Module Selection: A new task for Dialog Systems

Jan Nehring, Akhyar Ahmed, Lena A. Jäger

Abstract Many different dialog systems exist, which usually cover limited domains. This paper examines the Modular Dialog System Framework to combine many conversational agents to create a unified, diverse dialog system. The Modular Dialog System treats the underlying conversational agents as black boxes and works with any dialog system without further adaption. It also works with commercial frameworks, such as Google Dialogflow or IBM Watson Assistant, in which the inner workings are unknown company secrets. We propose a new task, Module Selection, choosing a conversational agent for a user utterance. Also, we propose an evaluation methodology for Modular Dialog Systems. Using the three available commercial frameworks, Google Dialogflow, Rasa, and IBM Watson Assistant, we create a dataset and propose three models that serve as a strong baseline for future research in Module Selection. Also, we examine the performance difference between a Modular Dialog System and the same dialog system implement in a single, monolithic system. We publish our dataset and source codes as open source.

1 Introduction

There are many reasons to combine dialog systems (DS). i) They often focus on a narrow task. Also, due to their architecture, they are often limited in what they can do. For example, a question-answering system such as DrQA [30] can answer questions only. A task-oriented dialog system can talk only about its task, and a chit-chat system such as DialoGPT [31] can do chit-chat only. A combination of multiple agents and technologies creates a more diverse DS. ii) The designers of a

Jan Nehring, Akhyar Ahmed
DFKI, Alt-Moabit 91c, 12049 Berlin, Germany e-mail: firstname.lastname@dfki.de

Lena Jäger
Department of Computational Linguistics, University of Zurich, Andreasstrasse 15, 8050 Zurich, Switzerland e-mail: jaeger@cl.uzh.ch

DS might want to combine several existing DS and save the effort of migrating them into a joint system. For example, multiple departments create chatbots in a company, and one wants to join these chatbots together into a unified system. iii) A new DS should combine several existing DS created using different technologies and cannot be implemented in a single joint DS. Alternatively, iv) a dialog system becomes so big that the performance of its NLU decreases because the model is not suitable to handle such large amounts of data. Nehring et al. [17] called these systems Modular Dialog Systems (MDS). These systems select the appropriate sub-DS for each incoming user utterance to generate the answer for the user. The experiments in this paper investigate use case ii), joining multiple existing DS.

In research, dialog systems often use machine learning for their dialog managers (DM). In industry applications, dialog systems often build on a different architecture: Popular dialog frameworks like Google Dialog Flow (GDF)[1] or IBM Watson Assistant (IWA)[2] rely on rule-based DM. Further, they rely heavily on the concept of intents. Since no established term exists yet in the literature to the best of our knowledge, we will refer to these systems as Intent-Based Dialog Systems (IBDS). IBDS usually only use machine learning for the Natural Language Understanding component, whereas the DM is rule-based. In this work, we investigate MDS for a combination of three IDBS.

Combining multiple IDBS into a unified architecture is a scarcely covered topic in the scientific literature. Although it is an issue in industry applications, see, e.g., the Google Mega Agent[3] or the concept of Skills in IWA[4], its solution is not trivial. To address this gap, we formally introduce the task *module selection* (MS), which brings the problem of multidomain dialog systems to IBDS (Section 4). Section 4 also proposes an evaluation methodology for this task. Further, we create a benchmark dataset for this task (section 5). For our experiment, we introduce and compare several models that provide a strong baseline for MS. We evaluate the models using our proposed evaluation framework. We publish all codes and our dataset on GitHub[5].

The scientific contributions of this work are i) to formally introduce the novel task MS and an evaluation framework for MS, ii) to present a dataset for the evaluation of models for MS, iii) to present three baseline models for MS, and iv) we investigate the performance drop between the MDS and non-modular dialog system, in which the DS is constructed as a single, monolithic system.

[1] https://cloud.google.com/dialogflow

[2] https://://www.ibm.com/products/watson-assistant

[3] https://cloud.google.com/dialogflow/es/docs/agents-mega

[4] https://cloud.ibm.com/docs/assistant?topic=assistant-skill-add

[5] https://github.com/jnehring/iwsds2023-modular-dialog-systems

2 Background

2.1 The Modular Dialog System Framework

The Modular Dialog System (MDS) framework [17] defines an architecture for combining several DS. Figure 1 shows the architecture of MDS. In this architecture, each DS is called a module. Each incoming user utterance is processed by an MS component that decides which module is appropriate to process this utterance. This module then produces the answer for the user. The MS task was first introduced as part of the MDS framework [17]. This paper extends this work, describes the task of MS, and proposes a dataset and an evaluation methodology.

