Representing Sentence Interpretations with Overlapping Box Embeddings

Anonymous ACL submission

Abstract

Most of the previous studies on sentence embeddings aim to obtain one representation per sentence. However, this approach is inadequate for handling the relations between sentences in cases where a sentence has multiple interpretations. To address this problem, we propose a novel concept, *interpretation embeddings*, which are the 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 four tasks: Recognizing Textual Entailment (RTE), Entailment Direction Prediction, Ambiguous RTE, and Conditional Semantic Textual Similarity (C-STS). In the RTE and Entailment Direction Prediction tasks, GumbelCSE outperforms other sentence embedding methods in most cases. In the Ambiguous RTE and C-STS tasks, 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, which 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.

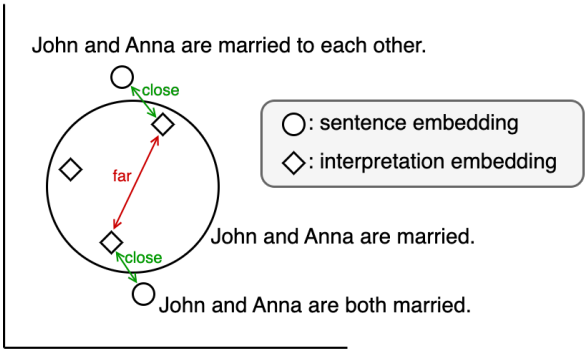

Figure 1: Conceptual diagram of interpretation embeddings

To address this problem, we propose *interpretation embeddings*, which are the 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, which is based on contrastive learning using Natural Language Inference (NLI) datasets. After obtaining sentence embeddings that include

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

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

Our proposed method is evaluated by conducting four experiments: Recognizing Textual Entailment (RTE), Entailment Direction Prediction (Yoda et al., 2024), Ambiguous RTE, and Conditional Semantic Textual Similarity (C-STS) (Deshpande et al., 2023). 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 four 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 one, which will be explained in detail in subsection 3.1. Several methods followed 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 learning sentence embeddings, enabling the adjustment of not only the number of embedding dimensions but also that of the layers.

Most recently, Large Language Models (LLMs) have been used for learning sentence embeddings, such as PromptEOL (Jiang et al., 2024). It 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 the 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

Ambiguity of a sentence 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).

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) created datasets that were annotated by many subjects. 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. In this study, we use AMBIENT to assess the effectiveness of our interpretation embedding method in handling the ambiguity of a sentence.

Semantic Textual Similarity (STS) is a task to predict the similarity between two sentences. Recently, Deshpande et al. (2023) proposed a Conditional Semantic Textual Similarity (C-STS) task, which aimed to predict sentence similarity under a condition indicated by a short sentence. Given the necessity of considering multiple interpretations in the C-STS task, we evaluate the quality of the interpretation embeddings obtained by the proposed method concerning this task.

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

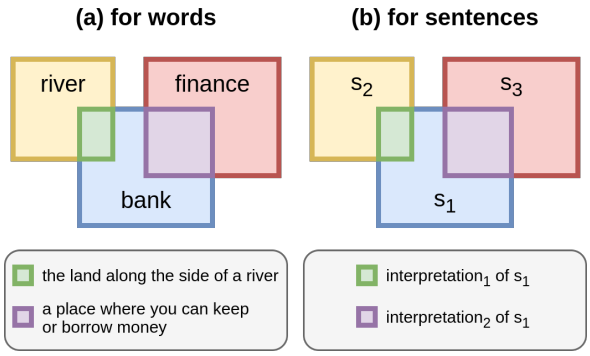

Figure 2: An explanation of interpretation embeddings comparing the situation for words

3 Proposed Method

We propose a new concept, *interpretation embeddings*, which are the representations of individual 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 be regarded as interpretation embeddings. 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 by 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 of Sentence Embeddings

We propose GumbelCSE, a sentence embedding method to learn box embeddings. First, we explain the basic concepts of box embeddings in 3.1.1. Second, we introduce related methods: SimCSE and GaussCSE in 3.1.2 and 3.1.3, respectively. Finally, we explain GumbelCSE in 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 due to the 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^- mean an instance (sentence), a positive instance for s_i, and a hard negative instance for s_i, respectively. Gao et al. (2021) use the training set 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 `[CLS]` in the final layer of BERT or the beginning-of-sentence token `<s>` in RoBERTa. Gaussian embeddings can represent asymmetric relationships 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 is calculated as 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 between two sentences becomes close to 1 for a pair of a premise and its entailment hypothesis, while 0 for other sentence pairs.

