Multi-Task Learning for Ambiguous Candidate Identification with Pre-trained Model

Daesik Jang* **and Hyewon Choi***
daesik0320@gmail.com, hyewon4999@gmail.com

Abstract

Recently, research using multimodal datasets containing image and text information has been conducted actively. One of them is the SIMMC2.1 dataset. It is a more complicated dataset than answering a conversation using only text because it should predict an answer after understanding the relationship between images and text. Therefore, there are limitations to answering a conversation only using text-based models such as BERT or GPT-2, so models with both image and language understanding abilities should be considered. We propose a new model that is effective for the ambiguous candidate identification task in DSTC11 SIMMC2.1 Tark. It consists of a simple pipeline model structure, which has two steps. The first step is to check whether there is ambiguity in the current user utterance, and the second step is to extract objects mentioned in the ambiguous utterance of the user. We suggest a new learning framework with a pre-trained image model and text model that is effective for the ambiguous candidate identification task. Experiments show that the proposed method can improve the model performance, and our model achieved 3rd place in sub-task 1 of the SIMMC2.1 track.

1 Introduction

Conversations with multiple contexts, and multimodal components, have been actively explored (Das et al., 2017; De Vries et al., 2017; Antol et al., 2015). When developing the conversational agent in the real world, visual information is essential for effective conversation agents so that they can easily adapt to user situations. In this paper, we present our ambiguous candidate identification model, which was submitted to the SIMMC2.1 challenge (Kottur et al., 2021). We focus on the ambiguous candidate identification task of SIMMC2.1, among other tasks. It is the task of detecting ambiguity in conversations and extracting objects mentioned in the user utterance predicted as ambiguous. For example, ambiguities may occur for several reasons, such as when the user does not give enough details to single out a unique referent (e.g., "What is the price of that t-shirt?" instead of "What is the price of the t-shirt on the left?") (Kottur et al., 2021). If a conversational agent needs to answer an ambiguous utterance of the user like the example, the agent may not be able to predict a correct answer effectively. Therefore, for an agent to generate a correct answer to the ambiguous utterance of the user, it is critical to find out the information about objects mentioned in the utterance and help the agent to generate the answer.

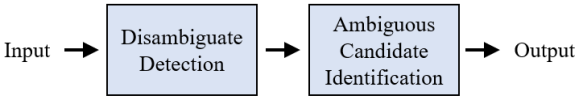

Figure 1: An illusion of the two-step pipeline model framework.

To solve this problem, we use the SIMMC2.1 dataset. For each scene in the data, there is a JSON file with every object's information including its name, ID, bounding box (bbox), location, and relationship. We use the objects' metadata(e.g., color, pattern, type) to determine whether the current user utterance is ambiguous and extract all objects mentioned in the user utterance predicted as ambiguous. Figure 1 is the structure of our proposed model. We construct a two-step pipeline model to solve the task. First, we detect whether the current user utterance is ambiguous using a binary classification model. Then, we use the dialogue history and object information to extract all objects in the ambiguous utterance of the user. This paper has two main contributions as follows:

