C-STS: Conditional Semantic Textual Similarity

Ameet Deshpande[⋆1,2] **Carlos E. Jimenez**[⋆1] **Howard Chen**[1]
Vishvak Murahari[1] **Victoria Graf**[1] **Tanmay Rajpurohit**[3]
Ashwin Kalyan[2] **Danqi Chen**[1] **Karthik Narasimhan**[1]

[1]Princeton University [2]The Allen Institute for AI [3]Georgia Tech
asd@cs.princeton.edu

Abstract

Semantic textual similarity (STS) has been a cornerstone task in NLP that measures the degree of similarity between a pair of sentences, with applications in information retrieval, question answering, and embedding methods. However, it is an inherently ambiguous task, with the sentence similarity depending on the specific aspect of interest. We resolve this ambiguity by proposing a novel task called conditional STS (C-STS) which measures similarity conditioned on an aspect elucidated in natural language (hereon, *condition*). As an example, the similarity between the sentences *"The NBA player shoots a three-pointer."* and *"A man throws a tennis ball into the air to serve."* is higher for the condition *"The motion of the ball."* (both upward) and lower for *"The size of the ball."* (one large and one small). C-STS's advantages are two-fold: (1) it reduces the subjectivity and ambiguity of STS, and (2) enables fine-grained similarity evaluation using diverse conditions. C-STS contains almost 20,000 instances from diverse domains and we evaluate several state-of-the-art models to demonstrate that even the most performant fine-tuning and in-context learning models (GPT-4, Flan, SimCSE) find it challenging, with Spearman correlation scores of < 50. We encourage the community to evaluate their models on C-STS to provide a more holistic view of semantic similarity and natural language understanding.[1]

1 Introduction

Natural language processing as a field has evolved over the years, with the architectures and training methods undergoing paradigm shifts (Elman, 1990; Hochreiter and Schmidhuber, 1997; Vaswani et al., 2017). But carefully designed evaluation tasks have stood the test of time and fueled model improvement over the years. Semantic textual similarity (STS) is one such instance introduced to measure the semantic similarity between two sentences. Several diverse STS datasets are popularly used, with expansions to multiple domains and languages (Agirre et al., 2012, 2016; Cer et al., 2017; Abdalla et al., 2021). It has also received attention in natural language understanding benchmarks (Wang et al., 2018) and has been a key driver for innovations in sentence embeddings (Gao et al., 2021).

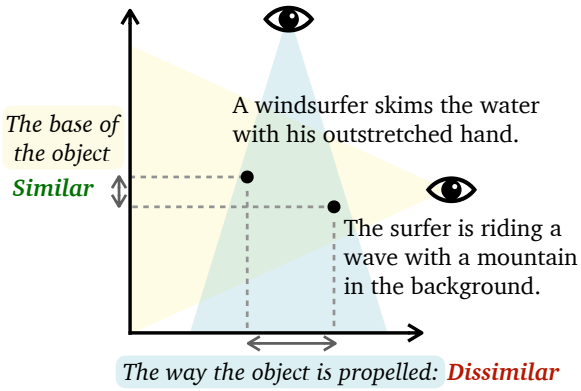

Figure 1: The C-STS task: two sentences are judged by their similarities based on free-form natural language conditions. The two sentences are more similar when judged by the condition 'The base of the object.' (yellow) as both windsurfing and surfing use a similar board but are dissimilar when judged by the condition 'The way the object is propelled.' (blue) because one is propelled by waves and the other by wind. Providing the conditions reduces the ambiguity of the task, and allows us to evaluate a grounded and multi-faceted notion of sentence similarity.

However, STS is inherently ill-defined, since the similarity between two sentences can vary wildly based on the attributes under focus (Figure 1). As noted in several studies, ambiguity in similarity judgement of pairs can be reduced with the help of context both for humans (De Deyne et al., 2016a,b) and models (Veit et al., 2017; Ye et al., 2022a; Lopez-Gazpio et al., 2017; Camburu et al., 2018). With the importance of STS both historically and in the evaluation of current models, we propose a

⋆: Equal Contribution
[1]Code: github.com/princeton-nlp/c-sts

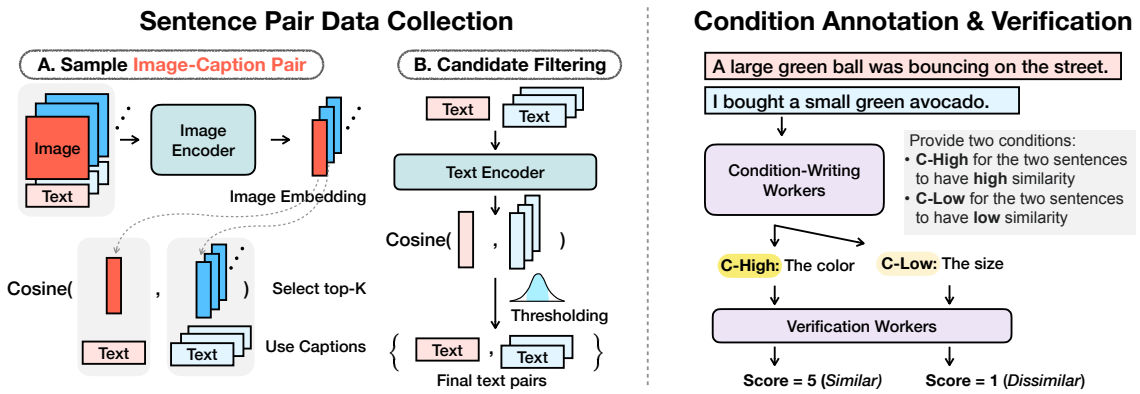

Figure 2: Illustrating the data collection process for collecting C-STS-2023. (Left) We first illustrate the sentence pair collection (§2.2.1) procedure. Step A: An image-caption pair is first sampled (red) from the dataset and the image is then fed into the image encoder to get the image embedding. The image embedding is compared against all other image embeddings in the dataset (blue) to find the top-K similar images. The original caption is then paired with the corresponding captions of the top-K similar images to generate sentence pairs. Step B: The sentence pairs are filtered based on textual similarity. (Right) We illustrate the condition annotation/verification (§2.2.2) procedure. Once the sentence pairs have been collected, they are sent to qualified Mechanical Turkers to get annotations and verify conditions.

new task called *conditional* STS (C-STS), which remedies this ambiguity by measuring the similarity as viewed through the lens of a third sentence, which is referenced as the *condition* (Figure 1).

C-STS, through the use of free-form natural language conditions, enables us to evaluate and probe natural language understanding for a myriad of fine-grained aspects. Figure 1 illustrates two conditions ("The base of the object" and "The way the object is propelled") which consequently help us evaluate sentence similarity of different aspects concerning water sports. Since conditions themselves are unconstrained grammatically correct sentences, they allow us to evaluate a precise, grounded, and multi-faceted notion of sentence similarity.

To comprehensively test models on C-STS, we create the C-STS-2023 dataset which includes almost 20,000 instances containing sentence pairs, a condition, and a scalar similarity judgement on the Likert scale (Likert, 1932). We find that even state-of-the-art sentence encoders and in-context models perform poorly on our task. While SimCSE and GPT-4 are among the best-performing models, their relatively poor Spearman's correlation score of 47.5% and 43.6% respectively, points to significant room for improvement (the performance on STS is typically saturated in the high eighties). We also propose new tri-encoder model and a quadruplet training loss which allows us to perform contrastive learning based on different conditions for the same sentence pair, and believe that C-STS should be tackled both with improved architectures and fine-tuning strategies. Our qualitative analysis shows that models find C-STS challenging when tested on different aspects of the same sentence pair rather than testing an unconditional and ambiguous notion of similarity. We hope that in addition to STS, future works evaluate on the C-STS task to comprehensively and benchmark semantic similarity assessment in language models.

