JCSE Journal of Computing Science and Engineering

Regular Paper

Journal of Computing Science and Engineering,
Vol. 16, No. 3, September 2022, pp. 165-177

Low-Dimensional Vector Representation Learning for Text Visualization Using Task-Oriented Dialogue Dataset

Taewook Hwang and Sangkeun Jung*
Computer Science & Engineering, Chungnam National University, Daejeon, Korea
taewook5295@gmail.com, hugman@cnu.ac.kr

Yoon-Hyung Roh
Language Intelligence Research Lab., Electronics and Telecommunications Research Institute, Daejeon, Korea
yhroh@etri.re.kr

Abstract
Text visualization is a complex technique that helps in data understanding and insight, and may lead to loss of information. Through the proposed low-dimensional vector representation learning method, deep learning and visualization through low-dimensional vector space construction were simultaneously performed. This method can transform a task-oriented dialogue dataset into low-dimensional coordinates, and based on this, a deep learning vector space can be constructed. The low-dimensional vector representation deep learning model found the intent of a sentence within a dataset and predicted the sentence components well in 3 out of 5 datasets. In addition, by checking the prediction results in the low-dimensional vector space, it was possible to improve the understanding of the data, such as identifying the structure or errors in the data.

Category: Natural Language Processing

Keywords: Natural language processing; Natural language understanding; Vector representation learning; Text visualization; Task-oriented dialogue dataset

I. INTRODUCTION

Natural language understanding (NLU) is a technology that represents the meanings of natural language in a structured form so that the machine can understand them. NLU is used in various fields such as dialogue systems, sentence structure analysis, and context analysis. In particular, NLU is one of the core technologies of conversational user interfaces, which are more widely used thanks to the prevalence of mobile phones and smart speakers. Recently, deep learning using various kinds of representation learning [1] has shown good performance in NLU.

High-quality data is essential for high-performance deep learning, including vector representation learning. Recently, a lot of raw-level natural language data has been collected through web crawling. The various deep learning models using this raw data show high performances. However, Well-tagged and processed data are needed for diverse NLU technologies, such as chatbots, dialog agents, and question answering.

Constructing well-tagged data requires a high-level

Open Access http://dx.doi.org/10.5626/JCSE.2022.16.3.165 http://jcse.kiise.org

Received 02 August 2022; **Accepted** 14 September 2022
*Corresponding Author

 pISSN: 1976-4677 eISSN: 2093-8020

insight and understanding of data. Moreover, construction and processing of natural language data are complicated due to the ambiguity of the language and domain portability, and there is no large corpus that can be used as a standard benchmark [2]. For this reason, data construction takes time and is costly.

Data visualization can be helpful in these situations. Visualization gives the insight to identify data characteristics and patterns within a short time, even by those who have no expertise in the domain. The insight enables the relevant persons to see the information that could not be known previously and helps them define new hypotheses, questions, and learning model structures for the data [3]. For visualization, the data needs to be transformed into low-dimensional coordinates in two or three dimensions. However, it is challenging to transform data with a large amount of information, and it is even more difficult to set specific transformation standards for natural language. To solve this problem, research using the vector space constructed through deep learning for visualization is being conducted.

In general, deep learning models show high performance only when a high-dimensional vector space is built and utilized. Therefore, the approach used in the existing studies for visualization of vector space is indirect visualization through dimensionality reduction techniques such as UMAP [4] or t-SNE [5]. However, dimensionality reduction methods are challenging to control and can differ from human intuition.

Therefore, we propose a low-dimensional semantic vector space construction method using semantic frames to parallel deep learning and visualization (Fig. 1). If a low-dimensional vector space is designed from the human point of view and deep learning is performed based on this space, it can help intuitive understanding of data information and deep learning operations. The core elements of the vector representation in this study are as follows.

- Recognizability: Low dimension vector spaces that can be recognized by humans.
- Controllability: Humans can easily control.
- Interpretability: Humans can intuitively understand and interpret semantic information.

To construct the interpretable, controllable, and recognizable low dimension semantic vector spaces, first, heuristic knowledge is grasped from the semantic frame mainly used in NLU. After that, the semantic frame is transformed into low dimension coordinates through a coordinate transformation algorithm reflecting the grasped heuristic knowledge to construct the correct answer vector space. Finally, the predicted semantic vector of each sentence calculated through the text reader and the correct answer coordinates are compared using a regression method to learn the model

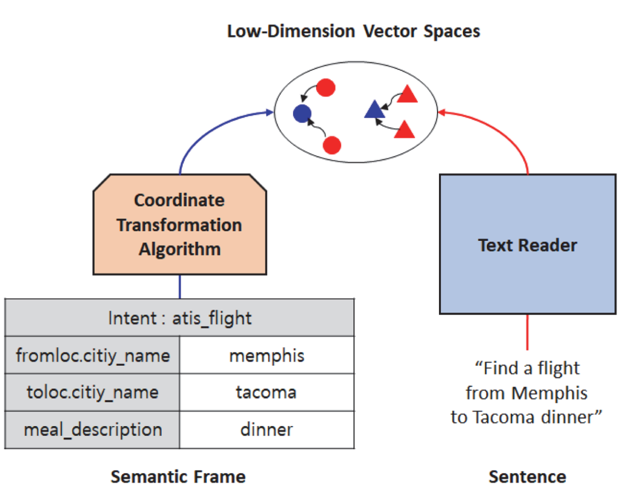


Fig. 1. Human interpretable low-dimensional semantic vector space construction method using semantic frames to parallel deep learning and visualization framework.

II. RELATED WORK

A. Vector Representation Learning

Most NLU technologies such as semantic role determination and syntax analysis can show performance-improving effects if only the input data are well embedded. Therefore, vector representation learning in NLU focuses on embedding technology that effectively vectorizes words and sentences [1].

In [6], the authors attempted to parse the semantic system by utilizing semantic and syntactic structures of sentences in learning. Kim et al. [7] attempted to represent the semantic by applying a method to inserted external data such as WordNet [8] and Macmillan dictionaries1 to the amount of information in the input data.

ELMo [9] proposed a method to obtain vector representations according to sentences through word embedding reflecting contexts using recurrent neural network with a pretrained language model.

