Annotating Training Data for Conditional Semantic Textual Similarity Measurement using Large Language Models

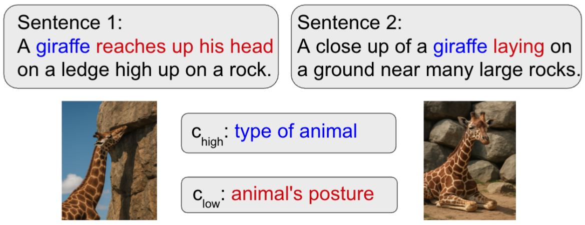
Gaifan Zhang[1] **Yi Zhou**[2] **Danushka Bollegala**[1]
[1] University of Liverpool [2] Cardiff University
sggzhan8@liverpool.ac.uk, zhouy131@cardiff.ac.uk, danushka@liverpool.ac.uk

Abstract

Semantic similarity between two sentences depends on the aspects considered between those sentences. To study this phenomenon, Deshpande et al. (2023) proposed the Conditional Semantic Textual Similarity (C-STS) task and annotated a human-rated similarity dataset containing pairs of sentences compared under two different conditions. However, Tu et al. (2024) found various annotation issues in this dataset and showed that manually re-annotating a small portion of it leads to more accurate C-STS models. Despite these pioneering efforts, the lack of large and accurately annotated C-STS datasets remains a blocker for making progress on this task as evidenced by the subpar performance of the C-STS models. To address this training data need, we resort to Large Language Models (LLMs) to correct the condition statements and similarity ratings in the original dataset proposed by Deshpande et al. (2023). Our proposed method is able to re-annotate a large training dataset for the C-STS task with minimal manual effort. Importantly, by training a supervised C-STS model on our cleaned and re-annotated dataset, we achieve a 5.4% statistically significant improvement in Spearman correlation. The re-annotated dataset is available at `https://LivNLP.github.io/CSTS-reannotation`.

1 Introduction

Semantic Textual Similarity (STS) is a fundamental Natural Language Processing (NLP) task to evaluate the semantic similarity between two given sentences (Agirre et al., 2012). However, the focus on the sentences can vary and affects the judgment of similarity. To address this, Deshpande et al. (2023) introduced a novel C-STS task, which measures the similarity between two sentences under a specified condition. In the C-STS dataset, each sentence pair has two conditions – a condition c_{low} producing a low semantic similarity, and a condition c_{high} a high semantic similarity, as shown in Figure 1. The similarity under each condition is rated on an ordinal scale from 1 (low similarity) to 5 (high similarity).



Sentence 1:
A giraffe reaches up his head on a ledge high up on a rock.

Sentence 2:
A close up of a giraffe laying on a ground near many large rocks.

c_{high}: type of animal

c_{low}: animal's posture

Figure 1: An example C-STS instance. The two sentences are compared under two different conditions, focusing on different aspects, resulting in a high (score of 5), and a lower (score of 1) semantic similarities. Images are only for visual cue.

While the C-STS task brings greater specificity to the aspects of sentences being compared, Tu et al. (2024) observed that both the conditions and human similarity ratings suffer from issues such as ambiguity and inaccuracy, introducing label noise into the task. Although recent methods (Li et al., 2024; Liu et al., 2025; Yoo et al., 2024) have advanced the modeling of C-STS, their performance is still limited by the dataset quality, with Spearman correlations generally remaining below 0.5. To reduce those identified annotation errors, Tu et al. (2024) re-annotated the validation portion of the dataset with the help of human annotators. However, as discussed later in §2.1, in addition to annotation errors in similarity ratings, we find that the conditions themselves can be problematic, such as expressing varying granularities and a high-level of subjectivity, further impacting the reliability of the dataset. Moreover, the validation data re-annotated by Tu et al. (2024) consists of only a small proportion (15%) of the C-STS dataset. Although it would be ideal to manually re-annotate the full C-STS dataset it is a costly task.