2 Methodology

The C-STS task requires sentence pairs, conditions which probe different aspects of similarity, and the similarity annotation for a given sentence pair and condition. We detail the technical details involved in creating our dataset.

2.1 Background: Semantic textual similarity

Semantic textual similarity (STS) (Agirre et al., 2012, 2013, 2014, 2015, 2016; Cer et al., 2017) is a task which requires machines to make similarity judgements between a pair of sentences ($\{s_1, s_2\}$). STS measures the unconditional semantic similarity between sentences because the annotator making the similarity assessment needs to *infer* what aspect of the sentences are being referred to. Formally, consider conditions ($c_i \in \mathcal{C}$) which reference different aspects of the sentences, then the

similarity of the two sentences is:

$$\frac{1}{\mathcal{C}}\sum_{i=1}^{\mathcal{C}} w_i \, \text{sim}_{c_i}(s_1, s_2), \quad \sum_i w_i = 1$$

Here, w_i is the weight assigned by the annotator to the condition c_i, and $\text{sim}_{c_i}(s_1, s_2)$ is the similarity of the sentences with respect to the condition. These weights are latent to the task and each annotator has their own interpretation of them which helps marginalize similarity, thus introducing ambiguity in the task. To remedy this, we introduce C-STS, which measures similarity conditionally based on an aspect that can be specified as a free-form natural language sentence.

2.2 Conditional semantic textual similarity

C-STS is a task comprising quadruplets containing two sentences, a natural language condition referencing an aspect of the conditions, and a similarity assessment ($\{s_1, s_2, c, \text{sim}\}$). Crucially, we do not place any constraints on c, and it can be any grammatically correct English sentence. This allows us to probe a myriad of fine-grained aspects related to the sentences, and we describe how we collect the dataset below.

2.2.1 Sentence Data Collection

We source sentence pairs ($\{s_1, s_2\}$) for our dataset from image-captioning datasets through a two-step process: (1) generate candidate text pairs through dense retrieval from the corresponding *image representations* and (2) filter out candidates that are irrelevant or ineffectual for our purposes. This step is completely automatic.

Image Retrieval Image-captioning datasets provide a compelling data source because image pair similarity and caption (text) pair similarity encode different semantics (Parekh et al., 2021). Image-representations thus serve as an informative latent variable which can represent their captions in ways that are not captured by text retrievers. Since current sentence representation models overlook aspects of conditional similarity, we utilize both the image and text to retrieve sentence pairs which form the foundation of our dataset.

We aim to derive sentence pairs (s_j, s_k) from an image-caption dataset $\mathcal{D}$ to aid in creating conditioning statements. To do this, we first generate a store of image pairs, or $\mathcal{P}_I$. Each pair, denoted by I_i, I_j, is such that I_j is amongst the top-k most similar images to I_i, determined by the cosine distance metric of their respective image representations obtained via an image encoder $E_I(\cdot)$. After establishing $\mathcal{P}_I$, we convert it into a sentence pair store ($\mathcal{P}_S$) by replacing each image in a pair with its corresponding caption. When each image $I_i \in \mathcal{D}$ is associated with a set of sentences $\{s\}_i$ we form the sentence pairs by taking the Cartesian product $\{s\}_i \times \{s\}_j$ for each image pair $I_i, I_j \in \mathcal{P}_I$.

Candidate Filtering After acquiring sentence pairs through image retrieval, we perform an additional step of quality assurance to find the best sentence pairs conducive for our task. Specifically, we rigorously select pairs of sentences for which the unconditional similarity is ambiguous, since this incentivizes the model to rely on the condition to reason about the conditional similarity.

To this end, we avoid high similarity sentence pairs by filtering out pairs with a high bag-of-words intersection over union (IOU) and avoid low similarity sentence by choosing sentences with a moderate or high cosine similarity of their SimCSE embeddings (Gao et al., 2021). The additional filters description for the candidate filtering process is in Appendix A.1.

2.2.2 Annotation Methodology

For each sentence pair in the store ($\mathcal{P}_S$), we wish to collect conditions and semantic similarity annotations for each sentence pair and condition triplet. We denote the two sentences by s_1 and s_2 and the condition with c. The triplet is denoted by $\{s_1, s_2, c\}$. Crucially, c is a free-form natural language sentence, which allows us to probe the multi-faceted nature of sentence similarity with a high level of control. We use Mechanical Turk for our human annotations and divide it into three stages.

Stage 1: Choosing a high-quality worker pool In the first stage, we design a qualification test to select workers who excel at our task. Specifically, we test two skills: (1) The quality of conditions they write for a given sentence pair and (2) semantic similarity judgements for a triplet ($\{s_1, s_2, c\}$). We choose a pool of 271 workers who perform well on both the tasks and request their annotation for subsequent stages. The exact qualification test used is provided in Appendices C.1 and C.2.

Stage 2: Condition annotation After sourcing sentence pairs ($\{s_1, s_2\}$) using the strategy discussed in the previous section, we provide them to the workers and instruct them to annotate each pair with one condition such that the similarity in

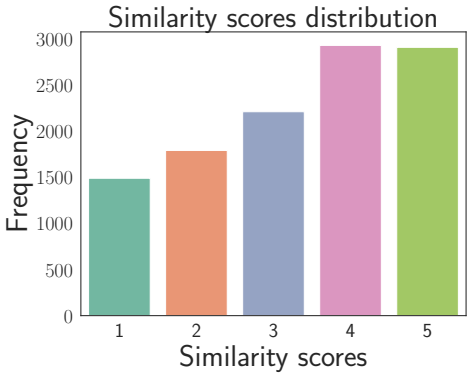

Figure 3: The distribution of similarity judgements on a Likert scale between $[1-5]$ as measured on the train partition.

its context is high (C-high) and one such that the similarity in its context is low (C-Low). Example:

s1 : A large green ball was bouncing on the street

s2 : I bought a small green avocado

C-High : The color of the object

C-Low : The size of the object

We do not place any constraints on the conditions other than that they should be grammatically correct sentences and that they should not be completely irrelevant to both sentences.

Stage 3: Condition verification and similarity assessment The output of annotations from the previous stage are triplets ($\{s_1, s_2, c\}$) with a binary similarity assessment (high or low). We now convert the similarity to a Likert scale (Likert, 1932) as is common with semantic textual similarity tasks (Agirre et al., 2012). In addition to assigning a similarity, we also use this stage to verify if the conditions from the previous stage are pertinent to the sentence pairs, which allows us to control the quality of the dataset. At the end of this stage, we have $\{s_1, s_2, c, \text{similarity}\}$ quadruplets which have passed a layer of human verification. The annotation guidelines are the same for all the stages as in Stage 1. We provide analysis on the dataset in Section §3.

3 Dataset Analysis

Dataset sources We use two image-caption datasets to generate sentence pairs using the method described in Section 2: the train split from the 2014 release of the MS COCO dataset (Lin et al., 2014) containing $\sim$ 83,000 images, and Flickr30K (Young et al., 2014) containing $\sim$ 30,000 images. Each dataset is processed separately and we do not intermix them during the retrieval stage. We use CLIP-ViT (Radford et al., 2021) to generate image embeddings and include the specific filtering criteria in Table 5 of Appendix A.1.

Dataset statistics To ensure high-quality faithful and diverse annotations, we collect a total of 20,000 instances and perform quality assurance (Section 5.3) which gives us a total of 18,956 instances as part of the C-STS-2023 dataset. Following standard practice, we create train, validation, and test partitions by splitting in a $60:15:25$ ratio. We present the distribution of similarity scores, which are discrete numbers between $[1, 5]$, in Figure 3. We also measure the inter-annotator agreement on a random sample of 100 examples with three independent annotations and find Fleiss' kappa score (Fleiss, 1971) to be 0.61 which implies substantial inter-annotator agreement.

