Tree-Structured Semantic Encoder with Knowledge Sharing for Domain Adaptation in Natural Language Generation

Bo-Hsiang Tseng,[†] **Paweł Budzianowski,**[†] **Yen-Chen Wu**[†] **Milica Gašić**[‡]
[†]University of Cambridge [‡] Heinrich Heine Univeristy Düsseldorf
bht26@cam.ac.uk, gasic@uni-duesseldorf.de

Abstract

Domain adaptation in natural language generation (NLG) remains challenging because of the high complexity of input semantics across domains and limited data of a target domain. This is particularly the case for dialogue systems, where we want to be able to seamlessly include new domains into the conversation. Therefore, it is crucial for generation models to share knowledge across domains for the effective adaptation from one domain to another. In this study, we exploit a tree-structured semantic encoder to capture the internal structure of complex semantic representations required for multi-domain dialogues in order to facilitate knowledge sharing across domains. In addition, a layer-wise attention mechanism between the tree encoder and the decoder is adopted to further improve the model's capability. The automatic evaluation results show that our model outperforms previous methods in terms of the BLEU score and the slot error rate, in particular when the adaptation data is limited. In subjective evaluation, human judges tend to prefer the sentences generated by our model, rating them more highly on informativeness and naturalness than other systems.

1 Introduction

Building open-domain Spoken Dialogue Systems (SDS) remains challenging. This is partially because of the difficulty of collecting sufficient data for all domains and the high complexity of natural language. Typical SDSs are designed based on a pre-defined ontology (Figure 1) which might cover knowledge spanning over multiple domains and topics (Young et al., 2013).

A crucial component of a Spoken Dialogue System is the Natural Language Generation (NLG) module, which generates the text that is finally presented to the user. NLG is especially challenging when building a multi-domain dialogue systems. Given a semantic representation (SR), the task for NLG is to generate natural language conveying the information encoded in the SR. Typically, an SR is composed of a set of slot-value pairs and a dialogue act consistent with an ontology. A dialogue act represents the intention of the system output and the slots provide domain-dependent information. Figure 2 presents examples of SRs with their corresponding natural language representations in various datasets.

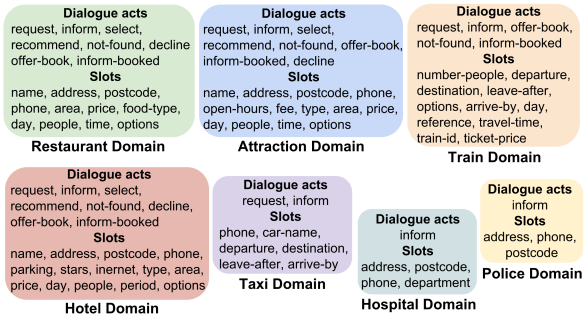


Figure 1: The ontology for multi-domain spoken dialogue systems.

The input semantics has its own hierarchical structure in which there are different sets of slot-value pairs under different dialogue acts across various domains. Modelling the semantic structure might be helpful for sharing information across domains and achieve better performance for domain adaptation. However, prior work encodes semantic representation in a flat way such as using a binary vector (Wen et al., 2015a,b) or using a sequential model such as an LSTM (Dušek and Jurcıcek, 2016; Tran and Nguyen, 2017). In that case, the structure of semantics is not fully captured by these encoding methods. This might limit models' performance especially when adapting to a new domain.

This paper investigates the possibility of lever-

Proceedings of the SIGDial 2019 Conference, pages 155–164
Stockholm, Sweden, 11-13 September 2019.

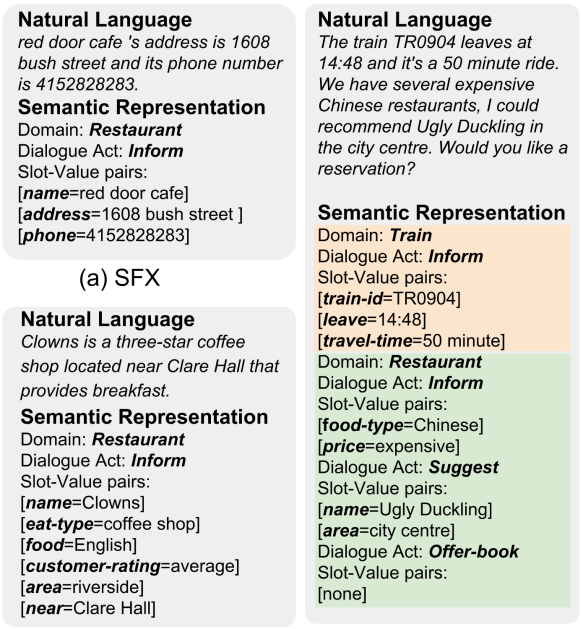


Figure 2: Examples of semantic representations in (a) SFX dataset (Wen et al., 2015b), (b) E2E dataset (Novikova et al., 2017) and (c) MultiWOZ dataset (Budzianowski et al., 2018).

