSTC2 dialogues are between a real system and users (Henderson *et al.*, 2014a), and thus include many system inconsistencies, especially based on speech recognition errors. A lot of these inconsistencies are recoverable , so the whole dialogue is not jeopardized. Based on an empiric analysis of the corpus, we list the most frequent types of inconsistencies found in human-computer interaction, with an example for each of them (see Table 1 for statistics):

1. **Intent misunderstood** (*Ex: U -> in which zone is it located? S -> Eraina is a great restaurant.*) Here, the system did not predict the correct user intent (*request*) and slot (*area*).
2. **Bad entity** (*Ex: U -> Mediterranean. S -> There is no Jamaican restaurant here.*) We distinguish this case from the previous one because in this case, the system actually predicted the correct intent and slot, but it then generated an incorrect output.
3. **Repetition** (*Ex: U -> gastropub. S -> What type of restaurant are you looking for? U -> Gastropub. S -> Ok let me look.*) The system forced the user to repeat the same utterance.
4. **"Bye" misunderstood** (*Ex: U -> goodbye. S -> The restaurant Paris serves French cuisine.*) This is a particular type of 1. where the system does not understand that the user wants to end the conversation.
5. **User insults** (*Ex: U -> i dont give a fuck if hk_fusion serves chinese food im looking for thai food. S -> am sorry but there is no fusion restaurant that matches your request.*) In addition to providing a clue on intent misunderstanding, this category also provides feedback about the user feeling, which is useful when considering commercial deployment (Kumar, 2010).
6. **Bad API call** (*Ex: U -> in the south. S -> api call = Kitchen, ANY_location, expensive*). As the API call is explicit, we can verify easily if it is consistent with the information given by the user.
7. **Incompleteness** (*Ex: U -> phone and address. S -> Phone is* ***) The system only responds to some of the slots requested by the user.
8. **Negation error** (*Ex: U -> no french food. S ->There are restaurants serving french food . What area do you want?*) The system is unable to process a negation in the user's request.

A lot of the above inconsistencies can be detected using simple rules (e.g., detecting that the system utterance repeats twice in a row). More elaborate rules may concern entity type mismatch between the user and system utterances, etc. The main source of information for all the rules is the dialogue history. However, a conversation history that is too large can become a source of noise for the system (Madotto *et al.*, 2018). According to several studies (Wolters *et al.*, 2009; Lai *et al.*, 2020), the longer a dialogue history is, the more difficult it becomes for the system to retrieve the most relevant

inconsistency type								total inconsistent	correct turns	total turns
1	2	3	4	5	6	7	8			
783	245	1,360	275	11	242	780	64	3,760	26,179	29,939

Table 1 – Number of inconsistencies of various types (see Section 3 for explanation of the individual types) in the bAbI corpus.

information, especially if the slots change during the dialogue. Therefore, our goal is to mimic the cognitive forget function (Bodner & Lindsay, 2003) during a dialogue (i.e., reproduce the same information filtering) and to define the optimal dialogue history size to remember. We note that the inconsistency annotation is not turn-independent. For example, in order to detect that the system says the same sentence twice and the user is bothered, we need to know turns t and $t+1$.[1] A legitimate question then is: To what extent can a reduction of the dialogue history size, possibly combined with the knowledge of the user's next turn, allow the system to better detect its own inconsistent behavior?