Qualitative analysis C-STS allows us to evaluate the generally fuzzy notion of sentence similarity with fidelity. We illustrate this in Table 1, where precise and discriminative conditions enable a targeted, fine-grained, and grounded definition of sentence similarity. The following is a representative instance where the conditions tease out nuanced and hidden similarities and differences between the two lexically similar sentences on surfing: Consider s_1: "A windsurfer skims the water..." and s_2: "The surfer is riding a wave.."). While the sentences are significantly dissimilar based on the condition "the way the object is propelled" as they talk about windsurfing and surfing respectively (the former uses a sail whereas the latter depends on the wave), they are very similar in context of the condition "the base of the object" as both windsurfing and surfing use a similar board.

Our diverse set of conditions provides broad support over the distribution of conditions and enables a holistic and multi-faceted evaluation of sentence similarity. For example, the conditions for the sentences on Tennis in Table 1 test similarity both on the sport being played (which requires understanding lexical and knowledge artifacts) as well as the number of people (which requires reasoning and commonsense capabilities).

4 Baselines

We evaluate our dataset on several baselines which can be categorized into (1) Finetuning baselines

Sentence 1	Sentence 2	Condition and Similarity
An older man holding a glass of wine while standing between two beautiful ladies.	A group of people gather around a table with bottles and glasses of wine.	*The people's demeanor*: 5 *The number of bottles*: 1
Various items are spread out on the floor, like a bag has been emptied.	A woman with a bag and its contents placed out before her on a bed.	*The arrangement of objects*: 4 *The surface the objects are on*: 1
A windsurfer skims the water with his outstretched hand.	The surfer is riding a wave with a mountain in the background.	*The base of the object*: 5 *The way the object is propelled*: 1
Female tennis player jumping off the ground and swinging racket in front of an audience	A young lady dressed in white playing tennis while the ball girl retrieves a tennis ball behind her.	*The sport being played*: 5 *The number of people*: 1

Table 1: Four examples from the C-STS validation set. Under different conditions, the same sentence pair can be separated into high similarity and low similarity. Scale from 1 (dissimilar) to 5 (similar).

which are trained using the full training partition and (2) Prompting baselines which are evaluated either in an in-context learning or zero-shot setting.

4.1 Finetuning baselines

We evaluate three sentence encoder models RoBERTa (Liu et al., 2019), a strong general-purpose Encoder-only model, supervised SimCSE (Gao et al., 2021) and unsupervised DiffCSE (Chuang et al., 2022). SimCSE and DiffCSE represent state-of-the-art sentence encoder models which are particularly strong on STS tasks. For both SimCSE and DiffCSE we use the RoBERTa pre-trained varieties.

Encoding configurations Encoder-only Transformer models, such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019), initially performed regression finetuning for STS tasks by simply concatenating the sentences and encoding them together before generating a prediction; let us call this type of architecture a cross-encoder. Recent approaches instead opt to encode sentences separately and compare their similarity using a distance metric, such as the cosine distance Reimers and Gurevych (2019); which we will call a bi-encoder. While DiffCSE and SimCSE are both designed for the bi-encoder setting in mind, we observe that they work well in the cross-encoder setting as well.

For our baselines, we evaluate each model in both settings. For the cross-encoder configuration, we encode the triplet containing the sentences and the condition ($\{s_1, s_2, c\}$), and the output is a scalar similarity score – $f_\theta(s_1; s_2; c)$. For the bi-encoder configuration (Reimers and Gurevych, 2019), the sentences of a pair are encoded independently along with the condition using a siamese network and their cosine similarity is computed – $\text{sim}_{\cos}(f_\theta(s_1; c), f_\theta(s_2; c))$.

In addition to the bi and cross encoder models, we propose tri-encoder models which encoder both the sentences and the condition. For this, we first encode all sentences of the triplet separately, with encoder $f_\theta(\cdot)$ as $\mathbf{s}_i = f_\theta(s_i)$, where $\mathbf{s}_i \in \mathbb{R}^h$. We then perform an additional transformation $h : \mathbb{R}^{2d} \rightarrow \mathbb{R}^d$ that operates on the condition and one each of the sentences. We finally compute the conditional similarity using the cosine distance as $\text{sim}_{\cos}\left(h\left(\mathbf{c}; \mathbf{s}_1\right), h\left(\mathbf{c}; \mathbf{s}_2\right)\right)$. We experiment with 2 functions for h, an MLP and the Hadamard product.

Objectives In addition to the standard MSE loss for regression, we use a quadruplet contrastive margin loss which we denote Quad. The Quad loss is defined as follows:

$$\begin{aligned}\text{Quad}(p_1, p_2, n_1, n_2) = &\max(M + d_{\cos}(p_1, p_2) \\ &- d_{\cos}(n_1, n_2), 0)\end{aligned}$$

Where $d_{\cos}$ is the cosine distance and M is a margin hyperparameter. Since each sentence pair in C-STS comes with two conditions (one with higher similarity and one with lower similarity) we represent the conditional encoding of each sentence in

the higher-similarity pair as p_1 and p_2 and represent the conditional encoding of each sentence in the lower similarity pair as n_1 and n_2.

We train all of our tasks for regression using, alternatively, mean squared error (MSE), Quad, and a linear combination of the quadruplet loss and MSE (Quad + MSE). Since we require a separate conditional encoding fore each sentence, the Quad and (Quad + MSE) objectives apply only the the bi-encoder and tri-encoder configurations.

Hyperparameters We evaluate the baselines on the test split for C-STS. We perform a hyperparameter sweep to select the best performing configuration and test using models trained with 3 random seeds, with further details in Appendix A.2. As a comparison for our training setting, we perform a similar hyperparameter sweep for the STS-B (Cer et al., 2017) dataset, with the validation split results and best hyperparameters shown in Table 9, showing that our finetuned baselines achieve very strong performance on traditional STS tasks.

4.2 In-context learning baselines

For in-context learning, we evaluate three types of models. An autoregressive language model (GPT-J (Wang and Komatsuzaki, 2021)), instruction finetuned Seq2Seq models (Flan-T5 models Chung et al. (2022)), and proprietary dialogue systems (ChatGPT-3.5 (OpenAI, 2022), ChatGPT-4 (OpenAI, 2023)). For ChatGPT-3.5 and ChatGPT-4 we use the OpenAI API with the static model versions `gpt-3.5-turbo-0301` and `gpt-4-0314`.

When evaluating few-shot models, each model input is composed of up to three parts: the instructions, K examples, and the query. Models are evaluated with $0, 2$, or 4 examples and using three different instruction prompts: no instructions, short instructions, which only provide a high-level description of the task and long instructions, shown in Figure 4, which resembles the annotation guidelines and is similar to the instructions used for the STS-B classification task in Wang et al. (2022).

For few-shot evaluation, we additionally always group a sentence pairs' two conditional similarity examples together, so models will always see contrasting pairs in the examples, but won't see a paired example for the query. We provide examples of the formats used for the input and output for more settings in Appendix B. As we did for the finetuned models, we also evaluate these models on the STS-B validation split, shown in Table 12, with instruction finetuned models and dialogue systems achieving relatively strong performance.

Instructions

Definition: Evaluate the similarity between the two sentences, with respect to the condition. Assign the pair a score between 1 and 5 as follows:
1 : The two sentences are completely dissimilar with respect to the condition.
2 : The two sentences are dissimilar, but are on a similar topic with respect to the condition.
3 : The two sentences are roughly equivalent, but some important information differs or is missing with respect to the condition.
4 : The two sentences are mostly equivalent, but some unimportant details differ with respect to the condition.
5 : The two sentences are completely equivalent with respect to the condition.

