Representing Sentence Interpretations with Overlapping Box Embeddings

Anonymous ACL submission

Abstract

Most of the previous studies on sentence embeddings aim to obtain a single representation per sentence. However, this approach is inadequate for handling the semantic relations between sentences when a sentence has multiple interpretations. To address this problem, we propose a novel concept, *interpretation embeddings*, which are representations of the interpretations of a sentence. We propose GumbelCSE, which is a contrastive learning method for learning box embeddings of sentences. The interpretation embeddings are derived by measuring the overlap between the box embeddings of the target sentence and those of other sentences. We evaluate our method on three tasks: Recognizing Textual Entailment (RTE), Entailment Direction Prediction, and Ambiguous RTE. On the RTE and Entailment Direction Prediction tasks, GumbelCSE outperforms baseline sentence embedding methods in most cases. In the Ambiguous RTE task, it is demonstrated that the interpretation embeddings are effective in capturing the ambiguity of meaning inherent in a sentence.[1]

1 Introduction

Sentence embeddings are vector representations of the meaning of a sentence, and they have been well-studied in the field of natural language processing (NLP) (Reimers and Gurevych, 2019; Gao et al., 2021; Jiang et al., 2024). Most of the previous studies aim to obtain one representation per sentence. However, this approach cannot handle the relations between sentences appropriately when a sentence has multiple interpretations. For example, the sentence "John and Anna are married" can be interpreted in two ways: "John and Anna are married to each other" and "John and Anna are both married." The former contradicts the sentence "John and Anna are not a couple," while the latter does not.

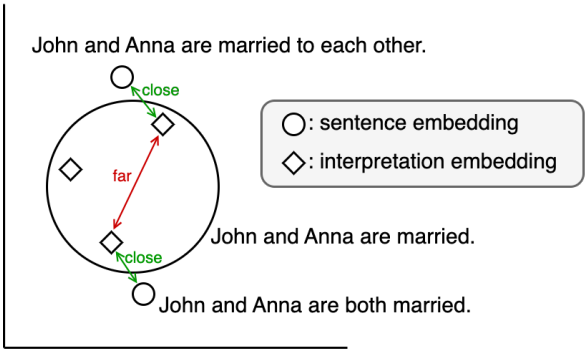

Figure 1: Conceptual diagram of interpretation embeddings

To address this problem, we propose *interpretation embeddings*, which are representations of the interpretations of a sentence. As illustrated in Figure 1, in our approach, an embedding of a sentence contains embeddings of multiple interpretations of the sentence, where each of the interpretation embeddings represents the individual meaning of the sentence. This allows us to compute the similarity between sentences more appropriately, even when a sentence has two or more meanings.

In this study, sentence embeddings are represented by box embeddings (Dasgupta et al., 2020), which represent items as hyperrectangles in a vector space. Intuitively, the box embeddings represent the meaning of a sentence not by a single point but by an area in a high-dimensional space. Then, interpretation embeddings are obtained by measuring the overlap of the box embeddings of the ambiguous sentence and other sentences, such as the sentences between "John and Anna are married" and "John and Anna are married to each other." We propose GumbelCSE for learning box embeddings of sentences; it is based on contrastive learning using natural language inference (NLI) datasets. After obtaining sentence embeddings that include mul-

[1]Our code will be made publicly available upon acceptance.

tiple interpretation embeddings, we also propose a method to extract the interpretation embeddings from the sentence embeddings.

Our proposed method is evaluated by conducting three experiments: Recognizing Textual Entailment (RTE), Entailment Direction Prediction (Yoda et al., 2024), and Ambiguous RTE. The effectiveness of our approach is demonstrated through these experiments.

The contributions of this paper are summarized as follows:

- We introduce a new concept, *interpretation embeddings*, which are the representations of interpretations to handle multiple meanings of a sentence.
- We propose a new sentence embedding method to learn box embeddings of sentences and interpretations.
- We empirically evaluate the effectiveness of our method through three different tasks.

2 Related Work

2.1 Sentence Embeddings

There have been numerous efforts to develop methods for learning sentence embeddings. For example, several methods using NLI datasets were proposed (Conneau et al., 2017; Reimers and Gurevych, 2019). Tsukagoshi et al. (2021) used definition sentences in a dictionary to train sentence embedding models.

Recently, the contrastive learning framework (Chen et al., 2020) has become a popular approach for the learning of sentence embeddings. SimCSE[2] (Gao et al., 2021) is a representative example of this approach that will be explained in detail in subsection 3.1. Several methods utilized SimCSE to obtain enhanced sentence embeddings. Yoda et al. (2024) extended SimCSE to learn Gaussian embeddings of sentences. Li et al. (2024) applied matryoshka representation learning (Kusupati et al., 2022) to learn sentence embeddings, enabling the adjustment of not only the number of embedding dimensions but also the number of layers.

Most recently, large language models (LLMs) have been used to learn sentence embeddings and achieved remarkable results. PromptEOL (Jiang et al., 2024) defines the hidden state of the next token of a prompt, "This sentence: `[text]` means in one word," as the sentence embedding of a sentence given as `[text]`, inspired by Jiang et al. (2022). It also has an in-context learning setting, which uses the definition sentences in a dictionary, inspired by Tsukagoshi et al. (2021).

The above sentence embedding methods define a single representation for a given sentence. In contrast, our method aims to represent a sentence with multiple vector representations.

2.2 Sentence-Level Ambiguity

The ambiguity of a sentence's meaning is an important issue in many NLP tasks, such as question answering (Min et al., 2020), event temporal relation extraction (Hu et al., 2024), text-to-SQL (Bhaskar et al., 2023), and machine translation (Lee et al., 2023; Pilault et al., 2023; Garg et al., 2024).