aging the semantic structure for NLG domain adaptation in dialogue systems. We present a generation model with a tree-structured semantic encoder that models the internal structure of the semantic representation to facilitate knowledge sharing across domains. Moreover, we propose a layer-wise attention mechanism to improve the generation performance. We perform experiments on the multi-domain Wizard-of-Oz corpus (MultiWOZ) (Budzianowski et al., 2018) and with human subjects. The results show that the proposed model outperforms previous methods on both automatic metrics and with human evaluation, suggesting that modelling the semantic structure can facilitate domain adaptation. To the best of our knowledge, this work is the first study exploiting the tree LSTM (Tai et al., 2015) to model the input semantics of NLG in spoken dialogue systems.

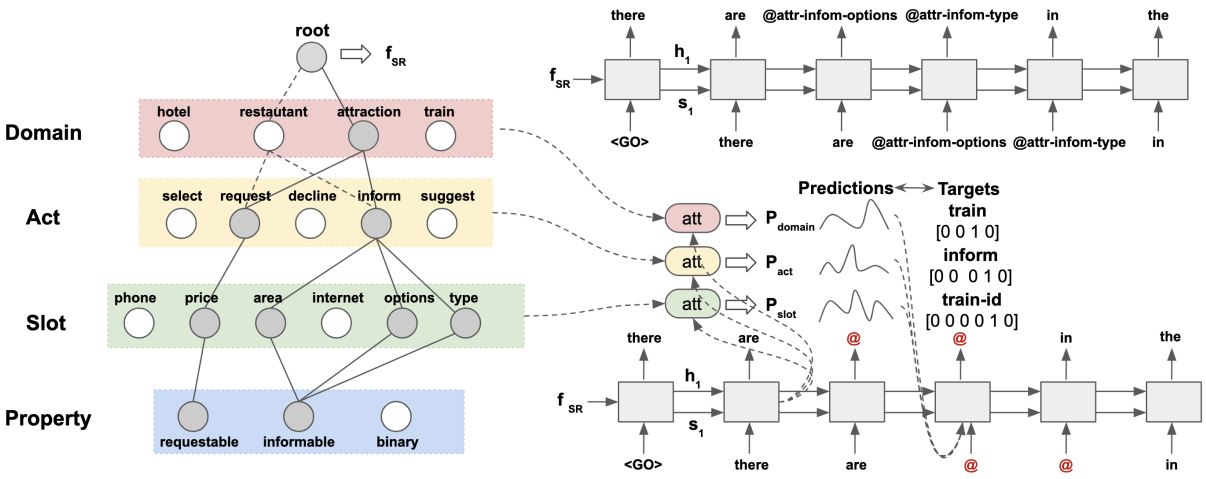
2 Related Work

Recently, recurrent neural network-based NLG models have shown their powerful capability and flexibility compared to traditional approaches that depend on hand-crafted rules in dialogue systems. A key development was the heuristic gate which turns off the slots that are already generated in the output sentence (Wen et al., 2015a). Subsequently, the semantically conditioned LSTM (SCLSTM) (Wen et al., 2015b) was proposed with an extra reading gate in the LSTM cell to let the model automatically learn to control the binary representation of the semantics during generation. The sequence-to-sequence (seq2seq) model (Cho et al., 2014; Sutskever et al., 2014) with attention mechanism (Bahdanau et al., 2014) that has achieved huge success in machine translation has also been applied to the NLG task. In (Dušek and Jurcıcek, 2016) the slot-value pairs in the semantics were treated as a sequence and encoded by LSTM. Based on the seq2seq model, in (Tran et al., 2017; Tran and Nguyen, 2017) the refinement gate was introduced to modify the input words and hidden states in the decoder by considering the attention result. Different training strategies were studied in prior work. The hierarchical decoding method was proposed by considering the linguistic pattern of the generated sentence (Su et al., 2018). The variational-based model was proposed to learn the latent variable from both natural language and semantics (Tseng et al., 2018). Lampouras and Vlachos (2016) proposed to use imitation learning to train NLG models, where the Locally Optimal Learning to Search framework was adopted to train against non-decomposable loss functions.

Domain adaptation has been widely studied in different areas such as machine translation (Koehn and Schroeder, 2007; Foster et al., 2010), part of speech tagging (Blitzer et al., 2006) and dialogue state tracking (Mrkšić et al., 2015) in spoken dialogue systems. In NLG for spoken dialogue systems, the trainable sentence planner proposed in (Walker et al., 2002; Stent et al., 2004) provides the flexibility of adapting to different domains. Subsequently, generators that can tailor user preferences (Walker et al., 2007) or learn their personality traits (Mairesse and Walker, 2008, 2011; Oraby et al., 2018) were proposed. To achieve multi-domain NLG, exploiting the shared knowledge between domains is important to handle unseen semantics. A multi-step procedure to train a multi-domain NLG model was proposed in (Wen et al., 2016). Adversarial learning is used in (Tran and Nguyen, 2018) in which two critics were introduced during model adaptation.