3.1.4 GumbelCSE

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

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

$$V_E = \sum_{j=1}^{M} e^{P(\mathbf{b}_j^+|\mathbf{b}_i)/\tau} \quad (8)$$

$$V_C = \sum_{j=1}^{M} e^{P(\mathbf{b}_j^-|\mathbf{b}_i)/\tau} \quad (9)$$

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

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

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

The design of this loss function draws inspiration from the work of Yoda et al. (2024). The probability $P(\mathbf{b}_i|\mathbf{b}_j)$ becomes close to 1 for a pair of a premise and its entailment hypothesis, while 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.

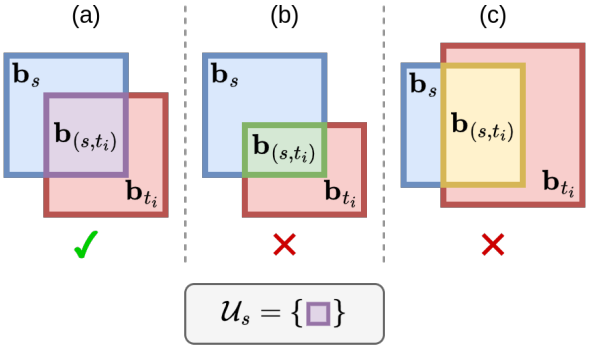

Figure 3: Extraction of interpretation embeddings

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 $\mathbf{b}_{(s,t_i)}$ are not appropriate interpretation embeddings. Therefore, $\mathcal{U}_s$ is formed by $\mathbf{b}_{(s,t_i)}$ 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 $\mathbf{b}_{t_i}$ are 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 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 the moderate overlap is found, as shown in Figure 3 (a), we add $\mathbf{b}_{(s,t_i)}$ to $\mathcal{U}_s$.

4 Experiments

Four experiments are conducted to evaluate GumbelCSE: RTE, Entailment Direction Prediction (Yoda et al., 2024), Ambiguous RTE, and C-STS (Deshpande et al., 2023). The experimental setups are described first in subsection 4.1, then the details of the experiments are presented in the following subsections.

4.1 Setup

The pre-trained BERT model (Devlin et al., 2019) `bert-base-uncased`[3] is utilized through all experiments. The number of dimensions of the output of the linear layer connected to the BERT model is set to 32, thereby enabling the training of the 16-dimensional box embeddings. This low-dimensional setting aims to reduce the memory and time costs associated with extracting interpretation embeddings.

During the training, the batch size is set to 512, the learning rate is $5e^{-5}$, and the temperature is 0.05, which are the same setting used in the training of SimCSE (Gao et al., 2021). The model is trained 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 hyperparameters are optimized using a development set of the RTE task. The model is validated every 100 steps, and the optimal model is chosen based on the Area Under the Curve (AUC) of the precision and the recall of the RTE task, which is the same setting as Yoda et al. (2024). The development set of SNLI is employed for the RTE and Entailment Direction Prediction tasks, while that of MNLI-mismatched[4] is utilized for the Ambiguous RTE and C-STS tasks. The number of instances in each of the development sets of SNLI and MNLI-mismatched is 10,000.

4.2 RTE

Task definition RTE is a task of classifying a pair 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[5], 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 set of SNLI and SICK is 10,000 and 4,927, respectively.

Model	SNLI	MNLI	SICK	Avg.
LINEAR	82.79	74.54	86.02	81.12
SimCSE*	74.96	**78.18**	86.11	79.75
GaussCSE*	76.64	76.85	83.15	78.88
GumbelCSE	**80.25**	73.74	**87.05**	**80.35**

Table 1: Accuracy of RTE. * indicates the results from Yoda et al. (2024).

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 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 `[CLS]` in the final layer of BERT. This is an ordinary BERT model fine-tuned for the RTE task. SimCSE predicts the label in the same way as our model, where the similarity between the premise and hypothesis is measured by the cosine similarity. Note that all models are trained or fine-tuned using the same dataset used to train SimCSE.

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, followed by SimCSE and GaussCSE. Given that the LINEAR model is fine-tuned for the RTE task, it outperforms the other CSE-based methods that learn task-agnostic sentence embeddings. However, GumbelCSE is almost comparable to LINEAR.

4.3 Entailment Direction Prediction

Task definition Entailment Direction Prediction is a task to predict the entailment direction between two given sentences s_1 and s_2. This is a binary classification task of which the 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

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

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

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

Model	SNLI	MNLI	SICK	Avg.
LENGTH*	92.63	82.64	69.14	81.47
GaussCSE*	97.38	91.92	86.22	91.84
GumbelCSE	**98.10**	**92.41**	**89.67**	**93.39**

Table 2: Accuracy of Entailment Direction Prediction. * indicates the results from Yoda et al. (2024).