BERT [10] learns using sentence data easily obtained from the web for masked language modeling and next sentence prediction. It was proved that the model is able to learn those sentence vector representations that apply to natural language processing in various fields.

In [11], the author modeled the co-semantic vector representations of sentences and semantic frames by actively modeling the correspondence between the sentences and the semantic frames in the NLU domain. In addition, using previous data, it successfully carried out an intent classification to classify sentences by measuring similarities between sentences and slot labeling to predict sentence components, thereby showing that it can be applied to various tasks of NLU [11].

B. Vector Space Visualization

Recently, various text visualization methods have been studied. In [12], the authors surveyed and classified text visualization research papers according to criteria, and through this, it can be seen that there are text visualization studies of various purposes and methods.

Among the various text visualization methods, this paper focuses on visualizing a vector space embedded with data. In general, a high-dimensional vector space is visualized by transforming it into a low-dimensional space through a dimensionality reduction technique. In the past, dimensionality reduction techniques such as DBSCAN [13] and MDS [14] were mainly used, but recently, UMAP [4] or t-SNE [5], which can effectively reduce high-dimensional information through mathematical techniques, are mainly used.

In order to gain insight into the vector space in which document information is embedded, the authors of [15] utilizes a dimensionality reduction technique to reduce a high-dimensional vector space to a low-dimensional one, enabling visual exploration. It visualizes neural document embeddings as configurable document maps, helping identify semantic features of data and semantic analysis.

Park et al. [16] proposed a visual analysis system to solve false-positive errors due to polysemy of natural language during text analysis. Using principal component analysis, they reduced the high-dimensional word embedding vector space to two dimensions and utilized it for concept-based document analysis.

III. DATA ANALYSIS

For the low dimension visualization proposed in this study, insights and understanding of data should be enhanced through structural and statistical analyses. In the case of low-dimensional spaces, the axis to express data information is quite limited compared to high-dimensional spaces. Therefore, only the core data information should be expressed through the axis, and other pieces of information should be expressed using distance, directions, and angles between data.

This paper used five datasets consisting of three datasets in Korean (Weather, Navi, and Restaurant) and two datasets in English (ATIS and SNIPS). Weather is a collection of speeches about weather, such as temperature and fine dust. Navi is a collection of speeches that can be requested to navigation devices, such as directions and search for nearby facilities during driving. Restaurant is a collection of speeches related to restaurants searching and reservations. ATIS is a collection of speeches regarding aircraft travel situations, such as flight reservation numbers and flight schedules. SNIPS is a collection of speeches that can be requested by an artificial intelligence assistant, such as music playback, restaurant reservations, and weather forecast information requests.

A. Data Structure

A dataset that is mainly used in NLU is called a task-oriented dialogue dataset, and it consists of intents, slot tags, and slot values called semantic frames. The intents are the subjects of the sentences, the slot tags are the names of the elements constituting the sentences, and the slot values are the actual values of the elements constituting the sentences.

Table 1 is an actual sentence appearing in the ATIS data set and the semantic frame of the sentence. There is one intent representing the sentence that exists, and there are many slot tags and slot values. In a semantic frame, the slot tags are subordinate to the intent, and slot values are subordinate to the slot tags. In addition, even if the same slot tags and slot values exist in two sentences, the semantic frames may differ according to the order in which individual slot values appear in the sentence.

B. Data Statistics

Data visualization is the visual representation of various numerical values based on statistical information. When data have been visualized, the statistical value such as the density, grouping, and distances can be seen directly. So, statistical information such as the distribution, composition, and scale of data should be understood first. To that end, statistical analysis was conducted to determine the distribution and density of data used in this study.

Table 2 shows the statistical distribution indicating how many slot tags are composed of Slot Tag Combinations (STC) in each dataset. STC are generally composed of 2–7 slot tags. One slot tag combination comprises many slot tags, which means that the amount of information is

Table 1. Example of semantic frame

Text	Find a flight from Memphis to Tacoma dinner
Semantic frame	atis flight (fromloc.city name=memphis, toloc.city name=tacoma, meal description=dinner)
Intent	atis flight
Slot tags	fromloc.city name, toloc.citiy name, meal description
Slot values	memphis, tacoma, dinner

Table 2. Dataset statistics

	Weather	Navi	Restaurant	ATIS	SNIPS
Number of trains	6,993	5,601	23,500	4,478	12,658
Number of tests	2,998	2,402	2,937	882	685
Number of intents	14	8	4	21	7
Number of slot tags	24	4	12	79	39
Number of slot tags combination	121	7	44	1,238	1,494

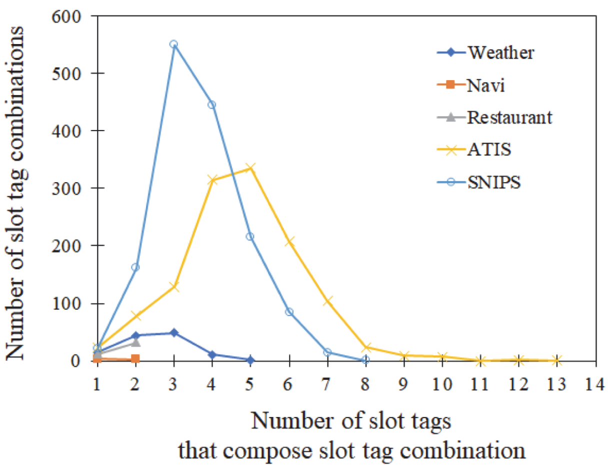


Fig. 2. Distribution of slot tag combinations.

large, and is relatively difficult to learn. On reviewing the distribution chart, it can be seen that the amounts of information of ATIS and SNIPS are larger than those of other data.

Fig. 2 shows the statistical distribution indicating how many slot tags are composed of STC in each dataset. It can be identified that the STC are generally composed of 2–7 slot tags. The more slot tags in the slot tag combination, the greater the amount of information, making it more difficult to learn. On reviewing the distribution chart, it can be seen that the amounts of information of ATIS and SNIPS are larger than those of other data.