NLI is also a fundamental task in which the ambiguity of meanings of sentences should be considered. The construction of an NLI dataset is often accompanied by disagreement in the annotation process, which is primarily attributed to ambiguity at the sentence level (Jiang and de Marneffe, 2022). Several attempts have been made to address this issue. Jiang et al. (2023) and Weber-Genzel et al. (2024) created NLI datasets annotated with labels and their corresponding explanations, which provided insight into the rationale behind the chosen labels. Pavlick and Kwiatkowski (2019) and Nie et al. (2020) re-annotated existing NLI datasets with many annotators and analyzed the relation between model performance and inter-annotator agreement. Meissner et al. (2021) and Zhou et al. (2022) proposed the paradigm of predicting the distribution of probabilities of the labels for a given pair of sentences. Liu et al. (2023) created the multi-labeled NLI dataset AMBIENT, which considered the interpretations of the sentences. Havaldar et al. (2025) created an NLI dataset where explicitly entailed, implicitly entailed, contradicted, and neutral hypotheses are associated with a premise. In this study, we use the NLI datasets created by Nie et al. (2020) and Liu et al. (2023) to assess the effectiveness of our interpretation embedding method in handling the ambiguity of a sentence.

3 Proposed Method

We propose a new concept, *interpretation embeddings*, which are the representations of individual

[2]SimCSE has two kinds of settings: unsupervised and supervised. In this paper, the term "SimCSE" refers to the supervised version.

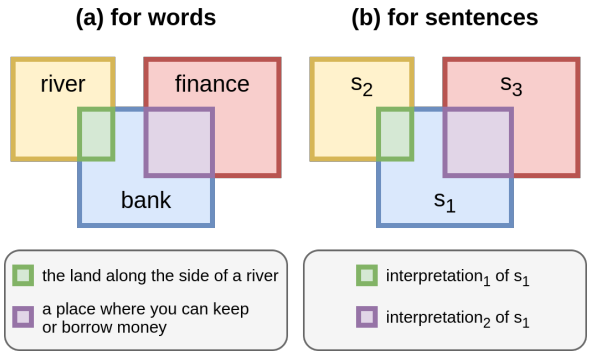

Figure 2: An explanation of interpretation embeddings compared to the situation for words

interpretations of a sentence. In this study, an interpretation embedding is represented by the overlap of the box embeddings (Dasgupta et al., 2020) of two sentences. As shown in Figure 2, in the case of words, an overlap of box embeddings can be regarded as a representation of a word sense. Similarly, in the case of sentences, we propose that the overlap of box embeddings can be regarded as a representation of an interpretation. The box embeddings of words are often studied (Onoe et al., 2021; Dasgupta et al., 2022; Oda et al., 2024), while those of sentences are not. In our proposed method, interpretation embeddings are obtained through two distinct steps. The first step involves training the box embeddings of sentences, which is explained in subsection 3.1. The second step entails retrieving the interpretation embeddings from the trained box embeddings of sentences, which is explained in subsection 3.2.

3.1 Learning Sentence Embeddings

We propose GumbelCSE, a sentence embedding method for learning box embeddings. First, we explain the basic concepts of box embeddings in subsection 3.1.1. Second, we introduce related methods, SimCSE (Gao et al., 2021) and GaussCSE (Yoda et al., 2024), in subsections 3.1.2 and 3.1.3, respectively. Finally, we explain GumbelCSE in subsection 3.1.4.

3.1.1 Box Embeddings

Box embeddings represent items as n-dimensional hyperrectangles. A box embedding $\mathbf{b}$ is constructed from two vectors: a center vector $\mathbf{c}$ and an offset vector $\mathbf{o}$. For each ith dimension, the area of a box embedding is defined as the interval $[c_i - o_i, c_i + o_i]$. Given two box embeddings $\mathbf{b}_x$ and $\mathbf{b}_y$, the asymmetrical similarity between them is defined as follows:

$$P(\mathbf{b}_x|\mathbf{b}_y) = \frac{\mathrm{Vol}(\mathbf{b}_x \cap \mathbf{b}_y)}{\mathrm{Vol}(\mathbf{b}_y)}. \quad (1)$$

Here, $\mathrm{Vol}(\mathbf{b})$ is the function that calculates the volume of $\mathbf{b}$, while $\mathbf{b}_x \cap \mathbf{b}_y$ is the overlap of $\mathbf{b}_x$ and $\mathbf{b}_y$. In this study, Gumbel Box (Dasgupta et al., 2020) is used for the calculation of the volume of box embeddings. More specifically, the Gumbel distribution is employed to calculate the volumes of box embeddings. This prevents the gradient from becoming zero during the training phase, which could occur if there is a lack of overlap between the box embeddings.

3.1.2 SimCSE

SimCSE (Gao et al., 2021) is a representative contrastive learning method for sentence embeddings. BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019) is used as an encoder that produces a vector representation of a sentence. This sentence encoder is fine-tuned utilizing a set of contrastive sentences. Each batch is constituted by M triplets (s_i, s_i^+, s_i^-), where s_i, s_i^+, and s_i^- indicate an instance (sentence), a positive instance for s_i, and a hard negative instance for s_i, respectively. Gao et al. (2021) used the training sets of SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) for constructing the above triplets, namely, using a premise as s_i, its entailment hypothesis as s_i^+, and its contradiction hypothesis as s_i^-. The loss for the ith instance is calculated by

$$-\log \frac{e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{M} \left(e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau} + e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_j^-)/\tau} \right)}, \quad (2)$$

where $\mathbf{h}$ is the embedding of s, $\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_j)$ is the cosine similarity between $\mathbf{h}_i$ and $\mathbf{h}_j$, and τ is the temperature.