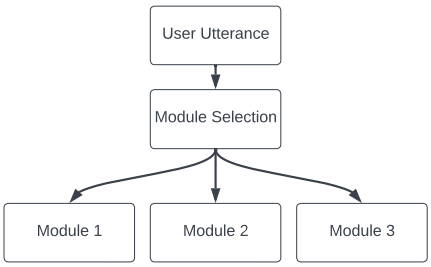


Fig. 1 Architecture of Modular Dialog System.

The modules of an MDS do not necessarily need to be IBDS. The architecture can handle other types of dialog systems, such as question-answering (see, e.g., [18]). However, this paper focuses on MDS consisting of multiple IBDS.

2.2 Google Dialogflow, IBM Watson Assistant and Rasa

This paper uses the off-the-shelf DS GDF, IWA, and Rasa modules to construct an MDS. All three are commercial products. GDF and IWA are closed source, whereas Rasa is open source. In GDF and IWA, authors of a dialog system define their chatbot in a browser-based user interface. In Rasa, the dialog designer works with text files and a command-line application.

The processing pipelines for user utterances of all three systems are similar and follow the architecture of IBDS. Rasa augments the rule-based DM by machine learning to allow conversation paths that the dialog designer did not anticipate. The exact implementations of the NLU of GDF and IWA are corporate secrets of the respective owners. Rasa uses the Dual Intent and Entity Transformer as NLU [4].

2.3 Evaluation of Dialog Systems

There are many ways how to evaluate a DS. User studies give insights into how efficient (duration) and effective (task success) a DS assists a user in solving a particular task; see [16] for an extensive review. However, these evaluations are time-consuming and expensive. Our work focuses on data-centric evaluation that does not require user studies. Such an evaluation method can only examine parts of the DS, such as the ID and ER components. These are text classification problems, and precision, recall, F1-scores, and accuracy are standard metrics used to evaluate models for ID and ED. [13, 14, 10].

3 Related Works

3.1 Combination of Dialog Systems

The scientific literature lists many approaches to how to combine multiple DS. They differ a lot; Many are concrete use-case examples that do not present a general framework [7, 19, 22, 24]. Often, the approach is tightly coupled to the types of DS that are combined. In general, it is hard to transfer the analysis results of one specific DS combination to another because, usually, the conversational agents differ from use case to use case.

A straightforward approach is to ask the user at the beginning of the dialog which domain he wants to talk about and limit the dialog to the respective sub-DS only, as in Clara [7]. Others use heuristics to rank the answers from each sub-DS [22, 24]. The DialPort framework [27, 28] connects multiple spoken dialog systems and knowledge sources and uses a Semi-Markov Decision Process [23] framework for selecting the suitable agent that can answer the user's utterance.

Joined models that combine several tasks, e.g., Song et al. [22], are popular in research. Although they often perform better than a modular approach, they tend to have a complex architecture. Also, it is only possible to combine existing DS by modifying them in a joined model. Other authors [19, 7] build similar systems using domain selection. Each DS is responsible for a domain (e.g., tourism, calendar, ...), and a domain selection component chooses the domain for each utterance. However, domain selection differs from MDS: The modules of the MDS can be from different domains but do not necessarily need to be. It is also possible that, e.g., one module is used for question answering, the other one to talk about a specific domain, and the third module for all other messages. Cercas Curry et al. [33] combined multiple agents with a simple priority list. The Black Box Agent Integration (BBAI) framework [32] is similar to MDS. However, they focus on the user perspective instead of other aspects, such as the combination of existing DS. Also, BBAI focuses on scalability, whereas our work focuses on evaluating and discussing the quality of modular and non-modular systems.

Multidomain dialog systems are often a combination of multiple DS, one for each domain. However, they only need to span multiple domains and can also be implemented in a single DS [15]. For an overview of the state-of-the-art, see DSCT8 [11] and DSTC9 [12].

3.2 Datasets

Our work builds on HWU64 [14], a dataset for NLU, ID, and ER. Section 5 describes this dataset in more detail. Another comparable dataset is CLINC150 [10] which contains additional out-of-scope utterances. Both datasets include several domains. By contrast, the NLU dataset Banking77 [5] contains many intents from a single domain.

