Addressing Slot-Value Changes in Task-oriented Dialogue Systems Through Dialogue Domain Adaptation

Tiziano Labruna
Fondazione Bruno Kessler,
Trento, Italy
Free University of Bozen-Bolzano
tlabruna@fbk.eu

Bernardo Magnini
Fondazione Bruno Kessler,
Trento, Italy
magnini@fbk.eu

Abstract

Recent task-oriented dialogue systems learn a model from annotated dialogues, and such dialogues are in turn collected and annotated so that they are consistent with certain domain knowledge. However, in real scenarios, domain knowledge is subject to frequent changes, and initial training dialogues may soon become obsolete, resulting in a significant decrease of the model performance. In this paper, we investigate the relationship between training dialogues and domain knowledge, and propose *dialogue domain adaptation*, a methodology aiming at adapting initial training dialogues to changes intervened in the domain knowledge. We focus on slot-value changes (e.g., when new slot-values are available to describe domain entities) and define an experimental setting for dialogue domain adaptation. First, we show that current state-of-the-art models for dialogue state tracking are still poorly robust to slot-value changes of the domain knowledge. Then, we compare different domain adaptation strategies, showing that simple techniques are effective to reduce the gap between training dialogues and domain knowledge.

1 Introduction

Conversational agents are receiving great attention in recent years, both in research and applications (McTear, 2020), mainly because of the progress achieved by neural approaches in modeling dialogue phenomena (Louvan and Magnini, 2020), (Balaraman et al., 2021). Particularly, we focus on task-oriented dialogue systems (Young et al., 2010), which are able to assist a user for specific tasks (e.g., booking a restaurant, taking an appointment, execute commands) in a certain conversational domain. Such data-driven dialogue systems typically learn a model from annotated training dialogues (e.g., (Budzianowski et al., 2018), (Du et al., 2020), (Price, 1990)), and such dialogues, in turn, are collected and annotated according to a certain domain scenario (e.g., restaurants in a certain town, songs in a certain music data-set, etc.). Once the model is trained, it is applied to understand new conversations in the same domain, or in a similar domain.

However, in real scenarios, domain knowledge is subject to frequent changes, and initial training dialogues may soon become obsolete, resulting in a significant decrease in the model performance. It is common that new domains (e.g., RESTAURANT) are added or removed for a certain conversational scenario, as well as slots, slot-values and instances. Such situations require the capacity of the dialogue system to adapt its behaviours to domain knowledge changes. Depending on the complexity of the changes that occur, domain adaptation of data-driven systems can be approached in two directions: (i) improving the model robustness, and (ii) adapting the training dialogues to the new situation. While the first direction has been largely explored through several techniques, including transfer learning (Louvan and Magnini, 2019), and zero-shot learning though schema-guided models (Wu et al., 2019a; Kim et al., 2020; Zhang et al., 2019; Heck et al., 2020; Balaraman and Magnini, 2021), and delexicalization (Henderson et al., 2014a,b; Yu et al., 2020), in this paper we take the second, less investigated, perspective, focusing on the relation between training dialogues and domain knowledge.

We have defined a new experimental setting for *dialogue domain adaptation*, where, given an initial conversational domain (KB-SOURCE), available training dialogues (D-SOURCE) are adapted to be as much as possible consistent to a modified knowledge base (KB-TARGET), resulting

Proceedings of Recent Advances in Natural Language Processing, pages 780–789
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_089

in new set of dialogues (D-TARGET). Then, we use a state-of-the-art model for dialogue state tracking and assess the performance of different adaptation strategies against a gold standard of manually adapted target dialogues. We run a number of experiments showing that: (i) current state-of-the-art models are very poorly robust to changes over the domain knowledge; (ii) a particular class of domain changes, i.e., slot-value changes, can be effectively addressed through simple dialogue domain adaptation techniques, which operate substitutions over D-SOURCE. Finally, as part of our study, we highlight that current component-based evaluation settings for task-oriented dialogue systems (i.e., slot filling, intent detection, dialogue state tracking, utterance generation) are not sensible to correctness of system responses, which, instead, is crucial to assess domain adaptability.

