Proceedings of ICSP2022

Transformer-based Natural Language Understanding and Generation

Feng Zhang[1,2],Gaoyun An[1,2*],Qiuqi Ruan[1,2]
[1]Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China
[2]Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China
22120336@bjtu.edu.cn, gyan@bjtu.edu.cn, qqruan@bjtu.edu.cn

2022 16th IEEE International Conference on Signal Processing (ICSP) | 978-1-6654-6056-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICSP56322.2022.9965301

***Abstract*—Facilitating the sharing of information between the two complementary tasks of Natural Language Understanding(NLU) and Natural Language Generation(NLG) is crucial to the study of Natural Language Processing(NLP). NLU extracts the core semantics from a given utterance, while NLG, in contrast, aims to construct the corresponding sentence based on the given semantics. However, model training for both research topics relies on manually annotated data, but the complexity of the annotation process involved makes it costly to acquire manually annotated data on a large scale. Also, in the existing research, few scholars have treated NLU and NLG as dual tasks. Indeed, both NLG and NLU can be approached as translation problems: NLU translates natural language into formal representations, while NLG converts formal representations into natural language. In this paper, we propose a Transformer-based Natural Language Understanding and Generation (T-NLU&G) model that jointly model NLU and NLG by introducing a shared latent variable. The model can help us explore the intrinsic connection between the natural language space and the formal representation space, and use this latent variable to facilitate information sharing between the two spaces. Experiment shows that our model achieves performance gains on both the E2E dataset and the Weather dataset, validates the feasibility and effectiveness of performance gains for the respective tasks via the T-NLU&G model, and is competitive with current state-of-the-art methods.**

***Index Terms*—natural language understanding,natural language generation; transformer, dual relationship**

I. Introduction

In the 21st century, when Internet technology is booming, people are relying more and more on the Internet. Posting and accessing the information on the Internet has become the main way of communicating with others. How to quickly extract the information people want from the dizzying amount of textual data, improve the efficiency of accessing information and infer the actual needs of users has become a pressing problem for researchers. In recent decades, benefit from the emergence of various data sets and the increase in computer computing power, the development of deep learning techniques has laid a good foundation for solving such problems and has achieved significant results in many fields, the representative one of which is natural language processing. NLU and NLG have been well studied as two separate tasks, but the dual relationship that exists between NLU and NLG cannot be ignored.

In order to facilitate information sharing between the NLU and NLG and to make use of the smaller data sample. We hypothesise a latent variable to jointly model NLU and NLG, through which the intrinsic connection between natural language and formal representation space can be investigated to benefit the performance of both tasks. This paper is based on Transformer to jointly model NLU and NLG as two dyadic tasks to deal with the mutual generation of natural language and computer formal language, ultimately achieving the effect of simultaneously improving the translation accuracy of both sides and being competitive on the E2E dataset with planar slot-value pairs and the Weather dataset with tree-structured formal representations.

In summary, our contributions are three-fold:

- Treating both NLU and NLG as translation tasks, the T-NLU&G is proposed to jointly model NLU and NLG to facilitate information sharing between the two parties and improve the performance of their respective tasks by introducing latent variables.
- The T-NLU&G is based on Transformer modeling [1] and the two-layer Transformer is chosen as the Encoder, and the experimental results outperform the choice of LSTM [2] as the Encoder.
- The T-NLU&G achieves the state-of-the-art performance on two challenging E2E [3] and Weather datasets [4].

II. RELATED WORK

A. NLU and NLG

NLU and NLG as separate tasks have been well studied. For example by considering NLU as a classification problem [5], and approaches to NLG vary from pipelined methods subsuming content planning and surface realisation [6] to more recent end-to-end modelling approaches [7], [8]. However, the success of both tasks is based on natural language and formal representation data pairs that require extensive manual annotation, and the complexity of the annotations involved makes the collection of these data costly.

B. Joint modeling of NLU and NLG

Inspired by [9], the joint modeling of both NLU and NLG as a translation problem achieved a performance improvement in each, but they still did not escape from the reliance on LSTM for NLP tasks. the introduction of the Transformer [1] marked a new stage in the development of NLP, which is a deep learning model based entirely on the self-attention, as it is suitable for parallelized computation, and the complexity of its model leads to higher accuracy and performance than the previously popular Recurrent Neural Networks(RNN) [10]. We use a two-layer Transformers structure Fig.1 as the model encoder to achieve results that outperform [9] on the E2E and Weather datasets.