An important dataset for multidomain dialog systems is MultiWoz [3]. However, this dataset targets DS with stochastical DM and is therefore not applicable to the MS task. Another related dataset is the Dialog Dodecathlon [21], which measures the performance of a DS in 12 different tasks, including question answering, persona grounding, empathetic dialog, and more.

4 Module Selection: A New Task

This section formally describes the task of MS. MS is the task of assigning user utterances to modules of an MDS. Given a modular dialog system with n modules $M_{1...n}$, the MS function MS assigns a module M_i (i) to an user utterance u. After MS decides on a module, this module can produce the answer to the given user utterance.

Different features are possible as inputs to an MS model. One noticeable feature is the text of the user utterance. In this case, the task resembles domain classification. Another prominent feature is the confidence scores of the intent classification of the modules. Other features one can think of are the dialog history or external knowledge, such as knowledge about the user.

We propose two measures to evaluate the quality of MS. First, one can directly evaluate the quality of the MS task, which is a classification task that can be evaluated using F1-scores. In the remainder of this paper, we call the F1-score of MS $\mathrm{F1}_{\mathrm{MS}}$.

The second measure is the quality of ID, which is also measured using F1-scores. We call it the F1-score of ID $\mathrm{F1}_{\mathrm{ID}}$. $\mathrm{F1}_{\mathrm{ID}}$ gives a more direct insight into the quality of the DS, while $\mathrm{F1}_{\mathrm{MS}}$ directly measures the quality of the MS step in the processing pipeline. For additional analysis, one can measure precision and recall of ID and MS.

Comparing the MDS to an analogous DS implemented as a single module can give an insight into whether the MDS architecture is appropriate or if it would make

more sense to implement a single module DS. We call the single-module implementation the *non-modular scenario*. Accordingly, the *modular-scenario* denotes the DS distributed over several modules. It is hard to implement both systems in practical applications to gain both numbers. However, in this paper, we calculated the F1-scores of ID in the non-modular scenario $F1_{\text{ID,nonmod}}$ for each of the DS Rasa, GDF, and IWA.

Often intent datasets are imbalanced. Therefore, in the remainder of this paper, we use micro-F1 scores to consider the class imbalance. Depending on the application, macro F1-scores can be a valid alternative.

5 Dataset Construction

In our experiment, we distinguish between two scenarios: In the *homogeneous* scenario, each module of the MDS uses the same DS technology. Each module uses a different DS technology in the *inhomogeneous* scenario. Therefore, we created three homogeneous datasets for each DS Rasa, GDF, and IWA. Then we created one inhomogeneous dataset consisting of three modules of Rasa, GDF, and IWA.

As a basis for our new dataset, we chose the dataset HWU64 [14], a dataset for ID and entity recognition. It contains 25,716 utterances from the home automation domain from 68 intents in 18 scenarios. One scenario is, for example, “alarm” with intents such as “set alarm”, “query alarm” and “remove alarm”. The creators did not name the dataset, but to our knowledge, it was first referred to as HWU64 by Casanueva et al., 2020 [5].

We split the data into four equally sized parts as shown in figure 2. $train_{ID}$ is the training data for the DS’ NLU. $train_{MS}$ is the training data for the MS. Finally there are a *valid* and a *test* set. We processed all samples by the three NLUs GDF, IWA, and Rasa and recorded the detected intents and their confidence scores.

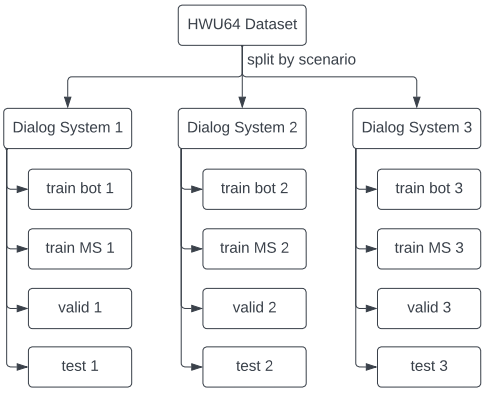


Fig. 2 Explanation of the dataset. We split the HWU64 dataset into three different DS. For each DS we assigned a train bot, train MS, valid and test part.

We randomly assigned the scenarios to the three dialog systems GDF, IWA, and Rasa. In our experiments, we found out that both $F1_{ID,mod}$ and $F1_{MS,mod}$ can vary a lot depending on this random assignment. Therefore, we repeated our dataset creation process five times to create five *splits*. Due to the random assignment, the modules have an inhomogeneous number of samples, intents, and scenarios for each agent and each split.