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 a longer sentence entails a shorter one.

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 in capturing asymmetric relations between sentences.

4.4 Ambiguous RTE

Task definition Ambiguous RTE is a task to classify a pair of a premise and a hypothesis into one of the 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 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 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 $\mathbf{b}_{(h,t_j)}$ 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 α_1 and α_2 are optimized using the development set by the grid search from 0.5 to 1.0 at intervals of 0.1. Also, β and n are 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 using not interpretation embeddings but sentence embeddings obtained by GumbelCSE.

Baselines We prepare two baseline models: LINEAR and SimCSE. LINEAR is a model that comprises a three-dimensional linear layer connected to the hidden state of `[CLS]` in the final layer of BERT. It is fine-tuned by 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

Model	ChaosNLI-S			ChaosNLI-M			AMBIENT		
	ent.	non.	both	ent.	non.	both	ent.	non.	both
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	–
GumbelCSE-sen	37.50	73.90	–	35.40	55.88	–	28.26	68.91	–
GumbelCSE-int	28.57	71.64	46.25	34.33	54.95	27.17	27.63	67.94	3.27

Table 3: F1 score of each class for Ambiguous RTE

is one of the three classes. SimCSE predicts the label in the same way explained in subsection 4.2.

Results The results of the Ambiguous RTE task are shown in Table 3. Note that SimCSE and GumbelCSE-sen are binary classifiers that do not classify a sample as the "both" class. The F1-scores of GumbelCSE-int for the "entailment" and "non-entailment" classes are almost comparable to those of GumbelCSE-sen (except for "entailment" in ChaosNLI-S), 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, which is especially fine-tuned for the Ambiguous RTE task. A comparison between SimCSE and GumbelCSE-sen is similar to a comparison between SimCSE and GumbelCSE in the RTE task. GumbelCSE-sen outperforms SimCSE in most cases, which is consistent with the results shown in Table 1.

4.5 C-STS

Task definition C-STS is a task to predict the similarity between two sentences s_1 and s_2 based on a condition c expressed by a short sentence. For example, the similarity between the following two sentences should be estimated high for the condition "The motion of the ball.", but low for the condition "The size of the ball." (Deshpande et al., 2023).

s_1: The NBA player shoots a three-pointer.
s_2: A man throws a tennis ball into the air to serve.

Datasets The development set of C-STS (Deshpande et al., 2023) and Linguistically C-STS (Tu et al., 2024), called LC-STS in this paper, are used for evaluation. LC-STS is created by re-annotating the development set of C-STS. The number of instances in the development set of C-STS and LC-STS is 2,834 and 2,620, respectively.

Method First, the set of interpretation embeddings of s_1 and s_2, $\mathcal{U}_{s_1}$ and $\mathcal{U}_{s_2}$, are extracted using the training set of C-STS as $\mathcal{T}$. Second, $\mathbf{b}'_{s1}$, an interpretation embedding of s_1 that is the most similar to the sentence embedding of c, is selected as shown in Equation (15). Here, sim is the symmetrical similarity of two box embeddings, which is defined as Equation (16).

$$\mathbf{b}'_{s1} = \mathrm{argmax}_{\mathbf{b}_{(s_1,t_i)} \in \mathcal{U}_{s_1}} \mathrm{sim}(\mathbf{b}_{(s_1,t_i)}, \mathbf{b}_c) \quad (15)$$

$$\mathrm{sim}(\mathbf{b}_x, \mathbf{b}_y) = \frac{P(\mathbf{b}_x|\mathbf{b}_y) + P(\mathbf{b}_y|\mathbf{b}_x)}{2} \quad (16)$$

The same process is applied to choose $\mathbf{b}'_{s2}$. Finally, the similarity between $\mathbf{b}'_{s1}$ and $\mathbf{b}'_{s2}$ is calculated as $\mathrm{sim}(\mathbf{b}'_{s_1}, \mathbf{b}'_{s_2})$. In addition to this method (denoted as GumbelCSE-con), we also evaluate another method, GumbelCSE-sen, which predicts the similarity between s_1 and s_2 as $\mathrm{sim}(\mathbf{b}_{s_1}, \mathbf{b}_{s_2})$ without the use of c.

The parameters α_1 and α_2 are optimized using the development set by the grid search from 0.5 to 1.0 at intervals of 0.1.