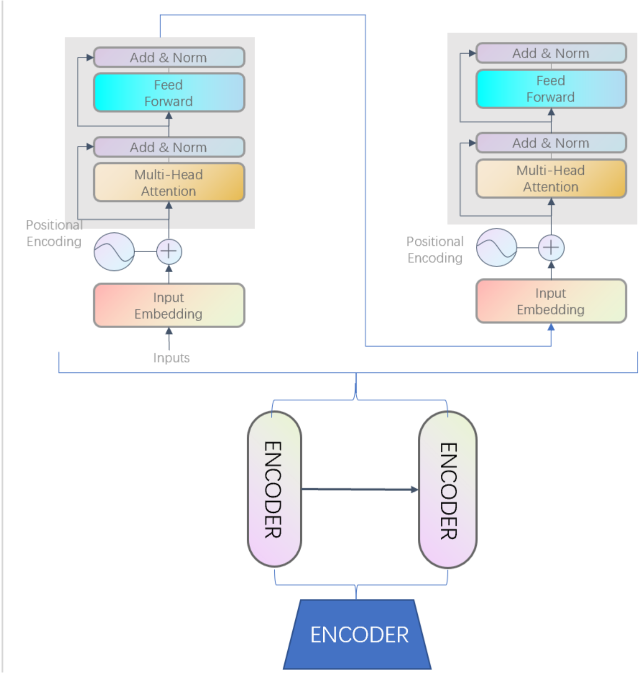

Fig. 1. The Encoder for the T-NLUG model is made up of two-layer of encoder for the Transformer.

III. Proposed Method

The key to the T-NLU&G is to convey information through the abstract latent variable z. The potential variable z can then help the natural language x to generate the formal representation y (or the formal representation y to generate the natural language x), and the whole process is NLU (or NLG).

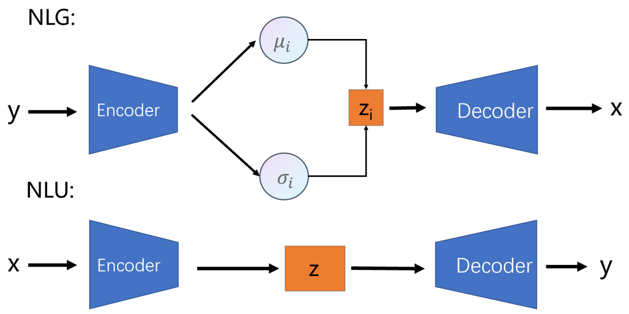

Fig. 2. The framework of the T-NLU&G model, where the encoder contains two layers of Transformer, and the decoder uses LSTM.x: natural language;y: formal representation;z: latent variable.

A. Model framework

Fig.2 shows the framework diagram of T-NLU&G.

NLG: The whole process is to first use y to infer z, and then use z and y to generate x. The steps within Encoder are to first encode y using a two-layer Transformer encoder, and after encoding, get a series of hidden vectors $\overline{\boldsymbol{h}}$ after averaging pooling operations, in order to to obtain the mean $\boldsymbol{\mu}_{y,z}$ and standard deviation $\boldsymbol{\sigma}_{y,z}$ of the joint y, z distribution, $\overline{\boldsymbol{h}}$, is then passed through two layers of feedforward neural networks. For $q(z|y)$ we choose a Gaussian distribution and Use NLG's Encoder to calculate the mean and variance of a Gaussian distribution.

$$\overline{\boldsymbol{h}} = Transformer - Encoder(y) \tag{1}$$

$$\boldsymbol{\mu}_{y,z} = \boldsymbol{W}_{\mu}\overline{\boldsymbol{h}} + \boldsymbol{b}_{\mu} \tag{2}$$

$$\boldsymbol{\sigma}_{y,z} = \boldsymbol{W}_{\sigma}\overline{\boldsymbol{h}} + \boldsymbol{b}_{\sigma} \tag{3}$$

where W and b represent the weights and biases of the neural network.