The paper is structured as follows. Section 2 provides basic background in task-oriented dialogue systems. Section 3 defines the conversational adaptation task, while in Section 4 we introduce the proposed *dialogue domain adaptation* approach. Section 5 introduces the new experimental setting, and Section 6 provides the results of our experiments and discusses them.

2 Task-oriented Dialogue Framework

This section provides background related to task-oriented dialogue systems, particularly about how domain knowledge is managed and about dialogue state tracking.

2.1 Domain Knowledge

According to most of the recent literature (Budzianowski et al., 2018; Bordes et al., 2017; Mrkšić et al., 2017), we assume a task-oriented dialogue between a system and an user, composed of a sequence of turns $\{t_1, t_2, ...t_n\}$. The goal of the dialogue is to retrieve a set of entities (possible empty) in a domain knowledge base (KB) that satisfy the user needs. A domain ontology O provides a schema for the KB, and typically represents entities (e.g., RESTAURANT, HOTEL, MOVIE) according to a pre-defined set of slots S (e.g., FOOD, AREA, PRICE, for the RESTAURANT domain), and values that a certain slot can assume (e.g., EXPENSIVE, MODERATE and CHEAP, for the slot PRICE). On the basis of the entities defined in the domain ontology, the application knowledge base, KB, is then populated with instances of such entities. As in most of the literature, we distinguish *informable slots*, which the user can use to constraint the search (e.g., AREA), and *requestable slots* (e.g., PHONENUMBER), whose values are typically asked only when a certain entity has been retrieved through the dialogue.

At each turn in the dialogue, both the user and the system may refer to facts in the KB, the user with the goal of retrieving entities matching his/her needs, and the system to propose entities that can help the user to achieve the dialogue goals.

2.2 Dialogue State Tracking

In a task-oriented system a dialogue state tracker (DST) maintains a distribution over the dialogue states based on the dialogue history. A dialogue state d_i for a turn t_i is typically represented as a set of slot-value pairs, such as {PRICE=MODERATE, FOOD=ITALIAN}, meaning that at t_i the system assumes that the user is looking for an Italian restaurant with a moderate price.

After being collected through Wizard of Oz, turns of each dialogue are annotated with the corresponding *dialogue state*, consisting of an intent and a set of slot-value pairs. The following is an example of the annotation provided in a portion of a MultiWOZ 2.0 dialogue:

USER: I would like a moderately priced restaurant in the west part of town.
INFORM(PRICE=MODERATE, AREA=WEST)
SYSTEM: There are three moderately priced restaurants in the west part of town. Do you prefer Indian, Italian or British?
REQUEST(FOOD)
USER: Can I have the address and phone number of the Italian location?
INFORM(PRICE=MODERATE, AREA=WEST, FOOD=ITALIAN)
REQUEST(ADDRESS, PHONE-NUMBER)

3 Dialogue Domain Adaptation: Task Definition

In this section we define *dialogue domain adaptation* (DDA) and its core properties. In

Dialogue Source	Dialogue Target
USER: I am looking for a **european** food restaurant in the expensive price range. Can you help with that?	USER: I am looking for a **osteria** food restaurant in the expensive price range. Can you help with that?
SYS: There are **5** of those. What area do you want to dine in?	SYS: There are **4** of those. What area do you want to dine in?
USER: In the centre of town please.	USER: In the centre of town please.
SYS: How about **eraina**? Shall I book you a table?	SYS: How about **Hosteria Il Malandrone**? Shall I book you a table?
USER: Yes, please. It will be just me and I 'd like to eat at 21:00 on the same day as my train.	USER: Yes, please. It will be just me and I 'd like to eat at 21:00 on the same day as my train.
SYS: OK, I 've got your booked. The reference number is **VMNDNKV2** and they 'll hold your table for 15 minutes.	SYS: OK, I 've got your booked. The reference number is **WPQHRNE4** and they 'll hold your table for 15 minutes.