4 Data and Experiments

4.1 The bAbI Corpus and Our Additional Annotation

To answer the question asked in Section 3, we use the bAbI dialogue corpus (Bordes *et al.*, 2017), which is a postprocessed version of the DSTC2 corpus (Henderson *et al.*, 2014a), consisting of 3,232 English dialogues between a human and a POMDP-based restaurant reservation system (Young *et al.*, 2013). Dummy API calls were added to simulate access to an external database. A dialogue turn contains either an exchange between the user and the system, or an API call and its result. Detailed statistics are provided in Henderson *et al.* (2014b). Based on the inconsistency types identified in Section 3, we automatically added inconsistency annotation to each dialogue turn by employing simple pattern-matching rules.[2] We conduct annotation evaluation on a sample of 150 dialogue turns by two linguists (with inter-annotator agreement in terms of Cohen's kappa of 0.76). We consider the annotated dataset as a silver-standard (computer annotation with human evaluation). For the evaluation, we choose labelling accuracy as the metric to reflect the annotation performance and obtain a 0.79 accuracy score. The accuracy metric is sufficient because the number of dialogues with and without inconsistencies is not overly imbalanced. Coming from the original DSTC2 corpus, each dialogue is also annotated according to the DSTC2 handbook guidelines[3] with a success mark on a satisfaction scale from 0 (unsatisfied) to 5 (satisfied) (Walker *et al.*, 1997). In total, 502 dialogues are failed (16%).

dialogue count	success	failure
with inconsistencies	1,715	420
without inconsistencies	1,020	82

Table 2 – Number of successful and failed dialogues with and without inconsistencies in bAbI data.

1. We assume that the system initiates the dialogue. Therefore, we take the next user utterance from the same turn. This is the case for DSTC2 (see Section 4).
2. The full code for the rules is available at https://github.com/DiaSER21/consistency.
3. https://github.com/matthen/dstc/blob/master/handbook.pdf

4.2 Correlation Between Failure and Inconsistencies

Unsurprisingly, almost all failed dialogues contain inconsistent system responses. The Fisher exact test (Fisher, 1936) shows that there is a very likely dependence between failed dialogues and the presence of inconsistency – dialogues with inconsistencies are ca. 3x more likely to fail (odds ratio 0.328, $p <$ 1e-20). [4] Most failed dialogue contain inconsistencies, but a much lower proportion of successful dialogues has them. Moreover, the number of inconsistencies in a dialogue is higher on average for the failed dialogues. There are many dialogues (around 15%) which can be considered as failed on closer inspection even though they are marked as successful. [5] This can explain why so many dialogues annotated in the original data as successful contain inconsistencies. The dialogue success is impacted not just by the presence of an inconsistency, but also by its relative position with respect to the key events in the transaction (e.g., API system call for fetching an answer, query for a confirmation etc.). This is why we felt justified in trying to gauge this impact.

We investigated which were the determining features in deciding whether a dialogue was a failure or not. We used Gaussian naïve Bayes (Chen *et al.*, 2009) from Scikit-Learn (Pedregosa *et al.*, 2011) to predict dialogue success. [6] Table 3 summarizes some of the different features used to improve the detection of failed dialogues. [7] If the dialogue contains inconsistencies already, they are more likely to occur again. We noticed that the types of inconsistencies are not that important to detect failed dialogues. We calculate unsuccessful dialogues' detection F1-score (unsuccessful counts as positive). The best results are achieved with simple TF-IDF-based textual features of user, system and API call inputs, coupled with the number of total inconsistencies and with the number of inconsistencies appearing before the first system's API call in the dialogue. The results confirm that inconsistencies have an influence on dialogue success.

features	precision	recall	F1-score
textual	0.56	0.52	0.53
textual + total inconsistencies	0.57	0.62	0.60
textual + total inconsistencies + inconsistencies before API call	**0.65**	0.57	**0.61**

Table 3 – Failed dialogue prediction with and without inconsistency annotation.

4.3 Models, Metrics and Experiments

Our rule-based automatic annotation (see Section 4.1) uses the whole annotated dialogue. However, we are not able to see future context in real use cases. Therefore, we raise a question about the possibility to match the performance by training a classification model based solely on the past context. We trained four different classifiers on our annotation to predict inconsistencies:

4. We note that although the dialogue inconsistencies are correlated with a higher chance of a dialogue failure, the correlation does not imply a strict cause-effect relationship, as users may be sufficiently motivated to put up with punctual inconsistencies if they feel that they can obtain what they want from the system.

5. For instance, the user never speaks during the dialogue, user requests are not satisfied, the system was unable to finish the dialogue, or there are numerous speech recognition errors.

6. Gaussian naïve Bayes worked better than other machine learning algorithms such as SVM, logistic regression, random forest and multilayer perceptron in our preliminary experiments.