3.1.3 GaussCSE

GaussCSE (Yoda et al., 2024) is an extension of SimCSE. It is designed to learn Gaussian embeddings of sentences, whereby each sentence is represented as a Gaussian distribution. A Gaussian embedding N is constructed from two vectors: a mean vector μ and a variance vector σ. These two vectors are the outputs of two linear layers, which are connected to the hidden state of the `[CLS]` token[3] in the final layer of BERT. Gaussian embeddings can

[3]The special token of the beginning of a sentence ⟨s⟩ when RoBERTa is used.

represent asymmetric relations between two sentences s_i and s_j using the following asymmetric similarity score:

$$\mathrm{sim}(s_i||s_j) = \frac{1}{1 + D_{\mathrm{KL}}(N_i||N_j)}. \quad (3)$$

Here, $D_{\mathrm{KL}}(N_i||N_j)$ is the Kullback-Leibler divergence from N_j to N_i.

The configuration of the triplets for training GaussCSE is identical to that of SimCSE, while the loss for the ith instance is calculated using Equation (7):

$$V_E = \sum_{j=1}^{M} e^{\mathrm{sim}(s_j^+||s_i)/\tau}, \quad (4)$$

$$V_C = \sum_{j=1}^{M} e^{\mathrm{sim}(s_j^-||s_i)/\tau}, \quad (5)$$

$$V_R = \sum_{j=1}^{M} e^{\mathrm{sim}(s_i||s_j^+)/\tau}, \quad (6)$$

$$l_i = -\log \frac{e^{\mathrm{sim}(s_i^+||s_i)/\tau}}{V_E + V_C + V_R}. \quad (7)$$

The objective of this loss function is to train Gaussian embeddings so that the similarity $\mathrm{sim}(s_j||s_i)$ becomes close to 1 for a pair (s_i, s_j) of a premise and its entailment hypothesis, while it is 0 for other sentence pairs.

3.1.4 GumbelCSE

We propose GumbelCSE, an extension of SimCSE for learning box embeddings of sentences. A box embedding $\mathbf{b}_s$ of a sentence s is the output of a linear layer, which is connected to the hidden state of the `[CLS]` token in the final layer of BERT. Here, $\mathbf{c}_s$ and $\mathbf{o}_s$ are obtained by splitting $\mathbf{b}_s$ in half. The asymmetric similarity between two box embeddings $\mathbf{b}_{s_i}$ and $\mathbf{b}_{s_j}$ is defined as Equation (1).

The triplets for training GumbelCSE are constructed in the same manner as those of SimCSE and GaussCSE. The loss for the ith instance is defined as Equation (12):

$$V_E = \sum_{j=1}^{M} e^{P(\mathbf{b}_{s_j}^+|\mathbf{b}_{s_i})/\tau}, \quad (8)$$

$$V_C = \sum_{j=1}^{M} e^{P(\mathbf{b}_{s_j}^-|\mathbf{b}_{s_i})/\tau}, \quad (9)$$

$$V_{R_1} = \sum_{j=1}^{M} e^{P(\mathbf{b}_{s_i}|\mathbf{b}_{s_j}^+)/\tau}, \quad (10)$$

$$V_{R_2} = \sum_{j=1}^{M} e^{P(\mathbf{b}_{s_i}|\mathbf{b}_{s_j}^-)/\tau}, \quad (11)$$

$$l_i = -\log \frac{e^{P(\mathbf{b}_{s_i}^+|\mathbf{b}_{s_i})/\tau}}{V_E + V_C + V_{R_1} + V_{R_2}}. \quad (12)$$

The design of this loss function draws inspiration from the work of Yoda et al. (2024). The probability $P(\mathbf{b}_{s_j}|\mathbf{b}_{s_i})$ becomes close to 1 for a pair (s_i, s_j) of a premise and its entailment hypothesis, while it is 0 for other pairs. In addition, a modification is made to obtain better box embeddings of sentences. We add V_{R_2} to learn the relation between a sentence and its hard negative sentence more clearly.

3.2 Extraction of Interpretation Embeddings

Let $\mathbf{b}_s$ be a box embedding of a sentence s. We extract $\mathcal{U}_s$, a set of box embeddings of multiple interpretations of the sentence s, from $\mathbf{b}_s$. As previously stated, we assume that $\mathbf{b}_s$ includes embeddings of multiple interpretations of s, and each interpretation can be represented by an overlap of box embeddings of s and another sentence.

First, a set of reference sentences, denoted as $\mathcal{T}$, is prepared. For each $t_i \in \mathcal{T}$, the overlap of $\mathbf{b}_s$ and $\mathbf{b}_{t_i}$, denoted as $\mathbf{b}_{(s,t_i)}$, is obtained as interpretation (box) embeddings. Obviously, all of $\mathbf{b}_{(s,t_i)}$ does not represent appropriate interpretation embeddings. Therefore, $\mathcal{U}_s$ is formed by the part of $\mathbf{b}_{(s,t_i)}$ that meets the following condition: $P(\mathbf{b}_{(s,t_i)}|\mathbf{b}_s)$ is greater than α_1 and smaller than α_2. That is, $\mathcal{U}_s$ is formalized as follows:

$$\mathcal{U}_s = \{\mathbf{b}_{(s,t_i)} \mid \alpha_1 < P(\mathbf{b}_{(s,t_i)}|\mathbf{b}_s) < \alpha_2\}. \quad (13)$$

$P(\mathbf{b}_{(s,t_i)}|\mathbf{b}_s)$ measures how much the two box embeddings overlap. α_1 and α_2 are hyperparameters, which are optimized using the development set.

