FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

ABSTRACT OF DOCTORAL THESIS

Vojtěch Hudeček

Low resource methods for dialogue systems applications

Institute of Formal and Applied Linguistics

Supervisor: Mgr. et Mgr. Ondřej Dušek, Ph.D.

Study Program:	Computer Science
Specialization:	Computational Linguistics

Prague 2023

The results of this thesis were achieved in the period of a doctoral study at the Faculty of Mathematics and Physics, Charles University in years 2017–2023.

Title: Low resource methods for dialogue systems applications

Author (Candidate): Mgr. Vojtěch Hudeček

Department:
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University
Malostranské náměstí 25
118 00 Prague 1, Czech Republic

Supervisor:
Mgr. et Mgr. Ondřej Dušek, Ph.D.
Institute of Formal and Applied Linguistics

Opponents:
Prof. M.Sc. Gabriel Skantze, Ph.D.
Department of Intelligent Systems
School of Electrical Engineering and C.S.
KTH Royal Institute of Technology
Lindstedtsvägen 24
100 44 Stockholm, Sweden

Ing. Petr Schwarz, Ph.D
Department of Computer Graphics and Multimedia
Faculty of Information Technology
Brno University of Technology
Božetěchova 2
612 00 Brno, Czech Republic

Chairman:
doc. Ing. Zdeněk Žabokrtský, Ph.D.
Institute of Formal and Applied Linguistics

The thesis defence will take place on February 9, 2024 at 10:30 a.m. in front of a committee for thesis defences in the branch Mathematical Linguistics at the Faculty of Mathematics and Physics, Charles University, Malostranské nám. 25, Prague 1, room S1.
The thesis can be viewed at the Study Department of Doctoral Studies of the Faculty of Mathematics and Physics, Charles University, Ke Karlovu 3,
Prague 2.

Disertační práce byla vypracována na základě výsledků získaných během doktorského studia na Matematicko-fyzikální fakultě Univerzity Karlovy v letech 2017–2023.

Název práce: Metody pracující s omezeným množstvím zdrojů pro využití v dialogových systémech

Autor (Kandidát): Mgr. Vojtěch Hudeček

Katedra:
Ústav formální a aplikované lingvistiky
Matematicko-fyzikální fakulta,
Univerzita Karlova
Malostranské náměstí 25
118 00 Prague 1, Česká republika

Vedoucí:
Mgr. et Mgr. Ondřej Dušek, Ph.D.
Ústav formální a aplikované lingvistiky

Oponenti:
Prof. M.Sc. Gabriel Skantze, Ph.D.
Department of Intelligent Systems
School of Electrical Engineering and C.S.
KTH Royal Institute of Technology
Lindstedtsvägen 24
100 44 Stockholm, Sweden

Ing. Petr Schwarz, Ph.D
Ústav Počítačové grafiky a multimédií
Fakulta informačních technologií
Vysoké učení technické
Božetěchova 2
612 00 Brno, Česká republika

Předseda komise:
doc. Ing. Zdeněk Žabokrtský, Ph.D.
Ústav formální a aplikované lingvistiky

Obhajoba se uskuteční 9. února 2024 v 10:30 v budově Ústavu formální a aplikované lingvistiky na adrese Malostranské nám. 25, Praha 1 v učebně S1.
Práce je k nahlédnutí na studijním oddělení Matematicko-fyzikální fakulty Univerzity Karlovy na adrese Ke Karlovu 3, Prague 2.

Acknowledgements

This work was supported by the European Research Council (Grant agreement No. 101039303 NG-NLG), and by the Charles University project SVV 260575. It used resources provided by the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth and Sports project No. LM2018101). Parts of the work were also supported by Charles University grants PRIMUS 19/SCI/10, GAUK 302120, project TL05000236 *AI asistent pro žáky a učitele* co-financed by the Technological Agency of the Czech Republic within the ÉTA 5 Programme and the ANRT CIFRE #2017/1543, HumanE-AI-Net project / EC Horizon 2020, Grant Agreement H2020-FETFLAG-2018-2020 no. 952026

1 Introduction

Human language is a convenient and natural means of communication for human beings. It is, therefore, desirable to implement an interface that mimics natural language and allows humans to interact with computers like they would with other human individuals.

To achieve this goal, we need to be able to transfer information between human users and the computer. Humans most often use speech or writing to encode and transfer information, so various techniques have been invented that deal with this kind of encoding, such as Automatic Speech Recognition (ASR), Optical Character Recognition (OCR), and Text-to-speech Synthesis (TTS). However, to efficiently transfer information, we need the ability to engage in a conversational exchange. A conversation (dialogue) offers additional means of communication such as clarification, information updates, or more effective encoding through context reference, etc.

In this work, we focus on the textual part of the problem, i.e. we do not care about encoding or decoding natural language in a signal such as speech. Put simply, the task of a Dialogue System (DS)(Jurafsky, 2000) is to generate the correct natural language response r given the natural language user utterance u and context c. In this work, we understand the dialogue as a *turn-taking* conversation, i.e. participants (user and system) communicate in alternating *turns*. In this work, we exclusively focus on two-party dialogues.

Despite some successful dialogue system deployments, current dialogue systems still suffer from several drawbacks. Usually, the DSs are tailored to specific applications, and applying them in other domains is hard. Typically, the system is customized to handle a set of predefined domains with a high success rate. A lot of effort goes into designing an ontology and handling domain-specific scenarios.

Ideally, a system would learn common behavioral patterns required to successfully finish the defined goal through conversational exchange. Another problem is that there seems to be a trade-off between interpretability and performance or scalability of the systems in the case of neural network-based models. In most cases, the more complex and capable the model is, the harder it is to interpret its behavior and explain its decisions. Hence, the vast majority of current dialogue systems deployed in production consist of multiple interconnected components that are rather conservative in terms of the used technology.

This thesis proposes solutions to some of these problems, especially in the task-oriented setting. For more detailed information, we refer the reader to the full text of this thesis, which contains necessary background information, related work, and a much more detailed description of the proposed methods and their results.

Datasets description

Here we briefly introduce the datasets that we use for experiments in our thesis.

MultiWOZ (**MW**) is an established task-oriented dataset introduced by Budzianowski et al. (2018). It has been released in several versions; the standard most commonly used nowadays are MultiWOZ 2.1 and MultiWOZ 2.2. MultiWOZ contains over 10,000 annotated dialogues and spans multiple domains – restaurant and hotel reservations, tourist attraction search, and taxi and train reservations. While some of the dialogues use only a single domain, most of them are multi-domain.

DSTC2 (Henderson et al., 2014) was introduced as a part of a challenge to improve state tracking within dialogue systems. It contains over 3,000 dialogues covering a single domain around restaurant reservations.

CamRest676 (**CR**) (Wen et al., 2017) is another crowd-sourced dialogue corpus gathered via the Wizard-of-Oz scheme. CamRest676, with its 676 conversations, is the smallest of the datasets used in this work, and it is also a single-domain dataset focused on helping users to find a restaurant in Cambridge, UK.

Schema-guided dialogue (**SGD**) is a large (more than 20,000 dialogues) multi-domain (around 20 domains covered) dataset containing a total of 45 API services based on a pre-defined schema. First, the data was collected via a simulator that interacts with the API services, and then the dialogues were paraphrased using crowd-sourcing.

ATIS (**AT**) (Hemphill et al., 1990) contains utterances taken from conversations about flight searches and reservations.

Cambridge SLU (CS) (Henderson et al., 2012) resembles the CamRest 676 dataset but is larger and focuses only on other user parts of the conversations. Therefore, Cambridge SLU is not a true dialogue dataset as it contains only single utterances and can be used solely for the NLU task.

Stanford Multidomain Dialogues (**SMD**) (Eric et al., 2017) contains concise dialogues between a driver and an in-car virtual assistant about appointments, navigation, and weather. The dataset assumes interaction with the database or external APIs.

