One Agent To Rule Them All: Towards Multi-agent Conversational AI

Anonymous ACL submission

Abstract

The increasing volume of commercially available conversational agents (CAs) on the market has resulted in users being burdened with learning and adopting multiple agents to accomplish their tasks. Though prior work has explored supporting a multitude of domains within the design of a single agent, the interaction experience suffers due to the large action space of desired capabilities. To address these problems, we introduce a new task BBAI: **B**lack-**B**ox **A**gent **I**ntegration, focusing on combining the capabilities of multiple black-box CAs at scale. We explore two techniques: *question agent pairing* and *question response pairing* aimed at resolving this task. Leveraging these techniques, we design One For All (OFA), a scalable system that provides a unified interface to interact with multiple CAs. Additionally, we introduce MARS: **M**ulti **A**gent **R**esponse **S**election, a new encoder model for question response pairing that jointly encodes user question and agent response pairs. We demonstrate that OFA is able to automatically and accurately integrate an ensemble of commercially available CAs spanning disparate domains. Specifically, using the MARS encoder we achieve the highest accuracy on our BBAI task, outperforming strong baselines.

1 Introduction

Influenced by the popularity of intelligent conversational agents (CAs), such as Apple Siri and Amazon Alexa, the conversational AI market is growing at an increasingly rapid pace and projected to reach a valuation of US $13.9 billion by 2025 (Market and Markets, 2020). These CAs have already begun to show great promise when deployed in domain-specific areas such as driver assistance (Lin et al., 2018), home automation (Luria et al., 2017), and food ordering (Frangoul, 2018) with platforms such as Pandora and Facebook today hosting more than 300,000 of these agents (Chaves and Gerosa, 2018; Nealon, 2018).

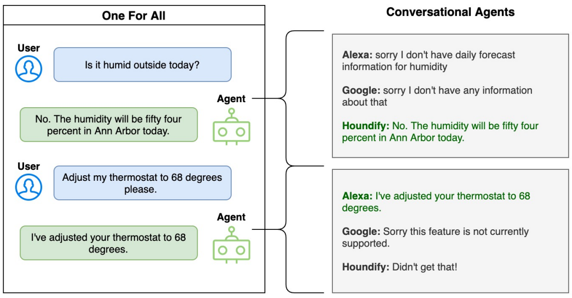

Figure 1: An example interaction using One For All which integrates multiple production black-box agents into a unified experience.

Most CAs are designed to be specialized in a single or set of specific domains. As such, users are required to interact with multiple agents in order to complete their tasks and answer their queries as shown in figure 1. E.g. A user may use Alexa for online shopping but engage with Google Assistant for daily news updates. Additionally, a given agent may be more proficient at a specific domain over another i.e A finance CA is better suited to answer finance questions. As a result, users are taxed with the burden of learning and adopting multiple agents leading to an increase in the cognitive load of interacting with agents, further discouraging the proliferation of their usage (Dubiel et al., 2020; Novick et al., 2018; Saltsman et al., 2019). This is escalated further as the number of conversational agents deployed into the market continues to increase. Therefore, the need arises for unifying multiple independent CAs through one conversational interface. This need has manifested in the commercial conversational AI industry with initiatives such as the Amazon Voice Interoperability Initiative (Amazon, 2019) which aims to create voice-enabled products that contain multiple, distinct, interoperable intelligent assistants on a single device. However, this interaction is still manual, requiring the user to orchestrate which agent is initiated. In addition, while it is possible to have distinct agents in a single device, users prefer interacting with a single agent over multiple (Chaves and Gerosa, 2018).

Prior work has explored in part combining the strengths of multiple agents in one system but they rely on direct access to the design and implementation details of the to-be-integrated agents. Subramaniam et al. (2018) and Cercas Curry et al. (2018) direct incoming user questions to a specific agent based on the candidate agents' internal knowledge graph and NLU architectures, respectively. However, in practice, the majority of the publicly available CAs are "black boxes" where their inner-workings contain highly-protected IP that is not accessible to the public. Additionally, Cercas Curry et al. (2018) facilitates their bot selection with a manual heuristic preference order that requires intimate knowledge of the agents to construct, and additional effort to maintain, thus not scaling well for the adaption of existing agents and introduction of new agents. Therefore, the task of integrating multiple production black-box CAs with a unified interface remains an open problem.