Figure 1: Example of dialogue domain adaptation. Words in bold indicate slot-values that have been adapted.

our setting we assume an initial conversational domain, represented in a KB-SOURCE, and corresponding annotated training dialogues D-SOURCE. Then, as in real application scenarios, we assume that a number of changes occur in KB-SOURCE, such that a new conversational domain KB-TARGET needs to be considered. *Dialogue domain adaptation* consists in the capacity to automatically produce new annotated dialogues D-TARGET, such that they maintain both the linguistic structure and the linguistic variability of the initial D-SOURCE dialogues, while, at the same time, being consistent with the new KB-TARGET.

Figure 1 provides a concrete example of DDA. Here we have a source dialogue, taken from the MultiWOZ data-set of restaurant booking conversations, mentioning restaurants in a certain region. Notice that it is implicitly assumed that the system responses are true facts in a KB-SOURCE (e.g., if the system says *What about Eraina?*, it means that KB-SOURCE contains a restaurant named *Eraina*). However, this might not be true in KB-TARGET, where the Eraina restaurant might not exist anymore. The target dialogue in Figure 1 is basically the same dialogue as the source dialogue, although adapted to be consistent with a KB-TARGET. DDA focuses on the automatic generation of such D-TARGET dialogues starting from D-SOURCE dialogues.

In the rest of the section we consider three core characteristics that affect DDA: domain changes, dialogue internal coherence, and KB-dialogue adherence.

3.1 Domain Changes

DDA strongly depends on the amount and types of changes that differentiate KB-SOURCE from KB-TARGET. Intuitively, the more the changes, the more the difficulty to adapt D-SOURCE to dialogues consistent to KB-TARGET.

As described in Section 2, we assume that domain knowledge is represented through a domain ontology, providing a schema that describes entities with slots and corresponding slot-values, and through a knowledge base, providing instances of domain entities. Accordingly, changes in domain knowledge may occur in four cases: (i) a domain is introduced or removed (e.g., adding HOTEL); (ii) a slot for a domain is introduced or removed (e.g., adding PARKING among the slots for RESTAURANT); (iii) a slot-value for an existing slot is introduced or removed (e.g., adding TUSCAN as slot-value for the slot FOOD of the RESTAURANT domain); (iv) an instance for a certain domain is introduced or removed (e.g., adding a new restaurant like BELLA NAPOLI with its features). In a concrete situation, such changes may occur either in an incremental way, as small changes of the domain knowledge reflecting modifications of the world, or as a consequence of a domain shift, when, for instance, a dialogue system is moved from one city to another.

In this paper we focus on *slot-value changes*, and we assume that, while moving from KB-SOURCE to KB-TARGET, both domains, slot names and number of instances are kept without any change.

3.2 Dialogue Internal Coherence

Human collected dialogues (as D-SOURCE dialogues) possess an internal coherence that needs to be preserved in D-TARGET dialogues. As an example, in the source dialogue in Figure 1, we assume that the *Eraina* restaurant mentioned by the system is coherent with the request of the user for a *european* restaurant. We assume that co-reference between anaphoric expressions and their references (e.g., *those* on the D-SOURCE) are kept consistent within the scope of a dialogue. Similarly, language variations (e.g., using different spellings for referencing the same entity) should be used consistently in the same dialogue.

Moreover, the semantic annotations of the dialogues need to respect the references of the utterance, even when anaphoric expressions occur. For example, if the user says *I want to book a table on the same day as my train arrival*, the annotation for "booking-day" has to be consistent to the referent mentioned in the previous part of the conversation.