Query

Input: Sentence 1: Elderly man sitting on a blue couch reading a paper.
Sentence 2: Older man riding public transportation while reading a newspaper.
Condition: The location of the man.
Output:

Figure 4: The full text input for the in-context learning setup with large language models, here in a zero-shot setting using 'long' instructions. Emphasis and section titles added for clarity.

5 Results

5.1 Evaluating sentence encoders on C-STS

We evaluate prominent sentence encoders on C-STS-2023 with three different encoding methods: cross-encoder, bi-encoder, and tri-encoder, across two model configurations. Our results indicate that all models perform relatively poorly on C-STS and demonstrate the need for significant progress on C-STS (Table 2).

As expected SimCSE and DiffCSE, which are fine-tuned with sentence similarity-based objectives naturally perform better than RoBERTa which is a vanilla pre-trained model across most model configurations and encoding architectures. For

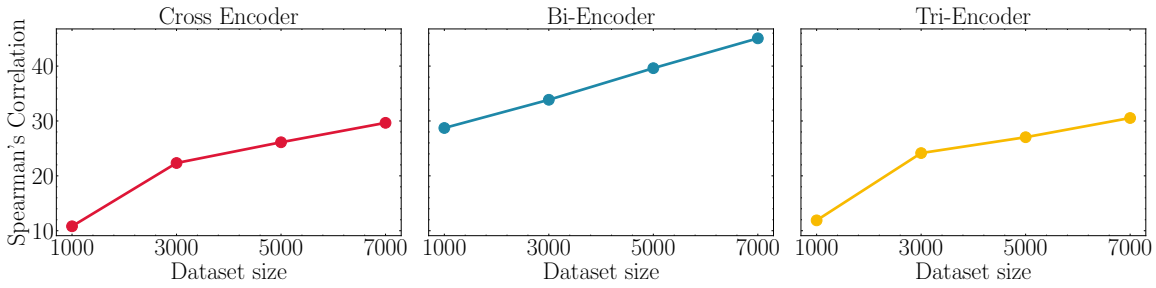

Figure 5: Model ($\text{SimCSE}_{\text{LARGE}}$) performance scaling as the dataset size increases. Across encoder types, the Spearman's correlation score increases as the dataset scales.

Encoding	Model	Spear.↑	Pears.↑
Cross Encoding	$\text{RoBERTa}_{\text{BASE}}$	$39.2_{\pm 1.3}$	$39.3_{\pm 1.3}$
	$\text{RoBERTa}_{\text{LARGE}}$	$40.7_{\pm 0.5}$	$40.8_{\pm 0.4}$
	$\text{DiffCSE}_{\text{BASE}}$	$38.8_{\pm 2.9}$	$39.0_{\pm 2.7}$
	$\text{SimCSE}_{\text{BASE}}$	$38.6_{\pm 1.3}$	$38.9_{\pm 1.2}$
	$\text{SimCSE}_{\text{LARGE}}$	$\mathbf{43.2}_{\pm 1.2}$	$\mathbf{43.2}_{\pm 1.3}$
Bi-Encoding	$\text{RoBERTa}_{\text{BASE}}$	$28.1_{\pm 8.5}$	$22.3_{\pm 14.1}$
	$\text{RoBERTa}_{\text{LARGE}}$	$27.4_{\pm 6.2}$	$21.3_{\pm 8.4}$
	$\text{DiffCSE}_{\text{BASE}}$	$43.4_{\pm 0.2}$	$43.5_{\pm 0.2}$
	$\text{SimCSE}_{\text{BASE}}$	$44.8_{\pm 0.3}$	$44.9_{\pm 0.3}$
	$\text{SimCSE}_{\text{LARGE}}$	$\mathbf{47.5}_{\pm 0.1}$	$\mathbf{47.6}_{\pm 0.1}$
Tri-Encoding	$\text{RoBERTa}_{\text{BASE}}$	$28.0_{\pm 0.4}$	$25.2_{\pm 1.0}$
	$\text{RoBERTa}_{\text{LARGE}}$	$20.3_{\pm 2.2}$	$18.9_{\pm 2.3}$
	$\text{DiffCSE}_{\text{BASE}}$	$28.9_{\pm 0.8}$	$27.8_{\pm 1.2}$
	$\text{SimCSE}_{\text{BASE}}$	$31.5_{\pm 0.5}$	$31.0_{\pm 0.5}$
	$\text{SimCSE}_{\text{LARGE}}$	$\mathbf{35.3}_{\pm 1.0}$	$\mathbf{35.6}_{\pm 0.9}$

Table 2: Fine-tuning results in Spearman r and Pearson r score for the three models (RoBERTa, DiffCSE, and SimCSE) on the test set.

Instruct.	Model	0-shot↑	2-shot↑	4-shot↑
	†$\text{SimCSE}_{\text{LARGE}}$ (All data)		**47.5**	
None	$\text{Flan-T5}_{\text{BASE}}$	-0.8	8.7	10.7
	$\text{Flan-T5}_{\text{LARGE}}$	0.0	12.3	12.8
	GPT-J	5.0	0.6	-0.6
	GPT-3.5	12.6	1.6	3.1
	GPT-4	**21.0**	**18.7**	**27.0**
Short	$\text{Flan-T5}_{\text{BASE}}$	6.8	9.1	6.7
	$\text{Flan-T5}_{\text{LARGE}}$	11.1	10.9	11.0
	GPT-J	-1.7	-2.4	1.1
	GPT-3.5	15.0	15.6	15.5
	GPT-4	**39.3**	**42.6**	**43.6**
Long	$\text{Flan-T5}_{\text{BASE}}$	11.3	5.1	8.8
	$\text{Flan-T5}_{\text{LARGE}}$	7.4	4.4	5.3
	GPT-J	-0.2	1.1	2.0
	GPT-3.5	9.9	16.6	15.3
	GPT-4	**32.5**	**41.8**	**43.1**

Table 3: Few-shot results on the test set using Spearman's correlation score. Generally, models perform much worse than their finetuned counterparts, with GPT-4 being the only evaluate model that achieves comparable performance to some finetuned baselines. †: Full fine-tuning

example, in the bi-encoder setting SimCSE-base outperforms RoBERTa-base by over 20 Spearman points. The performance on C-STS also varies significantly with the encoding architecture, with the bi-encoder architecture emerging as the most promising architecture. Taken together, these trends highlight the importance of critical modeling choices and training objectives towards proficiency on C-STS.

5.2 Few-shot performance on C-STS

We evaluate the few-shot C-STS capabilities of salient large language models in Table 3 across varying sets of instructions and the number of in-context examples. We find that similar to the sentence encoders, these language models have relatively poor performance across different configurations and pose the need for progress on C-STS.

Importantly, the state-of-the-art language model, GPT-4, perform relatively better than competing models (GPT-J, Flan-T5, GPT-3.5). For example, when using "short" instructions, GPT-4 outperforms GPT-3.5 and Flan models by over 20 points. This suggests that progress on C-STS correlates with the progress on existing benchmarks since GPT-4 is the strongest model. However, $\text{SimCSE}_{\text{LARGE}}$ beats GPT-4 even for the 4-shot scenario, showing the strength of sentence embedding models when given enough training data.