In order to explore this problem, we introduce the task BBAI: **B**lack-**B**ox **A**gent **I**ntegration that focuses on integrating multiple black-boxes CAs. We propose two techniques to tackling black-box multi-agent integration: (1) Question agent pairing and (2) Question response pairing. Intuitively, these two approaches can be viewed as a query-to-agent classification problem in contrast to that of an response selection problem. This formulation allows us to facilitate multi-agent integration whilst operating within the black-box constraints of the agents. Using these techniques we develop *One For All*, a novel conversational system that accurately and automatically unifies a set of black-box CAs spanning disparate domains. Additionally, we introduce MARS: **M**ulti **A**gent **R**esponse **S**election, a new encoder model for question response pairing that jointly encodes user question and agent response pairs. We evaluate these techniques on a suite of 19 publicly available agents consisting of Amazon Alexa[1], Google Assistant[2], SoundHound Houndify[3], Ford Adasa (Lin et al., 2018) and many more.

Specifically, this paper makes the following contributions:

- Formulation of the BBAI task that focuses on the challenge of integrating disparate black-box conversational agents into one experience. We construct a new dataset for this task, comprising of examples from a suite of 19 commercially deployed conversational agents. We publish our code and datasets. [4]

- We design *One For All*, a novel conversational system that accurately and automatically unifies a set of black-box CAs and introduce the MARS encoder model that outperforms strong state-of-art classification and ranking model baselines on the BBAI task.

- We conduct a thorough evaluation of various agent integration approaches showing that our MARS encoder outperforms strong baselines. We show that by facilitating the integration of multiple agents we can alleviate the needs for users to adopt multiple agents whilst facilitating the improvement and growth of agents over time.

2 BBAI: Black-Box Agent Integration Task Formulation

Building a unified interface for production agents spanning different domains presents several key challenges. First, most commercially available CAs are black-boxes, providing little to no information on their inner workings. Any approaches for agent integration must operate without relying of the internals of any given agent. Second, these conversational agents are constantly improved upon and expanded with new capabilities. The agent integration approaches need to be flexible and adaptive to these changes with relative ease. Given these constraints we assume the existence of the following information sources for the agent integration task:

1. User query/utterance: The question that the user asks the agent.

2. Agent skill representation: A textual representation that denotes what each agent is capable of. This can be in the form of example queries or a description of that agent.

3. Agent response: Each agent's response to the query asked.

[1] https://developer.amazon.com/en-US/alexa

[2] https://assistant.google.com/

[3] https://www.houndify.com/

[4] https://datasets.code

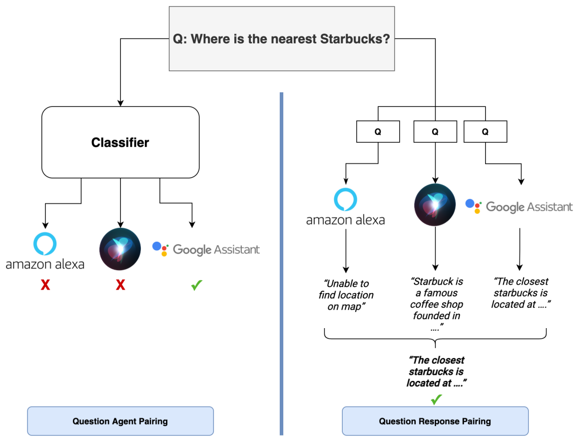

Figure 2: Overview of our proposed black-box agent integration techniques. In QA Pairing, the goal is to select the correct agent using information of the agent's capabilities. In QR Pairing, the goal is to select the correct agent response.

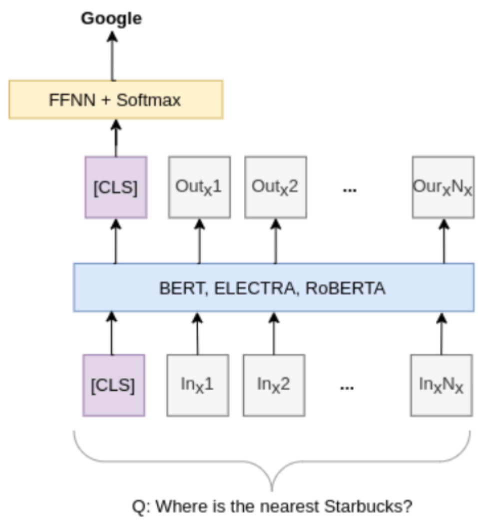

Figure 3: The transformer-based classification models in the OFA system. The models are trained on question agent pairs and tasked to predict a agent to route the given query to.