7. Features used: the user and system utterances transformed into word-based TF-IDF weights, system database API call with the same TF-IDF, total number of inconsistencies in the dialogue, number of inconsistencies happening before and after the API call, types of inconsistencies present.

	Bi-LSTM				DIET				WMM2Seq				GPT-2			
	binary		multi		binary		multi		binary		multi		binary		multi	
input	acc	F1	acc	F1	acc	F1	acc	F1	acc	F1	acc	F1	acc	F1	acc	F1
c	59.0	52.3	83.9	9.53	85.7	66.6	83.4	52.6	86.0	50.1	83.7	35.7	64.2	77.2	62.1	8.1
c+h_1	80.5	65.0	84.0	10.2	85.2	58.4	83.1	43.0	86.9	48.1	82.6	41.8	84.3	90.5	84.2	38.8
c+h_2	77.0	59.8	84.8	9.18	83.2	55.2	82.6	50.0	**87.0**	46.2	82.6	39.6	85.2	91.0	85.1	40.7
c+h_f	71.0	48.9	85.4	6.57	79.3	53.7	83.6	50.7	85.9	44.4	82.6	44.6	99.9	99.9	**98.4**	**93.7**
c+n	79.2	71.1	80.2	9.57	89.7	64.1	**88.9**	54.1	**90.2**	57.3	85.1	26.1	25.2	8.2	10.2	22.9
c+n+h_1	88.0	82.1	84.2	8.48	87.9	76.6	**87.6**	52.0	**89.1**	53.4	81.3	39.3	80.6	87.7	79.2	40.8
c+n+h_2	87.4	81.4	81.8	8.41	89.0	77.6	85.6	40.2	**89.1**	55.1	81.4	39.6	85.2	90.9	85.1	40.6
c+n+h_f	79.0	65.9	85.7	5.72	87.0	70.2	86.2	48.6	86.4	46.9	82.6	40.8	**99.9**	99.9	**98.5**	**93.5**

Table 4 – Inconsistency classification accuracy and weighted-averaged F1 scores (binary and multiclass mode) of our models. The most frequent baseline achieves accuracy 87%. We present results with various combinations of the input data. Possible inputs are: current turn (c), next user utterance (n), last 1 or 2 turns of dialogue history (h_1,h_2) or full history (h_f).

— **Bi-LSTM with attention** (Jang *et al.*, 2020) is a simple model for sequence/text classification but highly effective when it has to deal with long-term information such as dialogue history.
— **DIET classifier**, the dialogue intents entities transformer, is a transformer-based (Devlin *et al.*, 2019) dialogue intents classifier (Wu *et al.*, 2020) that outperforms most of recent classifiers in the user intention detection task.
— **WMM2Seq** (Chen *et al.*, 2019) is a memory network-based model that uses two different memory modules: context (dialogue history as episodic memory) and knowledge base (API calls as semantic memory) for generating system responses, one word at a time.
— **GPT-2** (Radford *et al.*, 2019) is a transformer-based architecture made of several transformer decoder blocks (Vaswani *et al.*, 2017), stacked one on top of the other. The architecture is pre-trained for language modeling on a huge corpus and is capable of effective finetuning for many downstream tasks. We finetune the model in a multitask setting, i.e. we optimize both inconsistency classification loss and response generation loss.

We use classification accuracy and weighted macro average of F1 scores as the evaluation metrics, and we train the models both for binary (inconsistency or not) and multiclass classification (predicting specific inconsistency type, or no inconsistency). We use the most frequent label prediction as a strong baseline (no inconsistency, present in 87% of the examples, i.e. accuracy 87%). The results are shown in Table 4 and discussed next. We use 2,117 dialogues for training and 1,115 for testing.