Additionally, we find that GPT-J, which is not instruction fine-tuned, performs much worse than the competing instruction fine-tuned models (Flan-T5, GPT-3.5, GPT-4). We also find that the presence of instructions primes the model to perform much better on C-STS and that the performance is robust to different instructions ("Short" and "Long") in the presence of two or more in-context examples. These observations highlight instruction fine-

Model	Sentence 1	Sentence 2	Condition	Output
Flan-T5-Base	A man taking a bite out of a sandwich at a table with someone else.	A man sitting with a pizza in his hand in front of pizza on the table.	Type of dish.	Pred: 4.5 Label: 1.0
GPT-3.5	A wooden bench surrounded by shrubbery and flowers on the side of a house.	A scene displays a vast array of flower pots in front of a decorated building.	The type of plants.	Pred: 0..0 Label: 3.0
GPT-4	Football player jumping to catch the ball with an empty stand behind him.	A football player preparing a football for a field goal kick, while his teammates can coach watch him.	The game being played.	Pred: 3.0 Label: 5.0
GPT-4	A giraffe reaches up his head on a ledge high up on a rock.	A giraffe in a zoo bending over the fence towards where impalas are grazing.	The height of the giraffe's head.	Pred: 1.0 Label: 1.0

Table 4: We show examples of model predictions evaluated on C-STS in the in-context setting ($K = 2$ with no instructions). We choose examples with different levels of accuracy, showcasing the different failure cases of model behavior.

tuning as a critical process to endow models with language proficiency.

5.3 Analysis

Scaling laws for C-STS We performs experiments to evaluate the effect of the quantity of C-STS data on sentence-embedding methods, here $\text{SimCSE}_{\text{LARGE}}$ (Figure 5). We notice that for all three encoder strategies, the performance monotonically increases as we increase the size of the training dataset. For example, for the SimCSE bi-encoder, the Spearman correlation increases from 30 when using a train set of 1000 examples to 45 for 7000 examples.

Furthermore, there is almost a linear increase in the performance of the models, especially the bi-encoder as we increase the amount of data. This quantitatively enforces the quality of the dataset, but also retroactively also makes that point that rather than relying on increasing the size of the dataset, we require better modeling strategies to improve performance on this novel task.

Qualitative Analysis We present predictions from different models in Table 4 to illustrate systematic pitfalls. For instance, Flan-T5 makes incorrect predictions even for straightforward instances and falsely predicts that both sentences talk about the same dish, even though the sentences clearly talk about sandwiches and pizza respectively. Additionally, GPT-3.5 incorrectly predicts that the two sentences are completely dissimilar when talking about the types of plants, even though both sentences are talking about flowering plants. Note that our annotation, unlike GPT-3.5, captures the nuance that the first sentence talks about *both shrubbery and flowers*, while the second sentence talks only about flowers, and therefore assigns a conservative similarity score of 3. The most proficient model on C-STS, GPT-4, is much better at capturing these nuances and accurately predicts, for instance, that the height of the giraffe's head (refer to the fourth example), is high in one sentence and low in another. GPT-4 is far from perfect though, and we outline a negative prediction (refer to the third example), where the model does not predict that the two sentences talk about the same game, even though they are very clearly about "Football".

More broadly, C-STS provides a lens into a model's ability to precisely and comprehensively understand and reason over the sentence pairs, and is well-suited to reveal systematic modeling issues.

6 Related Work

Historical perspectives of semantic similarities. Measuring semantic similarities is a long-standing problem spanning from cognitive science (Miller and Charles, 1991) to psychology (Tversky, 1977) where early attempts are made to quantify the subjective similarity judgements with information theoretical concepts. More recently, interest in semantic similarity has grasped popularity in the context of machine learning, with works in computer vision recognizing that the notion of similarity between images varies with the conditions (Veit et al., 2017) and can therefore be ambiguous (Ye et al., 2022b).

Textual similarity Tasks. Capturing textual similarity is also considered a fundamental problem in natural language processing. Works such as Agirre et al. (2012, 2016) define the textual semantic similarity tasks (STS), which is widely used in common benchmarks such as GLUE (Wang et al., 2018). Extensions to the STS setting have been proposed such as making the task broader with multilinguality (Cer et al., 2017) or incorporating relatedness (Abdalla et al., 2021). However, the loose definition of similarity has not been acknowledged as an issue explicitly. In contrast, our work tackles the ambiguity problem by collecting conditions and hence reduce subjectivity. To alleviate ambiguity, explanations play an important role in identifying the differences between the two sentences either in their syntactical structure (Lopez-Gazpio et al., 2017) or in natural language (Camburu et al., 2018), but the post-hoc nature of explanations prevents it from being used prior to the similarity judgement, rendering it a supplemental component as opposed to a paradigm change in the task setup. Beyond STS, works that leverage conditioning to enhance sentence representations obtain improved performance for retrieval (Asai et al., 2022) and embedding qualities (He et al., 2015; Su et al., 2022; Jiang et al., 2022), which corroborates with the observation that conditioning as a form of disambiguation benefits similarity measures.

7 Discussion

In this work, we propose conditional semantic textual similarity (C-STS), a novel semantic similarity assessment task that resolves the inherent ambiguity issues in STS. Given the importance of STS and what it enabled in terms of sentence embedding methods and evaluation, we believe that C-STS is a timely and necessary contribution to the natural language processing community. Rather than testing unconditional semantic similarity, the diversity of conditions in our dataset allows fine-grained evaluation. The same sentence pairs can be tested on a variety of different aspects represented by conditions, with similarities often varying significantly. C-STS poses a challenging hurdle to both fine-tuning and SOTA in-context learning models which struggle to capture the high-dimensional manifold of similarity. We believe that a combination of improved modeling and sophisticated fine-tuning strategies are required to push the boundaries on our task and we hope the community finds it useful.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif M. Mohammad. 2021. What Makes Sentences Semantically Related: A Textual Relatedness Dataset and Empirical Study. ArXiv:2110.04845 [cs].

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 252–263, Denver, Colorado. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. 2022. Task-aware Retrieval with Instructions. arXiv. ArXiv:2211.09260 [cs].

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In Advances in Neural Information Processing Systems, volume 31. Curran Associates, Inc.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo 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.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. DiffCSE: Difference-based contrastive learning for sentence embeddings. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4207–4218, Seattle, United States. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. arXiv preprint arXiv:2210.11416.

Simon De Deyne, Daniel J Navarro, Amy Perfors, and Gert Storms. 2016a. Structure at every scale: A semantic network account of the similarities between unrelated concepts. Journal of Experimental Psychology: General, 145(9):1228.

Simon De Deyne, Amy Perfors, and Daniel J Navarro. 2016b. Predicting human similarity judgments with distributional models: The value of word associations. In Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers, pages 1861–1870.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jeffrey L. Elman. 1990. Finding structure in time. Cognitive Science.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. Psychological Bulletin, 76:378–382.

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.

Hua He, Kevin Gimpel, and Jimmy Lin. 2015. Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1576–1586, Lisbon, Portugal. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural Computation.

Ting Jiang, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Liangjie Zhang, and Qi Zhang. 2022. Prompt-BERT: Improving BERT Sentence Embeddings with Prompts. arXiv. ArXiv:2201.04337 [cs].

Rensis Likert. 1932. A technique for the measurement of attitudes. Archives of psychology.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In European Conference on Computer Vision.

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.

I. Lopez-Gazpio, M. Maritxalar, A. Gonzalez-Agirre, G. Rigau, L. Uria, and E. Agirre. 2017. Interpretable Semantic Textual Similarity: Finding and explaining differences between sentences. volume 119, pages 186–199. ArXiv:1612.04868 [cs].

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. Language and Cognitive Processes.

OpenAI. 2022. Introducing ChatGPT. https://openai.com/blog/chatgpt.

OpenAI. 2023. Gpt-4. Accessed: 2023-05-23.

Zarana Parekh, Jason Baldridge, Daniel Cer, Austin Waters, and Yinfei Yang. 2021. Crisscrossed Captions: Extended Intramodal and Intermodal Semantic Similarity Judgments for MS-COCO. arXiv:2004.15020 [cs]. ArXiv: 2004.15020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In International Conference on Machine Learning.