3 Model

Our generation model is composed of two parts: (a) a tree-structured semantic encoder and (b) an LSTM decoder with additional gates. The tree-



(a) Semantically Tree-structured Encoder (b) LSTM-based Decoder

Figure 3: The overview of our generation model. The tree-structured semantic encoder (a) encodes semantic representation to obtain semantic embedding f_{SR}. Each node in the tree denotes a vector representation for that token. Grey node means it is activated during encoding with the corresponding token specified in the semantics. The LSTM-based decoder without layer-wise attention ((b), above) and with layer-wise attention ((b), below) takes f_{SR} as an initial state to generate natural language. The example utterance here is *"there are @attraction-inform-options @attraction-inform-type in the @attraction-inform-area, do you have a price range in mind?"*

structured semantic encoder extracts a semantic embedding from the semantics in a bottom-up fashion. The obtained embedding is then fed into the decoder as a condition to generate natural language with corresponding delexicalised tokens[1]. In addition, we further propose a layer-wise attention mechanism between the tree-structured semantic encoder and the decoder. The proposed attention mechanism further improves the model's ability to generate the correct information when adapting to a new domain with limited data.

3.1 Tree-Structured Semantic Encoder

There exists a hierarchical relationship between dialogue acts and slot-value pairs within various domains. Inspired by the tree-structured LSTM (Tai et al., 2015) that encodes natural language by capturing its syntactic properties, we propose a tree-structured semantic encoder to encode the semantic representation (SR) by exploiting its internal hierarchy.

3.1.1 Tree Hierarchy

Figure 3 (a) illustrates our tree-structured semantic encoder. The hierarchy of the tree represents the ontology with each layer symbolizing a different level of information. At each layer, a node denotes a possible type defined by the ontology. Given an SR, each slot-value pair is associated with a dialogue act (DA) within a domain. This relationship is modelled by the links between different layers in a tree as parents and children. For instance, the node denoting slot `name` is the child of the node denoting DA `suggest` and DA `suggest` is the child of the node representing domain `restaurant`. In addition, a slot can be *requestable*, *informable* or *binary*. Each of them behaves differently in natural language[2]. Each leaf node denotes a property that describes a slot. As a result, given an SR there is a one-to-one mapping between SR and its corresponding tree and a path from the root to a leaf node describes a slot-value pair along with its domain, DA, slot and property of slot information.

3.1.2 Semantic Representation Encoding

Given a tree representing an SR, each node j of the LSTM contains input, forget and output gates i_j, f_j and o_j respectively to obtain its hidden state and memory cell h_j and c_j. With a set of children $C(j)$, the non-leaf node j has two sources of input:

[1]Each value in a natural language utterance is replaced by a delexicalised token in the format `@domain-act-slot`. For instance, the informed restaurant *Golden House* will be replaced by the token `@restaurant-inform-name`. The mapping from values to delexicalised tokens is called delexicalisation. The inverse process is called lexicalisation.

[2]For instance, the utterance with a *requestable* slot `area` might be: *Which part of the city you are looking for?*. The utterance with the *informable* slot `area` might be: *There are several restaurants in the @restaurant-inform-area.*

(a) the token embedding e_j[3] and (b) children states h_k, c_k. The transition equations are as following:

$$\tilde{h}_j = \sum_{k \in C(j)} h_k,$$
$$\tilde{c}_j = \sum_{k \in C(j)} c_k,$$
$$i_j = \sigma(W_E^{(i)} e_j + U_E^{(i)} \tilde{h}_j + b_E^{(i)}),$$
$$f_j = \sigma(W_E^{(f)} e_j + U_E^{(f)} \tilde{h}_j + b_E^{(f)}),$$
$$o_j = \sigma(W_E^{(o)} e_j + U_E^{(o)} \tilde{h}_j + b_E^{(o)}),$$
$$g_j = \tanh(W_E^{(g)} e_j + U_E^{(g)} \tilde{h}_j + b_E^{(g)}),$$
$$c_j = i_j \circ g_j + f_j \circ \tilde{c}_j,$$
$$h_j = o_j \circ \tanh(c_j),$$

where k is the children index, $\tilde{h}_j$ and $\tilde{c}_j$ are the sum of children's hidden states and memory cells respectively.

The semantic embedding is obtained in a bottom-up fashion. Starting from the leaf nodes with their corresponding embeddings, the information is propagated from the property layer through the slot layer, act layer and domain layer to the root. The hidden state at the root is the final semantic embedding f_{SR} for the SR and it will be used to condition the decoder during generation.

During domain adaptation, the model might have seen some semantics in source domain (denoted by dash lines in the tree encoder in Figure 3) that shares a partial tree structure with the semantics in the target domain. For instance, the SR informing about options, type and area in restaurant domain shares partial tree structure with the SR informing about the same information in attraction domain. Modelling semantic structure by the tree encoder benefits knowledge sharing across domains.