The results show that, even if the baseline is strong (87%), it is outperformed by all the models. The best results (99%) are obtained by the GPT-2 based model when using the whole dialogue history (*h*) and the next user utterance (*nu*). When *h* is not used, the performance decreases; *nu* has a smaller effect. We believe that GPT-2 is capable of extracting input information that is most relevant for inconsistency classification, therefore it benefits from the long history. Indeed, when we examined the results, we observed that almost all GPT classification errors are related to the "incompleteness" inconsistency. These cases depend only on the immediate context (previous utterance). On the contrary, DIET and WMM2Seq obtained the best results (0.90) with the next user utterance and no history at all, even if the performance difference without the next user utterance is smaller than GPT-2's. Also, a simple Bi-LSTM outperforms the baseline when using both *h1* and *h2* with *nu* in binary mode but fails to pick up the necessary features in the multiclass mode.We observe that with

model	**bi-LSTM**	**DIET**	**WMM2Seq**	**GPT-2**
κ	0.74	0.76	0.67	**0.97**

Table 5 – Cohen's Kappa values for comparing the best models' predictions to the ground truth labels.

less information, WMM2Seq gets the highest accuracy after GPT-2. However, as GPT-2 training is both costly and needs the whole dialogue to get the best performance, the results confirm the need of predicting next user's utterance to have a more accurate model, in case where a smaller model is required or when the whole dialogue history is not available. We also compare the best model variants' predictions to ground-truth labels in binary mode and measure Cohen's Kappa (Ben-David, 2008) to asses that the models' performance is better than chance. The results are shown in Table 5.

5 Conclusion and perspectives

This work presents a new dataset based on the DSTC2/bAbI corpus that allows research on the task of detecting dialogue inconsistency, which has not been explored much so far. We conducted experiments that revealed a correlation between system turn inconsistencies and dialogue failures. This fact can be exploited in further research of dialogue modeling to prevent failures. Furthermore, we applied four different classifier architectures to automatically detect inconsistencies in the newly formed dataset. Among the explored architectures, the best performing were a GPT-2-based classifier and the WMM2Seq model. Interestingly, while GPT-2 strongly benefits from the provided history context, the WMM2Seq performed best when no history was used and next user utterance was available to the model, which makes it more suitable for the real world usecases. Access to the next utterance improved results across the board. With this set of experiments, we provide a first proof of the benefit we might gain by having dialogue systems to incorporate an oracle for predicting the next user turn, a step toward a future dialogue architecture with a dual system and user model. In future works, we will evaluate on more complex datasets in order to confirm the usefulness of this new feature when detecting system inconsistencies.

Acknowledgements

This work is supported by the HumanE-AI-Net project (EC Horizon 2020 grant no. 952026) [8] and AKIO Software [9] in the form of the DIASER microproject (*WP3 – Human AI Collaboration and Interaction*) and by Charles University grants PRIMUS 19/SCI/10, GAUK 302120, and SVV 260 575. We also want to thank the reviewers for their great remarks that helped us improving greatly the paper.

References

AUGUSTE J., BÉCHET F., DAMNATI G. & CHARLET D. (2019). Skip Act Vectors: integrating dialogue context into sentence embeddings. In *Tenth International Workshop on Spoken Dialogue*

8. https://www.humane-ai.eu
9. https://www.akio.com

Systems Technology, Syracuse, Italy. HAL : hal-02125259.

BEN-DAVID A. (2008). Comparison of classification accuracy using cohen's weighted kappa. *Expert Systems with Applications*, **34**(2), 825–832.

BODNER G. E. & LINDSAY D. S. (2003). Remembering and knowing in context. *Journal of Memory and Language*, **48**(3), 563–580.

BORDES A., BOUREAU Y.-L. & WESTON J. (2017). Learning end-to-end goal-oriented dialog.

CHEN J., HUANG H., TIAN S. & QU Y. (2009). Feature selection for text classification with naïve bayes. *Expert Systems with Applications*, **36**(3), 5432–5435.

CHEN X., XU J. & XU B. (2019). A Working Memory Model for Task-oriented Dialog Response Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 2687–2693, Florence, Italy: Association for Computational Linguistics. DOI : 10.18653/v1/P19-1258.