1. Contrastive loss (Chen et al., 2020; Kim et al., 2021) and binary cross-entropy loss are re-

*These authors contributed equally to this work

Proceedings of The Eleventh Dialog System Technology Challenge, pages 9–14

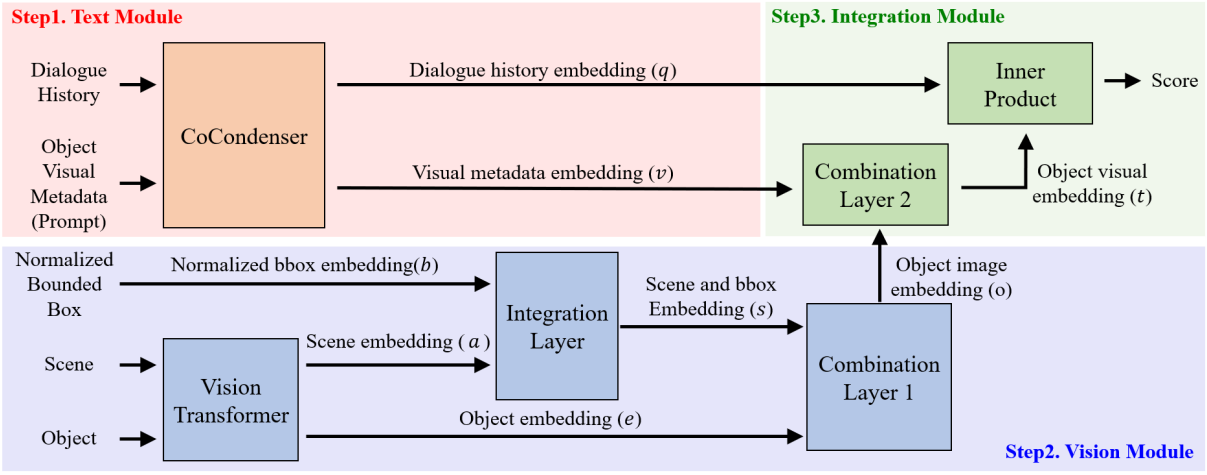

Figure 2: An illustration of the ambiguous candidate identification model using pre-trained model.

garded as different tasks to train a model using the MT-DNN (Liu et al., 2019) method.

2. An effective embedding generation for ambiguous candidate identification task is done by additionally inputting abundant linguistic information of metadata into the model using the prompt we created. Then, we efficiently adjust the encoding ratio according to data characteristics with a linear combination (Choi and Ko, 2022) to create object visual embedding.

Experiments show that our method can improve the model performance. As a result, we gained 3rd place in sub-task 1 of the SIMMC2.1 track.

2 Task and Data Description

DSTC11 Track 1 SIMMC2.1 consists of four sub-tasks. SIMMC 2.1 provides different datasets for each sub-task, such as the object metadata and the scene image corresponding to dialogue. Among them, we solve sub-task 1, which is the ambiguous candidate identification task. The ambiguous candidate identification task focuses on identifying the ambiguity in the current user utterance given dialogue context and scene information and finding object candidates corresponding to the ambiguity. This task is very effective for multi-modal interactive AI assistants to generate clear utterances from the next utterance.

3 Method

We solve the ambiguous candidate identification task with a two-step pipeline model. Figure 1 is the structure of our proposed model. We describe the disambiguate detection in 3.1 and the ambiguous candidate identification in 3.2.

3.1 Disambiguate Detection

First, disambiguate detection determines whether the current user utterance is ambiguous through binary classification using CoCondenser proposed in Gao and Callan (2022). The CoCondenser is a pre-trained language model based on an encoder suitable for contrastive learning. These Transformer (Vaswani et al., 2017) blocks are divided into three groups, early backbone encoder layers, late backbone encoder layers, and head layers. The head takes the `[CLS]` representation from the late representations, and the token representations from the early layers using a skip connection. The skip connection from the early layers to the head layers lets the model's `[CLS]` token have the global meaning of the input text. We use a pre-trained CoCondenser to generate `[CLS]` token embedding containing the global meaning of dialogue history. The input sequence is as follows:

`[CLS]` Current user utterance `[SEP]` Dialogue context `[SEP]`

Let $C \in \mathbb{R}^h$ be the final hidden vector of `[CLS]` token using CoCondenser, $W_{DD} \in \mathbb{R}^{2\times h}$ be a classification layer for disambiguate detection layer, and we minimize a binary cross-entropy loss.

$$f(x) = softmax\left(CW_{DD}^T\right) = [\hat{y_0}, \hat{y_1}] \quad (1)$$

$$\mathcal{L} = -\sum_{x \in D} y \ log\, \hat{y_1} + (1-y) \ log\, \hat{y_0} \quad (2)$$

where $f(x)$ is the final output of the model and $y \in \{0, 1\}$ is a label.

3.2 Ambiguous Candidate Identification

Figure 2 shows the entire structure of ambiguous candidate identification. The ambiguous candidate identification consists of three major parts. The first is the Text Module, the second is the Vision Module, and the last is the Integration Module which fuses the text and image information.