3.3 KB-Dialogue Adherence

The core assumption behind *dialogue domain adaptation* is that system utterances have to be as much as possible aligned with domain knowledge, meaning that the system responses should correspond to true facts in the domain knowledge. As an example, in Figure 1, the D-SOURCE dialogue reports that there are 5 restaurants in KB-SOURCE with certain characteristics, while the corresponding D-TARGET turn has been adapted reporting 4 restaurants in the KB-TARGET.

When the dialogue collection is carried on manually, KB-Dialogue adherence is supposed to be checked by humans, so that each system utterance is coherent to the KB. However, human mistakes may occur, for instance in case crowd workers in a Wizard of OZ setting make wrong queries to the domain KB. The relevance of KB-Dialogue adherence in our experimental setting will be discussed in Section 5.

4 Substitution-based DDA

We approach the dialog domain adaptation task described in section 4 through the substitution of slot-values in D-SOURCE with slot-values selected from KB-TARGET. Figure 2 depicts the elements of our experimental setting, highlighting the relationships between them.

The dataset D-SOURCE consists of both training and test dialogues, with the latter possibly containing a certain number of slot-values that are unseen in the training set. D-SOURCE dialogues are collected in a strong connection with KB-SOURCE, which has been quantified through a KB-Dialogue adherence measure. Given a certain KB-TARGET, which differs from KB-SOURCE for a proportion of slot-values estimated by the KB-overlap, the KB mapping defines the substitutions that need to be done for every slot-value, and, on the basis of this mapping, the adaptation process generates the D-TARGET data-set.

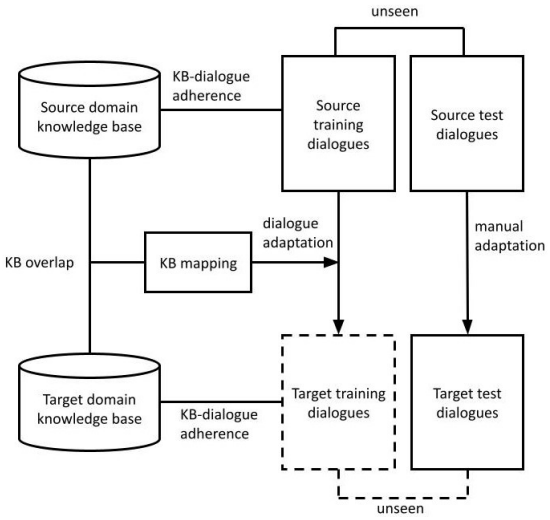

Figure 2: Scheme of *dialogue domain adaptation* methodology.

4.1 D-TARGET Generation

Starting from an annotated D-SOURCE dialogue, a KB-SOURCE and a KB-TARGET, the creation of a D-TARGET dialogue follows a general substitution-based procedure.

For each slot-value found in D-SOURCE, according to the semantic annotations provided in the dataset, the first step consists of checking whether the slot-value is known in KB-SOURCE. If it is known, then we try to substitute it with a corresponding slot-value in the KB-TARGET, otherwise we keep it as it is in D-TARGET.

In order to check whether the slot-value is known in KB-SOURCE, we compare the source slot-value with all slot-values in KB-SOURCE that have the same slot-name, applying a similarity function based on a variation of the Gestalt Pattern Matching algorithm (Black, 2004). We used a

threshold for deciding when to consider two slot-values as equal or different. The threshold has been determined by using a data-set of positive and negative examples and empirically finding the value that best separates the two sets (e.g., with the similarity value of 0.6, 10 slot-value pairs are classified as equal while they are different, and 8 pairs are classified as different while they are the same).
If at least one slot-value in KB-TARGET with similarity above the threshold is found, then we apply a mapping function that selects a target slot-value v_t to be used for substitution. We have defined three mapping functions.

RANDOM-DIALOGUE. For every slot-value v_i in a D-SOURCE dialogue, this mapping function randomly chooses one value from all the slot-values in the dialogue, both from User and System, that have the same slot-name as v_i. Then, the randomly chosen slot value is assigned to v_t.