The fact that the distributions and characteristics of the data are diverse despite being expressed in the same structure, semantic frame, means that diverse visualization results can appear depending on the dataset. A visualization method that is effective for one dataset may not suit the other datasets. In addition, in cases where the distribution is excessively concentrated on a specific value as with ATIS or SNIPS, the visualization may not be carried out properly, and quite biased visualization results may be produced. Given such facts, various visualization algorithms were proposed in this study, and attempts were made to find visualization methods suitable for individual datasets by comparing experimental performances.

IV. COORDINATE TRANSFORMATION ALGORITHMS FOR VISUALIZATION

For data visualization, understanding and insight into the relevant data are necessary. However, since visualization is performed in situations where data insight is insufficient, the characteristics or patterns that should be recognized cannot be identified. That is, there is a contradiction that visualization is performed for insight into data, but visualization is difficult due to lack of insight [3].

The three low-dimensional coordinate transformation algorithms proposed in this paper can be applied to all task-oriented dialog datasets with intents and slots. Because the data is transformed according to a consistent algorithm, the overall and detailed distribution of the data can be accurately identified, and comparisons between the data are possible. As this algorithm was developed based on the human point of view, it can help data set analysis by identifying outlier data or unusual data distributions that are not revealed in statistical data analysis.

A. Linear Coordinate Transformation

The first coordinate transformation algorithm is a linear coordinate transformation. This method converts all data into linear forms and is the most intuitive and suitable for human viewing among the three coordinate-transformation algorithms, but poses difficulties in predicting STC because the density of slot tags combinations is very high.

This method uses only two *s*; the x-axis and the y-axis. The number of all intents appearing in the data set is called n, the number of all STC is called m, the gap between intents on the x-axis is called x_{gap}, and the gap between slot tags on the y-axis is called y_{gap}. The y-coordinates of all intents are fixed at y_{gap}, and the x-coordinates are arranged at intervals of x_{gap} so that all intents can be placed between 0 and 1. After that, the x-coordinates of the STC belonging to individual intents become the x-coordinates of the relevant intents, and the y-coordinates are arranged at intervals of y_{gap} so that all STC can be arranged between 0 and 1.

$$x_{gap} = \frac{1}{n+1} \quad (1)$$

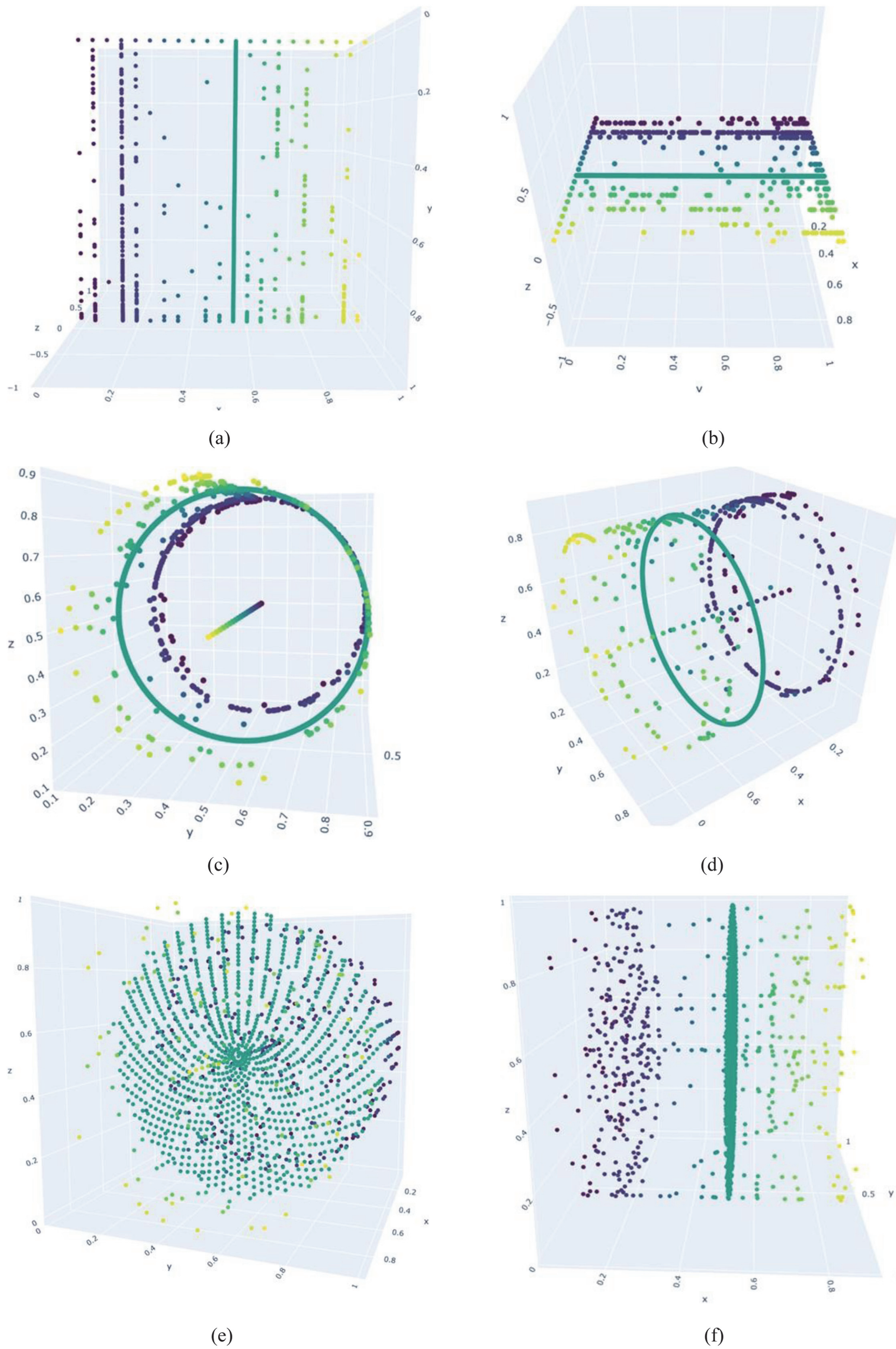


Fig. 3. Examples of coordinate transformation visualization (ATIS dataset). (a, c, e) Linear, circular, and spiral coordinate transformation visualization 1. (b, d, e) Linear, circular, and spiral coordinate transformation visualization 2.

$$y_{gap} = \frac{1}{m+2} \tag{2}$$

In Fig. 3, it can be seen that the intents are at the locations where the y-coordinate is 0, and below them, the STC belonging to individual intents are arranged. The parts with empty spaces in the middle are the locations of STC that appear in other intents but do not appear in the relevant intent. These parts are seen in other coordinate transformation algorithms.