The motivation for our method of extracting interpretation embeddings is as follows. As shown in Figure 3 (b), when the overlap of $\mathbf{b}_s$ and $\mathbf{b}_{t_i}$ is small, the meanings of these two sentences are extremely different, so the overlap may not represent an interpretation of s. As shown in Figure 3 (c), when the overlap of $\mathbf{b}_s$ and $\mathbf{b}_{t_i}$ is large, the meanings of the two sentences are similar and $\mathbf{b}_{(s,t_i)}$ is almost the same as $\mathbf{b}_s$; thus, $\mathbf{b}_{(s,t_i)}$ is unlikely to be an interpretation embedding. When a moderate overlap is found, as shown in Figure 3 (a), we add $\mathbf{b}_{(s,t_i)}$ to $\mathcal{U}_s$.

4 Experiments

Our proposed method is evaluated by three tasks: RTE, Entailment Direction Prediction (Yoda et al., 2024), and Ambiguous RTE. The experimental setups are described first in subsection 4.1; then, the details of the experiments are presented in the following subsections.

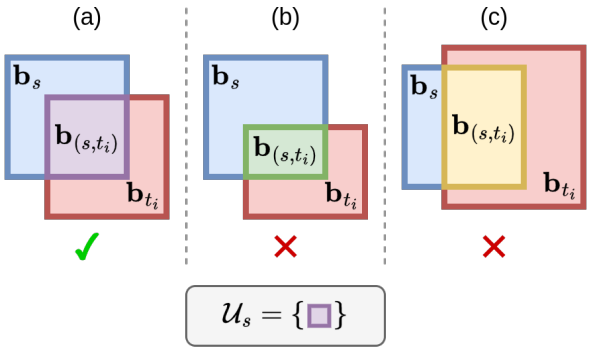

Figure 3: Extraction of interpretation embeddings

4.1 Setup

The pre-trained BERT model (Devlin et al., 2019) `bert-base-uncased`[4] and RoBERTa model (Liu et al., 2019) `roberta-base`[5] are utilized throughout all experiments. The number of dimensions of the output of the linear layer connected to the pre-trained model is set to 128, thereby enabling the training of the 64-dimensional box embeddings.[6]

During the training, the batch size is set to 512, the learning rate is $5e^{-5}$, and the temperature is 0.05; these are the same settings used in the training of SimCSE (Gao et al., 2021). The model is trained for 5 epochs using the training sets of SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) prepared by Gao et al. (2021), which consist of 275,601 triplets in total. The model is validated every 100 steps, and the optimal model is chosen based on Area Under the Precision-Recall Curve (AUPRC) of the RTE task; these are the same settings for the training of GaussCSE (Yoda et al., 2024). For the RTE and Entailment Direction Prediction tasks, we use the model that is optimized using the development set of SNLI. On the other hand, for the Ambiguous RTE task, we use the model that is optimized using MNLI-mismatched[7] because the test set of Ambiguous RTE task includes a part of the development set of SNLI.

4.2 RTE

Task definition RTE is a task that involves classifying a pair consisting of a premise and a hypothesis, $(p,\ h)$, into two classes: entailment or non-entailment.

Datasets Following Yoda et al. (2024), we use the test set of SNLI, MNLI-mismatched[8], and the test set of SICK (Marelli et al., 2014) for evaluation. As they are NLI datasets, the labels "neutral" and "contradiction" are converted to "non-entailment," while "entailment" remains unchanged. The number of instances in the test sets of SNLI and SICK is 10,000 and 4,927, respectively.

Method Following Yoda et al. (2024), GumbelCSE predicts the relation of (p, h) as entailment if $P(\mathbf{b}_h|\mathbf{b}_p)$ is greater than the threshold β; otherwise, it is non-entailment. β is optimized by the development set of SNLI.

Baselines We prepare three baseline models: LINEAR, SimCSE, and GaussCSE. LINEAR is a model that comprises a two-dimensional linear layer connected to the hidden state of the `[CLS]` token in the final layer of BERT or RoBERTa. This is an ordinary fine-tuning method for the RTE task. SimCSE and GaussCSE predict the label in the same way as our model, where the similarity between the premise and hypothesis is measured by the cosine similarity and Equation (3), respectively. Note that all models as well as our GumbelCSE are trained or fine-tuned using the same dataset.[9]

Results The results of the RTE task are shown in Table 1. Comparing three sentence embedding methods, GumbelCSE achieves the best performance on the average of the three datasets for both BERT and RoBERTa. Given that the LINEAR model is fine-tuned for the RTE task, it outperforms the majority of CSE-based methods that learn task-agnostic sentence embeddings. However, GumbelCSE with RoBERTa is better than LINEAR, while GumbelCSE with BERT is almost comparable to LINEAR.

4.3 Entailment Direction Prediction

Task definition Entailment Direction Prediction is a task that involves the prediction of the entailment direction between two given sentences s_1 and s_2. This is a binary classification task, where the

[4] https://huggingface.co/google-bert/bert-base-uncased

[5] https://huggingface.co/FacebookAI/roberta-base

[6] The influence of the number of dimension is addressed in Appendix C.

[7] MNLI provides two development sets, MNLI-matched and MNLI-mismatched, which respectively comprise samples of domains that are consistent and inconsistent with the training data.

[8] Recall that it is one of the development sets in MNLI, consisting of 10,000 samples.

[9] The details of the implementation of the baselines are described in Appendix A and B. These sections also describe the implementation details of the baselines for the Entailment Direction Prediction and Ambiguous RTE tasks.