Using this information we formulate the task of agent integration as given a query Q, a set of agents $A = \{a_1, a_2, \ldots, a_n\}$ and a set of agent responses $R = \{r_1, r_2, ..., r_n\}$ to query Q, determine the question-agent-response pair (Q, A_i, R_i) that resolves the query Q. Further, given the information available, we can taxonomize our approach into two techniques: (1) Question agent pairing where we preemptively select the agent for the query and (2) Question response pairing where evaluate the set of returned responses as depicted in Figure 2.

2.1 Question Agent Pairing

As shown in Figure 2, the goal of question agent pairing is, given a query Q and a set of agents $A = \{a_1, a_2, \ldots, a_n\}$, determine the question-agent pair (Q, A_i) that resolves the query Q. At its core, this can be viewed as a classification problem where the model learns the respective capabilities of each independent agent in order to predict which agent to use for a given question.

2.2 Question Response Pairing

As shown in Figure 2, the goal of question response pairing is, given a query Q and a set of agent responses $R = \{r_1, r_2, ..., r_n\}$, determine the question-response pair (Q, R_i) such that R_i resolves the query Q.

3 The One For All System

In this section, we present the design One For All (OFA), a scalable system that integrates multiple black-box CAs with a unified interface. We explain the various approaches implemented in One For All, detailing their inputs, outputs and training methodology.

3.1 Question Agent Pairing

In order to predict the best agent for a given query, knowledge of each agent's individual skill-set is required. However, as described in the task formulation in Section 2, the internal details of the agents are unavailable. Everyday users of these agents have no insight into the internal specifics of these agents. However, they are able to use these agents to accomplish tasks by building a mental model of each agents' respective capabilities through usage over time. We draw inspiration from this to determine the information we can use to represent an agent's skills without access to its internals.

3.1.1 Agent Skills Representation

Following the learning patterns described above, we model an agent's skill-set in two distinct ways:

(1) Query examples: Similar to building knowledge over time via agent interaction, an agents' query examples allows the model to learn what type of queries each agent is capable of resolving. For example, questions such as *"Where is the nearest gas station?"* and *"Direct me to Starbucks please"* will be amongst the query examples for a *"Directions"* agent.

(2) Agent descriptions:. These are textual summaries of an agent's capabilities. For example, a bank releases a new CA for its customers to use

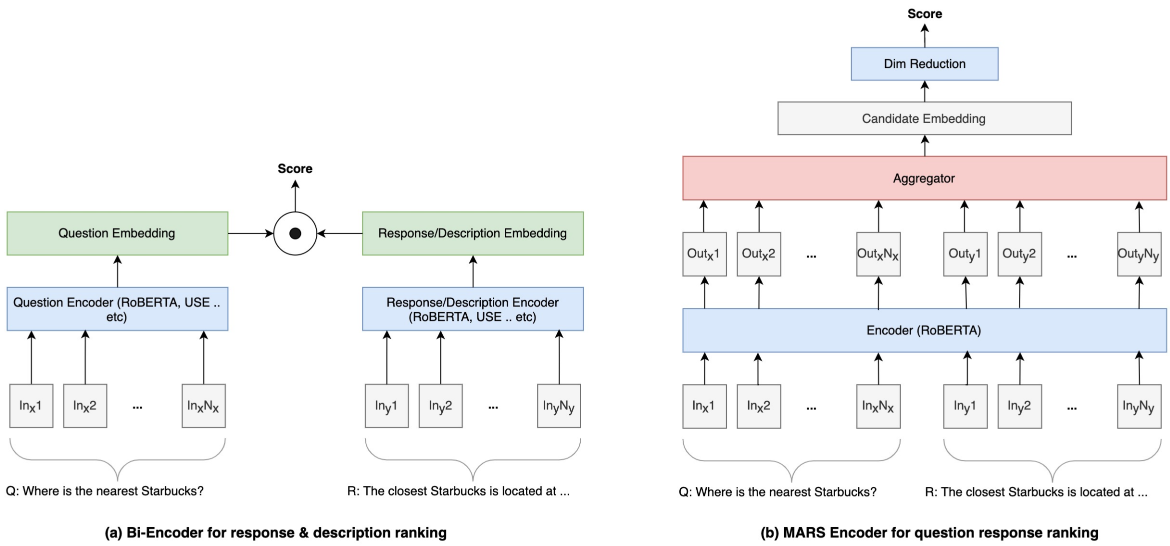

Figure 4: Overview of OFA approaches. (a) Bi-Encoder which is used for both QA and QR pairing encodes the question and candidate response/description separately and computes a ranking score via a dot product calculation. (b) Our MARS encoder jointly encodes the question and response into a single transformer and performs self-attention between the question and candidate response. To score a response we reduce the candidate embedding from a vector to a scalar score between 0...1 (Humeau et al., 2020).