B. Circular Coordinate Transformation

The second coordinate transformation algorithm proposed is a circular coordinate transformation. It is a method that transforms all data into circular forms and has intermediate intuitiveness and density among the three coordinate transformation algorithms.

This method utilizes all three-al coordinates of the x, y, and z-axes. The number of all intents appearing in the data set is called n, the number of all STC is called m, the gaps between intents on the x-axis is called x_{gap}, the radius of the circle is called r, and the angle between STC is called a. The y- and z-coordinates of all intents are fixed at 0.5, and the x-coordinates are arranged at intervals of x_{gap} so that all intents can be arranged between 0 and 1. After that, STC are arranged at angles of a at the circumferences of the circles centered around individual intents and of which the radius is r.

$$x_{gap} = \frac{1}{n+1} \tag{3}$$

$$r = 0.5 \tag{4}$$

$$a = \frac{360}{m} \tag{5}$$

It can be seen that there are intents at the center of Fig. 3, and STC are arranged at the circumferences of the circles that are centered around individual intents.

C. Spiral Coordinate Transformation

The last coordinate transformation algorithm is a spiral coordinate transformation. This method converts all data into spiral forms. It has the lowest intuitiveness among the three coordinate transformation algorithms, and also has the lowest density of STC.

Similar to the circular coordinate transformation algorithm, this method utilizes all three-al coordinates on the x, y, and z-axes. The number of all intents appearing in the data set is called n, the number of all STC is called m, and the gaps between intents at the x-axis is called x_{gap}. The number of STC to be arranged on the circumference of a spiral circle is called rc and initial rc is fixed at 16, called rc_{init}. The gap between spiral circles is called r_{gap} and can be obtained using the arithmetic progression summation formula. Since the rc_{init}. The gap between spiral circles is called r_{gap} rcinit is 16, STC are not arranged as much as the sum of 1 to 15. Therefore, 120, which is the sum of 1 to 15, is added in the arithmetic progression summation formula. The y- and z-coordinates of all intents are fixed at 0.5, and the x-coordinates are arranged at intervals of x_{gap} so that all intents can be arranged between 0 and 1. After that, as many STC as the rc are arranged on the circumferences of the spiral circles, which are centered on individual intents and of which the radius is r. The r increases by r_{gap}/rc each time STC are arranged one by one. The p in the r formula increases by 1 each time STC are arranged on the spiral circle one by one. The rc increases by 1 when all the STC in the number equal to the rc have been arranged on one spiral circle.

$$x_{gap} = \frac{1}{n+1} \tag{6}$$

$$rc_{init} = 16 \tag{7}$$

$$\frac{(i-1)*i}{2} + 120 \le total\ STC < \frac{i*(i+1)}{2} + 120 \tag{8}$$

$$r_{gap} = \frac{0.5}{i - rc_{init} + 1} \tag{9}$$

$$r = r_{gap} * (1 + rc - rc_{init}) + (\frac{r_{gap}}{rc} * p) \tag{10}$$

$$0 \le p < rc \tag{11}$$

The first slot tag combination is arranged at the distance of radius r_{gap}, and the 17th slot tag combination is arranged at the same angle as that of the first slot tag combination and at the distance of radius $2r_{gap}$. After that, the 34th slot tag combination is arranged at the distance of radius $3r_{gap}$, and all the STC to the last slot tag combination are arranged using the previous rule.

- 1^{st} STC : radius = r, angle = 0
- 2^{nd} STC : radius = $r + r/rc$, angle = $0 + 360/rc$
- 3^{rd} STC : radius = r + 2 r/rc, angle = 0 + 2 $360/rc$
- ...
- 17^{th} STC : radius = $2r$, angle = 0
- 18^{th} STC : radius = $2r + r/rc$, angle = $0 + 360/rc$

It can be seen that there are intents at the center of the

Fig. 3, and STC are arranged at the circumferences of the spiral circles that are centered around individual intents.

V. EXPERIMENT

The goal of the experiment is to confirm whether the vector expression learning using the low-al vector space shows satisfactory performance and improves the understanding of the experiment and data by visually confirming the experimental results. In order to verify the utility of the proposed methodology, an experiment corresponding to the following questions was performed.

- Do low-dimensional vector representation learnings show good performances?
- Can data be intuitively understood through visualization?

A. Model Structure

Fig. 4 is the diagram of the entire structure of the model. The model consists of a text reader and a coordinate transformation algorithm. The coordinate transformation algorithm converts all sentences into coordinates, and the Text Reader receives each sentence and outputs it in a 3D vector format. The Text Reader receives the sentence and outputs v_s, a 3D vector.

Two loss functions, the Euclidean distance and the cosine distance, were used. Euclidean distance measures distance in the vector space, and cosine distance measures direction in the vector space. When these two are used together, a more detailed vector space configuration is expected to be possible because both distance and direction are considered. L_e calculates the Euclidean distance between v_s and the coordinates. L_c calculates the cosine distance between v_s and the coordinates. In this case, the origin r of the cosine distance calculation is the intent coordinate of the relevant sentence. The sum of L_e and L_c is the final loss.

English and multilingual pretrained BERT models were used in the experiments. All models were fine-tuned for 30 epochs with batch size 32. The learning rate was set to 1e-5, and the maximum length of the input token was set to 40.

B. Sentence Search Performance

Table 3 shows the sentence search performance of all corpus and coordinate transformation algorithms used in this study. Sentence search measures the accuracy by comparing the intent and STC of the train sentence closest to the test sentence in the vector space. The distance between the train and test sentences was sorted in ascending order, and the accuracy was measured by selecting the top 1, 3, and 10 in the closest order among the sorted sentences.

Although excellent performances are shown in the case of intents, different results are shown depending on data and coordinate transformation algorithms in the case of STC.