3.2 Decoder

Figure 3 (b) presents the LSTM-based decoder with two introduced gates. The representation of the semantics, s_t, is initialised by the semantic embedding f_{SR} and then updated at each time step duration generation. Updating the semantics at each step is crucial to avoiding generating redundant or missing information in the SR. As in standard LSTMs, the transition equations of memory cell c_t are as following:

$$i_t = \sigma(W_D^{(i)} x_t + U_D^{(i)} h_{t-1} + b_D^{(i)}),$$
$$f_t = \sigma(W_D^{(f)} x_t + U_D^{(f)} h_{t-1} + b_D^{(f)}),$$
$$o_t = \sigma(W_D^{(o)} x_t + U_D^{(o)} h_{t-1} + b_D^{(o)}),$$
$$g_t = \tanh(W_D^{(g)} x_t + U_D^{(g)} h_{t-1} + b_D^{(g)}),$$
$$c_t = i_t \circ g_t + f_t \circ c_{t-1}.$$

The two introduced gates, reading gate r_t and writing gate w_t, are responsible for updating the semantic state s_t. The reading gate determines what information should be kept from the semantics at previous time step, while the writing gate decides what new information should be added into the current semantic state:

$$r_t = \sigma(W_D^{(r)} x_t + U_D^{(r)} h_{t-1} + V_D^{(r)} s_{t-1} + b_D^{(r)}),$$
$$w_t = \sigma(W_D^{(w)} x_t + U_D^{(w)} h_{t-1} + V_D^{(w)} s_{t-1} + b_D^{(w)}),$$
$$d_t = \tanh(W_D^{(d)} x_t + U_D^{(d)} h_{t-1} + V_D^{(d)} s_{t-1} + b_D^{(d)}),$$
$$s_t = w_t \circ d_t + r_t \circ s_{t-1}.$$

The hidden state h_t is then defined as the weighted sum of the memory cell and the semantic state with the output gate as weight:

$$h_t = o_t \circ \tanh(c_t) + (1 - o_t) \circ \tanh(s_t).$$

The probability of the word label y_t at each time step t is formed by a applying a softmax classifier that takes the hidden state h_t as input:

$$p(y_t | x_{<t}, f_{SR}) = \text{softmax}(W^{(s)} h_t).$$

The objective function is the standard negative log-likelihood:

$$J(\theta) = -\sum_t \log p(y_t | x_{<t}, f_{SR}). \quad (1)$$

3.3 Layer-wise Attention Mechanism

The semantic embedding obtained from the tree encoder contains high-level information regarding the semantic representation. However, the information in the tree is not fully leveraged during generation. Thanks to the hierarchical structure of a tree encoder with defined meaning for each layer, we can apply an attention mechanism to each layer to let the decoder concentrate on the different levels of information. We expect the decoder to leverage information regarding domain, dialogue act and slot from the hidden states in a tree to influence the generation process.

[3] All the domains, dialogue acts and slots appearing in an SR are viewed as tokens and encoded in the 1-hot vectors. The 1-hot vectors are then passed through an embedding layer to attain the token embeddings as inputs to the nodes.

Whenever the decoder generates the token `@`[4], the semantics s_t is used to drive an attention mechanism with hidden states in the different layers of the tree to obtain distributions over domains $p(d_t)$, dialogue acts $p(a_t)$ and slots $p(s_t)$ respectively:

$$p(d_t|x_{<t}, s_t) = \frac{\exp(\text{score}(s_t, h_d))}{\sum_{d' \in D} \exp(\text{score}(s_t, h_{d'}))},$$

$$p(a_t|x_{<t}, s_t) = \frac{\exp(\text{score}(s_t, h_a))}{\sum_{a' \in A} \exp(\text{score}(s_t, h_{a'}))},$$

$$p(s_t|x_{<t}, s_t) = \frac{\exp(\text{score}(s_t, h_s))}{\sum_{s' \in S} \exp(\text{score}(s_t, h_{s'}))},$$

where h_d, h_a and h_s are the hidden states of domain, dialogue act and slots in the tree encoder. D, A and S are the sets of domains, dialogue acts and slots defined in the ontology respectively. The score function used to calculate the similarity between two vectors is defined as following:

$$\text{score}(f, h) = f^T h.$$

The distributions $p(d_t)$, $p(a_t)$ and $p(s_t)$ are then used to predict domain, dialogue act and slot at time step t by taking the argmax operation to form the delexicalised tokens `@domain-act-slot` back into the generated sentence.

In order to avoid generating redundant or missing information in a given SR, the three predicted distributions are fed into next time step to augment the original input word [5] to condition the model on what information has already been generated.