RANDOM-KB. For every slot-value v_i in KB-SOURCE, this mapping function randomly chooses one value from all the slot-values in the KB-TARGET that have the same slot-name as v_i. Then, the randomly chosen slot value is assigned to v_t.

FREQUENCY-KB. For every slot-value v_i in KB-SOURCE, this mapping function chooses a value in KB-TARGET that has the same slot-name as v_i, on a frequency mapping (e.g., “indian” may correspond to “italian” if the proportion of instances for the two slot-value is similar). Then, the chosen slot value is assigned to v_t.
In order to generate a D-TARGET dialogue, we then go through all the slot-values in D-SOURCE and for each of them we check if there is a v_t mapping. If a mapping is found, we perform the substitution, otherwise we leave it as it is.

4.2 KB Overlap

We use *Knowledge Bases Overlap* as a measure that determines how much KB-SOURCE is equivalent to KB-TARGET. In order to assess this, all unique possible values for every slot of the domains in one KB need to be compared to all the unique values of the same slot for the other KB. For instance, if the slot is "restaurant-area", KB-SOURCE may have values ["north", "south", "east", "west"], while KB-TARGET may have values ["centre", "north", "south"]. In such case, the equal values for the slot would be 2, and the total different values would be 5, resulting in a KB overlap of 40%. In other words, the KB overlap indicates the percentage of changes that need to be done for changing from one KB to another.

4.3 Estimating KB-Dialogue Adherence

In accordance to what has been defined in paragraph 3.3, we intend the KB-Dialogue Adherence as the extent by which a dialogue is consistent to the content of the KB. In order to estimate this, we distinguish two cases:

- Case 1: the slot-values of one instance mentioned in the utterance correspond to the description of the instance in the KB (e.g., "The Old Cambridge is an expensive restaurant in the centre").
- Case 2: the system states a certain number of instances that meet certain conditions, and the KB actually contains the same number of instances (e.g. "There are 15 hotels with 4 stars in the north").

The adherence for each case is given by the percentage of the system’s utterances that complies with the respective condition, and the total KB-dialogue adherence is then calculated by averaging the two cases.

4.4 Unseen Slot-value Ratio

Given a dialogue data-set split into training and test set, the *unseen slot-value* ratio measures the number of slot-values that are present in the test set dialogues, but that are not present in the training set dialogues. This is an important indicator, which significantly affects the performance of a model, as, for every unseen slot-value, the model has to make a prediction over something for which it had not been trained on.

5 Experimental Setting

This section describes the experiments that we carried on to test *dialogue domain adaptation* based on slot-value substitutions. The experimental setting includes an initial KB-SOURCE and corresponding D-SOURCE dialogues; a set of handcrafted test D-SOURCE dialogues; few substitution algorithms that we experimented to produce different D-TARGET dialogues; and a state-of-art dialogue state

tracker to check the performance of different adaptation strategies.

5.1 KB-SOURCE and D-SOURCE

Our experimental setting is derived from the MultiWOZ 2.3 data-set (Han et al., 2020). Experimental D-SOURCE dialogues consists of the 10,438 MultiWOZ dialogues, with an average of 11.06 turns per dialogue, collected with the technique of the Wizard of Oz and spanning over 7 domains: Train, Attraction, Restaurant, Hotel, Police, Taxi, Hospital. Through dialogues the user asks information about things to do in Cambridge, such as restaurant or hotel reservation, request for train timing, information about an attraction, etc. The MultiWOZ knowledge base (i.e., our KB-SOURCE) presents an average of 525 instances per domain and 8.5 slots per instance.

Domain	Slot Type	Slot-value union	Inter-section	Overlap %
Attrac.	inf.	24	20	83.33
	all	862	139	16.13
Hosp.	inf.	118	0	0.00
	all	239	0	0.00
Hotel	inf.	18	18	100
	all	361	83	22.99
Police	inf.	2	0	0.00
	all	7	1	14.29
Rest.	inf.	47	15	31.91
	all	1173	129	11.00
Train	inf.	1226	184	15.01
	all	5703	699	12.26
All by slots	inf.	1435	237	16.52
	all	8345	1051	12.59
All by domains	inf.	2846	454	38.38
	all	15828	1963	12.78

Figure 3: Slot-value overlap between Cambridge KB-SOURCE and Pisa KB-TARGET .