2 Discovering dialogue slots

Getting raw, unlabeled data for dialogue system training is not difficult, especially if we restrict the target domain. In general, recording conversations in real life or artificial conditions is sufficient. A requirement for dialogue state labels makes this process much more costly. The sets of slots and their values typically must be designed by domain experts. This procedure consists of multiple tasks:

1. Determine which concepts need to be captured.
2. Define the captured concepts in a consistent way.
3. Label the occurrences of these concepts in the training data.

We present a novel approach to discovering a set of domain-relevant dialogue slots and their values given a set of dialogues in the target domain (such as transcripts from a call center). Our approach requires no manual annotation to tag slots in dialogue data. This substantially simplifies the dialogue system design and training process, as the developer no longer needs to design a set of slots and annotate their occurrences in the training data.

Most of the contents of this chapter were published at ACL 2021 (Hudeček et al., 2021).

2.1 Method overview

Figure 2.1 depicts a diagram describing our approach. Our slot discovery method has three main stages:

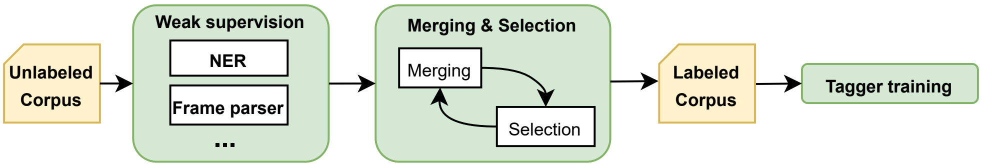

Figure 2.1: Illustration of our pipeline. First, we analyze an unlabeled in-domain corpus with supplied domain-agnostic linguistic annotation models, such as a frame-semantic parser or NER. This results in slot candidates. Next, we iteratively merge and select slot candidates to obtain domain-relevant slots. Finally, we use the resulting slot labels in the corpus to train a neural slot tagger.

1. We obtain weak supervision labels from automatic generic annotation. We obtain this annotation using domain-independent natural language taggers such as a semantic frame parser or a named entity recognizer (NER).
2. We identify domain-relevant slots based on the annotation labels by iteratively (a) merging and (b) ranking and selecting the most viable candidates
3. We use the discovered slots to train an independent slot tagger

Consider an example in Figure 2.2. First, we note that it helps to use multiple tagging models since a respective model might not capture some of the concepts. The set of tagged words from all the sources covers all the dialogue slot values (*cheap*, *Georgetown*). However, it contains irrelevant words (*restaurant*).

Therefore, to exploit the output of generic models, we need to polish and customize it to the specific domain.

Dialogue slots: `I am looking for a` `cheap` `restaurant in` `Georgetown` `.`
Semantic parser: `I am looking for a` `cheap` `restaurant` `in Georgetown.`
NER: `I am looking for a cheap restaurant in` `Georgetown` `.`

Figure 2.2: An utterance from the restaurant recommendation domain tagged with generic semantic parser (green) and Named Entity Recognition system (red). We provide a comparison with ground truth dialogue slot labels (blue).

2.2 Slot candidate identification by tagging semantic concepts

Our approach to selecting candidates for our method requires an initial pool of carefully chosen options representing coherent concepts. This step is critical to ensure the effectiveness of our selection process. We strive to gather as many candidates as possible to achieve this goal while preserving the above constraint. One of the key features of our method is its ability to merge several concepts into one, which means that we aim for high granularity and specificity in our input labels. As a result, we need to ensure that each candidate represents a unique, distinguishable concept.

Given that we cannot rely on human annotations, we use an automatic procedure to gather the initial set of candidates. This procedure combines multiple sequence tagging models to label the input corpus. This procedure aims to identify words or phrases in the text representing distinct concepts that can be used as candidate labels. We can use any sequence tagging NLP model that meets the following criteria: (1) a set of words with the same label indicates semantically coherent, distinct concepts, (2) no additional annotation is needed, and (3) the model is domain-independent.

2.3 Selection of slot candidates

In the previous step, we obtained a superset of all the slot candidates using weak supervision from the tagging models. Subsequently, we need to identify domain-relevant slots based on candidates provided by the automatic annotation. To achieve this, we design an iterative slot discovery procedure – in each iteration, we: (1) merge similar candidates, (2) rank candidates' relevance and eliminate irrelevant ones. Once no more frames are eliminated, the process stops and we obtain slot labels, which are used to train a slot tagger.

Standalone tagger We use the obtained labels to train a new, domain-specific slot tagger to improve performance. The tagger has no access to better labels than those derived by our method; however, it has a simpler task, as the set of target labels is now much smaller, and the domain is much narrower.

2.4 Slot Discovery Experiments

In this section, we provide a quantitative analysis of the results with respect to the NLU performance and quality of the discovered slots. We also evaluate the application of this method as a module in the end-to-end dialogue system model.

Evaluated systems We test multiple variants of our system. This gives us an idea about the contributions of all the individual methods we propose. Here we give an overview of all the system variants:

- *Ours-full* is the full version of our method (full annotation setup and trained slot tagger).
- *Ours-nothr* does not use the recall-increasing second-candidate rule in the slot tagger.
- *Ours-notag* excludes the slot tagger. This means that the outputs of input taggers are used directly to annotate the data.
- *Ours-nocl* further excludes the clustering step; slot candidate ranking and selection is performed over all candidates together.

We also compare to previous work of Chen et al. (2014)[1]. This method is similar to the variant *Ours-nocl* but does not merge similar frames and uses different ranking criteria. Essentially, they use the outputs of the input tagger directly after the selection step without further processing.

2.4.1 Results

We evaluate our approach to slot discovery by comparing the resulting slot labels to gold-standard supervised slot annotation.

Slot tagging is evaluated in Table 2.1. *Ours-full* (slot selection + trained tagger) outperforms all other approaches by a large margin, especially regarding recall.

Chen et al. (2014)'s method has a slightly higher precision, but our recall is much higher than theirs. A comparison between *Ours-notag* and *Ours-full* shows that applying the slot tagger improves both precision and recall. Tagger without the threshold decision rule (*Ours-nothr*) mostly performs better than the parser; however, using the threshold is essential to improve recall. Experiments on ATIS with NER as an additional annotation source proved that our method can benefit from it. As discussed above, using the trained tagging model is crucial to improve the recall of our method.

[1] We use our re-implementation of their approach.

method ↓ / dataset→	**CS**	**WH**	**WA**	**AT**
Tag-supervised*	0.724 ± .003	**0.742** ± .008	**0.731** ± .002	**0.848** ± .003
Dict-supervised*	**0.753** ± .005	**0.750** ± .018	0.665 ± .003	0.678 ± .002
weak supervision →	frames	frames	frames	frames,NER
Chen et al.	0.590 ± .001	0.382 ± .001	0.375 ± .001	0.616 ± .001
Ours-nocl	0.393 ± .011	0.122 ± .001	0.266 ± .008	0.677 ± .002
Ours-notag	0.664 ± .007	0.388 ± .002	0.383 ± .002	0.648 ± .003
Ours-nothr	0.569 ± .031	0.485 ± .032	0.435 ± .002	0.698 ± .004
Ours-full	**0.692** ± .008	**0.548** ± .004	**0.439** ± .001	**0.710** ± .002

Table 2.1: F1 score values with 95% confidence intervals for slot tagging performance comparison among different methods. The measures are evaluated using a manual slot mapping to the datasets' annotation, which is unnecessary for the methods. *Note that supervised setups are not directly comparable to our approach.

2.4.2 Error analysis

We conducted a manual error analysis of slot tagging to gain more insight into the output quality and sources of errors. We found that the tagger can generalize and capture unseen values.

One source of errors is the relatively low recall of the frame-semantic parsers. We successfully addressed this issue by introducing the slot tagger. However, many slot values remain untagged. This is expected as the input linguistic annotation quality inherently limits our method's performance. The candidate merging procedure causes another error (see also below). Due to frequent co-occurrence, two semantically unrelated candidates might be merged, and therefore, some tokens are wrongly included as respective slot fillers. Nevertheless, the merging step is required to obtain a reasonable number of slots for a dialogue domain.