CHERNYSHOVA E. (2018). *Expliciter et inférer dans la conversation : modélisation de la séquence d'explicitation dans l'interaction*. Theses, Université de Lyon. HAL : tel-02070720.

DEPPERMANN A. (2018). Inferential practices in social interaction: A conversation-analytic account. *Open Linguistics*, **4**(1), 35 – 55. DOI : https://doi.org/10.1515/opli-2018-0003.

DEVLIN J., CHANG M.-W., LEE K. & 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)*, p. 4171–4186, Minneapolis, Minnesota: Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

DZIRI N., KAMALLOO E., MATHEWSON K. & ZAIANE O. (2019). Evaluating coherence in dialogue systems using entailment. 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)*, p. 3806–3812, Minneapolis, Minnesota: Association for Computational Linguistics. DOI : 10.18653/v1/N19-1381.

ENGELBRECHT K.-P. & MÖLLER S. (2010). Sequential classifiers for the prediction of user judgments about spoken dialog systems. *Speech Communication*, **52**(10), 816 – 833. DOI : https://doi.org/10.1016/j.specom.2010.06.004.

FERNANDEZ R., LUCHT T., RODRIGUEZ K. & SCHLANGEN D. (2006). Interaction in task-oriented human-human dialogue: the effects of different turn-taking policies. In *2006 IEEE Spoken Language Technology Workshop*, p. 206–209. DOI : 10.1109/SLT.2006.326791.

FISHER R. A. (1936). Design of experiments. *British Medical Journal*, **1**(3923), 554–554. PMC2458144[pmcid].

GAO S., SETHI A., AGARWAL S., CHUNG T. & HAKKANI-TUR D. (2019). Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, p. 264–273, Stockholm, Sweden: Association for Computational Linguistics. DOI : 10.18653/v1/W19-5932.

HAM D., LEE J.-G., JANG Y. & KIM K.-E. (2020). End-to-End Neural Pipeline for Goal-Oriented Dialogue Systems using GPT-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 583–592, Online: Association for Computational Linguistics.

HANCOCK B., BORDES A., MAZARE P.-E. & WESTON J. (2019). Learning from dialogue after deployment: Feed yourself, chatbot! In *Proceedings of the 57th Annual Meeting of the Association*

for Computational Linguistics, p. 3667–3684, Florence, Italy: Association for Computational Linguistics. DOI : 10.18653/v1/P19-1358.

HENDERSON M., THOMSON B. & WILLIAMS J. D. (2014a). The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, p. 263–272, Philadelphia, PA, U.S.A.: Association for Computational Linguistics. DOI : 10.3115/v1/W14-4337.

HENDERSON M., THOMSON B. & WILLIAMS J. D. (2014b). The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, p. 263–272, Philadelphia, PA, U.S.A.: Association for Computational Linguistics. DOI : 10.3115/v1/W14-4337.

HORI C., PEREZ J., HIGASHINAKA R., HORI T., BOUREAU Y.-L., INABA M., TSUNOMORI Y., TAKAHASHI T., YOSHINO K. & KIM S. (2019). Overview of the sixth dialog system technology challenge: DSTC6. *Computer Speech & Language*, **55**, 1–25. DOI : https://doi.org/10.1016/j.csl.2018.09.004.

HOSSEINI-ASL E., MCCANN B., WU C.-S., YAVUZ S. & SOCHER R. (2020). A Simple Language Model for Task-Oriented Dialogue. *arXiv:2005.00796 [cs]*.

JANG B., KIM M., HARERIMANA G., KANG S.-U. & KIM J. W. (2020). Bi-lstm model to increase accuracy in text classification: Combining word2vec cnn and attention mechanism. *Applied Sciences*, **10**(17). DOI : 10.3390/app10175841.

KUMAR V. (2010). Customer relationship management. *Wiley international encyclopedia of marketing*.

LAI T. M., HUNG TRAN Q., BUI T. & KIHARA D. (2020). A simple but effective bert model for dialog state tracking on resource-limited systems. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 8034–8038. DOI : 10.1109/ICASSP40776.2020.9053975.