Model		SNLI	MNLI	SICK	Avg.
B	LINEAR	82.79	74.54	86.02	81.12
	SimCSE	75.53	74.22	71.08	73.61
	GaussCSE*	76.64	**76.85**	83.15	78.88
	GumbelCSE	**81.94**	73.91	**86.71**	**80.85**
R	LINEAR	81.75	73.82	83.13	79.57
	SimCSE	75.03	**77.77**	73.72	75.51
	GaussCSE*	76.37	77.74	82.95	79.02
	GumbelCSE	**80.95**	73.83	**87.52**	**80.77**

Table 1: Accuracy of RTE. "B" and "R" represent BERT and RoBERTa, respectively. * indicates the results from Yoda et al. (2024).

Model		SNLI	MNLI	SICK	Avg.
LENGTH*		92.63	82.64	69.14	81.47
B	GaussCSE*	97.38	91.92	86.22	91.84
	GumbelCSE	**98.01**	**92.93**	**90.18**	**93.70**
R	GaussCSE*	97.44	93.10	88.43	92.99
	GumbelCSE	**98.19**	**93.76**	**90.05**	**94.00**

Table 2: Accuracy of Entailment Direction Prediction. "B" and "R" represent BERT and RoBERTa, respectively. * indicates the results from Yoda et al. (2024).

goal is to determine whether s_1 entails s_2 or s_2 entails s_1.

Datasets We use 3,368, 3,463, and 794 sentence pairs labeled with "entailment" in the test set of SNLI, MNLI-mismatched, and the test set of SICK, respectively. In SICK, the labels for NLI are annotated for each direction of the sentence pairs. Instances labeled with the "entailment" tag for both directions have been excluded, following Yoda et al. (2024).

Method Similar to Yoda et al. (2024), GumbelCSE predicts that s_1 entails s_2 if $P(\mathbf{b}_{s_2}|\mathbf{b}_{s_1})$ is greater than $P(\mathbf{b}_{s_1}|\mathbf{b}_{s_2})$ and vice versa.

Baselines We prepare two baseline models: LENGTH and GaussCSE. LENGTH is a simple rule-based method that predicts that a longer sentence entails a shorter one. GaussCSE predicts the entailment direction in the same way as our model, where the similarity between s_1 and s_2 is measured by Equation (3).

Results The results of the Entailment Direction Prediction task are shown in Table 2. Both GaussCSE and GumbelCSE demonstrate superior performance compared to the naive baseline, LENGTH. Furthermore, GumbelCSE outperforms GaussCSE for all three datasets, substantiating the effectiveness of our GumbelCSE method in capturing asymmetric relations between sentences.

4.4 Ambiguous RTE

Task definition A new task, called Ambiguous RTE, is proposed to evaluate the effectiveness of interpretation embeddings. It is a task that involves classifying a pair consisting of a premise and a hypothesis into one of three classes: entailment, non-entailment, or both. The class "both" means that the relation between a premise and a hypothesis is ambiguous due to multiple interpretations of a sentence.

Datasets We use the test set of AMBIENT (Liu et al., 2023) and ChaosNLI (Nie et al., 2020) for evaluation and MNLI-mismatched for optimizing parameters. In these datasets, multiple NLI labels are given for each sentence pair, considering the ambiguity of the interpretation of a sentence. For example, the pair consisting of the premise "The cat was lost after leaving the house" and the hypothesis "The cat could not find its way" is labeled with both "entailment" and "neutral" (when the premise means "The cat is unable to be found"). These NLI labels are simplified to the three aforementioned coarse classes.

In ChaosNLI and MNLI-mismatched, the labels are voted by 100 and 5 annotators, respectively. Similar to the setting in Jiang and de Marneffe (2022), only the labels supported by 20 votes are used in ChaosNLI, while 2 votes are used in MNLI-mismatched.

The test set of ChaosNLI is divided into ChaosNLI-S and ChaosNLI-M, where the samples are derived from the development set of SNLI and MNLI-matched, respectively. The number of instances in the test set of AMBIENT, ChaosNLI-S, and ChaosNLI-M is 1,545, 1,514, and 1,599, respectively.

Method First, the sets of interpretation embeddings of p and h, $\mathcal{U}_p$ and $\mathcal{U}_h$, are extracted as described in subsection 3.2. Here, $\mathcal{T}$ (the set of reference sentences) is constructed from the n triplets randomly sampled in the training set of GumbelCSE. Second, for all pairs of the interpretation embeddings of p and h, namely $(\mathbf{b}_{(p,t_i)}, \mathbf{b}_{(h,t_j)}) \in \mathcal{U}_p \times \mathcal{U}_h$, $P(\mathbf{b}_{(h,t_j)}|\mathbf{b}_{(p,t_i)})$ is calculated. This probability evaluates how the interpretation embedding

Model		ChaosNLI-S			ChaosNLI-M			AMBIENT		
		ent.	non.	both	ent.	non.	both	ent.	non.	both
BERT	LINEAR	48.69	81.81	38.67	37.52	62.68	40.53	28.17	61.25	25.74
	SimCSE	24.54	70.40	–	34.36	56.72	–	26.50	51.86	–
	GaussCSE	31.81	71.83	–	34.31	57.82	–	28.57	62.49	–
	GumbelCSE-sen	38.63	73.22	–	32.05	56.14	–	32.56	66.83	–
	GumbelCSE-int	27.92	66.06	47.14	30.88	49.72	28.15	26.79	70.28	1.43
RoBERTa	LINEAR	52.71	82.26	35.84	36.36	62.13	37.59	30.25	67.17	18.28
	SimCSE	26.33	71.27	–	34.04	58.45	–	26.87	51.19	–
	GaussCSE	31.74	71.37	–	32.01	57.60	–	27.97	61.13	–
	GumbelCSE-sen	38.94	73.41	–	33.21	57.73	–	30.59	71.35	–
	GumbelCSE-int	31.98	67.33	50.33	32.40	50.95	27.25	29.99	69.41	2.28

Table 3: F1-score of each class for Ambiguous RTE. Note that SimCSE, GaussCSE and GumbelCSE-sen are binary classifiers that do not classify a sample as the "both" class.