Slot merging Although candidates in the CamRest676 data are merged into slots reasonably well, other datasets show a relatively low performance. The low RI scores result from errors in candidate ranking, which wrongly assigned high ranks to some rare, irrelevant candidates. These candidates do not appear in the reference mapping and are assumed to form singular "pseudo-slots". Nevertheless, this behavior barely influences slot tagging performance as the candidates are rare.

method	Slot F1	Joint Goal Accuracy	Entity Match Rate
Jin et al. supervised	0.967 ± .001	0.897 ± .002	0.869 ± .004
Jin et al. unsupervised	0.719 ± .002	0.385 ± .003	0.019 ± .002
Jin et al. weak-labels	0.709 ± .011	0.335 ± .008	0.269 ± .012
Ours-full (unsupervised)	**0.756** ± .004	**0.465** ± .007	**0.368** ± .008

Table 2.2: Evaluation on the downstream task of dialogue generation on CamRest676 data. We evaluate with respect to three state tracking metrics. The best results in an unsupervised setting are presented in bold.

2.4.3 Dialogue generation application

We explore the influence that our labels have on sequence-to-sequence dialogue response generation in an experiment on the CamRest676 data (see Table 2.2). We can see that our method provides helpful slot labels that improve dialogue state tracking performance. Our approach significantly improves all metrics compared to Jin et al. (2018)'s system used in a fully unsupervised setting. We achieve better results than Jin et al. (2018)'s system, especially regarding entity match rate, suggesting that our model can provide consistent labels throughout the dialogue. To make a fair comparison, we further evaluate Jin et al. (2018)'s system in a setting where it can learn from the labels provided directly by weak supervision (i.e., the frame-semantic parser, not filtered by our pipeline). We observe an improvement in entity match rate, but it does not match the improvement achieved with our filtered labels. Surprisingly, slot F1 and joint goal accuracy even decreased slightly, which suggests that label quality is important and the noisy labels obtained directly from weak supervision are not useful enough.

3
Dialogue modeling with less supervision

Dialogue modeling is a complicated task requiring the ability to communicate in a natural language and handle discrete decision processes representing the conversation logic. Various architectures have been proposed over the years, mostly relying on explicit data annotation on multiple levels to guide the model training process. One of the biggest challenges is to model task-oriented dialogue that requires interaction with external interfaces such as databases or API services. This aspect puts a hard constraint on the dialogue system architecture – it requires some explicit representation that allows one to communicate with external systems. Achieving this in a fully unsupervised setting is challenging due to a lack of model guidance in the form of structured labels.

To enable the model to interact with external sources of information via API, we annotated points in the training dialogues where interaction with an external API is needed.

In this chapter, we present a modeling approach that we previously published in Hudeček and Dušek (2022) and propose alternative unpublished extensions of a different base architecture using latent representations (Lubis et al., 2022).

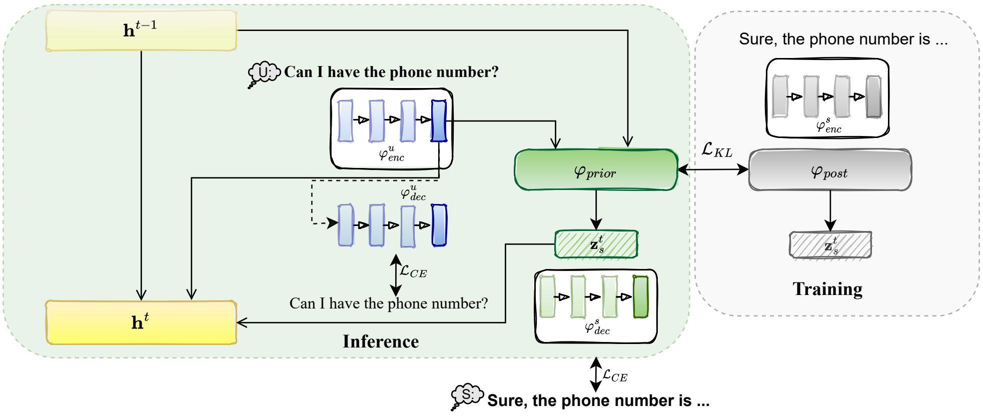

Figure 3.1: Visualization of our model architecture (one dialogue turn), described in Section 3.1. Yellow boxes represent the turn-level VRNN's hidden state h^t. The user utterance is represented as the last hidden state of the encoder network φ^u_{enc}, which is trained as an autoencoder along with the decoder φ^u_{dec}. The system utterance, encoded by the network φ^s_{enc}, is an input to the posterior network φ_{post} that helps to train the prior network φ_{prior} to construct meaningful latent variables $\mathbf{z}_s$, which initialize the system utterance decoder φ^s_{dec}. The training uses the whole architecture, including the posterior network φ_{post}, while only the part shaded in green is used for inference. $\mathcal{L}_{CE}$ stands for cross-entropy loss, $\mathcal{L}_{KL}$ for KL-divergence loss.

3.1 Task-Oriented dialogue with TO-VRNN

The VRNN model architecture (Chung et al., 2015) is designed to model sequences of observations coupled with latent states. A generative model can learn the conditional generative distribution of observations given the state. Moreover, although VRNN does not require a fixed set of states, it can be adjusted to model discrete states. The VRNN is great for modeling the discrete decision processes behind conversation exchanges, thanks to the above-mentioned properties. However, we propose some extensions for task-oriented dialogue modeling to distinguish between the user and system roles and incorporate the possibility of handling interaction with external interfaces.

TO-VRNN model description Figure 3.1 depicts our model's architecture. Following the original VRNN architecture, we employ a turn-level RNN that summarizes the context in its hidden state. In each dialogue turn, we model user and system utterances with separate autoencoders to account for different user and system behaviors. First, the user utterance, modeled with a standard vanilla autoencoder, is processed, and the last encoder hidden state $\varphi^u_{enc}(\mathbf{x}^t_u)$ provides the encoded representation used as an input for the next stage.

Next, the system part is used, which is realized by VAE with discrete latent variables $\mathbf{z}_s$ conditioned on the context RNN's hidden state $\mathbf{h}^{t-1}$ and the user utterance encoding $\varphi^u_{enc}(\mathbf{x}^t_u)$. Our model can thus be seen as a VRNN extended by an additional encoder-decoder module for input pre-processing.

To finalize the turn-processing step, we need to save the information into the turn-level network so it becomes part of the encoded context. The turn-level network state update looks as follows:

$$\mathbf{h}^{t+1} = \mathrm{RNN}([\varphi^u_{enc}(\mathbf{x}^t_u), \varphi_z(\mathbf{z}^t_s)], \mathbf{h}^t) \tag{3.1}$$

We train the model by minimizing a sum of the cross-entropy reconstruction loss on user utterances and the variational lower bound on system responses.

3.2 TO-VRNN Experiments

In this section, we evaluate the quality of responses generated by our model. We also inspect the model performance concerning dialogue success. We compare our models to baseline architectures s.a. LSTM (Hochreiter and Schmidhuber, 1997), VHRED (Serban et al., 2017), Transformer (Liu and Lane, 2016) and GPT-2 (Radford et al., 2019)

Database queries To include database information in the dialogues, we first identify all turns in the original datasets where database information is required, using handcrafted rules. We then create special database query turns based on the respective state annotation.

3.2.1 Results

Response quality Our architecture performs substantially better than (V)HRED (Serban et al., 2017), which commonly fails to acquire the necessary knowledge, especially on larger datasets. The attention-based versions perform better on BLEU but lose slightly on perplexity.