instead of having to visit the bank. Accompanied with this agent will be a semi-formal description of what this agent is capable of doing. This information is often publicly available in the agent's marketing materials.

Using these query examples and agent descriptions, we explore approaches for determining the agent best to resolve a given query. We describe in more details the dataset collection process in Section 4.

Question agent pairing using query examples QA pairing using query examples seeks to explore how best we can facilitate agent orchestration in a data constrained environment where only a few examples of the questions the agents can answer are present. This is similar to the use of text examples for the training of an intent classifier but at the agent level instead. Therefore, we treat this as a multi-label classification problem where a given query Q is mapped to a set of agents A. e.g Q: *'locate me some good places in Kentucky that serve sushi'* maps to the set of agents A: ["Alexa", "Google"] indicating that this query can be correctly answered by the agents Alexa and Google. Specifically, as shown in Figure 3, we train a multi-label classifier on top of state-of-the-art transformer models, BERT (Devlin et al., 2019), RoBerta (Liu et al., 2019) and Electra (Clark et al., 2020) to predict an agent A given a query Q.

Question agent pairing using agent descriptions While query examples are useful for understanding the capabilities of a given agent, they may not be readily available. When a new agent is introduced, users are unsure of the exact questions this agent can answer but they would typically have access to an explanation of its capabilities. As an alternative, we explore the use of such a description of the agents. For this task, we assume a textual description of an agent's capabilities, e.g. "Our productivity bot helps you stay productive and organized. From sleep timers and alarms to reminders, calendar management, and email ....".

In order to map a given query Q to an agent A described by description D_i, we treat this as a semantic similarity task. The intuition behind this is that for a given query Q the agent that is capable of answering a given question is likely to feature a agent description semantically similar to the question. We explore a suite of pre-trained and fine-tuned language models focusing on ranking the relevance of given description D_i to a query Q. Additionally, given the length of descriptions and the range of capabilities that may be described within a single description, we split the full description at the sentence level and use each sentence to represent a single skill S_i belonging to agent A. With this variation, the question-description similarity score is calculated as the $\max_i SemSim(Q, S_i)$.

For our BBAI task we consider the following state-of-art semantic retrieval based approaches whose utility map well to our problem domain:

BM25 This classic method measures keyword similarity and uses it to estimate the relevance of documents to a given search query (Robertson and Zaragoza, 2009). We encode the collection of agent descriptions and return the agent whose description is most relevant to the given query.

Universal Sentence Encoder (Cer et al., 2018) A sentence encoding model for encoding sentences into high dimensional vectors. We use the transformer model[5] for our experiment. As shown in part (a) of Figure 4, we encode the user query and the agent description and compute the dot product as a ranking score.

Roberta + STS (Reimers and Gurevych, 2019) We fine-tune Roberta-base on the STS benchmark dataset and use this model to encode our agent descriptions and user query. We compute the cosine similarity between the two vectors to compute a ranking score for each description as shown in Figure 4.

3.2 Question Response Pairing

Contrary to question agent pairing which selects the agent beforehand, question response pairing assumes that we provide each agent in the ensemble the opportunity to respond to the query Q and focus on selecting the best response from the set of returned responses. As such, we treat this as a response ranking problem of determining which question-response pair (Q, R_i) best answers the query Q. Prior work has shown strong performance on sentence pairing tasks such as this through the use of sentence encoders and language model fine-tuning (Henderson et al., 2019; Humeau et al., 2020; Reimers and Gurevych, 2019). We explore the use of these architectures in the domain of response selection with the goal of learning representations for correct question answering from diverse conversational agents.

BM25 Similar to our use of BM25 for question agent pairing we use it to rank each of our question response pairs.

USE and USE QA (Yang et al., 2019) We apply the USE model from our agent pairing task to rank agent responses. In addition, we consider USE QA, an extended version of the USE architecture specifically designed for question-answer retrieval applications. We use the Bi-Encoder architecture as shown in Figure 4 (a).