5.2 KB-TARGET

We decided to experiment DDA on a conversational domain with similar characteristics as MultiWOZ, simulating an application for the city of Pisa, in Italy. Pisa presents a number of characteristics that are very similar to Cambridge, such as the dimension, the presence of an important University with many departments spread all over the city, and the characterization of being a touristic city. The information necessary to create the Pisa KB-TARGET has been taken from a number of publicly available data-sets[1]. Starting from the MultiWOZ KB, as discussed in Section 3, we focused on slot-value changes, i.e., preserving all information but slot-values. The overlap between the resulting Pisa KB-TARGET and the initial Cambridge KB-SOURCE is shown in Table 3. We show both the breakdown slot-value overlap for single domains, as well as the aggregate overlap by slots and domains. Overall, 12.59% of the slot-values overlaps, indicating that the domain shift from Cambridge to Pisa has produced a drastic change in term of slot-values.

5.3 Test D-TARGET Dialogues

We created a Pisa test set (Pisa-T), using the test portion of the FREQUENCY KB Pisa data-set, which has then been manually revised with the aim of creating an error free data-set with respect to the Pisa KB, for what regards system messages. This means that every system utterance should tell the truth relatively to what is contained in the KB.

Test dialogues have been produced according to a semi-automatic procedure. First we apply substitutions to the original MultiWOZ test dialogues according to the frequency strategy described in section 4.1. In order to perform these corrections, we automatically identified the dialogues that showed a lack of adherence, and we adjusted them manually, both for the system utterances and for the semantic annotations, making sure that all test dialogues comply with the information reported in the respective KBs.

5.4 DDA Substitution Algorithms

We intend to compare different DDA substitution algorithms against the no-adaptation situation (i.e., the original MultiWOZ). We created three different training datasets that have been used for running experiments against the manually constructed test set.

[1] http://www.datiopen.it

No Adaptation Cambridge (NO ADAPT.). This is the original dataset from MultiWOZ 2.3. It is the baseline for our experiments, as no dialogue adaptation has been applied.

Random Selection from Dialogues (RANDOM-D). This substitution strategy is intended to preserve as much as possible the linguistic variety of D-SOURCE in D-TARGET. The dataset has been created in two steps: first, a preliminary Pisa data-set has been created with a frequency strategy, then all the slot-values in the dialogue have been randomly shuffled.

Random Selection from KB (RANDOM-KB). This substitution strategy is intended to take advantage of the alignment between KB-SOURCE and KB-TARGET, although with a basic random selection of target slot-values. This strategy does not preserve linguistic variety in D-SOURCE. Starting from the Cambridge dialogues, the substitutions have been done taking one random slot-value from KB-TARGET, only when the original slot-value was present in the KB-SOURCE.

Frequency-based Selection from KB (FREQ. KB). This substitution strategy is intended to take full advantage of the alignment between KB-SOURCE and KB-TARGET, choosing target slot-values that maximise their frequency in KB-SOURCE. This strategy does not preserve linguistic variety in D-SOURCE. Starting from the Cambridge dialogues, the substitutions have been done taking one slot-value, decided on the basis of a frequency strategy, from the KB-TARGET, only when the original slot-value was present in the KB-SOURCE.

5.5 DST Model and Evaluation Metrics

We compare the substitution strategies presented in Section 5.4 according to their capacity to provide training data for a dialogue state tracker. For all of our experiments we used the dialogue state tracking algorithm TRADE (Wu et al., 2019a). The algorithm is optimized for being used on multi-domains datasets like MultiWOZ, and it has actually been evaluated on this data-set (in its first version) for assessing the performance during the development of the algorithm.
The main evaluation metric is *joint goal accuracy*, largely used for DST, defined as the set of accumulated turn level goals up to a given turn in the dialogue. It indicates the model performance in predicting all slots in a given turn correctly and it is computed by the fraction of turns in a dialogue where all slots in a turn are predicted correctly.