Our models can generate relevant responses based on manual checks in most cases. As expected, only the models, including database turns, can predict correct entities. A relatively common error is informing about wrong slots, e.g., the model provides a phone number instead of an address or, even more frequently, provides wrong slot values.

model		CamRest676				MultiWOZ 2.1			
	db	BLEU	Ppl	MI	EMR	BLEU	Ppl	MI	EMR
LSTM	✗	3.90	5.34	–	–	0.92	8.23	–	–
Transformer	✗	4.98	7.72	–	–	0.95	6.95	–	–
GPT-2	✗	15.40	1.18	–	–	9.40	2.77	–	–
GPT-2	✓	13.89	1.80	–	–	9.56	2.43	–	–
HRED	✗	2.70	13.92	–	0.02	2.98	29.61	–	0.01
VHRED	✗	4.34	11.76	0.21	0.02	4.65	32.74	0.15	0.01
VHRED	✓	8.50	10.23	0.17	0.36	3.82	16.61	0.07	0.04
TO-VRNN-noattn	✗	12.98	4.64	0.29	0.01	7.18	9.16	**0.42**	0.02
TO-VRNN-noattn	✓	15.10	4.45	**0.34**	0.24	11.3	5.17	0.27	0.05
TO-VRNN-attn	✗	**17.37**	5.07	0.16	0.09	**12.28**	10.19	0.06	0.04
TO-VRNN-attn	✓	17.10	**4.23**	0.22	**0.81**	11.86	**6.03**	0.05	**0.08**
supervised *	✓	25.50	–	–	–	19.40	2.50	–	–

Table 3.1: Model performance in terms of BLEU for generated responses, Perplexity (Ppl), and Mutual Information (MI) between the generated response and the latent variables $\mathbf{z}_s$. We do not evaluate the database-enriched models on SMD as SMD's database structure does not map easily to our annotation style. We measure MI only for the models that use latent variables explicitly. The *db* column indicates systems that use database information. *Note that the supervised state-of-the-art scores are not directly comparable, as the systems use full turn-level supervision.(Qin et al., 2020);

Dialogue success The conventional definition of dialogue success or *success rate* reflects the ratio of dialogues in which the system captures all the mentioned slots correctly and provides all the requested information. We approximate tracking slot values turn-by-turn by checking for correct slot values upon database queries only, and we use this information to measure dialogue success.

Our system is not competitive with a fully supervised model but outperforms the baselines (VHRED, GPT). Upon inspection, we see that the system can often recognize correct slots. However, it has difficulties capturing the right values. However, the scores are promising, considering the minimal supervision of our training.

3.2.2 Latent Variable Interpretation

Explaining and interpreting the model behavior is crucial, especially in a setting without full supervision.

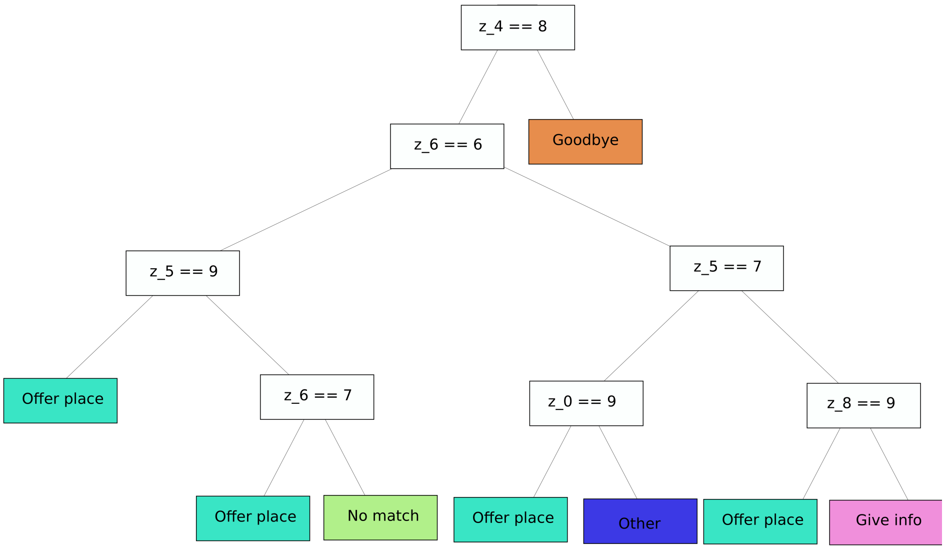

Figure 3.2: A visualization of a decision tree trained on the CamRest676 data to predict a system action from the contents of the latent variables. Each node represents a decision based on one latent variable value, and the leaf node colors represent different system actions. When the condition in a given node is fulfilled, the algorithm proceeds into the right subtree, left otherwise. For clarity, we limit the maximum tree depth to 4. The limit slightly lowers the accuracy – the pictured tree achieves an accuracy of 73% on the CamRest676 data.

Predictive power of the variables

To evaluate the predictive power of the obtained latent representations, we train a simple classifier that predicts the system action and current domain, using solely the obtained latent representations as input features. To put our results into perspective, we include several baselines: trivial random and majority class baselines and classifiers using representations obtained with other methods (HRED, VHRED, GPT). We use a decision tree (DT) classifier trained with the CART algorithm[1] and the *gini* split criterion due to its good interpretability. The results are shown in Table 3.2. Our classifier beats the random and majority baselines in all cases. It also outperforms classification based on (V)HRED and GPT representations. This demonstrates that our approach produces high-quality interpretable representations. We also observe that using attention harms the performance of the action classifier as it makes it possible for the models to bypass the latent variables. On CamRest676, the latent variables explain most of the annotated actions. Overall, we can observe that any hidden state taken from some trained model can explain some portion of the data. However, using our approach seems to perform better in this aspect. We also notice the influ-

[1] `https://scikit-learn.org/stable/modules/tree.html`

config	**CamRest676**	**MultiWOZ 2.1.**	
	gold	domain	action
random	0.17	0.14	0.09
majority	0.42	0.33	0.32
HRED	0.65	0.45	0.44
VHRED	0.52	0.36	0.32
GPT-2	0.65	0.60	0.55
TO-VRNN-attn	0.63	0.68	0.66
TO-VRNN-noattn	**0.75**	**0.70**	**0.69**
TO-VRNN-manual	0.59	–	–

Table 3.2: Accuracy of the domain and action decision-tree classifiers based on latent variables.

ence of the number of latent variables on the performance. In general, increasing the number of latent variables leads to a substantial performance improvement, which suggests that all the variables contribute with relevant information (see Table 3.2).

The information about domains and system actions is stored in categorical variables. It can be extracted by a simple classification model such as the decision tree, which allows us to interpret and explain the behavior of our model. For illustration, in Figure 3.2, we plot a DT with limited depth that achieves 73% accuracy when predicting the system action on the CamRest676 data. The aim is that latent variables hold high-level information, such as intents, actions, or domains. This helps interpretability but is insufficient for generating appropriate and factually correct responses – here, we need to incorporate correct slot values. This detailed information is captured and carried over via the attention mechanism in *TO-VRNN-attn.* Potential alternatives are copy mechanisms (Lei et al., 2018) or delexicalization on the generated outputs (Henderson et al., 2014; Peng et al., 2021b).

4 Sequence-to-Sequence Task-Oriented Dialogue Modeling

Using end-to-end trainable models instead of modular architectures can potentially offer more flexibility with respect to domain transfer, as only a single module needs to be adapted to new domains or use cases. In task-oriented dialogue, end-to-end implementations are dominated by sequence-to-sequence architectures based on language models. The LM-based approaches have taken over the benchmarks[1], demonstrating state-of-the-art performance.

However, these competitive models are fine-tuned on a large in-domain dataset, and domain transfer performance is not evaluated. In this chapter, we raise the question of how well these models can transfer the obtained skill of leading the dialogue to other domains. In other words, we want to discover if the models learn useful skills that can be beneficial in other domains or if the demonstrated behavior merely reproduces the patterns seen in the training portion of the data. We hypothesize that pre-training of these models can help to improve the performance. To confirm this hypothesis, we first describe our approach to end-to-end modeling with the GPT-2-based model (Kulhánek et al., 2021) in Section 4.1. We then describe our newly assembled and unified multi-domain dataset, designed for domain transfer experiments in Section 4.2 and detail our experimental results with AuGPT on this data in Section 4.3.