Roberta + STS We fine-tune Roberta-base on the STS benchmark dataset and use it to encode our question response pairs using the bi-encoder architecture in figure 4.

MARS encoder Pre-existing sentence pairing scoring models are tuned to score sentence pairs deemed semantically similar. However, in the case of conversational systems, an agent's response can be semantically similar but still incorrect. e.g Q: "What is the weather in Santa Clara today?", R: "Weather information is currently unavailable". These two sentences are semantically similar but the response does not resolve the query. In the MARS encoder we focusing on learning representation beyond similarity by also incorporating correctness of agent responses. Using the cross-encoder architecture (Humeau et al., 2020) shown in part (b) of Figure 4, we train a question response pair scoring model for the task of ranking responses to a given query Q generated by conversational agents. We concatenate both the input question and response performing full self attention on the entire input sequence. By passing both the question and agent response through a single transformer, the agent response is able to attend to user query and produce a more input sensitive representation of the question response embedding. Using the generated question response embedding vector we then convert it to a scalar score $S(Q, R_i)$ between 0..1 via a linear layer. Our training objective is to minimize the Cross-Entropy loss between the correct agent responses and the negative agent responses to the query Q.

4 Dataset Construction

For the task of BBAI, we construct a new dataset focusing on making it representative of real-world conversational agents at scale and covering a broad range of domains.

Using Amazon Mechanical Turk and scenario/paraphrasing-based prompts (Kang et al., 2018; Larson et al., 2019), we crowdsourced utterances across a range of agent skills/capabilities. These skills were extracted from public information sources describing each of the agents, in addition to observing their

[5] `https://tfhub.dev/google/universal-sentence-encoder-large/3`

Question	Agent Response			
	Alexa	Google	Houndify	Adasa
At how many miles will I run out of gas	"here's something I found on the web according to freakonomics.com previously when cars got 8 to 12 miles ...."	"on the website post Dash gazette.com they say some popular car models can make it between 30 and 50 miles ....",	Didn't get that!	"With your current fuel economy of 28 MPG, you should be able to cover about 532 miles with the fuel you have."
Is it gonna be warm Friday in Alhambra?	"here's something I found on the web according to Wikipedia. Org Cobra is one of the 100 selected cities in India which will be developed ...."	"No, it won't be hot Friday in Alhambra, California. Expect a high of 21 and a low of 6.",	"There will be a high of seventy degrees in Alhambra on Friday November twenty-seventh."	"Out of scope!"

Table 1: Sample question agent responses from the One For All dataset. Responses highlighted in green represent agent responses voted as correct by crowd workers.

capabilities. Our dataset is comprised of utterances across 37 broad domain categories. These include domains such as *Weather, Flight Information, Directions, Automobile*, etc. Crowd workers were paid \$0.12 for 5 utterances. These submitted utterances were then vetted by hand to ensure quality. Using the curated utterances, we then generated question responses by querying each agent to gather its response to the utterance.

In order to generate ground truth samples on which of the question-response pairs (Q, R_i) correctly resolves the query Q we launched a crowd-sourcing task asking workers to indicate the candidate responses that best answer the question shown. Five workers were assigned to each response selection task and majority voting (>2) was used to label the gold responses. As such for each query Q and the set of responses R we were able to gather the necessary question-agent pairs (Q, A_i) and question-response pairs (Q, R_i) needed evaluate our approaches.

Agent Descriptions We gather our agent descriptions by scraping the contents of each of the agent's public product pages and their built-in feature documentation web pages. We then manually clean, reformat and merge this data into a single document per agent. For our experiment, we focus only on extracting descriptions related to the built-in features of our agents.

Overall our dataset contains 5550 utterances with 19 question-response pairs per question (one from each of the 19 agents), 105,450 in total. The utterances are split into 3700 utterances (100 per domain) for the training set and 1850 (50 per domain) for the test set. The train and test sets respectively contain 2399 and 1186 utterances with at least one positive question-response pair. In the remaining examples, none of the agents were able to achieve annotator agreement (>= 3). A sample dataset example is shown in table 1 with responses from 4 of the 19 agents.

5 Results and Discussion

In this section we present and analyze the results of our experiments, detailing our insights and discussing the implications of each of our techniques.