During training, the error signals between predicted distributions and the true labels for domain, dialogue act and slot are added to the objective function. The objective function for the generation model with layer-wise attention mechanism is defined as following:

$$J_{att}(\theta) = J(\theta) - \sum_{t'} (\log p(d_{t'}|x_{<t'}, s_{t'}) + \log p(a_{t'}|x_{<t'}, s_{t'}) + \log p(s_{t'}|x_{<t'}, s_{t'})),$$

where $J(\theta)$ is the original objective function in equation 1 and t' is the index for the time step where each token `@` is generated.

[4]With the layer-wise attention mechanism, all values in the natural language are replaced by the same delexicalised token `@` instead of the tokens in the format `@domain-act-slot`, and the corresponding information regarding domain, dialogue act and slot will be used as signals to guide the decoder to predict the correct information.

[5]Only at the next time step of generating delexicalised token `@` the input is the concatenation of the word vector x_t and three predicted distributions. In any other time steps, the input is the word vector padded with zeros.

Table 1: The data statistics for each domain.

Domain	Restaurant	Hotel	Attraction	Train	Taxi
Examples	8.5k	6.6k	6.4k	11k	3.4k
Distinct SR	346	378	314	338	47
Dialogue acts	8	8	8	5	2
Slots	11	14	13	11	6

4 Experimental Results

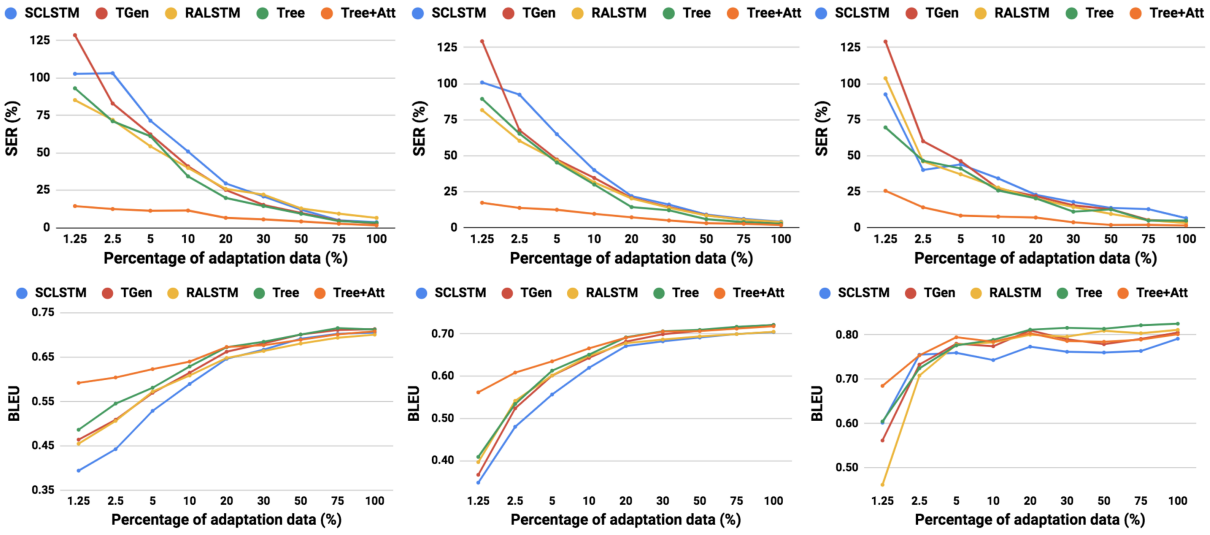
4.1 Dataset

We perform our experiments with the Multi-Domain Wizard-of-Oz (MultiWOZ) dataset (Budzianowski et al., 2018) that is a rich dialogue dataset spanning over 7 domains. There are 10438 dialogues and over $115k$ turns in total. The dataset contains a high level of complexity and naturalness which is suitable for developing multi-domain NLG models. There are multiple utterances in a single turn with an average of 18 words, 1.6 dialogue acts and 2.9 slots per turn. Some turns provide information for more than 1 domain. Comparing with previous NLG datasets which contain only 1 utterance in a turn with 1 dialogue act within 1 domain, the MultiWOZ dataset provides significantly more complexity and makes NLG more challenging. The number of examples, distinct semantic representation (SR) and numbers of dialogue acts and slots are reported in Table 1. The data split for train, dev and test is 3:1:1. The details of the ontology is presented in Figure 1.