6 Results and Discussion

Table 1 shows a summary of the results that we obtained from our experiments. We started from the original MultiWOZ 2.3 data-set, referred as "Cam" in the Table, based on the Knowledge Base "Cam-KB". We obtained a Joint Goal Accuracy of 0.490 for "Cam", which is aligned with the value reported for TRADE on MultiWOZ 2.3 (Wu et al., 2019b).
The NO ADAPT. experiment aimed at reproducing a zero adaptation situation, training a model on D-SOURCE and testing it on D-TARGET, which in our case was based on a KB-TARGET that differs of around 88% from the KB-SOURCE (see paragraph 4.2). As expected, the value for the unseen slots is much higher (more than 12 times) compared to the original setting. This contributed to a decrease of almost 75% in the Joint Goal Accuracy performance.
The remaining experiments were ran on three different D-TARGET datasets, created on the basis of different strategies, as explained in Section 5.4. RANDOM-D was made by substituting every source slot-value with a slot-value that was taken randomly from a list of all unique slot-values that are present in the dialogue. This means that for every substitution, there was the same chance of picking a very frequent value - such as "Indian food" - than picking a value that occurs only once - such as "south Caribbean spicy food". For this reason the Joint Goal Accuracy is very low, even if significantly better than in the no adaptation setting. A major improvement, however, happens when we use data-sets that are based on an adapted strategies based on the KB.
The RANDOM-KB strategy, in fact, has a Joint Goal Accuracy slightly lower than the original Cam experiment, while the frequency-based strategy even exceeds Cam with an goal accuracy over 50%. The difference in performance between these last two data-sets can be explained by considering that with RANDOM-KB we randomly changed the assignment of the slot-values to be substituted once for every dialogue, which can be beneficial for the model capability of generalizing, but that does not allow it to maximize the

Strategy	Training	Test	KB	Unseen slot-values	Joint Accuracy	KB train adherence	KB test adherence
—-	Cam	Cam-T	Cam-KB	1.22%	0.490	87.99%	91.66%
NO ADAPT.	Cam	Pisa-T	Pisa-KB	15.19%	0.131	39.22%	100%
RANDOM-D	Pisa	Pisa-T	Pisa-KB	2.27%	0.239	42.91%	100%
RANDOM-KB	Pisa	Pisa-T	Pisa-KB	4.78%	0.461	39.25%	100%
FREQ. KB	Pisa	Pisa-T	Pisa-KB	4.8%	0.502	83.96%	100%

Table 1: Results of the *dialogue domain adaptation* experiments. All experiments use the TRADE model. The first row corresponds to the original MultiWOZ 2.3 dataset tested over itself. The second row is the same dataset tested over our Pisa dataset, which has been manually ensured to be perfectly fitted to the KB-TARGET, and which has also been used for testing the other adaptation strategies.

learning for the specific configuration of the test set. On the other hand, FREQUENCY KB has been built with the same strategy of the test set, thus substituting for every slot-value one value from the KB-TARGET that has similar frequency than the value in the KB-SOURCE.

6.1 Linguistic Variability

The different substitutions approaches that we have adopted, radically diverge in the linguistic variability they produce in D-TARGET. If, on one side, for the RANDOM-D dialogues we substitute slot-values that are taken from the dialogue, this way preserving their variability (e.g., typos, synonyms, abbreviations, etc.), on the other side, for the RANDOM-KB and FREQUENCY KB, we always substitute slot-values that are present in the KB, which are only in their normalized version. This way we flatten the linguistic variability, and significantly reduce the total number of unique slot-values in the dialogue. This aspect is clearly highlighted by the different values for the Unseen Ratio. While RANDOM-D, which has a high number of variances for every slot-value (even if not as high as in Cam), produces a percentage of 2.27, RANDOM-KB and FREQUENCY KB show an Unseen Ratio percentage of more than the double. By substituting all possible variances of a slot-value with one unique value, in fact, the portion of slot-values that are not seen in the training, but that instead are present in the test, strongly increases.