Evaluation task: Similar to standard information retrieval evaluation measures, we denote accuracy as the metric *precision@1* and use it to evaluate both our question agent and question response pairing approaches. For question agent pairing this metric denotes: Given a set of N agents to the given query, whether the agent selected ultimately resolves the query successfully. For question response pairing it denotes: Given a set of N responses to the given query, whether the top-scoring response resolves the query successfully. For this evaluation, we test on examples with at least one valid agent response.

5.1 Question agent pairing

The results are summarized in tables 2 and 3. We find that for the QA pairing Roberta yields the best result with an accuracy of 69% in selecting the correct agent and 61.8% when scaled to 19 agents. Similarly, we see that we achieve can fair performance in extreme data scarce environments when using simple agent descriptions compared to that of query agent examples, with USE achieving 47.8% accuracy. Using agent descriptions offers greater flexibility in facilitating the improvement of agents over time compared to query examples since it only requires an update to the agent description. However, it still falls short when compared to using a single agent like Google or Alexa. Also, while consistent in learning to recognize the domain a given agent may be performant in, QA approaches fall short in a few cases:

	Method	Accuracy (n=4)	Agent Breakdown			
			Alexa	Google	Houndify	Adasa
Question Agent Pairing (QA Labels)	Bert	68.31	37.98	**40.93**	18.49	2.6
	Electra	67.86	35.28	**42.01**	20.11	2.6
	Roberta	**69.03**	34.92	**41.56**	20.65	2.87
Question Agent Pairing (Descriptions)	BM25	27.91	13.91	10.95	17.33	**57.81**
	USE	**47.84**	13.20	28.82	**52.42**	5.56
	Roberta+STS	39.40	18.94	22.35	**51.35**	7.36
Response Selection	BM25	51.07	28.64	24.69	14.81	**31.86**
	USE	72.89	**34.20**	27.65	22.98	15.17
	USE QA	75.49	**41.65**	36.45	17.95	3.95
	Roberta+STS	69.83	18.94	22.35	**51.35**	7.36
	MARS	**79.70**	37.34	**43.9**	15.71	3.05
Individual Agents	Alexa	49.37	**-**	-	-	-
	Google	**51.79**	-	-	-	-
	Houndify	34.82	-	-	-	-
	Adasa	4.12	-	-	-	-

Table 2: Performance breakdown of QA and QR approaches on our BBAI task when using our 4 largest agents Alexa, Google, Houndify and Adasa. **Note:** n = number of agents.

Method		Accuracy (n=19)	Agents
Question Agent Pairing (QA Labels)	Bert	59.10	Alexa, Google
	Electra	52.86	Houndify, Adasa
	Roberta	**61.88**	Recipe agent
Question Agent Pairing (Descriptions)	BM25	23.69	Dictionary agent
	USE	**43.59**	Task Manager
	Roberta+STS	36.67	Hotel agent, Stock agent
Response Selection	BM25	59.94	Math agent, Sports agent
	USE	64.42	Wikipedia agent
	USE QA	71.66	Mobile Account agent
	Roberta+STS	56.82	Banking agent
	OFA Encoder	**83.55**	Coffee shop agent
Individual Agents	Alexa	44.09	Event Search agent
	Google	**48.06**	Jokes agent
	Houndify	32.04	Reminders agent
	Adasa	3.45	Covid-19 agent

Table 3: Performance breakdown of QA and QR approaches on our BBAI task on all 19 commericial agents we show that the MARS encoder is able to scale and leverage the capabilities of new agents added to the ensemble without diminishing performance compared to other approaches.

Evaluation Performance per Domain (n=19)			
Domain	**MARS (QR)**	**USE (QA)**	**Roberta (QA)**
Weather	0.88	0.45	0.67
Directions	0.78	0.29	0.44
Auto	1.00	0.79	0.82
Restaurant Suggestion	0.79	0.5	0.68
Travel Suggestion	0.97	0.33	0.57
Time	0.81	0.54	0.76
Flight Info	0.83	0.61	0.7
Date	0.82	0.47	0.56

Table 4: Further breakdown of the best-performing approaches per technique on a subset of 8 out of the 37 domains. We find that our MARS encoder generalizes well across the various agent domains.

(1) Agent overlap - This is when a given domains' coverage is split between various agents. e.g The model learns that both Alexa & Google have proficiency handling some weather queries but it remains unclear about which one is best suited for the current query at hand.