The intent accuracy of Weather, Navi, Restaurant, and SNIPS shows over 90% performance in all coordinate transformation algorithms. ATIS shows more than 90% of linear coordinate transformation algorithm performance but less than 90% in circular and spiral.

The STC accuracy of Weather, Navi, and Restaurant has a coordinate transformation algorithm that shows a performance of more than 90%. On the other hand, in the case of ATIS and SNIPS, the performance is relatively low, and the performance deviation is huge depending on the coordinate transformation algorithm.

These results mean that appropriate coordinate transformation algorithms are different depending on the data and that the current coordinate transformation algorithms are insufficient to express all data or that changes in the composition of the data may be necessary.

C. Analysis with Visualization

In the visualization of Fig. 5, the blue dots are the coordinates calculated with the coordinate transformation algorithm, and the red dots are the v_s predicted by the deep learning model.

In the case of Weather, which showed high performance

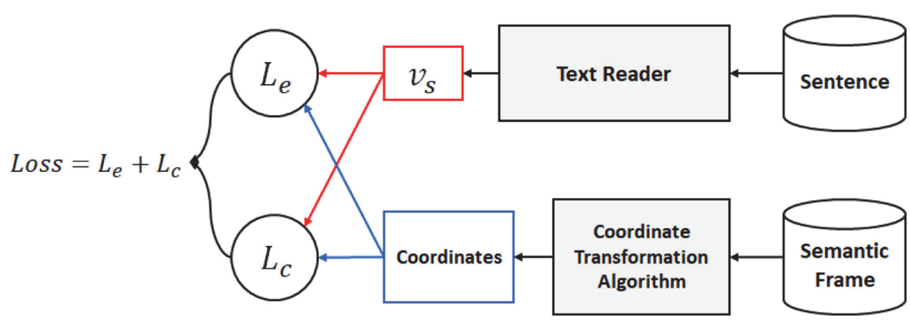


Fig. 4. Model structure.

Table 3. Top-*N* sentence search performance

		Intent accuracy (%)			STC accuracy (%)		
		Top 1	**Top 3**	**Top 10**	**Top 1**	**Top 3**	**Top 10**
Weather	Linear	**99.30**	**99.60**	**99.87**	76.52	94.06	**98.17**
	Circular	98.70	99.23	99.83	87.83	96.00	**98.07**
	Spiral	98.27	99.50	**99.87**	**94.96**	**96.76**	97.50
Navi	Linear	**99.83**	**99.96**	**100**	**99.67**	99.75	**100**
	Circular	99.75	99.92	**100**	99.58	99.71	**100**
	Spiral	**99.83**	**99.96**	**100**	**99.67**	**99.79**	**100**
Restaurant	Linear	97.31	97.51	99.35	91.93	95.44	97.14
	Circular	**97.48**	97.65	**99.63**	94.21	**95.85**	**97.34**
	Spiral	**97.48**	**97.79**	**99.63**	**95.13**	95.74	96.66
ATIS	Linear	**92.06**	**92.74**	**94.67**	**20.41**	31.52	46.83
	Circular	85.60	88.32	91.50	3.74	12.02	37.19
	Spiral	83.90	89.12	92.18	18.93	**39.46**	**47.17**
SNIPS	Linear	**98.25**	**99.27**	**100**	16.64	41.90	**65.40**
	Circular	98.10	99.12	**100**	14.89	35.91	59.42
	Spiral	95.77	96.93	98.83	**40.29**	**53.58**	60.29

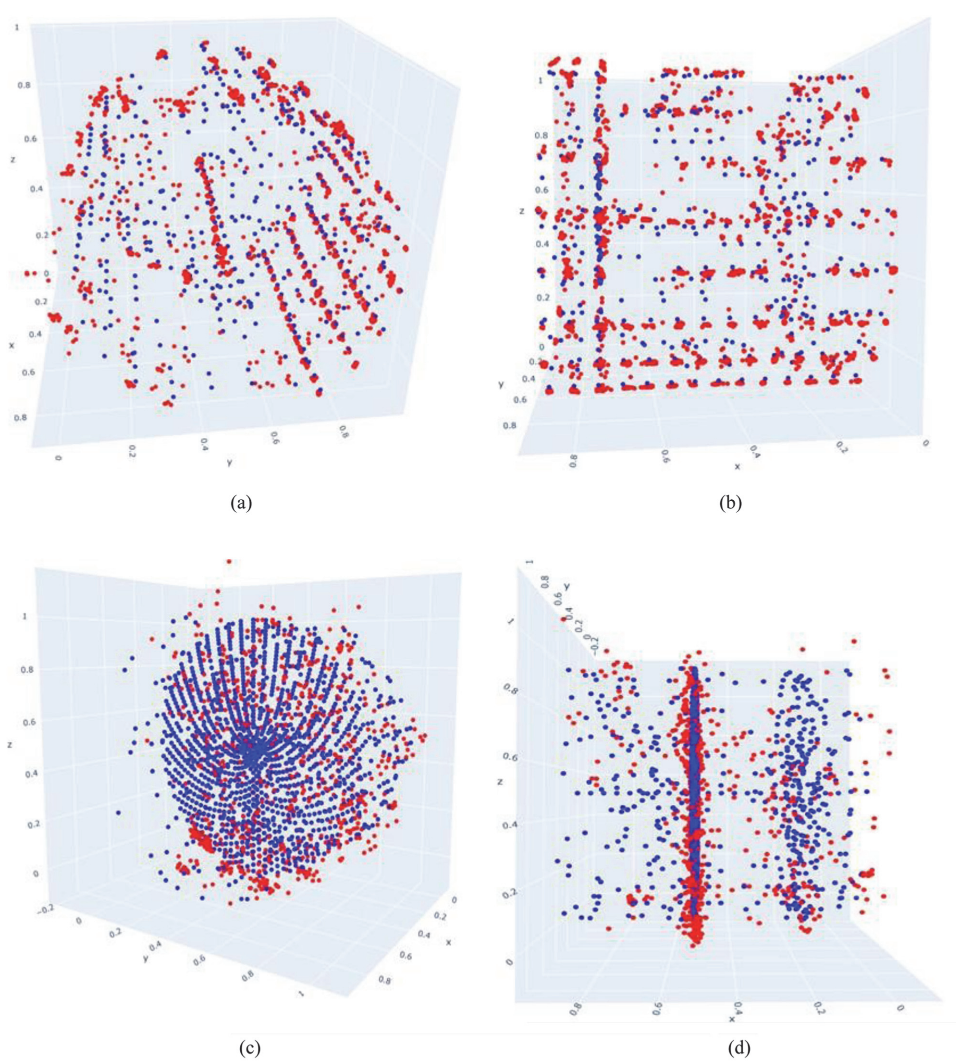


Fig. 5. Visualization of the correct answer (blue) of the coordinate transformation algorithm and the predicted value (red) of the model. (a) Weather spiral 1, (b) Weather spiral 2, (c) ATIS spiral 1, and (d) ATIS spiral 2.

in Table 3, it can be seen that the prediction was well made through Fig. 5(a) and 5(b). Although some errors are visible, most red dots are well clustered around the blue dots.