$\mathbf{b}_{(h,t_j)}$ subsumes $\mathbf{b}_{(p,t_i)}$, indicating the possibility that p entails h. Finally, (p, h) is classified as follows:

$$\begin{cases} \text{entailment} \\ \quad \text{if } \forall(\mathbf{b}_{(p,t_i)}, \mathbf{b}_{(h,t_j)}) \in \mathcal{U}_p \times \mathcal{U}_h \ P(\mathbf{b}_{(h,t_j)} | \mathbf{b}_{(p,t_i)}) > \beta \\ \text{non-entailment} \\ \quad \text{if } \forall(\mathbf{b}_{(p,t_i)}, \mathbf{b}_{(h,t_j)}) \in \mathcal{U}_p \times \mathcal{U}_h \ P(\mathbf{b}_{(h,t_j)} | \mathbf{b}_{(p,t_i)}) < \beta \\ \text{both} \\ \quad \text{otherwise} \end{cases} . \tag{14}$$

The parameter n, the number of triplets in $\mathcal{T}$, is set to 10,000. The parameters α_1 and α_2 are optimized using the development set through a grid search from 0.5 to 1.0 at intervals of 0.1. Also, β is optimized using the development set.

To evaluate the effectiveness of the use of interpretation embeddings, two methods are compared: GumbelCSE-sen and GumbelCSE-int. GumbelCSE-int is the aforementioned method, while GumbelCSE-sen classifies sentence pairs into entailment or non-entailment classes using not interpretation embeddings but sentence embeddings obtained by GumbelCSE.

Baselines We prepare three baseline models: LINEAR, SimCSE and GaussCSE. LINEAR is a model that comprises a three-dimensional linear layer connected to the hidden state of the `[CLS]` token in the final layer of BERT or RoBERTa. It is fine-tuned in two steps. First, it is fine-tuned by the training set of GumbelCSE, where the label is entailment or non-entailment. Then, it is fine-tuned by MNLI-mismatched, where the label is one of three classes. SimCSE and GaussCSE predict the label in the way explained in subsection 4.2. The parameter β for these baselines is also optimized using the same development data of GumbelCSE.

Results The results of the Ambiguous RTE task are shown in Table 3. The F1-scores of GumbelCSE-int for the "entailment" and "non-entailment" classes are almost comparable to those of GumbelCSE-sen, while GumbelCSE-int is additionally capable of classifying an ambiguous sentence pair as "both." This demonstrates the effectiveness of interpretation embeddings in comprehending the ambiguity of sentences. However, GumbelCSE-int could not outperform LINEAR except for ChaosNLI-S, which is especially fine-tuned for the Ambiguous RTE task. The comparison among SimCSE, GaussCSE, and GumbelCSE-sen is similar to the comparison on the RTE task, i.e., GumbelCSE-sen outperforms SimCSE and GaussCSE in most cases.

For GumbelCSE-int with both BERT and RoBERTa, α_1 and α_2 are optimized to 0.5 and 0.6, respectively, using the development data. This finding validates the effectiveness of our approach to derive interpretation embeddings by employing moderately similar sentences, as discussed in 3.1.4.

5 Analysis

5.1 Ablation Study

An ablation study is conducted to investigate the effectiveness of the components in the loss function in Equation (12). Tables 4 and 5 present the accuracy of the GumbelCSE models trained with several loss functions for the RTE and Entailment Direction Prediction tasks, respectively. These results demonstrate the effectiveness of the newly introduced V_{R_2} for the BERT-based GumbelCSE. In addition, it is found that V_C and V_{R_1} can also contribute to learn better representations.

	Loss function	SNLI	MNLI	SICK	Avg.
B	V_E	76.98	65.14	81.14	74.42
	V_E+V_C	80.53	73.78	84.86	79.72
	$V_E+V_C+V_{R_1}$	80.42	72.88	85.00	79.43
	$V_E+V_C+V_{R_1}+V_{R_2}$	**81.94**	**73.91**	**86.71**	**80.85**
R	V_E	77.17	66.72	81.88	75.26
	V_E+V_C	80.11	74.26	86.81	80.39
	$V_E+V_C+V_{R_1}$	**81.53**	**74.52**	86.97	**81.01**
	$V_E+V_C+V_{R_1}+V_{R_2}$	80.95	73.83	**87.52**	80.77

Table 4: Accuracy of RTE for several loss functions. "B" and "R" represent BERT and RoBERTa, respectively.

	Loss function	SNLI	MNLI	SICK	Avg.
B	V_E	97.98	92.29	86.65	92.31
	V_E+V_C	97.95	91.51	88.29	92.58
	$V_E+V_C+V_{R_1}$	97.80	92.20	88.41	92.81
	$V_E+V_C+V_{R_1}+V_{R_2}$	**98.01**	**92.93**	**90.18**	**93.70**
R	V_E	97.95	93.16	89.17	93.43
	V_E+V_C	97.86	92.52	89.55	93.31
	$V_E+V_C+V_{R_1}$	97.89	**93.94**	**90.18**	**94.00**
	$V_E+V_C+V_{R_1}+V_{R_2}$	**98.19**	93.76	90.05	**94.00**

Table 5: Accuracy of Entailment Direction Prediction for several loss functions.