Baselines We prepare two baseline models: SimCSE-sen and SimCSE-con. SimCSE-sen is a model that calculates the cosine similarity of the sentence embeddings of s_1 and s_2 encoded by SimCSE without c. SimCSE-con is a model that calculates the cosine similarity of the sentence embeddings of "s_1 `[SEP]` c" and "s_2 `[SEP]` c" encoded by SimCSE, which is called as "bi-encoder" in Deshpande et al. (2023).

Results The results of the C-STS task are shown in Table 4. GumbelCSE-con demonstrates the best performance on both datasets and evaluation metrics. This indicates that interpretation embeddings are an appropriate approach for the C-STS task.

A comparison of the models that consider the condition or not reveals that SimCSE-con unexpectedly performs poorer than SimCSE-sen despite the condition being taken into account. This may be because the insertion of the `[SEP]` harms the quality

Model	C-STS		LC-STS	
	Spear.	Pears.	Spear.	Pears.
SimCSE-sen	4.40	5.07	7.49	8.69
SimCSE-con	2.76	3.57	6.33	7.59
GumbelCSE-sen	6.75	7.13	10.46	**11.41**
GumbelCSE-con	**7.39**	**7.80**	**10.47**	**11.41**

Table 4: Spearman and Pearson correlations of C-STS

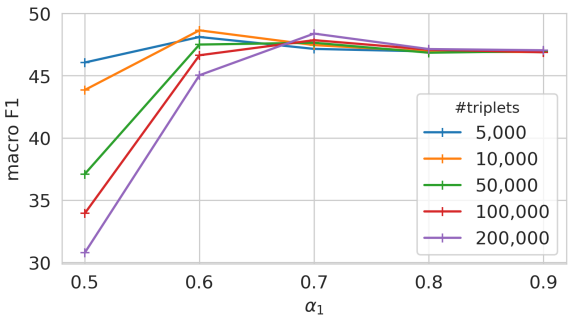

Figure 4: The macro F1 scores while varying α_1 from 0.5 to 0.9 in five settings

of the sentence embeddings of SimCSE. In contrast, GumbelCSE-con outperforms GumbelCSE-sen for the C-STS datasets, demonstrating the advantage of our method in terms of its ability to handle multiple interpretations of a sentence.

The performance of GumbelCSE-sen and GumbelCSE-con is almost the same for the LC-STS dataset. By the grid search, α_1 is determined to be 0.9 and α_2 to be 1.0 for GumbelCSE-con. It means that the number of extracted interpretation embeddings is relatively limited, suggesting that GumbelCSE-con is almost equivalent to GumbelCSE-sen, which handles the sentence embeddings only.

Additionally, GumbelCSE-sen outperforms SimCSE-sen for both datasets. It demonstrates that the box embeddings are a more appropriate representation of sentences than the single vectors for the C-STS task.

5 Analysis of Impact on 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. As mentioned in subsection 4.4, $\mathcal{T}$ is formed by sentences in triplets randomly sampled from the training data. The number of the triplets, n, is varied over {5,000, 10,000, 50,000, 100,000, 200,000}. Since each triplet comprises three sentences and duplicated sentences are removed, the number of reference sentences ($|\mathcal{T}|$) can be 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 shows that the macro F1 score of the Ambiguous RTE task of the models with different settings. The best 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 costs. When α_1 is set to a relatively small value (i.e., 0.5), the macro F1 score is significantly reduced as n is increased. 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 concerning the number of reference sentences, due to the decrease in the number of interpretation embeddings.

6 Conclusion

In this paper, we introduced a new concept *interpretation embeddings*, which represented the interpretations of a sentence. The interpretation embedding was created by overlapping the box embeddings of two sentences. Furthermore, we proposed GumbelCSE, which was a contrastive learning method for learning box embeddings of sentences, and the method for extracting interpretation embeddings from the box embedding of a sentence. We evaluated our method on four tasks: RTE, Entailment Direction Prediction, Ambiguous RTE, and C-STS. In the RTE and Entailment Direction Prediction tasks, GumbelCSE outperformed other sentence embedding methods in most cases. In the Ambiguous RTE and C-STS tasks, 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 the understanding of 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.

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.

Ameet Deshpande, Carlos Jimenez, Howard Chen, Vishvak Murahari, Victoria Graf, Tanmay Rajpurohit, Ashwin Kalyan, Danqi Chen, and Karthik Narasimhan. 2023. C-STS: Conditional semantic textual similarity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5669–5690, Singapore. Association for Computational Linguistics.

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.

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. PromptBERT: 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.

Jingxuan Tu, Keer Xu, Liulu Yue, Bingyang Ye, Kyeongmin Rim, and James Pustejovsky. 2024. Linguistically conditioned semantic textual similarity. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1161–1172, Bangkok, Thailand. 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.