On the other hand, ATIS predicted intents well but did not correctly predict STC. On reviewing Fig. 5(c) and 5(d), it can be seen that many STC of ATIS is concentrated on some intents, and the data density is relatively high because the number of STC is large. This is assumed to be the reason why the performance of ATIS to predict STC is shown to be quite low.

On reviewing the predicted visualization results, it can be seen that data can be well predicted when the data density is low and the data are evenly distributed. In addition, differences in performance are large among the types of coordinate transformation algorithms. So, appropriate coordinate transformation algorithms must be selected or developed according to data, and various coordinate transformation algorithms that can reflect the characteristics of data well are necessary. In addition, by checking the visualization results, whether predicted data are concentrated on some intents or STC, and which tags cannot be appropriately predicted can be directly identified.

VI. DISCUSSION

According to the experimental results, high performance and intuitive visualization are possible at the same time if a low-dimensional vector expression learning method is suitable for the dataset. However, there is a limitation that a coordinate transformation algorithm suitable for the dataset needs to be developed manually, and the proposed coordinate transformation algorithm lacks generalization.

Nevertheless, the low-dimensional vector representation learning method proposed in this study is expected to have high utility in data-related tasks. The experiments demonstrate that the composition and distribution of data can be checked, and normal and erroneous data can be easily distinguished. It is also expected to group new labels based on data distribution or verify errors in new data. Such usability is expected to help data augmentation or integration.

VII. CONCLUSION

In this study, we proposed constructing a low-dimensional semantic vector space using semantic frames to combine deep learning and visualization. Three coordinate transformation algorithms were proposed, which can be applied to all Task-Oriented Dialogue datasets with intents and slots. The proposed methods showed good sentence search performance, meaning sentence intent and slot information were found well. After learning, it was possible to understand and analyze the experimental results by visualizing the predicted results.

As a future study, we plan to study visualization technology that can reflect both human heuristic in- formation and machine coordinate space configuration technology by applying deep learning technology to the coordinate transformation algorithm. In addition, we intend to develop and proceed with a method that can increase versatility and conduct research to apply it to understanding, analyzing, and augmenting various data.

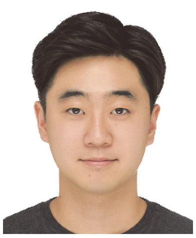
ACKNOWLEDGMENTS

This work was supported by Institute of Information Communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00004, Development of semi-supervised learning language intelligence technology and Korean tutoring service for foreigners). This work has supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022 R1F1A1071047).

REFERENCES

1. Y. Bengio, A. Courville, and P. Vincent, “Representation learning: a review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798-1828, 2013.
2. F. Brad, R. Iacob, I. Hosu, and T. Rebedea, “Dataset for a neural natural language interface for databases (NNLIDB),” 2017 [Online]. Available: https://arxiv.org/abs/1707.03172.
3. J. J. Van Wijk, “The value of visualization,” in *Proceedings of IEEE Visualization*, Minneapolis, MN, 2005, pp. 79-86.
4. L. McInnes, J. Healy, and J. Melville, “UMAP: uniform manifold approximation and projection for dimension reduction,” 2018 [Online]. Available: https://arxiv.org/abs/1802.03426.
5. L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, pp. 2579-2605, 2008.
6. Y. N. Chen, D. Hakanni-Tur, G. Tur, A. Celikyilmaz, J. Guo, and L. Deng, “Syntax or semantics? knowledge-guided joint semantic frame parsing,” in *Proceedings of 2016 IEEE Spoken Language Technology Workshop (SLT)*, San Diego, CA, 2016, pp. 348-355.
7. J. K. Kim, G. Tur, A. Celikyilmaz, B. Cao, and Y. Y. Wang, “Intent detection using semantically enriched word embeddings,” in *Proceedings of 2016 IEEE Spoken Language Technology Workshop (SLT)*, San Diego, CA, 2016, pp. 414-419.
8. G. A. Miller, *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.
9. M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” 2018 [Online]. Available: https://arxiv.org/abs/1802.05365.

10. J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” 2018 [Online]. Available: https://arxiv.org/abs/1810.04805.
11. S. Jung, “Semantic vector learning for natural language understanding,” *Computer Speech & Language*, vol. 56, pp. 130-145, 2019
12. M. Alharbi and R. S. Laramee, “SoS TextVis: an extended survey of surveys on text visualization,” *Computers*, vol. 8, no. 1, article no. 17, 2019. https://doi.org/10.3390/computers8010017
13. M. Ester, H. P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, Portland, OR, 1996, pp. 226-231.
14. A. Buja, D. F. Swayne, M. L. Littman, N. Dean, H. Hofmann, and L. Chen, “Data visualization with multidimensional scaling,” *Journal of Computational and Graphical Statistics*, vol. 17, no. 2, pp. 444-472, 2008
15. X. Ji, H. W. Shen, A. Ritter, R. Machiraju, and P. Y. Yen, “Visual exploration of neural document embedding in information retrieval: semantics and feature selection,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 6, pp. 2181-2192, 2019.
16. D. Park, S. Kim, J. Lee, J. Choo, N. Diakopoulos, and N. Elmqvist, “ConceptVector: text visual analytics via interactive lexicon building using word embedding,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 361-370, 2017.