5.2 Impact of Number of Reference Sentences

In our GumbelCSE method, interpretation embeddings are obtained by measuring the overlap between two box embeddings of the target sentence and reference sentences, where the set of reference sentences is denoted as $\mathcal{T}$. We analyze how the number of reference sentences influences the performance of the Ambiguous RTE task. The number of dimensions of the box embedding is set to 16 to reduce the memory and time costs associated with extracting interpretation embeddings. As mentioned in subsection 4.4, $\mathcal{T}$ is formed by sentences in triplets randomly sampled from the training data. The number of triplets, n, is varied over {5,000, 10,000, 50,000, 100,000, 200,000}. Since each triplet comprises three sentences and duplicate sentences are removed, the number of reference sentences ($|\mathcal{T}|$) is approximately $3 \times n$. The parameter α_1 is changed from 0.5 to 0.9 with a step size of 0.1, while α_2 is fixed at 1.0 to reduce the computational time required for analysis.

Figure 4 presents the macro-F1-scores of the Ambiguous RTE task of the models with different settings where BERT is used as the base model. The best macro-F1-score is obtained when $n = 10,000$ and $\alpha_1 = 0.6$. This demonstrates that a large number of reference sentences is not necessary to obtain a sufficient number of appropriate interpretation embeddings, resulting in the reduction of the computational cost. When α_1 is set to a relatively small value (i.e., 0.5), the macro-F1-score is significantly reduced as n increases. This is because the increase in the number of interpretation embeddings provides the opportunity for the "otherwise" condition in Equation (14) to be fulfilled, resulting in a substantial bias towards the "both" class. In contrast, when α_1 is set to a large value, the performance of the Ambiguous RTE task remains stable with respect to the number of reference sentences, due to the decrease in the number of interpretation embeddings.

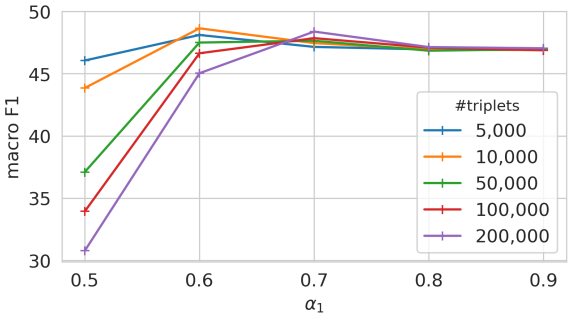

Figure 4: The macro-F1-scores obtained while varying α_1 from 0.5 to 0.9 in five settings

6 Conclusion

In this paper, we introduced a new concept *interpretation embeddings*, which represent the interpretations of a sentence. The interpretation embedding is created by overlapping the box embeddings of two sentences. Furthermore, we proposed GumbelCSE, which is a contrastive learning method for learning box embeddings of sentences, and the method for extracting interpretation embeddings of a sentence from the box embedding of a sentence. We evaluated our method on three tasks: RTE, Entailment Direction Prediction, and Ambiguous RTE. On the RTE and Entailment Direction Prediction tasks, GumbelCSE outperformed other sentence embedding methods in most cases. On the Ambiguous RTE task we proposed, it was demonstrated that interpretation embeddings are effective for understanding the multiple interpretations of a sentence. In the future, we plan to apply our method to more challenging tasks such as understanding metaphors or pragmatics.

Limitations

The bottleneck of our method is the substantial memory and time required for calculating the overlap of box embeddings to obtain interpretation embeddings. To mitigate this problem, the number of dimensions of box embeddings is set to a relatively low value (i.e., 16) in this paper. However, increasing this value could facilitate the representation of more subtle meanings of sentences. Another limitation is that our method has not yet been applied to real applications such as information retrieval.

References

Adithya Bhaskar, Tushar Tomar, Ashutosh Sathe, and Sunita Sarawagi. 2023. Benchmarking and improving text-to-SQL generation under ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7053–7074, Singapore. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 1597–1607. PMLR.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Shib Dasgupta, Michael Boratko, Siddhartha Mishra, Shriya Atmakuri, Dhruvesh Patel, Xiang Li, and Andrew McCallum. 2022. Word2Box: Capturing set-theoretic semantics of words using box embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2263–2276, Dublin, Ireland. Association for Computational Linguistics.

Shib Dasgupta, Michael Boratko, Dongxu Zhang, Luke Vilnis, Xiang Li, and Andrew McCallum. 2020. Improving local identifiability in probabilistic box embeddings. In *Advances in Neural Information Processing Systems*, volume 33, pages 182–192. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sarthak Garg, Mozhdeh Gheini, Clara Emmanuel, Tatiana Likhomanenko, Qin Gao, and Matthias Paulik. 2024. Generating gender alternatives in machine translation. In *Proceedings of the 5th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 237–254, Bangkok, Thailand. Association for Computational Linguistics.

Shreya Havaldar, Hamidreza Alvari, John Palowitch, Mohammad Javad Hosseini, Senaka Buthpitiya, and Alex Fabrikant. 2025. Entailed between the lines: Incorporating implication into nli. *Preprint*, arXiv:2501.07719.

Yutong Hu, Quzhe Huang, and Yansong Feng. 2024. Only one relation possible? modeling the ambiguity in event temporal relation extraction. *Preprint*, arXiv:2408.07353.

Nan-Jiang Jiang and Marie-Catherine de Marneffe. 2022. Investigating reasons for disagreement in natural language inference. *Transactions of the Association for Computational Linguistics*, 10:1357–1374.

Nan-Jiang Jiang, Chenhao Tan, and Marie-Catherine de Marneffe. 2023. Ecologically valid explanations for label variation in NLI. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10622–10633, Singapore. Association for Computational Linguistics.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2024. Scaling sentence embeddings with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3182–3196, Miami, Florida, USA. Association for Computational Linguistics.