Further, to compute the quality of the non-modular scenario $F1_{ID,nonmod}$ we processed the whole dataset once with each module.

It was expensive to process the dataset with its 25,716 utterances five times in the modular and three times in the non-modular scenario. Further, the dataset contains many samples for some intents: 1440 samples for the intent with the most samples, and 25% of the intents have 623 or more samples. This high number of samples is unrealistic in practical use cases. Therefore, we subsampled the dataset: In each split and each intent, we removed random samples, so each intent has a maximum of 100 samples. We repeated this process separately over the different splits such that each of the splits contained different samples. In this way, we reduced the size of the dataset by 75.36% for each split.

Like the HWU64 dataset, our new dataset cannot create a fully functioning dialog system because it does not contain any data for a DM or system responses. The user utterances are not embedded in the context of the dialog. As a result, models for MS based on this dataset cannot take the dialog history into account. Our new dataset shares this problem with NLU datasets such as HWU64, CLINC150, or Banking77. Further, we did not include the entities from the original dataset in our new dataset.

Table 1 shows the performance of the dialog systems in the non-modular scenario. A side result of our experiments is the comparison of ID of GDF, IWA, and Rasa (Table 1. IWA has the best-performing ID on our dataset. [14] reported similar results, although, in their experiment, the differences between GDF, IWA, and Rasa were less significant.

Table 1 Performance of the non-modular dataset.

Module	$F1_{ID,nonmod}$	$P_{ID,nonmod}$	$R_{ID,nonmod}$
GDF	79.87%	82.92%	78.48%
IWA	86.36%	86.93%	86.49%
RAS	76.26%	79.44%	75.34%

A dataset similar to this can be constructed with relatively low effort in real-life situations because the data already exists. If practitioners want to combine several of their already existing IDBS, then they already have training data for the NLU of their systems and can reuse this as training data for MS.

Figure 3 shows boxplots of the confidence values of the three DS over the five splits. The table shows confidence values for both in and out-of-domain samples for each DS. One can see that the average in-domain confidence values differ between the DS. In general, Rasa produces higher confidence values than the others. Across all dialog systems, out-of-domain samples can reach confidences as high as 1.0.

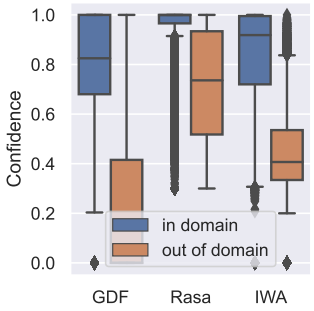


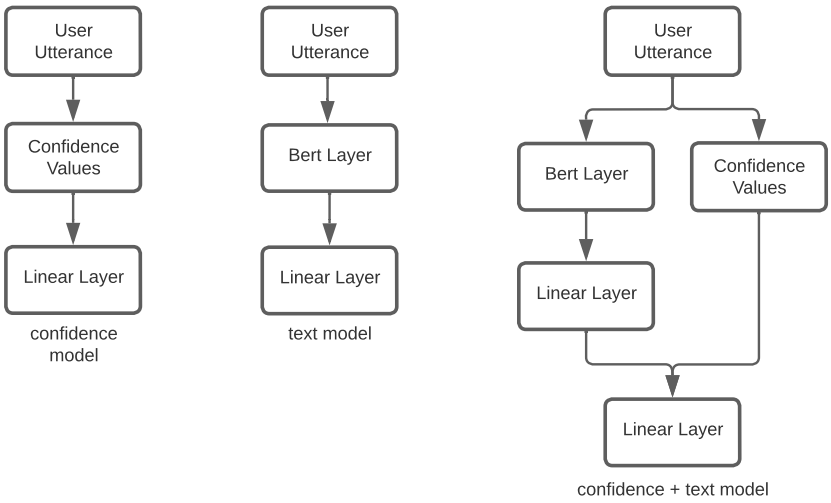
Fig. 3 Distribution of confidence values for Google Dialogflow, Rasa, and IBM Watson Assistant for in-domain and out-of-domain samples.

6 Baseline Evaluation

6.1 Models

We define three model architectures. The models use confidence values, text or text and confidence values as input features. Therefore we call the models *confidence*, *text* and *text + confidence*. Figure 4 shows the architecture of the models.

Fig. 4 Architectures of the three models *conf*, *text* and *conf+text*.



Our MDS consists of $n = 3$ modules. All three models use a user utterance as input and predict which of the three modules should answer the user utterance. We encode this prediction as one-hot encoding: The models produce a vector $y \in \mathbb{R}^n$.