4.2 Experimental Setup

The generators are implemented using the Pytorch library (Paszke et al., 2017). Our code is public[6]. The number of hidden units in the LSTMs is 100 with 1 hidden layer. The dropout rate is 0.25 and the Adam optimizer is used. The learning rate is 0.0025 for the models trained from scratch, and 0.001 for the models adapted from one domain to another in adaptation experiments. Beam search is used during decoding with a beam size 10. For automatic metrics, the BLEU scores and the slot error rate (SER) used in (Wen et al., 2015b) are reported. The SER is used to evaluate how accurate a generated sentence is in terms of conveying the desired information in the given semantic representation (SR). The SER is defined as: $(p + q)/N$, where p, q are the numbers of missing and redundant slots

[6]https://github.com/andy194673/TreeEncoder-NLG-Dialogue



(a) Restaurant to Hotel domain (b) Restaurant to Attraction domain (c) Train to Taxi domain

Figure 4: Domain adaptation experiments in three different settings. (a) adapting to hotel from restaurant domain. (b) adapting to attraction from restaurant domain. (c) adapting to taxi from train domain.

in a generated sentence, and N is the number of total slots that a generated sentence should contain. The results are averaged over 10 samples and 5 random initialised seeds. As explained above each delexicalised slot token in an utterance is in the format of `@domain-act-slot`. When calculating the SER, the predicted slot token is correct only if its domain, dialogue act and slot information are all correct. For example, if there is a desired slot `area` under dialogue act `inform` within `restaurant` domain in SR, the model needs to generate the token `@restaurant-inform-area`.

The tree-structured semantic encoder (Tree) and the variant with attention (Tree+Att) are compared against three baselines: (1) the semantically-conditioned LSTM (SCLSTM) that has an extra gate to update the binary vector of the semantic representation (Wen et al., 2015b); (2) TGen that is a seq2seq model with attention mechanism mapping SR into a word sequence (Dušek and Jurcıcek, 2016); (3) a refinement adjustment LSTM (RALSTM) that is an improved seq2seq model with a refinement gate and an adjustment gate in the decoder (Tran and Nguyen, 2017).

As the decoding method is slightly different between our model Tree+Att and baseline models[7], in order to guarantee the optimised baseline systems, we also trained baseline models in the same decoding way as Tree+Att to only predict `@` with three additional classifiers for domain, act and slot prediction. However, baseline models obtains better performance by the original decoding method so we keep that in the following experiments. All the models are optimized by selecting the best one based on the validation set result.

4.3 Automatic Evaluation

In order to examine the models' ability to share knowledge between domains, we performed experiments in three domain adaptation scenarios: (a) adapting to hotel from restaurant domain; (b) adapting to attraction from restaurant domain and (c) adapting to taxi from train domain. The adaptation models were fine-tuned with adaptation data based on the models trained on source domain[8]. The SER results are presented in the first row of Figure 4. Generally, our model without attention (Tree) performs similarly with RALSTM but better than TGen and SCLSTM. With the layer-wise attention mechanism, our model (Tree+Att) improves significantly and performs better than baselines at all different levels of adaptation data amount. Especially when the adaptation data used is only 1.25%, the SER is reduced from above 75% to around 25%. We found that this is because baseline models tend to predict the slots with the wrong dialogue act or in the wrong domain as the

[7]Tree+Att only generates token `@` and reply on attention results to form the complete slot token while baseline models directly generate slot tokens.

[8]All the multi-domain turns are removed in case the model have seen any examples related to target domain before adaptation.

Table 2: Human evaluation for utterance quality in three adaptation settings: Restaurant (Rest.) to Hotel domain; Restaurant to Attraction (Attr.) domain and Train to Taxi domain. Informativeness (Info.) and Naturalness (Nat.) are reported (rating out of 5).

Model	Rest. to Hotel		Rest. to Attr.		Train to Taxi	
	Info.	Nat.	Info.	Nat.	Info.	Nat.
SCLSTM	2.96	3.85	2.81	3.69	3.05	**4.26**
TGen	2.87	3.33	3.00	3.23	3.42	3.90
RALSTM	2.79	3.48	2.91	3.40	3.48	3.15
Tree	3.08	3.54	3.38	3.41	3.81	3.81
Tree+Att	**4.04**	**4.10**	**4.30**	**3.92**	**4.29**	3.78

limited adaptation data makes it difficult to learn the sentence pattern in the target domain. However, with the layer-wise attention mechanism, our model is able to pay attention on the information at different levels in the tree to make the correct predictions. (See more details in section 5 with error analysis and visualisation of attention distributions.) A similar trend can be observed in the BLEU results in the second row of Figure 4.

4.4 Human Evaluation

Because automatic evaluation such as BLEU may not consistently agree with human perception (Stent et al., 2005), we performed human testing via the Amazon Mechanical Turk service. We showed MTurk workers the generated sentences in adaptation experiments with adaptation data from 1.25% to 10% as we focus on the models' performance with limited adaptation data. Five models were compared together by showing, for each model, the 2 sentences with the highest probabilities out of the 10 generated sentences by beam search. The workers were asked to score each sentence from 1 (bad) to 5 (good) in terms of its informativeness and naturalness. The *informativeness* is defined as the degree to which the generated sentence contains all the information specified in the given semantic representation (SR) without conveying extra information and the *naturalness* is defined as whether the sentence is natural like human language. Ipeirotis et al. (2010) pointed out that malicious workers might take advantage of the difficulty of verifying the results and therefore submit answers with low quality. In order to filter out submissions with bad quality, we also asked them to score the ground truth sentence and an artificial sentence containing irrelevant information to the SR. If the worker gave ground truth sentence a low score (< 3) or gave the artificial sentence a high score (> 3) in terms of informativeness, the submission was discarded.