[1] `https://github.com/budzianowski/multiwoz#trophy-benchmarks`

The work presented in this chapter was published at the LREC conference and covered by Hudeček et al. (2022) [2]. For modeling, we use the AuGPT model (Kulhánek et al., 2021) to the development of which we contributed[3].

4.1 AuGPT Model

We choose the AuGPT model introduced by Kulhánek et al. (2021). The architecture utilizes the GPT-2 model (Radford et al., 2019) for both belief state prediction and response generation. Additionally, AuGPT introduces multiple training improvements. Instead of solely using cross-entropy loss for language modeling, AuGPT uses an additional training objective for state corruption detection. This modification aims to improve the robustness of the belief state prediction. Since GPT can work with any natural language sentences, applying this model to our dialogue datasets is straightforward.

4.2 Diaser corpus introduction

Motivated by the questions raised in this chapter, we created a collection of several well-established task-oriented dialogue datasets spanning several domains to yield one larger multi-domain corpus which we call *Diaser*. Specifically, we used: **MultiWOZ 2.2** (MultiWOZ), **Schema-guided dialogue** (SGD), **DSTC2** (DSTC) and **CamRest676** (CamRest). The merging process yields a dataset with over 37,000 dialogues, comprising more than 660,000 turns.

4.3 Experiments

In this chapter, we want to explore if sequence-to-sequence LM-based architectures can robustly learn the dialogue modeling capabilities and how well they can transfer them to other domains. To answer these questions, we conduct a series of experiments to train the model using only a small portion of the training data or even a different dataset with shared characteristics such as a task-oriented approach, regular user-system interaction, etc.

[2] This was a joint effort in which the author of this thesis focused on the data processing pipelines and AuGPT model training.

[3] The author of this thesis contributed to the model design and data preparation for the training

training			evaluation			metrics		
DSTC	**MW**	**SGD**	**DSTC**	**MW**	**SGD**	**Slot F1**	**JGA**	**BLEU**
✗	✓	✗	✗	✓	✗	0.89	0.53	18.61
✗	✗	✓	✗	✓	✗	0.16	0.02	4.01
✓	✓	✗	✗	✓	✗	0.89	0.55	19.67
✓	✗	✓	✗	✓	✗	0.17	0.03	5.68
✗	✓	✓	✗	✓	✗	0.89	0.52	19.92
✓	✓	✓	✗	✓	✗	0.90	0.54	21.09
✗	✓	✗	✗	✗	✓	0.04	0.01	5.63
✗	✗	✓	✗	✗	✓	0.59	0.21	28.17
✓	✓	✗	✗	✗	✓	0.03	0.01	5.51
✓	✗	✓	✗	✗	✓	0.58	0.21	27.96
✗	✓	✓	✗	✗	✓	0.63	0.23	27.54
✓	✓	✓	✗	✗	✓	0.63	0.22	27.72
✓	✓	✗	✓	✓	✓	0.28	0.12	15.30
✓	✗	✓	✓	✓	✓	0.55	0.22	27.28
✗	✓	✓	✓	✓	✓	0.65	0.25	25.13
✓	✓	✓	✓	✓	✓	0.70	0.28	**29.73**

Table 4.1: Performance of the AuGPT model trained and evaluated on various subsets of the unified dataset, namely DSTC, MultiWOZ (MW), and SGD. We omit CamRest since the data are very similar to DSTC.

Influence of training data on the target performance In Table 4.1, we can see a general pattern in the results, suggesting that the model fails to generalize across different datasets when a subset of data we evaluate is not included in the training.

Regarding the F_1 slot score and joint goal accuracy, the explanation for the poor model's accuracy can be found in a substantially different distribution of slots across SGD/MultiWOZ datasets and DSTC. The large difference in performance can also be seen in terms of BLEU, which suggests a vastly different language used in each dataset.

We get the best results when using all three datasets for training and obtain the model with better generalizing capabilities supported by the BLEU score of 21.09.

Finally, if we evaluate all three datasets, it is clear that the best-performing model is obtained when we train it on the full data.

5 Large Language Models for Task-Oriented Dialogue

As described in the previous chapter, pre-trained language models perform very well in end-to-end dialogue modeling. Despite this success, the widely used approach of fine-tuning pre-trained LM on a particular dataset still does not guarantee easy transferability of the learned knowledge, as we also show in Chapter 4. Large Language Models (LLMs) increased the model size by order of magnitude compared to the previous generation of pre-trained LMs.

We introduce an LLM-based TOD conversation pipeline which resembles other approaches based on LMs (Peng et al., 2021a; Yang et al., 2021) in Section 5.1. Instead of fine-tuning LMs, our method intentionally relies almost exclusively on using pre-trained LLMs as-is so we can test their out-of-the-box capabilities. The dialogue context and domain description are introduced to the model only by including them in the input prompt.

This work was published at the SIGDial 2023 conference (Hudeček and Dušek, 2023) and was extended in this chapter with more details and additional experiments.

5.1 Method

An overall description of the proposed pipeline is shown in Figure 5.1. The system consists of a pre-trained LLM and an (optional) context store in a vector database. Three LLM calls are performed in each dialogue turn, with specific prompts (see Section 5.1). First, the LLM performs domain detection and state

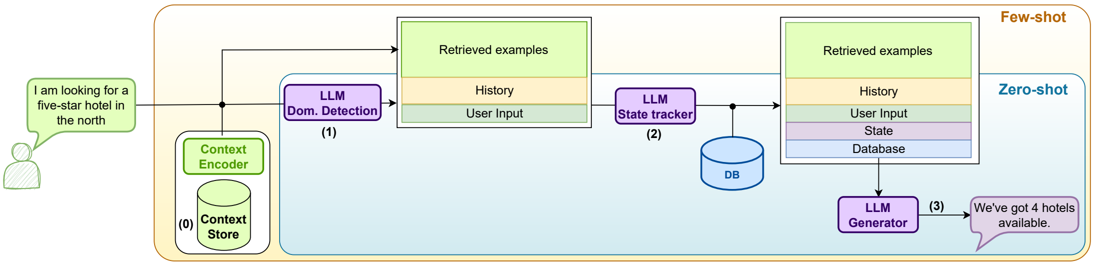

Figure 5.1: A detailed description of our proposed pipeline. (0) As a preprocessing step, we encode a subset of the training set that will be used to retrieve few-shot examples. Given a user input, we: (1) Detect the domain, retrieve relevant examples (in the few-shot setting), and construct an initial prompt. (2) Infer the belief state using LLM. Based on that, we retrieve database information and construct another prompt that includes the state and database results. (3) We ask the LLM to provide a final response.

tracking (Section 5.1). The updated belief state informs a database query, the results of which are used in the subsequent LLM-based response generation step. In the few-shot setting, the context store stores a limited number of examples from the training set, which are retrieved based on similarity with the conversation context and included in LLM prompts (see Section 5.1).

Prompt construction We aim to compare the raw capabilities of the selected LLMs. Therefore, we do not focus on prompt engineering techniques and choose universal prompts for all LLMs in this work. We choose simple, plain language statements as prompts, with no specific vocabulary, based only on a few preliminary tests. We define a single **domain detection prompt** for all examples, plus a pair of prompts for each domain in the given dataset: a **state tracking prompt** and a **response prompt**.

In addition to general instructions, each state tracking prompt contains a domain description, a list of relevant slots, the dialogue history, and the current user utterance.

Domain Detection and State Tracking We prompt the LM twice at each turn during state tracking, first to detect the active domain and then to output a belief state update. We then use the outputs to update the accumulated global belief state.

Our preliminary experiments showed that LLMs consistently struggle to output all active slot values at every turn. Therefore, we model only state updates, following the MinTL approach (Lin et al., 2020). The global belief state is then accumulated using these turn-level updates.