To address this data cleansing task, we use LLMs

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 27015–27027
November 4-9, 2025 ©2025 Association for Computational Linguistics

Dataset	Train	Validation	Test	Count
Deshpande et al. (2023)	11342	2834	4732	18908
Tu et al. (2024)		✓		2834
Ours	✓	✓		**14176**

Table 1: Dataset size comparison. The portions that have been re-annotated by Tu et al. and this work (ours) are indicated by ✓.

to (1) modify the conditions, and (2) re-annotate the similarity ratings between two sentences under the modified conditions, requiring minimum manual effort. LLMs have been successfully used to generate synthetic training data and to provide judgements for several related NLP tasks (Peng et al., 2023; Patel et al., 2024; Wei et al., 2024). It is noteworthy that prior work (Deshpande et al., 2023) using LLMs such as GPT-4 (OpenAI et al., 2023) and Flan-T5 (Chung et al., 2024) to predict C-STS have reported suboptimal performance where they observed numerous issues including semantically similar sentence pairs being incorrectly assigned with low similarity scores. While we also use LLMs to correct the conditions and similarity ratings, we aim to improve the effectiveness of the C-STS training data by improving annotation accuracy and increasing the number of high-quality and reliable instances, such that better C-STS models can be trained.

We cleaned the training dataset proposed by Deshpande et al. (2023), which accounts for 75% of the whole dataset, as demonstrated in Table 1. This provides more reliable training instances for the C-STS task. Since the test set labels have not been released, we do not modify the test instances. Our contributions in this paper are three-fold.

1. We first correct the errors and refine the expressions in the condition statements in the C-STS dataset (§2.1).

2. Next, we use two LLMs (i.e. `GPT-4o` and `Claude-3.7-Sonnet`) to independently obtain C-STS ratings, which we then combine with the original human ratings by averaging (§2.2).

3. To evaluate the usefulness of our LLM-cleansed dataset, we train a supervised C-STS model on it following the method proposed by Zhang et al. (2025).

Issue	Condition
Imbalanced Condition	number of # type of # color of #
Subjective Condition	The age of person. The color of animal. The number of people.
Inconsistent Phrasing Style	The all are food. Where the dog is visible from. The amount of stoves/ ovens. Type of room. The person's age.
Varying Granularity	The absence of tomato. The place of the object. The species of the one who's in the room.
Verbose Expression	The fact that they're both girls. String instrument being played. The players move to the position.
Grammatical Issue	The thing that fly.

Table 2: Common stand-alone condition issues.

Our evaluations show that the trained model obtains a Spearman correlation of 73.9% against the human-rated test data, thereby demonstrating the usefulness of our dataset when training C-STS models. Specifically, our cleaned and re-annotated dataset achieves a 5.4% statistically significant improvement measured in Spearman correlation.

2 C-STS Training Data Cleansing

Our data cleansing method for C-STS consists of two steps. In the first step (§2.1), we identify common issues with the conditions and use `GPT-4o` to refine those. In the second step (§ 2.2), we re-annotate the labels using both `GPT-4o` and `Claude-3.7-Sonnet`, due to their high performance on natural language understanding as demonstrated by Chatbot Arena leader-board (Zheng et al., 2023).[1] Empirically, we find that both of those LLMs generated ratings demonstrate a high level of agreement with the human C-STS ratings, resulting in Spearman correlations of 62% and 66% on the human-reannotated test set (ReTest) by Tu et al. (2024), respectively. Finally, we aggregate the human ratings in the original dataset with the two sets of LLM ratings.

2.1 Modifying the Conditions

We identify multiple issues in the conditions that impact the accuracy of the human annotations.

[1] `https://lmarena.ai/`

Issue	Sentence Pair	Condition
Ambiguous Condition	A climber with a yellow backpack walks along the ridge of a snowy mountainside. A person in a red hat with a huge backpack going hiking.	The climber.
Invalid Condition	A man wearing yellow and blue is riding a large, bucking bull. A bull rider, in full padding and wearing a helmet, rides a large brown and white bull.	Color of bull.
Unrelated Condition	Three hotdogs on buns with whole slices of relish sit on a white plate. A hot dog on a bun with a drop of ketchup on the table.	The number of dogs.

Table 3: Common condition issues that cause the judgment divergence related to sentences.

These issues fall into two categories: (1) conditions that are inherently ambiguous or misleading in their own (**stand-alone** condition issues), and (2) conditions that are misleading when interpretting the sentence semantics (**sentence-dependent** condition issues). Next, we describe those issues.