The final step is the Decoder decoding process using z and y to generate the natural language x. We first chose the Transformer structure for the decoder, but after several experiments, we found that the LSTM was a better choice for the decoder, so we finally used the LSTM with the attention mechanism as the decoder [11] and then passed a Fully-connected layer and Softmax processing were used to obtain the final output probability distribution of the generated words:

$$\boldsymbol{q}_i = attention(\boldsymbol{g}_i^x, \boldsymbol{H}) \tag{4}$$

$$p(x_i) = softmax(\boldsymbol{W_v}[\boldsymbol{q}_i \oplus \boldsymbol{g}_i^x \oplus \boldsymbol{z}] + \boldsymbol{b}_v) \tag{5}$$

where⊕ represents addition, $\boldsymbol{g}_i^x$ is the corresponding decoder hidden state, **H** is the hidden vector obtained by the encoderthe potential vector **z** is sampled from the approximate posterior using the reparameterization technique [12] and $p(x_i)$ is the output probability distribution in step i.

NLU: The process is roughly the opposite of NLG. The main difference is that the meaning of the natural language x is ambiguous and polysemous, whereas the formal representation y is precisely defined. In other words, multiple utterances can correspond to a single formal representation at the same time. To highlight the fact that the formal representation y output from NLU has little variability, the potential vector z is used directly as the mean vector $\boldsymbol{\mu}_{x,z}$ during decoding, rather than sampling from $q(z|x)$ as in Eqs.1-3. The remaining steps are identical to Eqs.4-5 in NLU. After obtaining the potential vector z, the NLU decoder was used to predict the formal representation y from the potential variable z and the natural language x. Since formal representations can be classified according to their structure, we considered two cases in dialogue systems. In the first case, the Weather dataset is used. y is a representation in the form of a tree structure [13]. Then, a linearised sequence y is generated using an LSTM decoder through the standard Attention mechanism. In the second case, we have selected the E2E dataset and y is the form of a set of slot-value pairs. z is used to get the final probability distribution of the slots:

$$p(y_s) = softmax(w_s z + b_s) \tag{6}$$

where $p(y_s)$ is the distribution of predicted values for slot s.

B. Optimize the T-NLU&G model

In this section, we illustrate how the T-NLU&G can be trained in a semi-supervised way when we only have a small fraction of the annotated data.

Optimised data pairs: Given a set of data pairs $(\star, \bullet)$,the optimization objective is to maximize the log-likelihood of the joint probability p$(\star, \bullet)$,so integrate out the potential variable z between $\star$ and $\bullet$

$$logp(\star, \bullet) = log \int_z p(\star, \bullet, z) \tag{7}$$

The standard variational autoencoder is used to derive a target based on a variational lower bound:

$$\begin{aligned}\mathcal{L}_{\star,\bullet} &= \mathbb{E}_{q(\star|\bullet)} logp(\bullet|z,\star) + \mathbb{E}_{q(\star|\bullet)} logp(\star|z,\bullet) \\ &- KL[q(z|\star)||p(z)]\end{aligned} \tag{8}$$

When ($\star$,$\bullet$) is (x,y)(or(y,x)), the first term is the NLU(or NLG) model; the second term is a reconstruction of the natural language x(or formal representation y); the last term represents the KL scatter of the prior and posterior probabilities of the hidden variable z, also known as the relative entropy.

Optimisation of one-sided data: When we get data $\star$ or $\bullet$ that has not been manually annotated,the optimization objective of the model is the logarithm of the marginal likelihood p($\star$) or p($\bullet$).

$$logp(\star) = log \int_\bullet \int_z p(\star, \bullet, z) \tag{9}$$

In the case of only unannotated $\star$, neither z nor $\bullet$ is inferred, and so the objective is built from the variational lower bound on the margins.

$$\mathcal{L}_\star = \mathbb{E}_{q(\bullet|z,\star)} \mathbb{E}_{q(z|\star)} logp(\star|z,\bullet) - KL[q(z|\star)||p(z)] \tag{10}$$

When $\star$ is the natural language x($\bullet$ is the formal representation y), the first term is the autoencoder reconstruction of x using the NLU-NLG path, and the second term is the KL scatter.When $\star$ is the foamal representation y($\bullet$ is the natuarl language x), the first term is an autoencoder reconstruction of y using the NLU-NLG path.