Each index in this vector represents one module. The target values that the neural networks learn from is a vector $y_{\text{target}} \in \mathbb{R}^n$ that is 0 in all places, except for the index of the target module, which is 1.

Each module produces a confidence value for each utterance, a real number between 0 and 1. In a real-world system, one would send each incoming user utterance to the modules and collect the confidence values of the modules' ID. In our experiments, we can use the confidence values from our dataset. We concatenate the confidence values to a vector $x_{\text{conf}} \in \mathbb{R}^n$ to create the confidence features. The *confidence* model (Figure 4, left) maps these confidence values directly to the output, through a linear transformation $y = x_{\text{conf}}A^T + b$. $x_{\text{conf}} \in \mathbb{R}^n$ is the input vector, $y \in \mathbb{R}^n$ is the output vector, $b \in \mathbb{R}^n$ is the bias vector, and $A \in \mathbb{R}^{n \times n}$ is a matrix. The dimensionality of x, A, and b can vary.

Model *text* (Figure 4, center) uses the user utterance as feature. It uses a standard BERT for sequence classification architecture [6], which uses a linear layer on top of a pre-trained BERT model. The trainable parameters are the parameters of the BERT model and parameters A and b of the linear layer.

Model *text + confidence* (Figure 4, right) is a combination of the former two models. In the figure, the left branch is the *confidence* model and the right branch is the *text* model. Both produce an output of length n, which the model concatenates to a vector $x_2 \in \mathbb{R}^{2n}$. x_2 is then mapped to y using another linear transformation. The trainable parameters are the BERT layer and the three linear layers.

All models are trained on the $train_{MS}$ section of the dataset using Cross-Entropy Loss [1]. Training examples are encoded in one-hot encoding, meaning that they are n dimensional vectors with 0 on every position except for the position of the label, which is 1. We perform a grid search using the *valid* dataset to find optimal values for batch size (32) and learning rate ($5e^{-5}$). Metrics F1_{ID} and F1_{MS}, along with respective precision and recall values, are calculated on *test* part of the dataset. We trained and evaluated the models for each split and averaged the metrics.

6.2 Results

6.2.1 Homogeneous dataset

Table 2 shows the performance of our models on the homogeneous datasets. Over both tasks MS and ID, the performance of models *text* and *text-confidence* are on-par with each other, while the performance of the *confidence* model is much lower across both tasks.

For the MS task, the performance of the *confidence* model differs, while the other models show very similar values. This is no surprise because the confidence values are different between the different DS, but the textual inputs are the same. We assume that the *text+confidence* learns to rely much more on the textual features and mostly ignores the confidence features, which explains why the values are so similar.

Table 2 F1-scores dialog systems in the homogeneous dataset for models confidence, text, and text + confidence. The scores are averaged over the five splits; the numbers in brackets denote the standard deviation.

Task	Dialog System	Model	F1	Precision	Recall
MS	GDF	confidence	24.46% (9.58)	24.50% (11.27)	31.22% (9.99)
		text	**92.71 % (0.62)**	**92.87% (0.72)**	**92.71% (0.62)**
		text + confidence	92.47% (0.41)	92.60% (0.43)	92.48% (0.41)
MS	RAS	confidence	24.02% (6.35)	19.97% (6.60)	36.25% (9.60)
		text	**92.36% (1.22)**	**92.51% (1.08)**	**92.36% (1.22)**
		text + confidence	91.90% (1.16)	92.11% (1.06)	91.91% (1.17)
MS	IWA	confidence	19.33% (8.17)	19.70% (10.46)	32.32% (10.74)
		text	91.97% (1.38)	**92.23% (1.19)**	91.95% (1.41)
		text + confidence	**92.15% (0.94)**	92.21% (0.93)	**92.16% (0.93)**
ID	GDF	confidence	22.13% (8.19)	27.64% (12.18)	23.23% (8.25)
		text	**73.08% (1.30)**	**78.44% (1.20)**	**70.51% (1.21)**
		text + confidence	72.83% (1.09)	78.41% (1.04)	70.18% (1.09)
ID	RAS	confidence	20.44% (5.23)	19.89% (6.51)	26.98% (7.06)
		text	**73.28% (0.57)**	**76.09% (0.58)**	**72.59% (0.77)**
		text + confidence	73.17% (0.68)	76.03% (0.44)	72.46% (0.90)
ID	IWA	confidence	19.28% (8.08)	16.99% (6.98)	27.66% (9.79)
		text	**66.02% (13.93)**	65.52% (15.46)	**69.27% (11.10)**
		text + confidence	65.93% (13.51)	**65.66% (14.87)**	69.23% (10.57)

The results of the *confidence* models are low and show a large standard deviation across the different splits. We conclude that the *confidence* models are susceptible to the respective dataset split and, because of their unreliable performance, are even less helpful than their low average performance already suggests.