7 Conclusion

Domain knowledge in conversational agents is subject to frequent changes, and this leads to the necessity of continuously updating training dialogues in order to keep them consistent with domain knowledge. As collecting conversational dialogues by hand requires a significant effort, approaches for automatically updating are required. In this paper we have proposed *dialogue domain adaptation*, a methodology for operating changes to an initial training dialogue, so that it becomes adherent to a modified domain knowledge. The experiments that we conducted reveal a twofold evidence: they demonstrate that zero adaptation results in a significant loss in DST performance, and they show that simple substitution-based adaptation methods bring instead effective results. Moreover, the experiments on different adaptation methods showed diverse phenomena. While the best performance is obtained using a frequency strategy - which maps the most frequent slot-value of the source domain to the most frequent slot-value of the target domain - a random strategy based on KB values performed slightly worse, and a severe drop in the results occurred when using a random strategy based on dialogue values. Linguistic variability is perhaps an important factor that emphasises this difference, and it will be an interesting topic to be explored in further works.

References

V. Balaraman and B. Magnini. 2021. Domain-aware dialogue state tracker for multi-domain dialogue systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:866–873.

Vevake Balaraman, Seyedmostafa Sheikhalishahi, and Bernardo Magnini. 2021. Recent neural methods on dialogue state tracking for task-oriented dialogue systems: A survey. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGdial 2021, Singapore and Online, July 29-31, 2021*, pages 239–251. Association for Computational Linguistics.

Paul E Black. 2004. Ratcliff/obershelp pattern recognition. *Dictionary of algorithms and data structures*, 17.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *ICLR*. OpenReview.net.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 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. Association for Computational Linguistics.

Yuheng Du, Shereen Oraby, Vittorio Perera, Minmin Shen, Anjali Narayan-Chen, Tagyoung Chung, Anushree Venkatesh, and Dilek Hakkani-Tur. 2020. Schema-guided natural language generation. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 283–295, Dublin, Ireland. Association for Computational Linguistics.

Ting Han, Ximing Liu, Ryuichi Takanobu, Yixin Lian, Chongxuan Huang, Wei Peng, and Minlie Huang. 2020. Multiwoz 2.3: A multi-domain task-oriented dataset enhanced with annotation corrections and co-reference annotation. *arXiv preprint arXiv:2010.05594*.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGdial 2020, 1st virtual meeting, July 1-3, 2020*, pages 35–44. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014a. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014b. The third dialog state tracking challenge. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 324–329.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582, Online. Association for Computational Linguistics.

Samuel Louvan and Bernardo Magnini. 2019. Leveraging non-conversational tasks for low resource slot filling: Does it help? In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, SIGdial 2019, Stockholm, Sweden, September 11-13, 2019*, pages 85–91. Association for Computational Linguistics.

Samuel Louvan and Bernardo Magnini. 2020. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 480–496, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Michaael McTear. 2020. *Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots*. Morgan and Claypool Publishers.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788. Association for Computational Linguistics.

P. J. Price. 1990. Evaluation of spoken language systems: the atis domain. In *HLT*.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019a. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.

Lijun Wu, Yiren Wang, Yingce Xia, Fei Tian, Fei Gao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019b. Depth growing for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5558–5563, Florence, Italy. Association for Computational Linguistics.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.

Boya Yu, Konstantine Arkoudas, and Wael Hamza. 2020. Delexicalized paraphrase generation. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 102–112, Online. International Committee on Computational Linguistics.

Jianguo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S. Yu, Richard Socher, and Caiming Xiong. 2019. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *CoRR*, abs/1910.03544.