C. Training model

T-NLU&G$_1$: When we have a series of annotated data pairs (x,y), not only can the T-NLU&G model be jointly optimized in a supervised manner, but also the separate unannotated x and y can be used to optimize NLU and NLG individually.The loss function:

$$\mathcal{L}_1 = \sum_{(x,y)\sim(X,Y)} (\mathcal{L}_{x,y} + \mathcal{L}_{y,x}) + \sum_{(x,y)\sim(X,Y)} (\mathcal{L}_x + \mathcal{L}_y) \tag{11}$$

T-NLU&G$_2$: The data can be used to optimize a semi-supervised model when the additional data x or y that have not been manually annotated is available.The loss function:

$$\mathcal{L}_2 = \sum_{(x,y)\sim(X,Y)} (\mathcal{L}_{x,y} + \mathcal{L}_{y,x}) + \sum_{x\sim X} \mathcal{L}_x + \sum_{y\sim Y} \mathcal{L}_y \tag{12}$$

IV. EXPERIMENTS

A. Datasets

Our proposed model is experimented on the E2E dataset [2] with planar slot-value pairs and the Weather dataset [3] with a tree-structured representation, using different proportions of manually annotated training data (5%, 10%, 25%, 50%, 100%) to evaluate the model).

B. Implementation Details

The encoder is a two-layer Transformer, the number of LSTM cells in the decoder is set to 300, the dimensionality of the potential space is 150. the learning rate of the optimizer Adam [15] is 1e-3. The models were all subjected to multiple experiments and the average of the NLU and NLG results was used as the final result of the model.

C. Evaluations

Decoupled: Supervised training of NLG and NLU models alone.

Augmentation: Fine-tune the pre-trained model by adding unannotated data [16] to the Decoupled model.

JUG: [9] used a joint modeling NLU and NLG model, but the Encoder was chosen as the LSTM.

Step 1+2: [17]Two sets of models, *teather* (NLU, NLG and Autoencoder) and *student* (NLU and NLU), were pre-trained for weak and clean data respectively, and the parameters of the *student* model were updated with the step size of each random gradient iteration of the *teather* model, Fine-tune the *student* model parameters by combining the clean and weak datasets.

T-NLU&G$_1$: According to Eqs.11, only the annotated data were used to jointly optimise the NLU, NLG and Autoencoder.

T-NLU&G$_2$: According to Eqs.12, both annotated and unannotated data were used in a semi-supervised manner to jointly optimise NLU, NLG and Autoencoder.

From the TABLE I-IV, we can see that the **T-NLU&G$_1$** achieves better results than when modelling NLU and NLG alone, and outperforms model **Step1+2** overall. when adding additional unannotated data, our model **T-NLU&G$_2$** achieves further performance improvements and significantly all other models, especially at low proportions of annotated data, where the difference between T-NLUG2 and The difference between the **Decoupled** model is greater

V. CONCLUSION

We propose a model for jointly modelling NLU and NLG that facilitates information sharing between the NLU and NLG by introducing a latent variable between natural language and formal representation. Experiments demonstrate that the model achieves performance gains on the E2E and Weather datasets and that easily accessible unannotated data can be used to train

TABLE I
NLU RESULTS ON THE E2E DATASET. THE TABLE SHOWS THE RESULTS IN TERMS OF JOINT ACCURACY (%) [14] AND F1 SCORES (IN PARENTHESES) FOR DIFFERENT PERCENTAGES OF THE ANNOTATED TRAINING DATA.

Model/Data	**5%**	**10%**	**20%**	**50%**	**100%**
Decoupled	52.77(0.874)	62.23(0.902)	69.37(0.924)	73.68(0.935)	76.12(0.942)
Augmentation	54.71(0.878)	62.54(0.902)	68.91(0.922)	73.84(0.935)	-
JUG$_{semi}$	68.09(0.921)	**70.33(0.925)**	73.79(0.935)	75.46(0.939)	-
Step1+2	56.33	-	-	72.45	-
T-NLU&G$_1$	60.26(0.897)	66.22(91.17)	72.27(0.922)	74.93(0.934)	**78.38(0.945)**
T-NLU&G$_2$	**68.31(0.915)**	70.72(0.924)	**74.35(0.935)**	**76.93(0.941)**	-

TABLE II
NLG RESULTS ON THE E2E DATASET. THE TABLE SHOWS THE RESULTS FOR BLEU-4 AND SEMANTIC ACCURACY (%) (IN PARENTHESES) FOR DIFFERENT PERCENTAGES OF THE ANNOTATED TRAINING DATA.