We generate, delexicalized outputs which allow us to evaluate the success rate and compare it to previous works. The prompt specifies that the model should provide entity values as delexicalized placeholders, and any few-shot examples are constructed accordingly.

Context Storage It has been shown that enriching prompts with specific examples (i.e. *few-shot prompting*) boosts LM performance (Madotto et al., 2020; Brown et al., 2020). To apply this knowledge efficiently in our pipeline, we introduce a storage that contains encoded dialogue contexts. Once the relevant examples are retrieved, we include them in the prompt to guide the model better. When constructing the context store, we employ only a few training examples, ensuring we evaluate in a truly few-shot setting.

5.2 LLM Experiments

To obtain a broad overview of the current LLMs' capabilities, we compare several models spanning different numbers of trainable parameters and different training methods. We also experiment with four variants of the base setup, using either zero-shot or few-shot operations and either predicted or oracle belief states. We use the following models for the evaluation: **Tk-Instruct-11B**(Wang et al., 2022), **ChatGPT**[1], **Alpaca-LoRA-7B** (Touvron et al., 2023; Hu et al., 2021), **GPT-NeoXT-Chat-Base-20B** (Black et al., 2022) and **OPT-IML-30B** (Iyer et al., 2022).

Evaluated variants We test four setup variants for each pair of model and dataset. Specifically, we use zero-shot (without examples) or few-shot (including examples) prompts (*-zs-* vs. *-fs-*) and either generated or oracle belief states (*-gbs* vs. *-obs*).

Human Evaluation

For human evaluation, we perform a small-scale in-house interaction study on MultiWOZ. The annotators were given goal descriptions sampled from the MultiWOZ test set and concise instructions on how to proceed. Since the MultiWOZ goal often involves tasks in multiple domains, we ask annotators to evaluate each domain in the dialogue distinctly. At the end of each dialogue, the annotators are asked to answer these questions:

[1] https://openai.com/blog/chatgpt

model	few	oracle		**MultiWOZ 2.2**		
	shot	BS	BLEU	JGA	Slot-F1	Success
Supervised SotA	✗	✗	19.90♣	0.60♢	–	0.82♡
Alpaca-LoRA-7B-*fs-gbs*	✓	✗	5.53	0.06	0.08	0.06
Tk-Instruct-11B-*fs-gbs*	✓	✗	6.56	0.16	0.33	0.19
GPT-NeoXT-20B-*fs-gbs*	✓	✗	2.73	0.05	0.04	0.05
OPT-IML-30B-*fs-gbs*	✓	✗	4.40	0.03	0.03	0.04
ChatGPT-*fs-gbs*	✓	✗	6.77	**0.27**	**0.51**	0.44
Alpaca-LoRA-7B-*fs-obs*	✓	✓	5.96	–	–	0.41
Tk-Instruct-11B-*fs-obs*	✓	✓	**6.91**	–	–	0.46
GPT-NeoXT-20B-*fs-obs*	✓	✓	2.92	–	–	0.28
OPT-IML-30B-*fs-obs*	✓	✓	5.40	–	–	0.28
ChatGPT-*fs-obs*	✓	✓	6.84	–	–	**0.68**

Table 5.1: Evaluation of the chosen LLMs concerning widely used TOD measures on the MultiWOZ dataset. We list only the few-shot variants that use 10 examples per domain in the context storage, two selected for the prompts. We also provide supervised state-of-the-art results to put the numbers in context: ♣Sun et al. (2022), ♢Huang et al. (2023), ♡Feng et al. (2023).

1. *How many of the sub dialogues/domains were handled successfully?* (corresponding to dialogue success)

2. *How many clarifications or corrections were needed?*

3. *Was all the provided information captured correctly?* (corresponding to JGA)

Results

Belief State Tracking The belief state tracking results overview is given in Table 5.1 (*JGA* and *Slot-F1*). We hypothesize that the performance could be further improved by careful model-specific prompt customization and perhaps task re-formulation; nevertheless, this is not the goal of this work.

Dialogue-level performance Results for dialogue success are provided in Table 5.1, and there is again a large gap between prompted LLMs and supervised custom models' performance. The contribution of retrieved examples is more obvious when we supply the oracle belief state, which helps consistently for all the models.

	ChatGPT	**Tk-Instruct**
dialogues	25	25
clarify / dial	1.08	1.68
succesful subdialogues	81%	71%
succesful dialogues	76%	64%

Table 5.2: Human evaluation results for ChatGPT and Tk-Instruct-11B models. We evaluate the conversation on the sub-dialogue level i.e. each domain in the dialogue is evaluated separately.

We also explore the influence of the context storage size on the dialogue success rate. The results are given in Figure 5.2. The biggest improvement can be achieved by supplying just a few examples instead of zero-shot prompting, but increasing the size of the example pool for retrieval does not yield further performance gains.

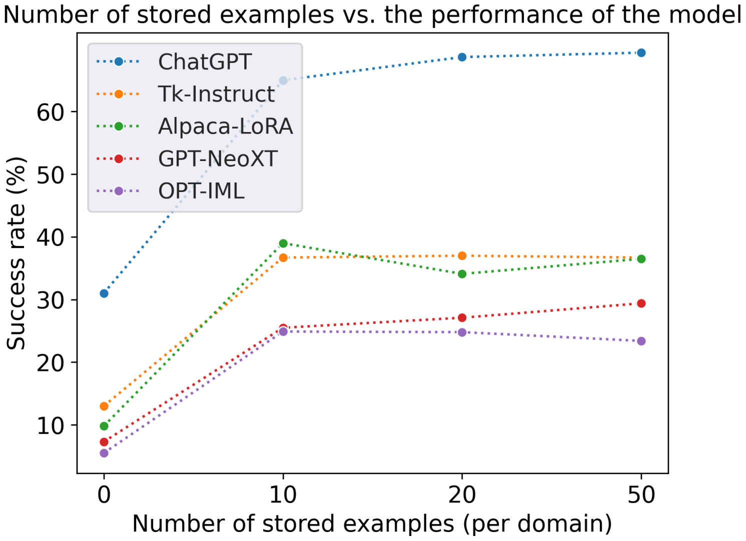

Figure 5.2: The influence of the number of examples per domain available for few-shot retrieval and response generation performance of the model in terms of the dialogue success on MultiWOZ 2.2 data with the oracle state supplied. Note that this does not represent the number of examples selected for the prompt, which is fixed to two.

Human Evaluation We can see that in a real interaction with a human user and allowing for clarification or correction, the models perform better than the strict automatic evaluation. Furthermore, the models are often successful in multiple sub-dialogues, even if a part of the dialogue fails.

6 Conclusion

In this thesis, we discussed the problems of data annotation needed for task-oriented dialogue modeling. We proposed multiple approaches to address how current systems rely on extensive data annotation.

Decreasing the amount of required supervision Automatic data labeling procedures are desirable since they can save expensive resources by providing valuable insights into the gathered conversation data. In Chapter 2, we proposed a pipeline method for the unsupervised discovery of dialogue slot schema and automatic labeling. The method iteratively refines a set of input candidates to obtain semantically coherent concepts. We showed that the method can successfully exploit the inputs and refine the initial candidate pool to outperform other approaches.

Training dialogues from unlabeled data The ultimate goal is to train task-oriented dialogue models solely from an unlabeled input corpus. As an intermediate step toward this goal, we introduced a novel usage of latent variable models for TOD generation in Chapter 3. Our latent action models based on variational training show promising performance when outperforming other baseline approaches, even with many more parameters when the same amount of training data is presented. Moreover, our model creates representations in a discrete latent space that can be used to predict the system's actions successfully.

Less data for end-to-end TOD models Another direction is using full supervision but minimizing the required training data. We explored this path in Chapters 4 and 5. The power of pre-trained large language models is great, and they can achieve good performance with only a fraction of the training set available to them. We observed the importance of examples in the in-context learning approach. Using LLMs promises to bridge the gap between academic datasets and real-world use cases as LLMs can handle various conversation behaviors very well.