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022. Prompt-BERT: Improving BERT sentence embeddings with prompts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8826–8837, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2022. Matryoshka representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 30233–30249. Curran Associates, Inc.

Jaechan Lee, Alisa Liu, Orevaoghene Ahia, Hila Gonen, and Noah Smith. 2023. That was the last straw, we need more: Are translation systems sensitive to disambiguating context? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4555–4569, Singapore. Association for Computational Linguistics.

Xianming Li, Zongxi Li, Jing Li, Haoran Xie, and Qing Li. 2024. Ese: Espresso sentence embeddings. *Preprint*, arXiv:2402.14776.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah Smith, and Yejin Choi. 2023. We're afraid language models aren't modeling ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 790–807, Singapore. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Johannes Mario Meissner, Napat Thumwanit, Saku Sugawara, and Akiko Aizawa. 2021. Embracing ambiguity: Shifting the training target of NLI models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 862–869, Online. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143, Online. Association for Computational Linguistics.

Kohei Oda, Kiyoaki Shirai, and Natthawut Kertkeidkachorn. 2024. Learning contextualized box embeddings with prototypical networks. In *Proceedings of the 9th Workshop on Representation Learning for NLP (RepL4NLP-2024)*, pages 1–12, Bangkok, Thailand. Association for Computational Linguistics.

Yasumasa Onoe, Michael Boratko, Andrew McCallum, and Greg Durrett. 2021. Modeling fine-grained entity types with box embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2051–2064, Online. Association for Computational Linguistics.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Jonathan Pilault, Xavier Garcia, Arthur Bražinskas, and Orhan Firat. 2023. Interactive-chain-prompting: Ambiguity resolution for crosslingual conditional generation with interaction. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–483, Nusa Dua, Bali. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Hayato Tsukagoshi, Ryohei Sasano, and Koichi Takeda. 2021. DefSent: Sentence embeddings using definition sentences. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 411–418, Online. Association for Computational Linguistics.

Leon Weber-Genzel, Siyao Peng, Marie-Catherine De Marneffe, and Barbara Plank. 2024. VariErr NLI: Separating annotation error from human label variation. In *Proceedings of the 62nd Annual Meeting of*

the Association for Computational Linguistics (Volume 1: Long Papers), pages 2256–2269, Bangkok, Thailand. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Shohei Yoda, Hayato Tsukagoshi, Ryohei Sasano, and Koichi Takeda. 2024. Sentence representations via Gaussian embedding. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 418–425, St. Julian's, Malta. Association for Computational Linguistics.

Xiang Zhou, Yixin Nie, and Mohit Bansal. 2022. Distributed NLI: Learning to predict human opinion distributions for language reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 972–987, Dublin, Ireland. Association for Computational Linguistics.

A Implementation Details of Baselines

SimCSE is implemented with the same settings as the original paper (Gao et al., 2021). The epoch, the batch size, the learning rate, and the temperature are set to 3, 512, $5e^{-5}$, and 0.05, respectively. We validate the model every 250 steps on the development set of STS-B (Cer et al., 2017) and choose the best checkpoint. The best validation score (Spearman's correlation) achieved by our implemented SimCSE, where BERT serves as the base model, is 86.1, which is almost the same as the score 86.2 described in (Gao et al., 2021).

GaussCSE is also implemented with the same settings as the original paper (Yoda et al., 2024). The epoch, the learning rate, and the temperature are set to 3, 16, $5e^{-5}$, and 0.05, respectively. The dimensions of both the mean vector and the variance vector are set to 768. We validate the model every 3200 steps on the RTE task using the development set of SNLI and choose the best checkpoint. The best validation scores (AUPRC) achieved by our implemented GaussCSE based on BERT and RoBERTa are 66.19 and 66.91, respectively. These scores are almost the same as 66.21 and 66.31 reported in (Yoda et al., 2024).

LINEAR in the RTE task is implemented as follows. The epoch, the batch size, and the learning rate are set to 5, 512, and $5e^{-5}$, respectively. The cross-entropy loss is chosen as the loss function. We validate the model every 100 steps on the RTE task using the development set of SNLI and choose the best checkpoint. The best validation scores (accuracy) of the LINEAR models obtained by fine-tuning BERT and RoBERTa are 83.18 and 82.01, respectively.

LINEAR in the Ambiguous RTE task is implemented as follows. First, the model is trained using relatively large SNLI and MNLI datasets with the same settings as for the RTE task. Second, it is re-trained using the small MNLI-mismatched dataset. The epoch, the batch size, and the learning rate are set to 5, 128, and $5e^{-5}$, respectively. The cross-entropy loss is chosen as the loss function. In the second training phase, the model is not validated using the development data; instead, the model obtained after the final epoch is used.

Box dim.	16	32	64	128	256
AUPRC	75.86	77.41	**77.95**	77.10	76.23

Table 6: Validation scores for several dimensions of box embeddings

B Training Time

All models described in section 4 are trained using a single GPU, NVIDIA RTX A6000 48GB. The time required to complete the training of GumbelCSE, GaussCSE, and SimCSE are about 120, 200, and 20 minutes, respectively. The fine-tuning of LINEAR in the RTE task takes approximately 60 minutes, while the second fine-tuning of LINEAR in the Ambiguous RTE task using the MNLI-mismatched dataset takes about one minute.

C Influence of Dimensions of Box Embeddings

This section investigates how the dimensions of box embeddings influences the performance of GumbelCSE. Table 6 shows AUPRC of GumbelCSE, where BERT is used as the base model, in the validation data of the RTE task when the dimensions of $\mathbf{b}_s$ is set to {16, 32, 64, 128, 256}. It is found that a moderate size of box embeddings, specifically 64, achieves the best AUPRC.