Taewook Hwang

Taewook Hwang received an B.S. in Computer Science & Engineering from Chungnam National University, Daejeon, South Korea in 2019. He is currently pursuing a Ph.D. in Computer Science & Engineering from Chungnam National University, Daejeon, South Korea.

Sangkeun Jung

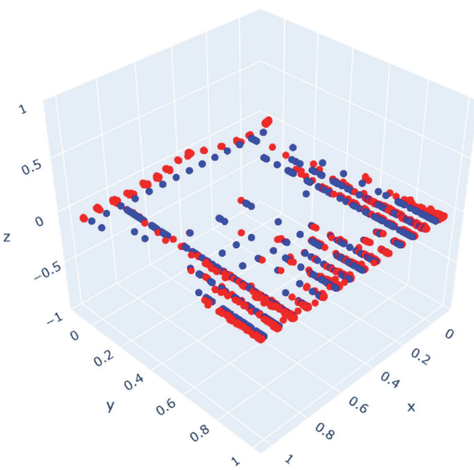
Sangkeun Jung received his B.S., M.S., and Ph.D. in computer engineering from POSTECH, Pohang, South Korea, from 2004 to 2010. From 2010 to 2012, he was a researcher at the Samsung Electronics, Suwon, South Korea. From 2012 to 2014, he was a researcher at ETRI, Daejeon, South Korea. Moreover, from 2014 to 2018, he was a researcher at SK Telecom, Seoul, South Korea. Since 2018, he has been a professor of computer science and engineering at Chungnam National University, Daejeon, South Korea. His research interests include natural language processing, machine learning, and deep learning.

Yoon-Hyung Roh

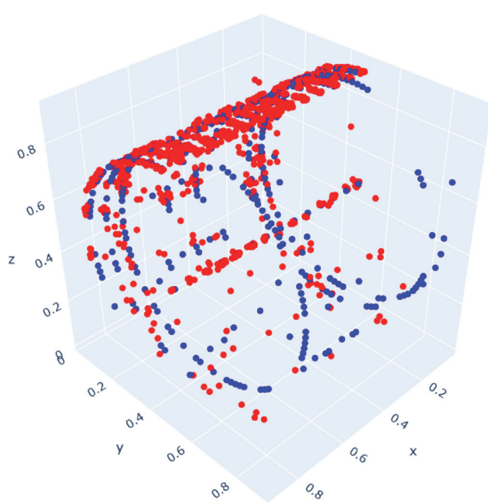
Yoon-Hyung Roh received an M.S. in computer science from KAIST, Daejeon, South Korea in 2000. Since 2000, he has been a researcher at ETRI, Daejeon, South Korea. His research interests include natural language processing, dialog systems, and machine learning.

I. APPENDIX. VISUALIZATION OF EXPERIMENT RESULTS

The Figure A.1, A.2, A.3, A.4 and A.5 are experiment results such as the Figure 5.

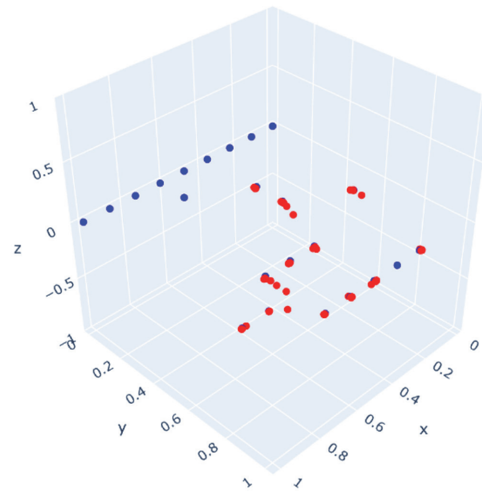


(a) Weather linear

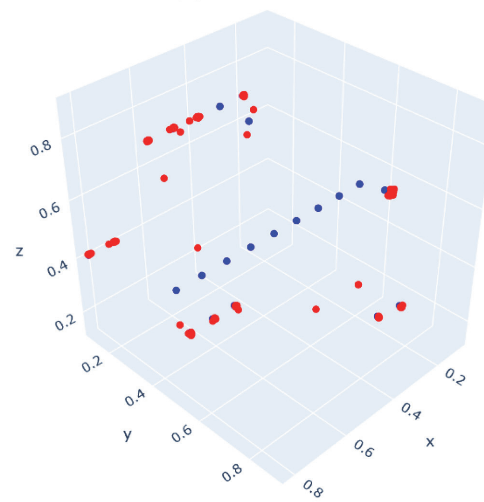


(b) Weather circular

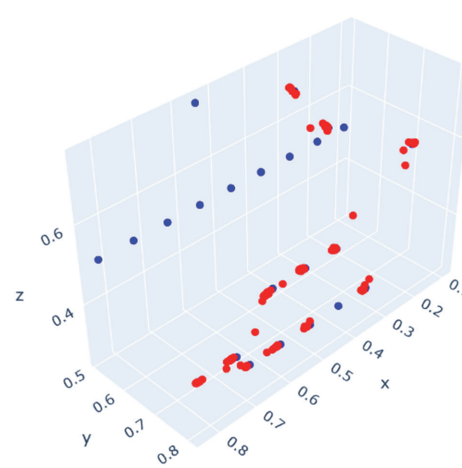
Fig. A.1: Visualization of the correct answer(blue) of the coordinate transformation algorithm and the predicted value(red) of Weather dataset.