3.2.1 Text Module

We use the pre-trained language model CoCodenser in the Text Module. The Text Module is used for encoding both the dialogue history and object visual metadata. The input of dialogue history is the same as the input format of Disambiguate Detection. Object visual metadata is composed of prompts to contain rich metadata information. We have structured the prompts for two domains: Fashion and Furniture. The examples of prompts are as below.

Fashion metadata

- Absence of "Sleeve Length"
  - This <Type> with <Pattern> pattern and colored <Color>.
  - Ex) This is jean with denim pattern and colored dark blue.
- Presence of "Sleeve Length"
  - This <Sleeve Length> sleeve <Type> with <Pattern> pattern and colored <Color>.
  - Ex) This is long sleeve blouse with plaid pattern and colored red, white, yellow.

Furniture metadata

- This is <Color> <Type>.
- Ex) This is red area rug.

For the fashion data, the configured prompt varies depending on the presence or absence of "Sleeve Length" in metadata. We create the dialogue history embedding and visual metadata embedding using `[CLS]` token embedding in the last layer.

3.2.2 Vision Module

The Vision Module uses a scene of the current user utterance and an object included in it to encode visual information corresponding to that utterance. We use Vision Transformer (ViT) (Dosovitskiy et al., 2020) to encode a scene and an object effectively. Scene embedding and object embedding are created using the class token embedding of the last layer with a pre-trained ViT without further fine-tuning. Then, we add the normalized bbox information to supplement the object's relative position information to the scene embedding. It is necessary to normalize the bbox as shown in Equations (3-6) with the method proposed in YOLO (Redmon et al., 2016). The normalized bbox information provides relative positioning information for the object in the scene.

$$x_{center} = \frac{\left(x + \frac{width}{2}\right)}{width\ of\ whole\ image} \tag{3}$$

$$y_{center} = \frac{\left(y + \frac{height}{2}\right)}{height\ of\ whole\ image} \tag{4}$$

$$w_{ratio} = \frac{width}{width\ of\ whole\ image} \tag{5}$$

$$h_{ratio} = \frac{height}{height\ of\ whole\ image} \tag{6}$$

$$b = [x_{center};\ y_{center};\ w_{ratio};\ h_{ratio}] \tag{7}$$

As shown in Equation (7), the normalized bbox embedding b is generated through concatenation. Then, the bbox embedding b is concatenated with the scene embedding $a \in \mathbb{R}^h$ to produce the embedding $s \in \mathbb{R}^h$ using the learnable parameter $W_{Integration} \in \mathbb{R}^{h \times (h+4)}$.

$$s = W_{Integration} \cdot [a; b] \tag{8}$$

The embedding s and the object embedding $e \in \mathbb{R}^h$ are concatenated as in Equation (9) to obtain the weight $m_1 \in \mathbb{R}^h$ through a learnable parameter, $W_\gamma \in \mathbb{R}^{h \times 2h}$. The object image embedding $o \in \mathbb{R}^h$ is produced through a linear combination using the weight m_1.

$$m_1 = \sigma\left(W_\gamma \cdot [e; s]\right) \tag{9}$$

$$o = m_1 \circ e + (1 - m_1) \circ s \tag{10}$$

where $\circ$ is the element-wise product.

3.2.3 Integration Module

The Visual metadata embedding, $v \in \mathbb{R}^h$, and the object image embedding, o, are linearly combined.

$$m_2 = \sigma\left(W_\alpha \cdot [v; o]\right) \tag{11}$$

$$t = m_2 \circ v + (1 - m_2) \circ o \tag{12}$$

When extracting the ambiguous candidate, the importance of visual metadata and object image information may vary depending on the situation. Thus, we use a linear combination to dynamically

adjust the amount of the visual metadata and object image information to generate an embedding $t \in \mathbb{R}^h$ that is effective for extracting ambiguous candidates. The final score is calculated dialogue history embedding $q \in \mathbb{R}^h$ and inner product with the embedding t containing the object visual information created in Equations (11) and (12) as in Equation (13) and (14).

$$\hat{y} = q \cdot t \tag{13}$$

$$score = sigmoid\,(\hat{y}) \tag{14}$$

At inference time, if the $score$ value exceeds the threshold value δ, it is determined as an ambiguous candidate.