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.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. One Embedder, Any Task: Instruction-Finetuned Text Embeddings. arXiv. ArXiv:2212.09741 [cs].

Amos Tversky. 1977. Features of similarity. Psychological Review, 84:327–352.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008.

Andreas Veit, Serge Belongie, and Theofanis Karaletsos. 2017. Conditional Similarity Networks. arXiv. ArXiv:1603.07810 [cs].

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. EMNLP 2018, page 353.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. CoRR, abs/1910.03771.

Han-Jia Ye, Yi Shi, and De-Chuan Zhan. 2022a. Identifying ambiguous similarity conditions via semantic matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16610–16619.

Han-Jia Ye, Yi Shi, and De-Chuan Zhan. 2022b. Identifying Ambiguous Similarity Conditions via Semantic Matching. ArXiv:2204.04053 [cs].

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics, 2:67–78.

A Appendix

A.1 Sentence Pair Generation Details

Here we include some further details about sourcing sentence pairs from image-caption datasets.

As discussed in Section 2, we use a variety of metrics to quantitatively characterize the sentence pairs, and then to filter with the goal of removing pairs with excessively high or low unconditional similarity. The general criteria we consider are defined as follows:

- IOU - This is computed by taking the intersection over union of the bag of words for each sentence, after stopword removal. It represents the lexical similarity and overlap of a sentence pair.
- d_{text} - The cosine distance of the pair's SimCSE embeddings. We chose SimCSE due to its ubiquity and effectiveness.
- ratio - This is the ratio of the shorter sentence's word count to the longer sentence's word count in a given pair.
- length - This is the character length of the shortest sentence in a pair.

Using these criteria, we filter the sentence pairs based upon thresholds (exact values shown in Table 5) where sentences are *rejected* if they violate any of these criteria. These thresholds were selected based primarily manual inspection of samples on their margins. Criteria such as ratio and length are used primarily to facilitate comparison. Sentences with very different lengths are more difficult to compare, as are sentences that are very short or contain few details.

	COCO	Flickr30K
k	64	128
IOU	≤ 0.12	≤ 0.2
d_{text}	≥ 0.4	≥ 0.4
ratio	≥ 0.7	≥ 0.7
length	≥ 50	≥ 48

Table 5: The list of filters criteria and values used for each dataset. Sentence pairs that violate any criterion are discarded.

A.2 Evaluation Details

Implementation Details All models, with the exception of the ChatGPT systems, are trained and or evaluated in PyTorch using the Huggingface Transformers library (Wolf et al., 2019) and pre-trained weights repository. We use the STS-B dataset as distributed on `https://huggingface.co/docs/datasets` as part of the GLUE (Wang et al., 2018) evaluation benchmark.

Model	C-STS		STS-B	
	Spear.	Pears.	Spear.	Pears.
$\text{DiffCSE}_{\text{BASE}}$	0.88	0.50	84.44	85.06
$\text{RoBERTa}_{\text{BASE}}$	-0.43	-0.09	35.15	48.18
$\text{RoBERTa}_{\text{LARGE}}$	-1.77	-2.39	7.29	15.08
$\text{SimCSE}_{\text{BASE}}$	1.66	0.84	85.14	86.80
$\text{SimCSE}_{\text{LARGE}}$	1.87	1.35	88.09	88.90

Table 6: Zero-shot Bi-Encoder models evaluation results on C-STS and STS-B validation data. These results verify that strong performance on STS tasks do not translate to C-STS, suggesting substantial room for improvement for fine-grained sentence embedding models.

Zero-shot Bi-encoder Performance As an initial comparison, we evaluate bi-encoder models without finetuning on both C-STS and STS-B. Models, like SimCSE and DiffCSE are trained for sentence encoding using bi-encoder contrastive losses, and expectedly perform very well on zero-shot evalution for STS-B. As shown in Table 6, we see that strong performance on STS-B does not translate to good performance on C-STS, suggesting that models fail entirely to incorporate the provided conditioning statement. These results suggest that current approaches to training sentence encoders may be too specialized to existing tasks for evaluation, such as STS-B.

Finetuning Baselines For evaluation of the finetuning baselines on C-STS, we perform a hyperparameter sweep to select the best training settings for each model and encoding method before evaluating on the test split of C-STS. We show the hyperparameter values used in the sweep in Table 7, and the final hyperparameter values chosen in Table 8. We evaluate 3 random seeds using the best validation configuration to evaluate on the test data, with final results reported in Table 2.

We additionally perform an extensive evaluation of our models on STS-B. We perform a comparable validation sweep as shown in Table 7, reporting the

Batch Size	{32}
Encoding Type	{Cross, Bi-, Tri-}
Epochs	{3}
Learning Rate	{1e-5, 3e-5}
Objective	{MSE, Quad, Quad + MSE}
Pooler Type	{[CLS] w/ MLP, * w/o MLP}
Seed	{42}
Weight Decay	{0, 0.1}

Table 7: Hyperparameter sweep done for C-STS validation for finetuning models. The same sweep, with the exception of the Encoding Type and Objective hyper parameters are done for STS-B.

best performing hyperparameters and their performance in Table 9.

Lastly, we perform a data ablation training a RoBERTa$_{\text{BASE}}$ model alternatively on only the condition and only the sentence pair. The model trained to predict similarity based on the condition statement alone recovers non-trivial performance, but falls well behind the full-input baseline.

Prompting Baselines We report more details of results of the prompting baselines for the validation sets of C-STS and STS-B.

For comparison to validation performance of other models, we include the validation performance for C-STS in Table 11, which largely mirrors performance on the test set. We notice, expectedly, that models frequently output non-numerical responses in settings where there are no instructions to do so, or no in-context examples to follow.

On STS-B validation performance, models generally perform much better than on C-STS, with some models performing comparably to finetuned models. Since STS-B is included as a task in Natural Instructions v2 (Wang et al., 2022), it is likely to be recognizable to Flan-T5 models, which counts Natural Instructions v2 in its training data. Likewise, STS-B is comprised of long-existing and popular datasets, which plausibly exist in the the corpora used to train ChatGPT models.

Processing Prompting Baseline Generations For parsing prompting model generations, we allow for a maximum of 20 generation tokens. The output is stripped of non-numeric characters and errant punctuation before being cast to a float. For example, the response "The Answer is 2.0." is processed as 2.0 and counts as a valid prediction. If the cast fails, we mark the answer invalid and replace the predictions by a number $y \sim U\,[1, 5]$.

B Prompt Examples

All prompts for the prompting baselines may consist of instructions, examples, and a query, though we include evaluations for no instructions and no examples in our results. Figure 6 shows an prompt example for the *short* instructions and $K = 2$.

Instructions

On a scale between 1 and 5, how similar are the following two sentences with respect to the condition provided? Respond only with a score between 1 and 5.

Examples

Input: Sentence 1: A bunch of blue buses parked in a parking lot in front of a housing community.
Sentence 2: Two buses, one blue and one red and white, are going to different destinations.
Condition: The type of transportation.
Output: 4.0
Input: Sentence 1: A bunch of blue buses parked in a parking lot in front of a housing community.
Sentence 2: Two buses, one blue and one red and white, are going to different destinations. Condition: The number of buses.
Output: 1.0

Query

Input: Sentence 1: The skater is descending the wooden wall beside the slope .
Sentence 2: A boy skateboards off a ramp covered in graffiti .
Condition: The location.
Output:

Figure 6: We show the full input for 2-shot setting with *short* instructions.

C Crowdsourcing Guidelines

C.1 Condition Annotation

We provide the complete condition annotation guidelines used for Mechanical Turk data collection in Figure 7.

C.2 Condition Verification

We provide the complete verification guidelines used for Mechanical Turk data collection in Figure 8.