Future research directions

Thanks to the tremendous recent progress in large language models pre-training and applications, unsupervised dialogue modeling will likely improve greatly in the near future. LLMs can be used to obtain better representations and, together with contrastive learning objectives, achieve previously unseen performance in dialogue structure discovery and schema induction. Therefore, our method proposed in Chapter 2 can be improved with stronger and more capable initial candidate identification and representation models.

The modeling capabilities of the Transformer-based models will likely achieve significant improvements in human-machine interaction with little to no need to provide in-domain data examples. Their performance can contribute to creating more powerful latent variable models, which we discussed in Chapter 3.

As for using the pre-trained LLMs, which we discussed in Chapter 5, we can expect a lot of applications based on in-context learning augmented with retrieval mechanisms to guide the model with certain dialogue flows. For task-oriented dialogue, in particular, some challenges need to be addressed, such as the tendency of LLMs to produce answers not grounded in the context introduced in prompt, so-called hallucinations.

Bibliography

Black, S. – Biderman, S. – Hallahan, E. – Anthony, Q. – Gao, L. – Golding, L. – He, H. – Leahy, C. – McDonell, K. – Phang, J. – others. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*. 2022.

Brown, T. – Mann, B. – Ryder, N. – Subbiah, M. – Kaplan, J. D. – Dhariwal, P. – Neelakantan, A. – Shyam, P. – Sastry, G. – Askell, A. – others. Language models are few-shot learners. *Advances in neural information processing systems*. 2020, 33, p. 1877–1901.

Budzianowski, P. – Wen, T.-H. – Tseng, B.-H. – Casanueva, I. – Ultes, S. – Ramadan, O. – Gašić, M. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 5016–5026, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1547. Available at: `https://aclanthology.org/D18-1547`.

Chen, Y.-N. – Wang, W. Y. – Rudnicky, A. I. Leveraging frame semantics and distributional semantics for unsupervised semantic slot induction in spoken dialogue systems. In *Proceedings of IEEE SLT*, p. 584–589, 2014. doi: 10.1109/SLT.2014.7078639.

Chung, J. – Kastner, K. – Dinh, L. – Goel, K. – Courville, A. C. – Bengio, Y. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, p. 2980–2988, 2015.

Eric, M. – Krishnan, L. – Charette, F. – Manning, C. D. Key-Value Retrieval Networks for Task-Oriented Dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, p. 37–49, Saarbrücken, Germany, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5506. Available at: `https://aclanthology.org/W17-5506`.

Feng, Y. – Yang, S. – Zhang, S. – Zhang, J. – Xiong, C. – Zhou, M. – Wang, H. Fantastic Rewards and How to Tame Them: A Case Study on Reward Learning for Task-oriented Dialogue Systems. *arXiv preprint arXiv:2302.10342*. 2023.

HEMPHILL, C. T. – GODFREY, J. J. – DODDINGTON, G. R. The ATIS spoken language systems pilot corpus. In *Proceedings of the workshop on Speech and Natural Language - HLT '90*, p. 96–101, Hidden Valley, Pennsylvania, 1990. doi: 10.3115/116580.116613. Available at: https://www.aclweb.org/anthology/H90-1021/.

HENDERSON, M. – GAŠIĆ, M. – THOMSON, B. – TSIAKOULIS, P. – YU, K. – YOUNG, S. Discriminative spoken language understanding using word confusion networks. In *Proceedings of IEEE SLT*, p. 176–181, 2012. doi: 10.1109/SLT.2012.6424218.

HENDERSON, M. – THOMSON, B. – YOUNG, S. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, p. 360–365, December 2014. doi: 10.1109/SLT.2014.7078601.

HOCHREITER, S. – SCHMIDHUBER, J. Long Short-Term Memory. *Neural Comput.* November 1997, 9, 8, p. 1735–1780. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. Available at: https://doi.org/10.1162/neco.1997.9.8.1735.

HU, E. J. – SHEN, Y. – WALLIS, P. – ALLEN-ZHU, Z. – LI, Y. – WANG, S. – WANG, L. – CHEN, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*. 2021.

HUANG, T. – HALBE, S. A. – SANKAR, C. – AMINI, P. – KOTTUR, S. – GERAMIFARD, A. – RAZAVIYAYN, M. – BEIRAMI, A. Robustness through Data Augmentation Loss Consistency. *Transactions on Machine Learning Research*. 2023. Available at: https://openreview.net/forum?id=a1meaRy1bN.

HUDEČEK, V. – DUŠEK, O. Learning Interpretable Latent Dialogue Actions With Less Supervision. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, p. 297–308, Online only, November 2022. Association for Computational Linguistics. Available at: https://aclanthology.org/2022.aacl-main.24.

HUDEČEK, V. – DUŠEK, O. Are Large Language Models All You Need for Task-Oriented Dialogue? In *Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue*, p. 216–228, Prague, Czechia, September 2023. Association for Computational Linguistics. Available at: https://aclanthology.org/2023.sigdial-1.21.

Hudeček, V. – Dušek, O. – Yu, Z. Discovering Dialogue Slots with Weak Supervision. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, p. 2430–2442, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.189. Available at: https://aclanthology.org/2021.acl-long.189.

Hudeček, V. – Schaub, L.-p. – Stancl, D. – Paroubek, P. – Dušek, O. A Unifying View On Task-oriented Dialogue Annotation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, p. 1286–1296, Marseille, France, June 2022. European Language Resources Association. Available at: https://aclanthology.org/2022.lrec-1.137.

Iyer, S. – Lin, X. V. – Pasunuru, R. – Mihaylov, T. – Simig, D. – Yu, P. – Shuster, K. – Wang, T. – Liu, Q. – Koura, P. S. – others. OPT-IML: Scaling Language Model Instruction Meta Learning through the Lens of Generalization. *arXiv preprint arXiv:2212.12017*. 2022.

Jin, X. – Lei, W. – Ren, Z. – Chen, H. – Liang, S. – Zhao, Y. – Yin, D. Explicit State Tracking with Semi-Supervision for Neural Dialogue Generation. In *Proceedings of ACM CIKM*, p. 1403–1412, 2018. doi: 10.1145/3269206.3271683.

Jurafsky, D. *Speech & language processing*. Pearson Education India, 2000.

Kulhánek, J. – Hudeček, V. – Nekvinda, T. – Dušek, O. AuGPT: Auxiliary Tasks and Data Augmentation for End-To-End Dialogue with Pre-Trained Language Models. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, p. 198–210, Online, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.nlp4convai-1.19. Available at: https://aclanthology.org/2021.nlp4convai-1.19.

Lei, W. – Jin, X. – Kan, M.-Y. – Ren, Z. – He, X. – Yin, D. Sequicity: Simplifying Task-oriented Dialogue Systems with Single Sequence-to-Sequence Architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 1437–1447, Melbourne, Australia, 2018. doi: 10.18653/v1/P18-1133. Available at: https://www.aclweb.org/anthology/P18-1133.

Lin, Z. – Madotto, A. – Winata, G. I. – Fung, P. MinTL: Minimalist Transfer Learning for Task-Oriented Dialogue Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 3391–3405, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.273. Available at: https://aclanthology.org/2020.emnlp-main.273.

Liu, B. – Lane, I. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*. 2016.

Lubis, N. – Geishauser, C. – Lin, H.-c. – Niekerk, C. – Heck, M. – Feng, S. – Gasic, M. Dialogue Evaluation with Offline Reinforcement Learning. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, p. 478–489, Edinburgh, UK, September 2022. Association for Computational Linguistics. Available at: https://aclanthology.org/2022.sigdial-1.46.

Madotto, A. – Liu, Z. – Lin, Z. – Fung, P. Language models as few-shot learner for task-oriented dialogue systems. *arXiv preprint arXiv:2008.06239*. 2020.