3.2.4 Training Method

We train a model using the binary cross-entropy loss and contrastive loss. In binary cross-entropy loss, we train positive samples to have scores close to 1 and negative ones to 0.

$$\mathcal{L}_{BCE} = -y\,log\,\sigma\,(\hat{y}) + (1-y)\;log\;(1-\sigma\,(\hat{y})) \tag{15}$$

After obtaining scores using positive and negative pairs of dialogue history and objects, a contrastive loss is calculated.

$$\mathcal{L}_{Contrastive} = -log\,\frac{exp\,(q \cdot t_+/\tau)}{\sum_k exp\,(q \cdot t_k/\tau)} \tag{16}$$

We adopt the multi-task learning framework, MT-DNN, and train the model with both binary cross-entropy loss and contrastive loss.

4 Experiments

4.1 Training Setup

We utilized CoCondenser pre-trained with Wikipedia data and pre-trained ViT. We obtained the class token embedding of the last layer from the pre-trained ViT, which was released in Dosovitskiy et al. (2020). The optimizer used for training is Adam (Kingma and Ba, 2014) and we use the linear scheduler with a learning rate of 5e-5. We trained our model for 10 epochs on one RTX 3090.

4.2 Comparison with other models

Table 1 shows the performance results of our proposed model and other models on devtest dataset and teststd dataset. The proposed model was evaluated using the two steps pipeline model. Our model achieved 3rd place in both devtest dataset and teststd dataset. The results prove that our learning method is effective for the ambiguous candidate identification task.

TeamID	Devtest F1-score	Teststd F1-score
1	70.31%	67.26%
2 (Our model)	66.22%	65.17%
3	62.45%	63.84%
4	68.47%	70.50%

Table 1: The performance comparison on devtest and teststd dataset.

Type	Loss function	Precision	Recall	F1-score
1	Contrastive Loss	29.28%	93.73%	44.62%
2	Binary Cross-entropy Loss	62.23%	69.46%	66.20%
3	Contrastive Loss + Binary Cross-entropy Loss	**63.64%**	**76.10%**	**69.28%**

Table 2: Experimental results according to loss type on devtest dataset.

Type	Loss function	Positive	Negative
1	Contrastive Loss	12.76	2.93
2	Binary Cross-entropy Loss	2.81	-3.88
3	Contrastive Loss + Binary Cross-entropy Loss	6.06	-8.30

Table 3: Experimental results of averaging the positive and negative samples $\hat{y}$ value on devtest dataset.

4.3 Effect of loss type

Table 2 is the model performance according to the loss type, and Table 3 presents the score values derived at inference time from the model trained with each loss type. If a score value obtained from Equation (14) is 0.5 or higher, we regard the value as an ambiguous candidate. Positive and negative scores in Table 3 are the results of averaging $\hat{y}$ values obtained from Equation (13) of the positive and negative samples of all evaluation data during inference.

First, Type 1 contrastive loss shows relatively low performance with an F1-score of 44.62%. Then, recall is a very high value of 93.73%. In Type 1, most positive and negative $\hat{y}$ values were higher than 0. We can see that the contrastive loss simply distances a positive $\hat{y}$ value from a negative one without any directional information about an object.

Second, Type 2 binary cross-entropy loss is an F1-score of 66.20%, achieving relatively higher performance than Type 1. The recall of Type 2 is 69.46%, which is lower than that of Type 1. Unlike the contrastive loss, the binary cross-entropy learns positive scores as close to 1 and negative scores as close to 0. In this way, it helps the model absorb the directional information about an object. Therefore,

we can see the Type 2 scores in Table 3 evenly distributed as positive sample $\hat{y}$ values as positive numbers and negative sample $\hat{y}$ values as negative numbers based on 0.

Finally, Type 3 is a model with binary cross-entropy loss and contrastive loss using the method proposed in MT-DNN. The result was an F1-score of 69.28%, which is about 3%p higher than that of Type 2. In MT-DNN multi-task learning framework, models can have benefits of both contrastive loss and binary cross-entropy loss. That is, while learning directional information about an object, positive $\hat{y}$ values are evenly distributed as positive numbers and negative $\hat{y}$ values as negative numbers. Also, we can see that the contrastive loss distances a positive score from a negative one with directional information about an object, unlike Type 1. This can extract ambiguous candidates more effectively than Type 1 and Type 2 when identifying them.