Model	Modeling Type	Learning Rate	Weight Decay	Transform	Objective	Tri-Encoder Op.	Spearman	Pearson
RoBERTa$_{\text{BASE}}$	Cross Encoder	3.0e-05	0.10	True	MSE	-	41.02	40.95
	Bi Encoder	3.0e-05	0.10	True	MSE	-	37.93	37.17
	Tri Encoder	3.0e-05	0.00	False	Quad + MSE	Concat	28.70	27.50
RoBERTa$_{\text{LARGE}}$	Cross Encoder	1.0e-05	0.10	True	MSE	-	40.21	40.49
	Bi Encoder	1.0e-05	0.10	True	Quad + MSE	-	35.81	33.25
	Tri Encoder	1.0e-05	0.00	True	MSE	Hadamard	21.82	21.46
DiffCSE$_{\text{BASE}}$	Cross Encoder	3.0e-05	0.10	False	MSE	-	39.73	39.84
	Bi Encoder	3.0e-05	0.00	False	MSE	-	42.18	41.85
	Tri Encoder	3.0e-05	0.10	False	Quad + MSE	Hadamard	30.60	29.59
SimCSE$_{\text{BASE}}$	Cross Encoder	3.0e-05	0.10	True	MSE	-	33.91	34.90
	Bi Encoder	3.0e-05	0.10	False	MSE	-	45.67	45.55
	Tri Encoder	3.0e-05	0.10	False	Quad + MSE	Hadamard	33.06	32.35
SimCSE$_{\text{LARGE}}$	Cross Encoder	1.0e-05	0.10	True	MSE	-	44.31	44.42
	Bi Encoder	1.0e-05	0.10	False	MSE	-	47.70	47.41
	Tri Encoder	1.0e-05	0.00	True	MSE	Hadamard	34.46	34.95

Table 8: Fine-tuning models' results over *validation* split. We show the best performing configuration selected over the validation split which was the final configuration used to report each models' test performance.

Model	Encoding	Learning Rate	Transform	Objective	Spearman	Pearson
RoBERTa$_{\text{BASE}}$	Cross Encoder	3.0e-05	True	MSE	90.54	90.55
	Bi Encoder	3.0e-05	False	MSE	87.23	86.73
DiffCSE$_{\text{BASE}}$	Cross Encoder	3.0e-05	False	MSE	89.75	89.82
	Bi Encoder	3.0e-05	False	MSE	88.08	87.66
RoBERTa$_{\text{LARGE}}$	Cross Encoder	3.0e-05	True	MSE	91.49	91.58
	Bi Encoder	3.0e-05	False	MSE	87.79	87.25
SimCSE$_{\text{BASE}}$	Cross Encoder	3.0e-05	True	MSE	89.50	89.65
	Bi Encoder	3.0e-05	False	MSE	89.69	89.30
SimCSE$_{\text{LARGE}}$	Cross Encoder	3.0e-05	True	MSE	91.73	91.78
	Bi Encoder	1.0e-05	False	MSE	90.70	90.56

Table 9: Validation performance of best sweep setting on STS-B.

Data Ablation	Spear.	Pears.
Condition Only	28.21	28.62
Sentence Only	9.98	9.51
Baseline	40.11	40.21

Table 10: When finetuned only with condition statement, RoBERTa$_{\text{BASE}}$ model can recover non-trivial performance, but falls well behind the baseline. Training on only the sentence pairs proves to be even less informative. We report the best validation performance over the same hyperparameter grid described in Section 4.1.

Instruction	Model	0-shot			2-shot			4-shot		
		Invalid	Pears.	Spear.	Invalid	Pears.	Spear.	Invalid	Pears.	Spear.
None	Flan-T5$_{\text{BASE}}$	98.24	2.30	0.69	10.09	10.54	9.66	1.13	7.51	7.11
	Flan-T5$_{\text{LARGE}}$	87.12	-1.90	-3.04	10.55	10.96	11.69	5.08	12.54	13.12
	Flan-T5$_{\text{XL}}$	99.54	-1.89	0.61	84.76	-1.89	5.40	69.09	-1.89	5.99
	GPT-J	28.76	0.88	2.53	16.41	-2.12	-0.68	0.85	-0.51	-0.84
	GPT-3.5	65.24	5.80	11.21	17.57	3.96	3.91	2.43	6.49	6.31
	GPT-4	59.17	9.01	16.69	4.98	16.10	15.56	0.60	26.74	26.59
Short	Flan-T5$_{\text{BASE}}$	0.00	6.01	5.62	0.00	3.31	4.23	0.00	5.45	5.39
	Flan-T5$_{\text{LARGE}}$	0.00	8.45	8.39	0.00	8.16	8.05	0.00	8.33	8.71
	Flan-T5$_{\text{XL}}$	0.00	2.30	2.08	0.00	9.34	9.52	0.00	8.58	8.73
	GPT-J	39.87	0.95	0.30	38.39	-3.06	-1.41	5.75	0.68	-1.41
	GPT-3.5	0.00	12.91	11.13	0.04	16.63	17.62	0.07	12.60	13.76
	GPT-4	0.00	38.77	39.47	0.00	39.76	41.25	0.00	41.52	42.05
Long	Flan-T5$_{\text{BASE}}$	0.00	5.66	4.90	0.00	6.18	5.23	0.00	6.70	6.07
	Flan-T5$_{\text{LARGE}}$	0.00	5.44	5.07	0.00	5.68	4.61	0.00	5.82	5.75
	Flan-T5$_{\text{XL}}$	0.00	5.35	5.43	0.00	6.07	6.16	0.00	4.53	5.15
	GPT-J	92.17	-1.98	-1.58	2.29	-1.79	0.55	8.05	-1.89	0.05
	GPT-3.5	0.00	10.24	8.42	0.00	16.82	15.46	0.00	16.60	15.70
	GPT-4	0.00	33.48	33.04	0.00	39.08	39.53	0.00	42.26	42.38

Table 11: Validation performance for prompting baselines on C-STS.

Instruction	Model	0-shot			2-shot			4-shot		
		Invalid	Pears.	Spear.	Invalid	Pears.	Spear.	Invalid	Pears.	Spear.
None	Flan-T5$_{\text{BASE}}$	93.60	0.06	0.39	21.27	0.06	28.60	12.07	-0.37	32.36
	Flan-T5$_{\text{LARGE}}$	78.27	-0.85	5.65	8.93	32.39	36.99	1.27	35.40	42.39
	Flan-T5$_{\text{XL}}$	91.27	1.97	7.31	86.40	2.82	7.07	81.93	2.82	-0.82
	GPT-J	15.40	-1.12	2.66	16.87	0.95	1.84	13.20	2.79	6.57
	GPT-3.5	96.93	-4.17	1.61	0.07	63.86	64.83	0.00	74.96	76.15
	GPT-4	63.20	-2.40	20.10	0.00	76.70	75.92	0.00	86.16	86.25
Short	Flan-T5$_{\text{BASE}}$	0.00	79.55	79.25	0.00	79.51	79.37	0.00	79.04	78.88
	Flan-T5$_{\text{LARGE}}$	0.00	75.87	75.45	0.00	76.07	76.89	0.00	76.37	77.32
	Flan-T5$_{\text{XL}}$	0.00	65.01	64.83	0.00	52.00	58.76	0.00	52.91	57.83
	GPT-J	15.47	0.17	7.98	54.07	5.40	2.62	29.20	1.24	7.53
	GPT-3.5	0.00	86.58	86.78	0.00	83.69	83.13	0.00	85.12	84.91
	GPT-4	0.00	88.20	88.95	0.00	88.38	88.44	0.00	89.02	88.96
Long	Flan-T5$_{\text{BASE}}$	0.00	81.51	81.15	0.00	81.38	81.07	0.00	81.51	81.12
	Flan-T5$_{\text{LARGE}}$	0.00	69.60	68.31	0.00	68.69	66.67	0.00	66.44	64.23
	Flan-T5$_{\text{XL}}$	21.33	2.82	20.25	0.87	28.28	41.78	0.40	32.04	48.85
	GPT-J	3.80	4.42	3.34	42.93	2.03	5.09	37.87	0.41	6.75
	GPT-3.5	0.00	86.28	86.59	0.00	86.16	85.81	0.00	87.08	86.90
	GPT-4	0.00	89.57	89.76	0.00	90.01	89.95	0.00	90.73	90.65

Table 12: Validation performance for prompting baselines on STS-B.