LI M., ROLLER S., KULIKOV I., WELLECK S., BOUREAU Y.-L., CHO K. & WESTON J. (2020). Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 4715–4728, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.428.

LIN Z., KANG X., LI G., JI F., CHEN H. & ZHANG Y. (2020). "wait, i'm still talking!" predicting the dialogue interaction behavior using imagine-then-arbitrate model.

LITMAN D., SWERTS M. & HIRSCHBERG J. (2006). Characterizing and predicting corrections in spoken dialogue systems. *Computational Linguistics*, **32**(3), 417–438. DOI : 10.1162/coli.2006.32.3.417.

MA M., HUANG L., XIONG H., ZHENG R., LIU K., ZHENG B., ZHANG C., HE Z., LIU H., LI X., WU H. & WANG H. (2019). Stacl: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 3025–3036, Florence, Italy: Association for Computational Linguistics. DOI : 10.18653/v1/P19-1289.

MADOTTO A., WU C.-S. & FUNG P. (2018). Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 1468–1478, Melbourne, Australia: Association for Computational Linguistics. DOI : 10.18653/v1/P18-1136.

PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V. *et al.* (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, **12**, 2825–2830.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019). Language models are unsupervised multitask learners.

SAMMUT C. & WEBB G. I., Éds. (2010). *POMDPs*, In C. SAMMUT & G. I. WEBB, Éds., *Encyclopedia of Machine Learning*, p. 776–776. Springer US: Boston, MA. DOI : 10.1007/978-0-387-30164-8_642.

SHANG G., TIXIER A. J.-P., VAZIRGIANNIS M. & LORRÉ J.-P. (2020). Speaker-change aware crf for dialogue act classification.

SUKHBAATAR S., SZLAM A., WESTON J. & FERGUS R. (2015). End-to-end memory networks. In C. CORTES, N. D. LAWRENCE, D. D. LEE, M. SUGIYAMA & R. GARNETT, Éds., *Advances in Neural Information Processing Systems 28*, p. 2440–2448. Curran Associates, Inc.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, p. 5998–6008, Long Beach, CA, USA.

VERON M. (2019). Lifelong learning et systèmes de dialogue : définition et perspectives. In *Rencontres des Etudiants Chercheurs en Informatique pour le Traitement Automatique des Langues*, Toulouse, France. HAL : hal-02301064.

WALKER M. A., LITMAN D. J., KAMM C. A. & ABELLA A. (1997). PARADISE: A framework for evaluating spoken dialogue agents. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, p. 271–280, Madrid, Spain: Association for Computational Linguistics. DOI : 10.3115/976909.979652.

WELLECK S., WESTON J., SZLAM A. & CHO K. (2019). Dialogue Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 3731–3741, Florence, Italy: Association for Computational Linguistics. DOI : 10.18653/v1/P19-1363.

WESTON J., CHOPRA S. & BORDES A. (2015). Memory networks. In *3rd International Conference on Learning Representations (ICLR2015)*, San Diego, CA, USA.

WHITNEY D., ROSEN E., MACGLASHAN J., WONG L. L. S. & TELLEX S. (2017). Reducing errors in object-fetching interactions through social feedback. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, p. 1006–1013. DOI : 10.1109/ICRA.2017.7989121.

WOLTERS M., GEORGILA K., MOORE J. D., LOGIE R. H., MACPHERSON S. E. & WATSON M. (2009). Reducing working memory load in spoken dialogue systems. *Interact. Comput.*, **21**(4), 276–287. DOI : 10.1016/j.intcom.2009.05.009.

WU C.-S., HOI S. C., SOCHER R. & XIONG C. (2020). TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 917–929, Online: Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.66.

YOUNG S., GAŠIĆ M., THOMSON B. & WILLIAMS J. D. (2013). POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, **101**(5), 1160–1179. DOI : 10.1109/JPROC.2012.2225812.

ZHANG W., FENG Y., MENG F., YOU D. & LIU Q. (2019). Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 4334–4343, Florence, Italy: Association for Computational Linguistics. DOI : 10.18653/v1/P19-1426.