Model/Data	**5%**	**10%**	**20%**	**50%**	**100%**
Decoupled	0.693(83.74)	0.723(87.33)	0.784(92.52)	0.793(94.91)	0.813(96.98)
Augmentation	0.747(84.79)	0.770(90.13)	0.806(94.06)	0.815(96.04)	-
JUG$_{semi}$	**0.814(90.47)**	**0.792(94.76)**	0.819(95.59)	0.827(98.42)	-
Step 1+2	0.775	-	-	0.822	-
T-NLU&G$_1$	0.752(85.59)	0.749(87.03)	0.763(92.06)	0.784(0.973)	**0.831(98.77)**
T-NLU&G$_2$	0.809(91.18)	0.779(90.59)	**0.822(95.97)**	**0.826(98.50)**	-

TABLE III
NLU MATCHING ACCURACY (%) ON THE WEATHER DATASET.

Model/Data	**5%**	**10%**	**20%**	**50%**	**100%**
Decoupled	73.46	80.85	86.00	88.45	**90.68**
Augmentation	74.77	79.84	86.24	88.69	-
JUGsemi	79.19	83.22	87.46	**89.17**	-
Step 1+2	80.36	-	-	87.77	-
T-NLU&G1	72.51	78.56	84.84	87.86	88.63
T-NLU&G2	**82.12**	**84.56**	**87.66**	88.85	-

TABLE IV
NLG'S BLEU RESULTS ON THE WEATHER DATASET.

Model/Data	**5%**	**10%**	**20%**	**50%**	**100%**
Decoupled	0.632	0.667	0.703	0.719	**0.725**
Augmentation	0.635	0.677	0.703	0.727	-
JUGsemi	0.670	0.701	0.725	**0.733**	-
Step 1+2	0.672	-	-	0.717	-
T-NLU&G1	0.627	0.671	0.711	0.721	0.722
T-NLU&G2	**0.678**	**0.716**	**0.730**	0.727	-

an optimized model, improving the utilization of the data and avoiding the need to acquire expensive manually annotated data.

ACKNOWLEDGMENT

This work was supported partly by the National Natural Science Foundation of China (62072028 and 61772067).

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, arXiv preprint arXiv: 1706.03762.

[2] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[3] Novikova J, Dušek O, Rieser V. The E2E dataset: New challenges for end-to-end generation[C]//18th SIGDIAL Conference. 2017: 201-206.

[4] Semantic Parsing on Freebase from Question-Answer Pairs Berant J, Chou A, Frostig R, et al. Semantic parsing on freebase from question-answer pairs[C]//Proceedings of the 2013 conference on empirical methods in natural language processing. 2013: 1533-1544.

[5] Zhang X, Wang H. A joint model of intent determination and slot filling for spoken language understanding[C]//IJCAI. 2016, 16(2016): 2993-2999.

[6] Stent A, Prasad R, Walker M. Trainable sentence planning for complex information presentations in spoken dialog systems[C]//Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04). 2004: 79-86.

[7] Dusek O, Novikova J, Rieser V. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge[J]. Computer Speech Language, 2020, 59: 123-156.

[8] Wen T H, Gasic M, Mrksic N, et al. Semantically conditioned lstm-based natural language generation for spoken dialogue systems[J]. arXiv preprint arXiv:1508.01745, 2015.

[9] Bo-Hsiang Tseng, Jianpeng Cheng, Yimai Fang, David Vandyke:A Generative Model for Joint Natural Language Understanding and Generation. ACL 2020: 1795-1807.

[10] Mrkšić N, Séaghdha D O, Wen T H, et al. Neural belief tracker: Data-driven dialogue state tracking[C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL. 2017: 1777-1888.

[11] Kingma D P, Welling M. Auto-encoding variational bayes[J]. arXiv preprint arXiv:1312.6114, 2013.

[12] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[C]// 3rd International Conference on Learning Representations, ICLR. 2015: 1-15.

[13] Banarescu L, Bonial C, Cai S, et al. Abstract meaning representation for sembanking[C]//Proceedings of the 7th linguistic annotation workshop and interoperability with discourse. 2013: 178-186.

[14] Berant J, Chou A, Frostig R, et al. Semantic parsing on freebase from question-answer pairs[C]//Proceedings of the 2013 conference on empirical methods in natural language processing. 2013: 1533-1544.

[15] Kingma D P, Ba J. Adam: A method for stochastic optimization[C]//3rd International Conference on Learning Representations, ICLR. 2015: 1-15.

[16] Lee D H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks[C]//Workshop on challenges in representation learning, ICML. 2013, 3(2): 896.

[17] Chang E, Demberg V, Marin A. Jointly improving language understanding and generation with quality-weighted weak supervision of automatic labeling[J]. arXiv preprint arXiv:2102.03551, 2021.