(2) Query variation - While an agent's examples or descriptions may allude to proficiency in a given domain, it may still fail when asked certain query variations. e.g Figure 1 shows a case where Alexa is capable of handling weather queries but fails when a condition like humidity is asked for. Another example is when a similar question in asked in a different or more complex way. Both Houndify & Alexa are known to be proficient at answering age related questions but for question like *"How old I will be on September 28, 1995 if I was born on March 29, 1967?"*, Alexa is unable to answer as opposed to Houndify.

These cases are further highlighted when inspecting QA pairing performance at the domain level in table 4. We find that the QA approaches struggle with domains such as *"travel suggestion"* and *"Directions"* which are heavily split in coverage.

5.2 Question response pairing

In overall performance we find that our MARS encoder outperforms strong baselines, achieving 83.55% accuracy on the BBAI task. We note that our MARS encoder outperforms the best single performing agent (Google Assistant) by 32%. This shows the utility and power of OFA in not only alleviating the need for you users to learn and adopt multiple agents but also validating that multiple agents working collectively can achieve significantly more than single agents working in isolation.

When inspecting the performance of MARS at the domain level we see in Table 4 that it is able to maintain its high performance across the varying domains unlike the QA approaches. This advantage comes from the ability to select an agent at the response level allowing the system to catch cases in which an agent once deemed proficient fails or another agent improves.

5.3 Agent pairing vs Response pairing

We now describe the trade-offs between agent pairing and response pairing. Question response pairing greatly outperforms agent pairing in terms of accuracy, given that it is privy to the final responses from each of the agents. However, in practice this comes with additional networking, compute, and latency costs, having to send the query to each of the agents and await their response. Given that the querying of agents is done in parallel, the latency cost is equal to that of the slowest agent. Question response pairing also better supports agent adaptation. With response pairing, a system can seamlessly add or remove an agent without diminishing the experience as show by MARS in table 3. In addition, as conversational agents are upgraded to offer a more diverse feature-set such as new domain support or improved responses, they can instantly be integrated into a response pairing approach.

5.4 Scalability

We evaluate our approaches on a suite of 19 commercially deployed agents spanning 37 broad domain categories. As shown in table 2 we examine performance when using the 4 largest agents in terms of domain sport and popularity (Alexa, Google Assistant, Houndify and Ford Adasa) showing improvement upon single agent use in both QA and QR approaches. When scaled up to 19 agents, MARS encoder improves even further by leveraging the new capabilities of the additional agents and is the only approach that does not decrease in performance as the number of agents and domains scale. This improvement is achieved via the more input sensitive representations that the MARS encoder is able to learn by encoding both the question and response in a single transformer.

6 Related Work

Ensemble approaches to solving complex tasks in the context of NLP are widely used (Deng and Platt, 2014; Araque et al., 2017). In dialogue systems, recent attempts at ensemble approaches and multi-agent architectures include Cercas Curry et al. (2018) and Subramaniam et al. (2018). AlanaV2 (Cercas Curry et al., 2018) demonstrated an ensemble architecture of multiple bots using a combination of rule-based machine learning systems built to support topic-based conversations across domains. It was built to be an open domain bot supporting topic based conversations. Specifically, AlanaV2's architecture utilizes a variety of ontologies and NLU pipelines that draw information from a variety of web sources such as reddit. However, its agent selection approach is guided by a simple priority bot list. Subramaniam et al. (Subramaniam et al., 2018) describe their conversational framework that employs an *Orchestrator Bot* to understand the user query and direct them to a domain-specific bot that handles subsequent dialogue. In our work, we expand up the multi-agent goal by focusing on the integration of black-box conversational agents at scale.

6.1 Response Selection

This is the task of selecting the most appropriate response given context from a pool of candidates. It is a central component to information retrieval applications and has become a focus point in the evaluation of dialogue systems. (Sato et al., 2020; Henderson et al., 2019; Wang et al., 2020). Prior work has shown strong performance on sentence pairing tasks through the use of sentence encoders and language model fine-tuning (Henderson et al., 2019; Humeau et al., 2020; Reimers and Gurevych, 2019). In our work we explore the task of response selection using it as one of the basis for integrating black-box conversation agents.