The other models (*text* and *text + confidence* show consistent results (low standard deviation) for all splits. The exception from the rule is the ID task using the IWA DS: Its standard deviation is significantly higher (10.57-15.46) than for the other DS (0.44 - 1.3). This result cannot stem from the MS because the quality of MS for IWA is consistent / has a low standard deviation. We further analyzed the results of IWA and found out that, although IWA has high F1-scores in ID, on three subsets, the f1 scores drop below 20%: This is the case for agent 2 in split 3 and 4 (f1 scores of 8% and 13%) and agent 0 in split 2 (20%). IWA is a black box; therefore, we cannot explain why the performance drops on these subsets. This result is counter-intuitive since IWA had the strongest performing ID in the non-modular scenario (see table 1).

Table 3 shows the f1-scores of ID between of the non-modular and the modular scenario for model *text*. The performance metrics are copied directly from tables 1 and 2. The table also shows the difference between the two f1-scores. The performance is generally lower in the modular scenario. This is obvious because the MS introduces an additional source of error. At the same time, we expect the f1-scores of ID of the individual modules get higher than $F1_{ID,nonmod}$ because the models need to

differentiate between fewer intents, which is generally easier. However, this is just an assumption; we do not present data to support this hypothesis.

Table 3 F1-score of ID in the non-modular and the modular scenario on the homogeneous dataset with MS model *text*.

Module	$F1_{ID,nonmod}$	$F1_{ID,mod}$	Difference
GDF	79.87%	73.08%	6.79%
IWA	86.36%	66.02%	17.34%
RAS	76.26%	73.28%	2.98%

6.2.2 Inhomogeneous dataset

Table 4 shows the performance of the dialog systems on the inhomogeneous dataset. The results are similar to the homogeneous dataset: The performance of *text* and *text+confidence* model are high across both tasks, while the performance of the *confidence* model is low. Not surprising, for the models *text* and *text-confidence*, the performance of the MS task is very similar for the homogeneous and the inhomogeneous dataset because the textual features are the same between both datasets. Surprisingly, the performance in the ID task achieves higher results in the inhomogeneous datasets than in all three homogeneous datasets. Also, the standard deviations are low for the *text* and *text-confidence* models, we cannot see the high standard deviation of IWA from the homogeneous dataset in our experiment, although IWA is processing one third of the inhomogeneous dataset.

Table 4 F1-scores dialog systems in the inhomogeneous dataset for models conf, test and joined.

Task	Model	F1	Precision	Recall
MS	confidence	26.08% (6.82)	22.37% (7.47)	38.19% (4.80)
MS	text	92.21% (1.07)	**92.40% (0.96)**	92.23% (1.05)
MS	text + confidence	**92.32% (0.86)**	92.38% (0.83)	**92.32% (0.87)**
ID	confidence	25.29% (4.46)	24.21% (6.63)	31.03% (3.95)
ID	text	76.82% (0.59)	79.65% (0.71)	**76.04% (0.57)**
ID	text + confidence	**76.89% (0.26)**	**79.79% (0.54)**	75.98% (0.21)

6.3 Discussion of the experiments

As expected, the modular scenario introduces an error compared to the non-modular scenario (see table 3). The drop in $F1_{ID}$ ranges from 2.98% - 17.34%, depending on

the models used for ID and MS and is therefore highly dependent on the models. So unfortunately, we cannot quantify the performance drop from the non-modular to the modular scenario. Our analysis indicates that IWA usually has high f1 scores in ID, but underperforms dramatically on a few subsets. These low-performing subsets happen to be in the modular scenarios. We argue that the cause of the low performance is not the MDS, but the specific module IWA. We argue that including a non-functioning module in an MDS results in a non-functioning MDS. As long as the single modules show strong performance, as do GDF and RAS, the performance drop is between 2.98% and 6.79%.

We showed that the models *text* and *text + confidence* perform best for MS. We can show that the user utterances text is a good feature for the MS task and provides a strong baseline. The *confidence* model shows poor performance in MS. Our data does not show if the problem of the *confidence* model is the much more simple model or the input features. The distribution of confidence values (see figure 3) indicates that the confidence values are not very reliable features.