(a) Navi linear

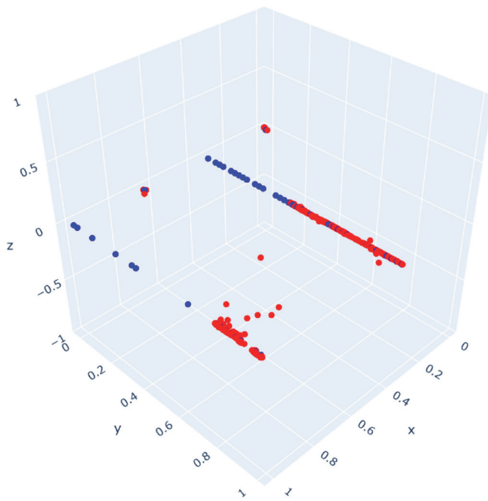


(b) Navi circular

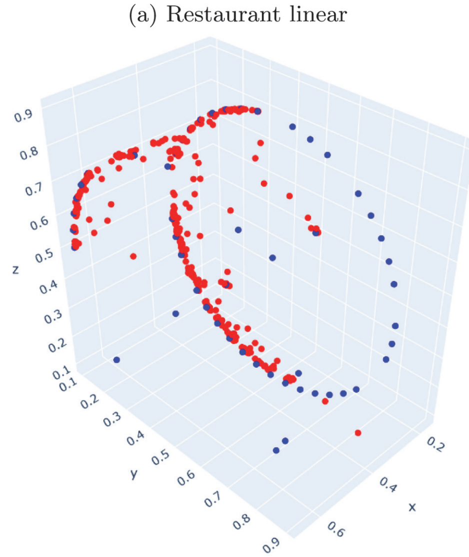


(c) Navi spiral

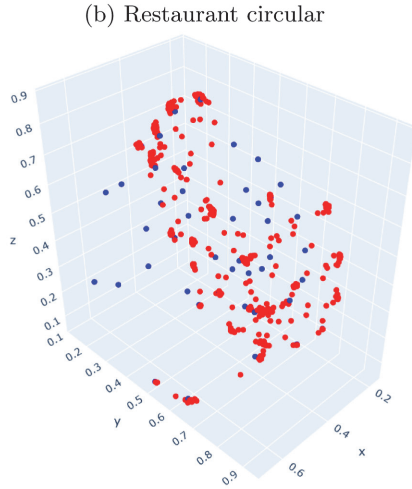
Fig. A.2: Visualization of the correct answer(blue) of the coordinate transformation algorithm and the predicted value(red) of Navi dataset.



(a) Restaurant linear

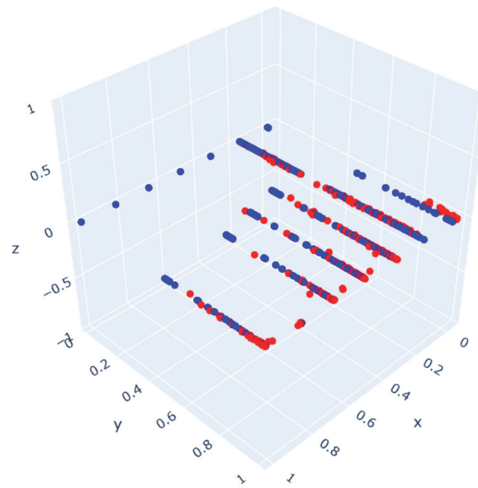


(b) Restaurant circular

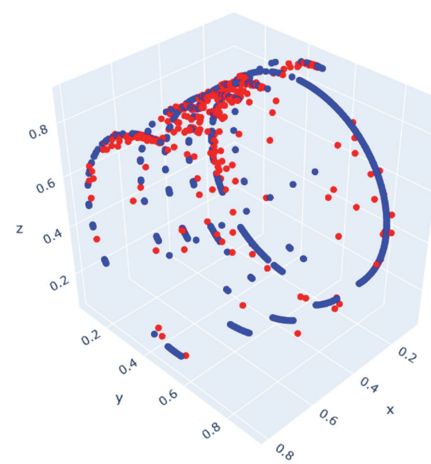


(c) Restaurant spiral

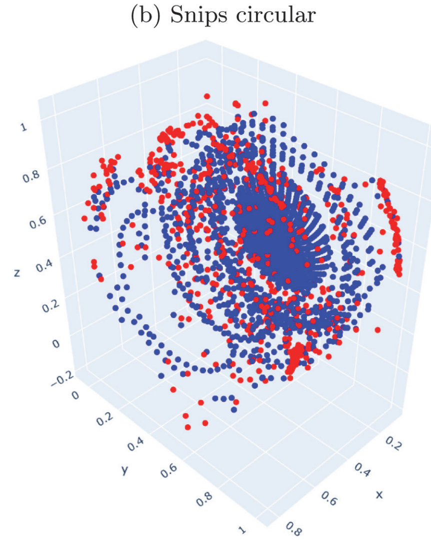
Fig. A.3: Visualization of the correct answer(blue) of the coordinate transformation algorithm and the predicted value(red) of Restaurant dataset.



(a) Snips linear

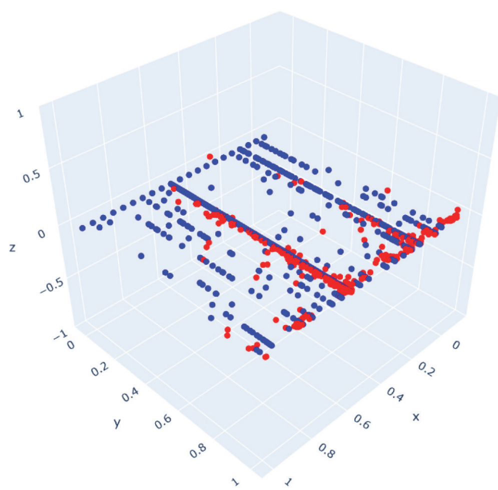


(b) Snips circular

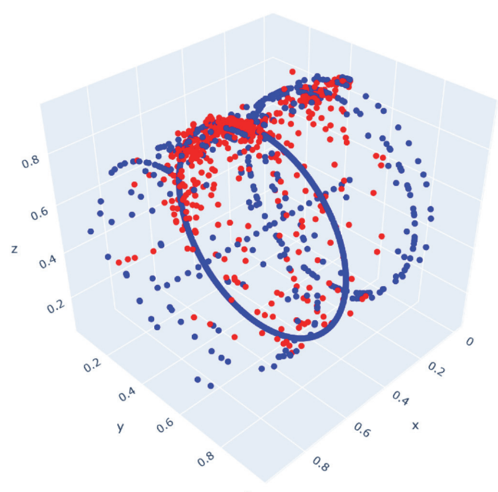


(c) Snips spiral

Fig. A.4: Visualization of the correct answer(blue) of the coordinate transformation algorithm and the predicted value(red) of Snips dataset.



(a) ATIS linear



(b) ATIS circular

Fig. A.5: Visualization of the correct answer(blue) of the coordinate transformation algorithm and the predicted value(red) of ATIS dataset.