7 Conclusion

The rapid proliferation of conversational agents calls for a unified approach to interacting with multiple CAs. Key challenges of building such an interface lies in that most commercial CAs are black-boxes with hidden internals. This paper introduces BBAI a new task of agent integration that focuses on unifying black-boxes CAs across varying domains. We explore two task techniques, question agent pairing and question response pairing and present One For All, a scalable system that unifies multiple black-box CAs with a centralized user interface. Using a combination of commercially available conversational agents, we evaluate a variety of approaches to multi-agent integration through One For All. Our MARS encoder achieves 88.5% accuracy on BBAI and outperforms the best single agent configuration by over 32%. These results demonstrate the power of One For All which can leverage state-of-the-art NLU approaches to enable multiple agents to collectively achieve more than any single conversational agent in isolation eliminating the need for users to learn and adopt multiple agents.

References

Amazon. 2019. Amazon and leading technology companies announce the voice interoperability initiative.

Oscar Araque, Ignacio Corcuera-Platas, J. Fernando Sánchez-Rada, and Carlos A. Iglesias. 2017. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77:236 – 246.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Amanda Cercas Curry, Ioannis Papaioannou, Alessandro Suglia, Shubham Agarwal, Igor Shalyminov, Xu Xinnuo, Ondrej Dusek, Arash Eshghi, Ioannis Konstas, Verena Rieser, and Oliver Lemon. 2018. Alana v2: Entertaining and informative open-domain social dialogue using ontologies and entity linking. In *1st Proceedings of Alexa Prize (Alexa Prize 2018)*.

Ana Paula Chaves and Marco Aurelio Gerosa. 2018. Single or multiple conversational agents?: An interactional coherence comparison. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 191:1–191:13, New York, NY, USA. ACM.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

Li Deng and John Platt. 2014. Ensemble deep learning for speech recognition. In *Proc. Interspeech*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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.

Mateusz Dubiel, Martin Halvey, Leif Azzopardi, and Sylvain Daronnat. 2020. Interactive evaluation of conversational agents: Reflections on the impact of search task design. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*, ICTIR '20, page 85–88, New York, NY, USA. Association for Computing Machinery.

Anmar Frangoul. 2018. Here's how robots are transforming takeout deliveries. https://www.cnbc.com/2018/08/02/virtual-assistants-and-robotic-deliveries-are-html.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2019. Convert: Efficient and accurate conversational representations from transformers.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Yiping Kang, Yunqi Zhang, Jonathan K. Kummerfeld, Parker Hill, Johann Hauswald, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2018. Data collection for dialogue system: A startup perspective. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 1311–1316. Association for Computational Linguistics.

Shih-Chieh Lin, Chang-Hong Hsu, Walter Talamonti, Yunqi Zhang, Steve Oney, Jason Mars, and Lingjia Tang. 2018. Adasa: A conversational in-vehicle digital assistant for advanced driver assistance features. In *Proceedings of the 31st ACM Symposium on User Interface Software and Technology (UIST 2018)*, UIST-31, Berlin, Germany. ACM.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Michal Luria, Guy Hoffman, and Oren Zuckerman. 2017. Comparing social robot, screen and voice interfaces for smart-home control. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 580–628, New York, NY, USA. ACM.

Market and Markets. 2020. Conversational ai market.

Gary Nealon. 2018. Using facebook messenger and chatbots to grow your audience.

David Novick, Laura J. Hinojos, Aaron E. Rodriguez, Adriana Camacho, and Mahdokht Afravi. 2018. Conversational interaction with multiple agents initiated via proxemics and gaze. In *Proceedings of the 6th International Conference on Human-Agent Interaction*, HAI '18, page 356–358, New York, NY, USA. Association for Computing Machinery.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Thomas L. Saltsman, Mark D. Seery, Cheryl L. Kondrak, Veronica M. Lamarche, and Lindsey Streamer. 2019. Too many fish in the sea: A motivational examination of the choice overload experience. *Biological Psychology*, 145:17–30.

Shiki Sato, Reina Akama, Hiroki Ouchi, Jun Suzuki, and Kentaro Inui. 2020. Evaluating dialogue generation systems via response selection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 593–599, Online. Association for Computational Linguistics.

Sethuramalingam Subramaniam, Pooja Aggarwal, Gargi B. Dasgupta, and Amit Paradkar. 2018. Cobots - a cognitive multi-bot conversational framework for technical support. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pages 597–604, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Weishi Wang, Shafiq R. Joty, and Steven C. H. Hoi. 2020. Response selection for multi-party conversations with dynamic topic tracking. *CoRR*, abs/2010.07785.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Multilingual universal sentence encoder for semantic retrieval.