7 Conclusion

We have presented *module selection* as a novel task. Further, we have presented a dataset to evaluate module selection models and an evaluation framework. We have presented three models and evaluated them using the dataset and the methodology. We could show that in 2/3 experiments, the MDS showed only slightly weaker performance compared to the non-modular scenario. In the third case, we argue that the problem lies in the modules and not the MDS framework.

We have shown that text is, at least in our dataset, a more reliable feature for module selection than confidence scores. The models serve as a strong baseline for future work in module selection.

Our presented evaluation framework does not directly evaluate the quality of the DS. Instead, it measures the quality of ID in the modular scenario. Like NLU datasets like HWU64, CLINC150 or Banking77, it does not take dialog history into account. Dialogs often span more than one turn, and neglecting the dialog context is a critical limitation. Nevertheless, this evaluation framework is a practical way of evaluating the quality of module selection in modular dialog systems.

Acknowledgements

This work was partially supported by the German Federal Ministry of Education and Research under grant 01|S20043 (Lena A. Jäger) and by German Federal Ministry for Economic Affairs and Energy under grand 01MK20011R (Jan Nehring).

References

1. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.
2. Bobrow, D. G., Kaplan, R. M., Kay, M., Norman, D. A., Thompson, H. S., and Winograd, T. (1977). GUS, A Frame-Driven Dialog System. *Artificial Intelligence*, 8(2):155–173.
3. Budzianowski, P., Wen, T.-H., Tseng, B.-H., Casanueva, I., Ultes, S., Ramadan, O., and Gašić, M. (2018). 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*, pages 5016–5026, Brussels, Belgium, October-November. Association for Computational Linguistics.
4. Bunk, T., Varshneya, D., Vlasov, V., and Nichol, A. (2020). DIET: Lightweight Language Understanding for Dialogue Systems. *arXiv*.
5. Casanueva, I., Temčinas, T., Gerz, D., Henderson, M., and Vulić, I. (2020). Efficient Intent Detection with Dual Sentence Encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online, July. Association for Computational Linguistics.
6. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
7. D'Haro, L. F., Kim, S., Yeo, K. H., Jiang, R., Niculescu, A. I., Banchs, R. E., and Li, H. (2015). CLARA: A Multifunctional Virtual Agent for Conference Support and Touristic Information. In *Natural Language Dialog Systems and Intelligent Assistants*, pages 233–239. Springer International Publishing, oct.
8. Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. *34th International Conference on Machine Learning, ICML 2017*, 3:2130–2143, jun.
9. Jurafsky, D. and Martin, J. H. (2021). *Speech and Language Processing (3rd edition draft)*.
10. Larson, S., Mahendran, A., Peper, J. J., Clarke, C., Lee, A., Hill, P., Kummerfeld, J. K., Leach, K., Laurenzano, M. A., Tang, L., and Mars, J. (2019). An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China, November. Association for Computational Linguistics.
11. Li, J., Peng, B., Lee, S., Gao, J., Takanobu, R., Zhu, Q., Huang, M., Schulz, H., Atkinson, A., and Adada, M. (2020). Results of the Multi-Domain Task-Completion Dialog Challenge. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, Eighth Dialog System Technology Challenge Workshop*.
12. Li, J., Zhu, Q., Luo, L., Liden, L., Huang, K., Shayandeh, S., Liang, R., Peng, B., Zhang, Z., Shukla, S., Takanobu, R., Huang, M., and Gao, J. (2021). Multi-domain Task-oriented Dialog Challenge II at DSTC9. In *AAAI-2021 Dialog System Technology Challenge 9 Workshop*.
13. Liu, J., Li, Y., and Lin, M. (2019a). Review of Intent Detection Methods in the Human-Machine Dialogue System. *Journal of Physics: Conference Series,* 1267:12059, jul.
14. Liu, X., Eshghi, A., Swietojanski, P., and Rieser, V. (2019b). Benchmarking Natural Language Understanding Services for building Conversational Agents. In *Proceedings of the Tenth International Workshop on Spoken Dialogue Systems Technology (IWSDS)*, Ortigia, Siracusa (SR), Italy, apr. Springer.
15. Mrkšić, M, Séaghdha, D. O., Thomson, Blaise, Gašić, M., Su, P. H., Vandyke, D., Wen, T.H., and Young, S. (2015). Multi-domain Dialog State Tracking using Recurrent Neural Networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 794–799, Beijing, China. Association for Computational Linguistics.

16. Möller, S. (2005). *Quality of Telephone-Based Spoken Dialogue Systems*. Kluwer Academic Publishers, Boston.
17. Nehring, J. and Ahmed, A. (2021). Normalisierungsmethoden für Intent Erkennung Modularer Dialogsysteme. In Benjamin Weiss Stefan Hillmann, editor, *Tagungsband der 32. Konferenz. Elektronische Sprachsignalverarbeitung (ESSV-2021), March 3-5, Berlin, Germany*. TUDpress.
18. Nehring, J., Feldhus, N., Kaur, H., and Ahmed, A. (2021). Combining Open Domain Question Answering with a Task-Oriented Dialog System. In *Proceedings of the 1st Workshop on Document-grounded Dialogue and Conversational Question Answering (DialDoc 2021)*, pages 38–45, Online, aug. Association for Computational Linguistics.
19. Planells, J., Hurtado, L.-F., Segarra, E., and Sanchis, E. (2013). A Multi-domain Dialog System to integrate heterogeneous Spoken Dialog Systems. Technical report.
20. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.
21. Shuster, K., Ju, D., Roller, S., Dinan, E., Boureau, Y., and Weston, J. (2020). The Dialogue Dodecathlon: Open-Domain Knowledge and Image Grounded Conversational Agents. In Dan Jurafsky, et al., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2453–2470. Association for Computational Linguistics.
22. Song, Y., Li, C.-T., Nie, J.-Y., Zhang, M., Zhao, D., and Yan, R. (2018). An Ensemble of Retrieval-Based and Generation-Based Human-Computer Conversation Systems. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4382–4388. International Joint Conferences on Artificial Intelligence Organization.
23. Sutton, R., Precup, D., and Singh, S. (1999). Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial Intelligence*, 112:181–211.
24. Tanaka, R., Ozeki, A., Kato, S., and Lee, A. (2019). An Ensemble Dialogue System for Facts-Based Sentence Generation. *arXiv*.
25. Williams, J. D. and Young, S. (2007). Partially observable Markov decision processes for spoken dialog systems. *Computer Speech Language*, 21(2):393–422.
26. Young, S., Gašić, M., Thomson, B., and Williams, J. D. (2013). POMDP-Based Statistical Spoken Dialog Systems: A Review. *Proceedings of the IEEE*, 101(5):1160–1179.
27. Zhao, T., Lee, K., and Eskenazi, M. (2016a). DialPort: A General Framework for Aggregating Dialog Systems. In *Proceedings of the Workshop on Uphill Battles in Language Processing: Scaling Early Achievements to Robust Methods*, pages 32–34, Austin, TX, November. Association for Computational Linguistics.
28. Zhao, T., Lee, K., and Eskenazi, M. (2016b). DialPort: Connecting the spoken dialog research community to real user data. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 83–90.
29. Clarke, C., Peper, J., Krishnamurthy, K., Talamonti, W., Leach, K., Lasecki, W., Kang, Y., Tang, L., and Mars, J. (2022). One Agent To Rule Them All: Towards Multi-agent Conversational AI. In Findings of the Association for Computational Linguistics: ACL 2022, pages 3258–3267, Dublin, Ireland. Association for Computational Linguistics.
30. Chen, D., Fisch, A., Weston, J. , Bordes, A. (2017). Reading Wikipedia to Answer Open-Domain Questions. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.
31. Zhang, Y., Sun, S., Galley, M., Chen, Y., Brockett, C., Gao, X., Gao, J., Liu, J., and Dolan, B. (2020). DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 270–278, Online. Association for Computational Linguistics.
32. Clarke, C., Peper, J., Krishnamurthy, K., Talamonti, W., Leach, K., Lasecki, W., Kang, Y., Tang, L., and Mars, J. (2022). One Agent To Rule Them All: Towards Multi-agent Conversational AI. In Findings of the Association for Computational Linguistics: ACL 2022, pages 3258–3267, Dublin, Ireland. Association for Computational Linguistics.

33. Cercas Curry, A., Papaioannou, I., Suglia, A., Agarwal, S., Shalyminov, I., Xinnuo, X., Dusek, O., Eshghi, A., Konstas, I., Rieser, V., and Lemon, O. (2018). Alana v2: Entertaining and informative opendomain social dialogue using ontologies and entity linking. In 1st Proceedings of Alexa Prize (Alexa Prize 2018).