The results pertaining to informativeness and naturalness are reported in Table 2 in three adaptation settings: Restaurant (Rest.) to Hotel domain; Restaurant to Attraction (Attr.) domain and Train to Taxi domain. For informativeness, our models (both Tree+Att & Tree) outperform all baseline models in the different settings. This result is consistent with the slot error rate of the automatic evaluation reported in Figure 4 and indicates that the tree-structured semantic encoder does help the model to produce utterances with the correct information. For naturalness, Tree+Att performs the best in two settings, while SCLSTM performs better when adapting to taxi domain. This might be because SCLSTM is good at generating utterances with simple patterns and the taxi domain is relatively easy due to its low number of combinations of SR[9]. When adapting to more complex domains such as hotel or attraction, our models provide both informative and natural utterances. Table 3 presents example semantic representations with corresponding ground truth sentence and the top-1 utterance generated by each model.

5 Error Analysis and Observation

In order to investigate what type of testing data our model performs better on, we divide all test set into two subsets - *seen* and *unseen*. If the semantics of a testing example appear in the training set, the example is defined as *seen*. Otherwise, the example is marked as *unseen*. Table 4 reports the number of seen and unseen examples and the number of wrong utterances (at least 1 missing or redundant slot) generated by each model with different amount of adaptation data when adapting from restaurant to hotel domain. With more adaptation data, more SRs of testing examples appear in the training set. We observe that our model obtains better generalisation ability for unseen SRs. For instance, with 1.25% adaptation data, Tree+Att generates 134 wrong utterances out of 902 unseen semantics (14.8%). However, the baseline models such as SCLSTM produces 729 wrong sentences out of 902 semantics (80.5%). We hypothesize that our model is more capable of learning sentence patterns from source domain and generate correct content for domain adaptation. For example, when adapting from restaurant to hotel domain (see Table 3 - Hotel column), Tree+Att cor-

[9]There are only 2 dialogue acts and 6 slots in taxi domain.

Table 3: Example semantic representations (SR) with ground truth sentences in two adaptation settings with 1.25% adaptation data and the top-1 sentences generated by each model. Both are adapted from restaurant domain. The slot-value pairs are in bold. Each generated sentence is followed by a brief description to explain if the sentence correctly conveys the information in the SR.

Domain	**Attraction**	**Hotel**
Semantic Representation	Dialogue Act: `Inform` Slot-Value pairs: [Area: **west**] [Options: **five**] [Type: **colleges**] Dialogue Act: `Request` Slot-Value pairs: [Price=?]	Dialogue Act: `Inform` Slot-Value pairs: [Options=**two**] Dialogue Act: `Select` Slot-Value pairs: [type1=**guesthouse**] [type2=**hotel**]
Ground Truth	*there are **five colleges** in the **west**.* *do you mind paying an entrance fee ?*	*i have **two**, would you* *prefer a **guesthouse** or **hotel** ?*
SCLSTM	*what type of place are you looking for ?* (miss 3 slots & request wrong)	*what area would you* *like to stay in ?* (miss 3 slots)
TGen	*there are located in the . do you have a price* *range in mind ?* (miss 3 slots)	*i have found options. would you* *prefer or ?* (miss 3 slots)
RALSTM	*we have **five colleges** in the **west** area . do you* *have an attraction type in mind ?* (request wrong)	*i have **two** options.* *do you have a preference ?* (miss 2 slots)
Tree	*there are **five colleges** in the **west** . do you have an* *area of town you would prefer ?* (request wrong)	*i have found **two** options for you.* *do you have a preference ?* (miss 2 slots)
Tree+Att	*there are **five colleges** in the **west** .* *do you have a price range in mind ?* (correct)	*i have **two** options for you. would* *you prefer **guesthouse** or **hotel** ?* (correct)

Table 4: Error analysis - number of examples in the testing set and the number of wrong generated utterances (at least 1 missing or redundant slot) by each model in different adaptation data scenarios. The testing example is defined as seen if its semantics appears in the training set.

Percentage	**1.25%**		**5%**		**10%**		**50%**	
Testing	**seen**	**unseen**	**seen**	**unseen**	**seen**	**unseen**	**seen**	**unseen**
examples	439	902	858	483	1069	272	1330	11
SCLSTM	248	729	307	412	302	190	111	5
TGen	309	741	176	353	178	168	102	6
Tree+Att	10	134	31	103	60	55	76	3

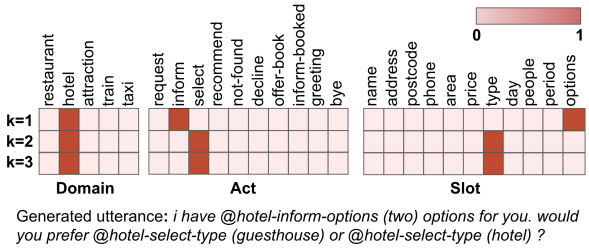


Figure 5: The visualisation of the layer-wise attention distributions over domains, acts and slots at each time step k when slot token is generated and the generated utterance with lexicalised values in the parentheses. The color shades signify the attention weight.

rectly learns to generalize from the training sentence: "*i have two options for you, would you prefer American or Chinese*" in restaurant domain. However, SCLSTM fails to produce a similar sentence pattern.