Mukherjee, S. – Hudeček, V. – Dušek, O. Polite Chatbot: A Text Style Transfer Application. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, p. 87–93, 2023.

Peng, B. – Li, C. – Li, J. – Shayandeh, S. – Liden, L. – Gao, J. Soloist: Building Task Bots at Scale with Transfer Learning and Machine Teaching. *Transactions of the Association for Computational Linguistics*. 2021a, 9, p. 807–824. doi: 10.1162/tacl_a_00399. Available at: https://aclanthology.org/2021.tacl-1.49.

Peng, B. – Li, C. – Li, J. – Shayandeh, S. – Liden, L. – Gao, J. SOLOIST: Building Task Bots at Scale with Transfer Learning and Machine Teaching. *Trans. Assoc. Comput. Linguistics*. 2021b, 9, p. 907–824. doi: 10.1162/tacl_a_00399. Available at: https://doi.org/10.1162/tacl_a_00399.

Plátek, O. – Bělohlávek, P. – Hudeček, V. – Jurčíček, F. Recurrent neural networks for dialogue state tracking. *arXiv preprint arXiv:1606.08733*. 2016.

Plátek, O. – Hudeček, V. – Schmidtová, P. – Lango, M. – Dušek, O. Three Ways of Using Large Language Models to Evaluate Chat. *arXiv preprint arXiv:2308.06502*. 2023.

Qin, L. – Xu, X. – Che, W. – Zhang, Y. – Liu, T. Dynamic Fusion Network for Multi-Domain End-to-end Task-Oriented Dialog. In Jurafsky, D. – Chai, J. – Schluter, N. – Tetreault, J. R. (Ed.) *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, p. 6344–6354, Online, 2020. doi: 10.18653/v1/2020.acl-main.565. Available at: https://doi.org/10.18653/v1/2020.acl-main.565.

Radford, A. – Wu, J. – Child, R. – Luan, D. – Amodei, D. – Sutskever, I. – others. Language models are unsupervised multitask learners. *OpenAI blog*. 2019, 1, 8, p. 9.

Schaub, L.-P. – Hudecek, V. – Stancl, D. – Dusek, O. – Paroubek, P. Définition et détection des incohérences du système dans les dialogues orientés tâche. (We present experiments on automatically detecting inconsistent behavior of task-oriented dialogue systems from the context). In *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, p. 142–152, Lille, France, 6 2021. ATALA. Available at: https://aclanthology.org/2021.jeptalnrecital-taln.13.

Serban, I. – Sordoni, A. – Lowe, R. – Charlin, L. – Pineau, J. – Courville, A. – Bengio, Y. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the AAAI conference on artificial intelligence*, 31, 2017.

Straka, M. – Mediankin, N. – Kocmi, T. – Žabokrtský, Z. – Hudeček, V. – Hajič, J. SumeCzech: Large Czech News-Based Summarization Dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). Available at: https://aclanthology.org/L18-1551.

Sun, H. – Bao, J. – Wu, Y. – He, X. Mars: Semantic-aware Contrastive Learning for End-to-End Task-Oriented Dialog. *arXiv preprint arXiv:2210.08917*. 2022.

Touvron, H. – Lavril, T. – Izacard, G. – Martinet, X. – Lachaux, M.-A. – Lacroix, T. – Rozière, B. – Goyal, N. – Hambro, E. – Azhar, F. – Rodriguez, A. – Joulin, A. – Grave, E. – Lample, G. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*. 2023.

Wang, Y. – Mishra, S. – Alipoormolabashi, P. – Kordi, Y. – Mirzaei, A. – Arunkumar, A. – Ashok, A. – Dhanasekaran, A. S. – Naik, A. – Stap, D. – others. Super-NaturalInstructions:Generalization via Declarative Instructions on 1600+ Tasks. In *EMNLP*, 2022.

Wen, T.-H. – Vandyke, D. – Mrksić, N. – Gašić, M. – Rojas-Barahona, L. M. – Su, P.-H. – Ultes, S. – Young, S. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of EACL*, p. 438–449, 2017. Available at: https://www.aclweb.org/anthology/E17-1042.

Yang, Y. – Li, Y. – Quan, X. Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. In *Proceedings of the AAAI Conference on Artificial Intelligence*, p. 14230–14238, 2021.

List of Publications

We first present list of publications in which the author of this thesis is the main author, followed with more publications to which the author contributed. The number of citations was computed using Google Scholar.

Total number of citations: 148

HUDEČEK, V. – DUŠEK, O. Are Large Language Models All You Need for Task-Oriented Dialogue? In *Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue*, p. 216–228, Prague, Czechia, September 2023. Association for Computational Linguistics. Available at: `https://aclanthology.org/2023.sigdial-1.21`

- In this work we use large language models and in-context learning to model task-oriented dialogue.
- Citations: 23

HUDEČEK, V. – DUŠEK, O. – YU, Z. Discovering Dialogue Slots with Weak Supervision. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, p. 2430–2442, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.189. Available at: `https://aclanthology.org/2021.acl-long.189`

- This work presents pipeline method for unsupervised slot discovery.
- Citations: 22

HUDEČEK, V. – DUŠEK, O. Learning Interpretable Latent Dialogue Actions With Less Supervision. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, p. 297–308, Online only, November 2022. Association for Computational Linguistics. Available at: `https://aclanthology.org/2022.aacl-main.24`

- In this publication we introduce our dialogue model with variational training and discrete latent space.
- Citations: 1

Hudeček, V. – Schaub, L.-p. – Stancl, D. – Paroubek, P. – Dušek, O. A Unifying View On Task-oriented Dialogue Annotation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, p. 1286–1296, Marseille, France, June 2022. European Language Resources Association. Available at: `https://aclanthology.org/2022.lrec-1.137`

- In this work we present novel dialogue corpus and related experiments.
- Citations: 0

Kulhánek, J. – Hudeček, V. – Nekvinda, T. – Dušek, O. AuGPT: Auxiliary Tasks and Data Augmentation for End-To-End Dialogue with Pre-Trained Language Models. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, p. 198–210, Online, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.nlp4convai-1.19. Available at: `https://aclanthology.org/2021.nlp4convai-1.19`

- This work presents the AuGPT model and related experiments. It also describes the DSTC 8 challenge participation.
- Citations: 53

Straka, M. – Mediankin, N. – Kocmi, T. – Žabokrtský, Z. – Hudeček, V. – Hajič, J. SumeCzech: Large Czech News-Based Summarization Dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). Available at: `https://aclanthology.org/L18-1551`

- This publication describes a dataset that can be used for training of summarization models of news in Czech language and related experiments.
- Citations: 25

Plátek, O. – Bělohlávek, P. – Hudeček, V. – Jurčíček, F. Recurrent neural networks for dialogue state tracking. *arXiv preprint arXiv:1606.08733.* 2016

- In this publication we introduce an RNN-based model used to dialogue state tracking as a sequence modeling task.
- Citations: 13

Schaub, L.-P. – Hudecek, V. – Stancl, D. – Dusek, O. – Paroubek, P. Définition et détection des incohérences du système dans les dialogues orientés tâche. (We present experiments on automatically detecting inconsistent behavior of task-oriented dialogue systems from the context). In *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, p. 142–152, Lille, France, 6 2021. ATALA. Available at: `https://aclanthology.org/2021.jeptalnrecital-taln.13`

- This work discusses the inconsistencies and problems found in dialogue corpora
- Citations: 5

Mukherjee, S. – Hudeček, V. – Dušek, O. Polite Chatbot: A Text Style Transfer Application. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, p. 87–93, 2023

- In this work we explore the abilities of language models for text style transfer.
- Citations: 5

Plátek, O. – Hudeček, V. – Schmidtová, P. – Lango, M. – Dušek, O. Three Ways of Using Large Language Models to Evaluate Chat. *arXiv preprint arXiv:2308.06502*. 2023

- In this work we propose ways of evaluating dialogue qualities with large language models and describe our submission to the DSTC 11 challenge.
- Citations: 1