2.1.1 Stand-alone Condition Issues

Imbalanced Conditions: Certain condition types occur far more frequently than the others, resulting in a highly imbalanced distribution (see Appendix A), biasing model training and evaluation. For example, the condition types *number of #* and *type of #* take 16.7% and 16.6% of the dataset, respectively.

Subjective Conditions: Some conditions introduce discrepancies with the human similarity ratings because different annotators can interpret the same condition differently. As a result, different annotators can assign contradicting similarity ratings to the same sentence pair. For example, when comparing the two numbers 2 and 3 (in the case of condition *number of #*), one annotator might consider the numerical closeness (i.e. 2 is closer to 3) as an indication of high similarity, while another may regard this as an inequality (i.e. 2 is not equal to 3), assigning a low similarity. Appendix B presents examples of such subjectivity and inconsistency in human similarity judgments. This annotation noise in the original C-STS dataset reduces the reliability of model evaluations.

Inconsistent Phrasing Styles: The phrasing of some conditions is inconsistent, ranging from full sentences to fragmented sentences or phrases. Moreover, they lack uniformity in both stopword usage and their grammatical structure.

Varying Granularity: Conditions range from very general to overly specific. This divergence affects how the models interpret those conditions.

Verbose Expressions: Conditions can sometimes have over-complex expressions, including words that overly elaborate sentence structures.

Grammatical Issues: Obvious English grammatical errors exist in some of the conditions.

Table 2 shows examples of the above-mentioned issues.

2.1.2 Sentence-dependent Condition Issues

Ambiguous Conditions: Tu et al. (2024) found that conditions presented as singletons without associated entity features to be ambiguous, lacking a clear specification of the aspects being compared.

Invalid Conditions: Tu et al. (2024) showed that some of the conditions to be invalid, as they require information that cannot be inferred from the sentences based on those conditions.

Unrelated Conditions: Some conditions contain typos or imprecise expressions. Although comprehensible by humans, such issues could mislead embedding model judges.

Table 3 shows examples of the above-mentioned issues. We also observe overlaps of sentences and conditions between the training and test sets (see Appendix C for details), which can overestimate the generalisability of the models. To standardise the condition expressions and improve their specificity and accuracy to reduce ambiguity, we use `GPT-4o` to refine the conditions. The complete prompt, along with examples before/after the modified conditions, is provided in Appendix D. Specifically, we instruct `GPT-4o` using a prompt that provides explicit guidelines and constraints. The prompt requires that conditions to be clear, specific, and semantically grounded, discouraging vague references (e.g., “animal”) in favour of more precise formulations (e.g., “species of animal”). We also remove redundant stopwords (e.g., “the”) and maintain a uniform phrasing style across all conditions. Additionally, the prompt requests a justification for any substantive modifications.

2.2 Re-annotating the Similarity Ratings

After refining the conditions, we use LLMs to re-annotate the similarity ratings in the training set. Specifically, we use `GPT-4o` and `Claude-3.7-Sonnet` with a few-shot prompt, providing five examples covering similarity ratings (1–5), each accompanied by a human-written justification. We also require LLMs to give corresponding justifications for their similarity ratings. This design serves two purposes: (1) it helps the LLM to understand the scoring rubric in a conditional STS context; and (2) it encourages the generation of not only a similarity rating but also a justification, which serves as a self-check mechanism to reduce hallucinations and improve the annotation quality. We use the same five-point rating scale proposed by Deshpande et al. (2023) and instruct the LLMs to only return a JSON-formatted object instead of a natural language commentary. The complete prompt, along with examples before/after re-annotating the similarity ratings under the modified conditions is provided in Appendix E.

Our preliminary analysis of the condition patterns and human ratings showed that the condition type *number of #* takes the largest proportion in the dataset and has a serious problem of subjectivity as described in section 2.1.1. Therefore, we provide additional clarification and instructions to LLMs along with the general scoring definition by Deshpande et al. (2023). We adopt the re-annotation strategy of Tu et al. (2024), assigning high similarity scores to sentence pairs that contain the same counted number and low similarity scores when the numbers differ. If the numbers cannot be counted explicitly, the annotation relies on the approximate quantities and follows the general similarity definition. This adjustment improves the consistency and interpretability of the dataset on this specific condition type.

To further increase the reliability of the annotations, we combine the original human ratings with multiple LLM-predicted ratings. