Dual Inference for Improving Language Understanding and Generation

Shang-Yu Su⋆ **Yung-Sung Chuang**⋆ **Yun-Nung Chen**
National Taiwan University, Taipei, Taiwan
{b05901033,f05921117}@ntu.edu.tw y.v.chen@ieee.org

Abstract

Natural language understanding (NLU) and Natural language generation (NLG) tasks hold a strong dual relationship, where NLU aims at predicting semantic labels based on natural language utterances and NLG does the opposite. The prior work mainly focused on exploiting the duality in model training in order to obtain the models with better performance. However, regarding the fast-growing scale of models in the current NLP area, sometimes we may have difficulty retraining whole NLU and NLG models. To better address the issue, this paper proposes to leverage the duality in the inference stage without the need of retraining. The experiments on three benchmark datasets demonstrate the effectiveness of the proposed method in both NLU and NLG, providing the great potential of practical usage. [1]

1 Introduction

Various tasks, though different in their goals and formations, are usually not independent and yield diverse relationships between each other within each domain. It has been found that many tasks come with a dual form, where we could directly swap the input and the target of a task to formulate into another task. Such structural duality emerges as one of the important relationship for further investigation, which has been utilized in many tasks including machine translation (Wu et al., 2016), speech recognition and synthesis (Tjandra et al., 2017), and so on. Previous work first exploited the duality of the task pairs and proposed supervised (Xia et al., 2017) and unsupervised (reinforcement learning) (He et al., 2016) learning frameworks in machine translation. The recent studies magnified the importance of the duality by revealing exploitation of it could boost the learning for both tasks.

Natural language understanding (NLU) (Tur and De Mori, 2011; Hakkani-Tür et al., 2016) and natural language generation (NLG) (Wen et al., 2015; Su et al., 2018) are two major components in modular conversational systems, where NLU extracts core semantic concepts from the given utterances, and NLG constructs the associated sentences based on the given semantic representations. Su et al. (2019) was the first attempt that leveraged the duality in dialogue modeling and employed the dual supervised learning framework for training NLU and NLG. Furthermore, Su et al. (2020) proposed a joint learning framework that can train two modules seamlessly towards the potential of unsupervised NLU and NLG. Recently, Zhu et al. (2020) proposed a semi-supervised framework to learn NLU with an auxiliary generation model for pseudo-labeling to make use of unlabeled data.

Despite the effectiveness showed by the prior work, they all focused on leveraging the duality in the *training* process to obtain powerful NLU and NLG models. However, there has been little investigation on how to leverage the dual relationship into the inference stage. Considering the fast-growing scale of models in the current NLP area, such as BERT (Devlin et al., 2018) and GPT-3 (Brown et al., 2020), retraining the whole models may be difficult. Due to the constraint, this paper introduces a dual inference framework, which takes the advantage of existing models from two dual tasks without re-training (Xia et al., 2017), to perform inference for each individual task regarding the duality between NLU and NLG. The contributions can be summarized as 3-fold:

- The paper is the first work that proposes a dual inference framework for NLU and NLG to utilize their duality without model re-training.
- The presented framework is flexible for diverse trained models, showing the potential of

[1] The source code and data are available at https://github.com/MiuLab/DuaLUG.

⋆The first two authors contributed to this paper equally.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4930–4936
November 16 - 20, 2020. ©2020 Association for Computational Linguistics

practical applications and broader usage.

- The experiments on diverse benchmark datasets consistently validate the effectiveness of the proposed method.

2 Proposed Dual Inference Framework

With the semantics space $\mathcal{X}$ and the natural language space $\mathcal{Y}$, given n data pairs $\{(x_i, y_i)\}_{i=1}^n$ sampled from the joint space $\mathcal{X} \times \mathcal{Y}$, the goal of NLG is to generate corresponding utterances based on given semantics. In other words, the task is to learn a mapping function $f(x; \theta_{x\to y})$ to transform semantic representations into natural language.

In contrast, the goal of NLU is to capture the core meaning from utterances, finding a function $g(y; \theta_{y\to x})$ to predict semantic representations given natural language utterances. Note that in this paper, the NLU task has two parts: (1) intent prediction and (2) slot filling. Hence, x is defined as a sequence of words ($x = \{x_i\}$), while semantics y can be divided into an intent y^I and a sequence of slot tags $y^S = \{y_i^S\}$, ($y = (y^I, y^S)$). Considering that this paper focuses on the inference stage, diverse strategies can be applied to train these modules. Here we conduct a typical strategy based on maximum likelihood estimation (MLE) of the parameterized conditional distribution by the trainable parameters $\theta_{x\to y}$ and $\theta_{y\to x}$.

2.1 Dual Inference

After obtaining the parameters $\theta_{x\to y}$ and $\theta_{y\to x}$ in the training stage, a normal inference process works as follows:

$$f(x) = \arg\max_{y' \in \mathcal{Y}} \left\{ \log P\left(y' \mid x; \theta_{x\to y}\right) \right\},$$

$$g(y) = \arg\max_{x' \in \mathcal{X}} \left\{ \log P\left(x' \mid y; \theta_{y\to x}\right) \right\},$$

where $P(.)$ represents the probability distribution, and x' and y' stand for model prediction. We can leverage the duality between $f(x)$ and $g(y)$ into the inference processes (Xia et al., 2017). By taking NLG as an example, the core concept of dual inference is to dissemble the normal inference function into two parts: (1) inference based on the forward model $\theta_{x\to y}$ and (2) inference based on the backward model $\theta_{y\to x}$. The inference process can now be rewritten into the following:

$$f(x) \equiv \arg\max_{y' \in \mathcal{Y}} \{\alpha \log P(y' \mid x; \theta_{x\to y}) + (1-\alpha) \log P(y' \mid x; \theta_{y\to x})\}, \quad (1)$$

where α is the adjustable weight for balancing two inference components.

Based on Bayes theorem, the second term in (1) can be expended as follows:

$$\begin{aligned} &\log P(y' \mid x; \theta_{y\to x}) \\ &\quad = \log\left(\frac{P(x \mid y'; \theta_{y\to x}) P(y'; \theta_y)}{P(x; \theta_x)}\right), \\ &\quad = \log P(x \mid y'; \theta_{y\to x}) \\ &\qquad + \log P(y'; \theta_y) - \log P(x; \theta_x), \end{aligned}$$

where θ_x and θ_y are parameters for the marginal distribution of x and y. Finally, the inference process considers not only the forward pass but also the backward model of the dual task. Formally, the dual inference process of NLU and NLG can be written as:

$$\begin{aligned} f(x) \equiv \arg\max_{y' \in \mathcal{Y}} \{&\alpha \log P(y' \mid x; \theta_{x\to y}) \\ &+ (1-\alpha)(\log P(x \mid y'; \theta_{y\to x}) \\ &+ \beta \log P(y'; \theta_y) - \beta \log P(x; \theta_x))\}, \\ g(y) \equiv \arg\max_{x' \in \mathcal{X}} \{&\alpha \log P(x' \mid y; \theta_{y\to x}) \\ &+ (1-\alpha)(\log P(y \mid x'; \theta_{x\to y}) \\ &+ \beta \log P(x'; \theta_x) - \beta \log P(y; \theta_y))\}, \end{aligned}$$

where we introduce an additional weight β to adjust the influence of marginals. The idea behind this inference method is intuitive: the prediction from a model is reliable when the original input can be reconstructed based on it. Note that this framework is flexible for any trained models ($\theta_{x\to y}$ and $\theta_{y\to x}$), and leveraging the duality does not need any model re-training but inference.

2.2 Marginal Distribution Estimation

As derived in the previous section, marginal distributions of semantics $P(x)$ and language $P(y)$ are required in our dual inference method. We follow the prior work for estimating marginals (Su et al., 2019).

Language Model We train an RNN-based language model (Mikolov et al., 2010; Sundermeyer et al., 2012) to estimate the distribution of natural language sentences $P(y)$ by the cross entropy objective.

Masked Prediction of Semantic Labels A semantic frames x contains an intent label and some slot-value pairs; for example, {*Intent: "atis_flight", fromloc.city_name: "kansas city", toloc.city_name:*

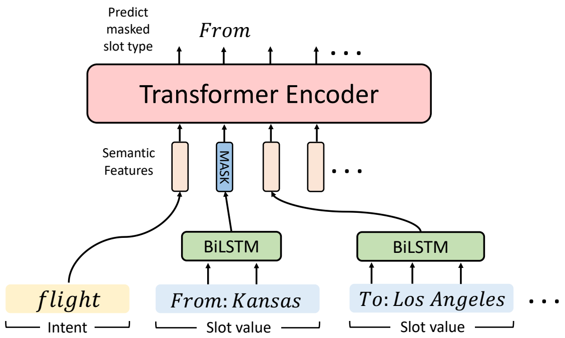

Figure 1: The proposed model for estimating the density of a given semantic frame.

"los angeles", depart_date.month_name: "april ninth"}. A semantic frame is a parallel set of discrete labels which is not suitable to model by autoregressiveness like language modeling. Prior work (Su et al., 2019, 2020) simplified the NLU task and treated semantics as a finite number of labels, and they utilized masked autoencoders (MADE) (Germain et al., 2015) to estimate the joint distribution. However, the slot values can be arbitrary word sequences in the regular NLU setting, so MADE is no longer applicable for benchmark NLU datasets.

Considering the issue about scalability and the parallel nature, we use non-autoregressive masked models (Devlin et al., 2018) to predict the semantic labels instead of MADE. The masked model is a two-layer Transformer (Vaswani et al., 2017) illustrated in Figure 1. We first encode the slot-value pairs using a bidirectional LSTM, where an intent or each slot-value pair has a corresponding encoded feature vector. Subsequently, in each iteration, we mask out some encoded features from the input and use the masked slots or intent as the targets. When estimating the density of a given semantic frame, we mask out a random input semantic feature three times and use the cumulative product of probability as the marginal distribution to predict the masked slot.

3 Experiments

To evaluate the proposed methods on a fair basis, we take two simple GRU-based models for both NLU and NLG, and the details can be found in Appendix D. For NLU, accuracy and F1 measure are reported for intent prediction and slot filling respectively, while for NLG, the evaluation metrics include BLEU and ROUGE-(1, 2, L) scores with multiple references. The hyperparameters and other training settings are reported in Appendix A.

Dataset	#Train	#Test	Vocab	#Intent	#Slot
SNIPS	13084	700	9076	7	72
ATIS	4478	893	1428	25	130
E2E NLG	42063	4693	3210	-	16

Table 1: The statistics of the datasets.

3.1 Datasets

The benchmark datasets conducted in our experiments are listed as follows:

- **ATIS** (Hemphill et al., 1990): an NLU dataset containing audio recordings of people making flight reservations. It has sentence-level intents and word-level slot tags.
- **SNIPS** (Coucke et al., 2018): an NLU dataset focusing on evaluating voice assistants for multiple domains, which has sentence-level intents and word-level slot tags.
- **E2E NLG** (Novikova et al., 2017): an NLG dataset in the restaurant domain, where each meaning representation has up to 5 references in natural language and no intent labels.

We use the open-sourced *Tokenizers*[2] package for preprocessing with byte-pair-encoding (BPE) (Sennrich et al., 2016). The details of datasets are shown in Table 1, where the vocabulary size is based on BPE subwords. We augment NLU data for NLG usage and NLG data for NLU usage, and the augmentation strategy are detailed in Appendix C.

3.2 Results and Analysis

Three baselines are performed for each dataset: (1) Iterative Baseline: simply training NLU and NLG iteratively, (2) Dual Supervised Learning (Su et al., 2019), and (3) Joint Baseline: the output from one model is sent to another as in Su et al. (2020)[3]. In joint baselines, the outputs of NLU are intent and IOB-slot tags, whose modalities are different from the NLG input, so a simple matching method is performed (see Appendix C).

For each trained baseline, the proposed dual inference technique is applied. The inference details are reported in Appendix B. We try two different approaches of searching inference parameters (α and β):

[2] https://github.com/huggingface/tokenizers

[3] In our NLU setting, it is infeasible to flow the gradients though the loop for training the models jointly.