We will be hosting more HITs in the future and we invite you to attend those.
Please send any feedback you have to: placeholder@gmail.com

Task summary

Our goal is to understand the similarity of a sentence pair based on a condition.
Concretely, for a sentence pair (S1 and S2), provide one condition (C-High) such that S1 and S2 have high similarity, and one condition (C-Low) such that they have low similarity.

As an example:

S1: A large green ball was bouncing on the street.
S2: I bought a small green avocado.

C-High: The color (High Similarity because it is green in both sentences)
C-Low: The size (Low Similarity because it is large in the first and small in the second sentence)

Conditions are English phrases which are used to choose an aspect of the sentence.

Guidelines for conditions

You are allowed to use the internet to understand the sentences, but the conditions need to be written by you. The following guidelines need to be followed.

1. **Conditions should be grammatically correct** English phrases or sentences.
2. **Avoid conditions which reference missing information** that cannot be inferred from sentences. For example, avoid the following condition, because the color of the animal in S2 is unclear.
    a. S1: Brown bears attacked people in the night.
    b. S2: Some dogs were barking on the road.
    c. C-High: The color of the animal.

    But the following is a good condition because it can be inferred that the game is chess:
    d. S1: Black ultimately reached an endgame two pawns up.
    e. S2: Now the white king comes just in time to support his pawn.
    f. C-High: The game being played.
3. **Conditions should reference aspects or attributes of sentences and not the values.** For example, the following ("The color is green") is an incorrect condition because it directly mentions "green", which is the value of the attribute "color":
    a. S1: A large green watermelon.

Figure 7: Annotation Guidelines

    b. S2: A green avocado in the basket.
    c. C-High: The color is green.

   Instead, the same condition can correctly be written as: “The color of the fruit”.
4. **Avoid conditions which explicitly use words like “sentences”**. For example, instead of saying “the color in the sentence”, just say “The color”.
5. **Avoid vague conditions which do not help narrow down a specific aspect of the sentence**. For example, avoid conditions which simply say “The activity”, which does not help narrow down the aspect. Instead use more informative words like “the sport” or “the hobby” as much as possible.
6. **Whenever possible, try to write conditions which refer to abstract similarity**.
   Consider the following sentences:
    a. Two women are celebrating a goal.
    b. A couple is eating a tasty meal.

   A condition which is more abstract is preferred:
    c. *Abstract condition*: The sentiment of the people.

   Although a more literal condition is valid, it is less preferred:
    d. *Literal condition*: The number of people.

Examples

We provide good and bad examples of conditions for sentence pairs, along with the reasoning.

Good examples

All the following conditions are valid because they follow our guidelines.

Sentence 1	Sentence 2	Condition	Similarity	Explanation
The moon looked incredible!	The car was completely covered in snow.	The color of the object.	High	The color is white in both cases. This is a good condition because it references the color of the object without explicitly mentioning it.
A group of people wearing helmets and riding on bikes.	A group of bikers are gathered together and taking pictures.	The speed of the cyclists.	Low	The group of cyclists is moving in the first sentence whereas they are not in the second. Hence their speeds are dissimilar.
Three people are holding a ladder while another climbs it.	Three people are listening to music in a car.	The number of people.	Low	There are four people in the first sentence but only three in the second sentence.

Bad examples

All the following conditions are invalid because they ignore one or more of our guidelines.

Sentence 1	Sentence 2	Condition	Reason for invalidity of condition
Egyptians appeased gods with offerings and prayers.	People in this era put faith in specific gods to protect their lives.	The culture involved.	It violates guideline 2. The culture in the second sentence cannot be inferred and is missing information.
An adult elephant is playing in the river.	A boulder is rolling down the hill.	The size of the object is large.	It violates guideline 3. The condition should have been "The size of the object", without explicitly referring to it being "large".
A guitarist is playing on a bench.	A man in a green hat is playing the guitar on the road.	The instrument in the sentence.	It violates guideline 4. The condition would be good if "in the sentence" was removed so that it is just "The instrument".
A middle-aged man is helping construct a grass hut.	Three men work on a roof.	The activity.	This condition is too vague and does not reference a specific aspect. A better condition would be: "The type of construction".
A man on top of a partially completed roof laying down more shingles.	A man in a hard hat and safety gear stands in a construction site.	The number of people.	While this condition is valid, it violates guideline 6, which says that an abstract condition should be considered wherever possible. A better condition would have been, "*The occupation of the man*", which is "construction worker" in both cases.

We will be hosting more HITs in the future and we invite you to attempt those.
Please send any feedback you have to: placeholder@gmail.com

Task summary

Our goal is to understand the similarity of a sentence pair based on a condition.
Concretely, for a sentence pair (**S1** and **S2**), and a condition '**C**', provide a score which indicates the similarity of **S1** and **S2** with respect to **C**.

As an example:

S1: A large green ball was bouncing on the street.
S2: I bought a small green avocado.
C: The size of the object

Similarity: 1 (Low Similarity because it is large in the first and small in the second sentence)

Guidelines for annotating similarity

Part 1

Given two sentences and a condition, **first check if the condition applies to both the sentences**. If the condition does not apply even to one of the sentences, please check the box provided to indicate the same. For example:

S1: A small dog happily runs across the street.
S2: I bought a small green avocado.
C: The sentiment

In the above example, the condition does not make sense for S2 because there is no sentiment that can be inferred from it.

Part 2

If the condition makes sense, given two sentences and a condition, please ascribe a similarity score for the sentences when interpreted with respect to the condition.
The score has to be one of the following numbers: {1, 2, 3, 4, 5}.
Tips:

- **Please avoid overusing the middle range score (3) as much as possible.**
- **Feel free to use the extreme scores (1 and 5) if they make sense to you.**

The following is the meaning of the different scores:

1. **Score = 1**: *The two sentences are completely dissimilar with respect to the condition.*
   For example:

Figure 8: Verification Guidelines

S1: A man cooks in the kitchen.
S2: A woman is riding a bike on the road.
C: The gender (Man and woman are dissimilar with respect to gender)

2. **Score = 2**: *The two sentences are dissimilar, but are on a similar topic with respect to the condition.*
   For example:
   S1: A man plays the guitar.
   S2: A little girl listens to the violin.
   C: The instrument (Both are string instruments, similar but different instruments)
3. **Score = 3**: *The two sentences are roughly equivalent, but some important information differs or is missing with respect to the condition.*
   For example:
   S1: A small crowd gathered around the injured person.
   S2: A crowd jumps up and down to the tunes played by an artist.
   C: Number of people
   (While both are crowds, it is important and unclear how many people there are.)
4. **Score = 4**: *The two sentences are mostly equivalent, but some unimportant details differ with respect to the condition.*
   For example:
   S1: The little girl plays the jazz guitar.
   S2: The guitar looked nice and shiny.
   C: The instrument (Guitar in both cases, but the exact type is different and unimportant)
5. **Score = 5**: *The two sentences are completely equivalent as they mean the same thing with respect to the condition.*
   For example:
   S1: Three boys play on the playground.
   S2: There are 3 girls near the fountain.
   C: The number of people (3 and three are strictly equivalent)