4.4 Ablation study

In Table 4, the Text Module of Figure 2 uses the same method to encode dialogue history and object visual metadata. We tested by setting the Vision Module encoding method differently.

Method	Model	Precision	Recall	F1-score
1	CoCondenser + Scene embedding + Normalized bbox	43.44%	51.15%	46.98%
2	CoCondenser + Scene embedding + Normalized bbox **+ Object embedding**	57.74%	58.02%	57.88% (+10.9%p)
3	CoCondenser + Scene embedding + Normalized bbox + Object embedding **+ Visual metadata** **(Our model)**	**63.64%**	**76.01%**	**69.28%** **(+22.3%p)**

Table 4: The experiments to examine the vision module encoding method on devtest dataset.

First, we encode image information using scene embedding and normalized bbox information. The result is an F1-score of 46.98%. Second, using object embedding together improved performance by about 11%p. These evaluation results demonstrate that we can create richer object image embedding by using the image information of an object itself with the scene information. Finally, when we use our own prompt, there is as much as 11%p performance improvement with an F1-score of 69.28%. This shows that information on each object's prompt plays a significant role in resolving the ambiguous candidate identification tasks.

4.5 Effect of encoding method

When combining two embeddings, we used a linear combination instead of concatenating and projecting them through weights. The structure helps the model dynamically adjust the encoding ratio depending on the situation. A simple weighted method concatenates two embeddings and then projects them through the weights W to create one embedding.

Combination layer 1	Combination layer 2	F1-score
simple weight	simple weight	66.74%
linear combination	simple weight	67.37%
linear combination	linear combination	**69.28%**

Table 5: The effect of the linear combination on devtest dataset.

We evaluated three forms to prove the contribution of our proposed linear combination. First, when using the simple weights for the combination layers 1 and 2, an F1-score of 66.74% shows relatively low performance. Second, when using a linear combination for combination layer 1 and simple weights for combination layer 2, an F1-score of 69.28% achieves higher performance than when applying simple weights to both layers. Finally, using a linear combination for both layers gives the highest performing an F1-score of 69.28%. These results prove our hypothesis that our proposed linear combination method effectively solves ambiguous candidate identification tasks by dynamically adjusting the encoding ratio according to the data characteristics.

5 Conclusion

Our proposed model places 3rd in the ambiguous candidate identification task of the SIMMC2.1 track. There are two main contributions. First, we consider both the contrastive loss and the binary cross-entropy loss and use a multi-task learning framework, MT-DNN. By learning a model in this way, it can have both benefits of contrastive loss and binary cross-entropy loss. We have demonstrated through testing that our proposed learning method is effective for ambiguous candidate identification. Second, by using our prompt to further input metadata information into a model, we create richer embeddings for successful ambigu-

ous candidate identification. Then, we use a linear combination to adjust the encoding ratio according to the data characteristics to create embeddings suitable for object extraction. Through ablation studies, evaluation results show that visual metadata information using prompts improve performance. In particular, model performance increased by about 11%p through visual metadata information (prompt), proving the effectiveness of our prompts.

We suggest a novel two-step pipeline system for the ambiguous candidate identification task. In the future, we are planning to work on an End2End system instead of a two-step pipeline system.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Hyewon Choi and Youngjoong Ko. 2022. Effective fake news video detection using domain knowledge and multimodal data fusion on youtube. *Pattern Recognition Letters*, 154:44–52.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 326–335.

Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5503–5512.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853.

Bosung Kim, Hyewon Choi, Haeun Yu, and Youngjoong Ko. 2021. Query reformulation for descriptive queries of jargon words using a knowledge graph based on a dictionary. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 854–862.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Satwik Kottur, Seungwhan Moon, Alborz Geramifard, and Babak Damavandi. 2021. Simmc 2.0: A task-oriented dialog dataset for immersive multimodal conversations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4903–4912.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.