Learning Scheme	**NLU**		**NLG**			
	Accuracy	**F1**	**BLEU**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
ATIS						
Iterative Baseline	84.10	94.26	16.08	35.10	11.94	33.73
+ DualInf(α=0.5, β=0.5)	85.07	93.84	**17.38**	**36.40**	**13.33**	**35.09**
+ DualInf(α^*, β^*)	**85.57**	**94.63**	16.06	35.19	11.93	33.75
Dual Supervised Learning	82.98	94.85	16.98	38.83	15.56	37.50
+ DualInf(α=0.5, β=0.5)	83.68	94.89	**20.69**	**40.62**	**17.72**	**39.31**
+ DualInf(α^*, β^*)	**84.26**	**95.32**	17.05	38.82	15.57	37.42
Joint Baseline	81.44	90.37	21.00	39.70	18.91	38.48
+ DualInf(α=0.5, β=0.5)	81.21	88.42	**22.60**	**41.19**	**20.24**	**39.88**
+ DualInf(α^*, β^*)	**85.88**	**90.66**	20.67	39.41	18.68	38.16
SNIPS						
Iterative Baseline	96.58	96.67	15.49	34.32	13.75	33.26
+ DualInf(α=0.5, β=0.5)	**97.07**	**96.70**	**16.90**	**35.43**	**15.18**	**34.41**
+ DualInf(α^*, β^*)	96.88	96.76	15.46	34.21	13.78	33.14
Dual Supervised Learning	96.83	96.71	15.96	36.69	15.39	35.73
+ DualInf(α=0.5, β=0.5)	**96.88**	**96.80**	**18.07**	**37.63**	**16.75**	**36.67**
+ DualInf(α^*, β^*)	95.34	96.68	16.08	36.97	15.62	36.04
Joint Baseline	97.18	94.57	17.15	36.32	15.68	35.36
+ DualInf(α=0.5, β=0.5)	**97.27**	95.59	**18.56**	37.87	17.25	36.90
+ DualInf(α^*, β^*)	95.54	**96.06**	18.26	**38.16**	**17.70**	**37.40**
E2E NLG						
Iterative Baseline	-	94.25	24.98	44.60	19.40	37.99
+ DualInf(α=0.5, β=0.5)	-	94.29	25.34	44.82	19.73	38.23
+ DualInf(α^*, β^*)	-	**94.55**	**25.35**	**44.87**	**19.74**	**38.30**
Dual Supervised Learning	-	94.49	24.73	45.74	19.60	39.91
+ DualInf(α=0.5, β=0.5)	-	**94.53**	**25.40**	**46.25**	**20.18**	**40.42**
+ DualInf(α^*, β^*)	-	94.47	24.67	45.71	19.56	39.88
Joint Baseline	-	93.51	25.19	44.80	19.59	38.20
+ DualInf(α=0.5, β=0.5)	-	93.43	**25.57**	45.11	**19.90**	38.56
+ DualInf(α^*, β^*)	-	**93.88**	25.54	**45.17**	19.89	**38.61**

Table 2: For NLU, accuracy and F1 measure are reported for intent prediction and slot filling respectively. The NLG performance is reported by BLEU, ROUGE-1, ROUGE-2, and ROUGE-L of models (%). All reported numbers are averaged over three different runs.

- DualInf(α=0.5, β=0.5): simply uses α=0.5 and β=0.5 to balance the effect of backward inference and the influence of the marginal distributions.

- DualInf(α^*, β^*): uses the best parameters α=α^* and β=β^* searched by using validation set for intent prediction, slot filling, language generation individually. The parameters α and β ranged from 0.0 to 1.0, with a step of 0.1; hence for each task, there are 121 pairs of (α, β).

The results are shown in Table 2. For ATIS, all NLU models achieve the best performance by selecting the parameters for intent prediction and slot filling individually. For NLG, the models with (α=0.5, β=0.5) outperform the baselines and the ones with (α^*, β^*), probably because of the discrepancy between the validation set and the test set. In the results of SNIPS, for the models mainly trained by standard supervised learning (iterative baseline and dual supervised learning), the proposed method with (α=0.5, β=0.5) outperform the others in both NLU and NLG. However, the model trained with the connection between NLU and NLG behaves different, which performs best on slot F-1 and ROUGE with (α^*, β^*) and performs best on intent accuracy and ROUGE with (α=0.5, β=0.5).

For E2E NLG, the results show a similar trend as ATIS, better NLU results with (α^*, β^*) in NLU and better NLG performance with (α=0.5, β=0.5).

In summary, the proposed dual inference technique can consistently improve the performance of NLU and NLG models trained by different learning algorithms, showing its generalization to multiple datasets/domains and flexibility of diverse training baselines. Furthermore, for the models learned by standard supervised learning, simply picking the inference parameters (α=0.5, β=0.5) would possibly provide improvement on performance.

4 Conclusion

This paper introduces a dual inference framework for NLU and NLG, enabling us to leverage the duality between the tasks without re-training the large-scale models. The benchmark experiments demonstrate the effectiveness of the proposed dual inference approach for both NLU and NLG trained by different learning algorithms even without sophisticated parameter search on different datasets, showing the great potential of future usage.

Acknowledgments

We thank reviewers for their insightful comments. This work was financially supported from the Young Scholar Fellowship Program by Ministry of Science and Technology (MOST) in Taiwan, under Grant 109-2636-E-002-026.

References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. 2015. Made: Masked autoencoder for distribution estimation. In *International Conference on Machine Learning*, pages 881–889.

Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Proceedings of INTERSPEECH*, pages 715–719.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990.*

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Shang-Yu Su, Chao-Wei Huang, and Yun-Nung Chen. 2019. Dual supervised learning for natural language understanding and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5472–5477.

Shang-Yu Su, Chao-Wei Huang, and Yun-Nung Chen. 2020. Towards unsupervised language understanding and generation by joint dual learning. In *Proceedings of the 58th Annual Meeting of the 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. In *Proceedings of The 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. 2017. Listening while speaking: Speech chain by deep learning. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 301–308. IEEE.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yingce Xia, Tao Qin, Wei Chen, Jiang Bian, Nenghai Yu, and Tie-Yan Liu. 2017. Dual supervised learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3789–3798. JMLR. org.

Su Zhu, Ruisheng Cao, and Kai Yu. 2020. Dual learning for semi-supervised natural language understanding. *arXiv preprint arXiv:2004.12299*.

A Training Details

In all experiments, we use mini-batch Adam as the optimizer with each batch of 48 examples on Nvidia Tesla V100. 10 training epochs were performed without early stop, the hidden size of network layers is 200, and word embedding is of size 50. The ratio of teacher forcing is set to 0.9.

B Inference Details

During inference, we use beam search with beam size equal to 20. When applying dual inference, we use beam search to decode 20 possible hypotheses with the primal model (e.g. NLG). Then, we take the dual model (e.g. NLU) and the marginal models to compute the probabilities of these hypotheses in the opposite direction. Finally, we re-rank the hypotheses using the probabilities in both directions (e.g. NLG and NLU) and select the top-1 ranked hypothesis.

To make the NLU model be able to decode different hypotheses, we need to use the auto-regressive architecture for slot filling, as described in Appendix D.

C Data Augmentation

NLU → NLG As described in 3.2, the modality of the NLU outputs (an intent and a sequence of IOB-slot tags) are different from the modality of the NLG inputs (semantic frame containing intent (if applicable) and slot-value pairs). Therefore, we propose a matching method: for each word, the NLU model will predict an IOB tag ∈ {O, B-slot, I-slot}, we simply drop the I- and B- and aggregate all the words with the same slot then combine it with the predicted intent.

For example, if given the word sequence:

[*which, flights, travel, from, kansas,*
city, to, los, angeles, on, april, ninth],

the NLU predicts the IOB-slot sequence:

[O, O, O, O, B-fromloc.city_name,
I-fromloc.city_name,
O, B-toloc.city_name, I-toloc.city_name, O,
B-depart_date.month_name,
B-depart_date.day_number]

and a corresponding intent ”atis_flight”, we transform the sequences into a semantic frame:

{intent[atis_flight],
fromloc.city_name[kansas city],
toloc.city_name[los angelos],
depart_date.month_name[april ninth]}.

The constructed semantic frames can then be used as the NLG input.

NLG → NLU The NLG dataset (E2E NLG) is augmented based on IOB schema and direct matching. For example, a semantic frame with the slot-value pairs:

{name[Bibimbap House], food[English],
priceRange[moderate], area[riverside],
near[Clare Hall]}

corresponds to the target sentence “*Bibimbap House is a moderately priced restaurant who's main cuisine is English food. You will find this local gem near Clare Hall in the Riverside area.*”. The produced IOB slot data would be

[Bibimbap:B-Name, House:I-Name is:O a:O
moderately:B-PriceRange, priced:I-PriceRange,
restaurant:O, who's:O, main:O, cuisine:O, is:O,
English:B-Food food:O. You:O, will:O, find:O,
this:O, local:O, gem:O, near:B-Near,
Clare:I-Near, Hall:I-Near, in:O, the:O,
Riverside:B-Area, area:I-Area].

D Model Structure

For NLU, the model is a simple GRU (Cho et al., 2014) with a word and last output as input at each timestep i and a linear layer at the end for intent prediction based on the final hidden state:

$$o_i = \mathrm{GRU}([w_i, o_{i-1}]).$$

The model for NLG is almost the same but with an additional encoder for encoding semantic frames, where slot-value pairs are encoded into semantic vectors for basic attention, the mean-pooled semantic vector is used as initial state. We borrow the encoder structure in Zhu et al. (2020) for our experiments. At each timestep i, the last predicted word and the aggregated semantic vector from attention are used as the input:

$$o_i = \mathrm{GRU}([h_i^{Attn}, o_{i-1}] \mid h_{mean}).$$