Figure 5 shows the example of visualisation of layer-wise attention distributions over domains, acts and slots generated by the Tree+Att model. The model is confident of generating the correct slot tokens with the distinct peaks indicated by the dark red color in the attention distributions even though the adaptation data used is simply 1.25%.

6 Conclusion and Future Work

This paper investigates the possibility of leveraging internal structure of input semantics for NLG domain adaptation in dialogue systems. The proposed tree-structured semantic encoder is able to capture the structure of semantic representations and facilitate knowledge sharing across domains. In addition, we have proposed a layer-wise attention mechanism between the tree-structured semantic encoder and the decoder to enhance the performance. Our proposed model was evaluated on the complex multi-domain MultiWOZ dataset. The automatic evaluation results show that our model is more efficient in terms of adaptation data usage and outperforms previous methods by reducing the slot error rate up to 50% when the adaptation data is limited. What is more, human judges rate our model more highly than previous methods. Future work will explore a tree encoder exploiting both semantic representation and context information in end-to-end dialogue systems.

Acknowledgments

Bo-Hsiang Tseng is supported by Cambridge Trust and the Ministry of Education, Taiwan. This work was partly funded by an Alexander von Humboldt Sofja Kovalevskaja grant.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 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.

Kyunghyun Cho, Bart Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

Ondrej Dušek and Filip Jurcıcek. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 45.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 451–459. Association for Computational Linguistics.

Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. 2010. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the second workshop on statistical machine translation*, pages 224–227. Association for Computational Linguistics.

Gerasimos Lampouras and Andreas Vlachos. 2016. Imitation learning for language generation from unaligned data. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1101–1112.

François Mairesse and Marilyn Walker. 2008. Trainable generation of big-five personality styles through data-driven parameter estimation. *Proceedings of ACL-08: HLT*, pages 165–173.

François Mairesse and Marilyn A Walker. 2011. Controlling user perceptions of linguistic style: Trainable generation of personality traits. *Computational Linguistics*, 37(3):455–488.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 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)*, volume 2, pages 794–799.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Saarbrücken, Germany. ArXiv:1706.09254.

Shereen Oraby, Lena Reed, Shubhangi Tandon, TS Sharath, Stephanie Lukin, and Marilyn Walker. 2018. Controlling personality-based stylistic variation with neural natural language generators. *arXiv preprint arXiv:1805.08352*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 341–351. Springer.

Amanda Stent, Rashmi Prasad, and Marilyn Walker. 2004. Trainable sentence planning for complex information presentation in spoken dialog systems. In *Proceedings of the 42nd annual meeting on association for computational linguistics*, page 79. Association for Computational Linguistics.

Shang-Yu Su, Kai-Ling Lo, Yi-Ting Yeh, and Yun-Nung Chen. 2018. Natural language generation by hierarchical decoding with linguistic patterns. *arXiv preprint arXiv:1808.02747*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations

from tree-structured long short-term memory 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 1: Long Papers)*, volume 1, pages 1556–1566.

Van-Khanh Tran and Le-Minh Nguyen. 2017. Natural language generation for spoken dialogue system using rnn encoder-decoder networks. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 442–451.

Van-Khanh Tran and Le-Minh Nguyen. 2018. Adversarial domain adaptation for variational neural language generation in dialogue systems. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1205–1217.

Van-Khanh Tran, Le-Minh Nguyen, and Satoshi Tojo. 2017. Neural-based natural language generation in dialogue using rnn encoder-decoder with semantic aggregation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 231–240.

Bo-Hsiang Tseng, Florian Kreyssig, Paweł Budzianowski, Iñigo Casanueva, Yen-Chen Wu, Stefan Ultes, and Milica Gasic. 2018. Variational cross-domain natural language generation for spoken dialogue systems. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 338–343.

Marilyn A Walker, Owen C Rambow, and Monica Rogati. 2002. Training a sentence planner for spoken dialogue using boosting. *Computer Speech & Language*, 16(3-4):409–433.

Marilyn A Walker, Amanda Stent, François Mairesse, and Rashmi Prasad. 2007. Individual and domain adaptation in sentence planning for dialogue. *Journal of Artificial Intelligence Research*, 30:413–456.

Tsung-Hsien Wen, Milica Gašic, Dongho Kim, Nikola Mrkšic, Pei-Hao Su, David Vandyke, and Steve Young. 2015a. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 275.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Conference on North American Chapter of the Association for Computational Linguistics (NAACL)*. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015b. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.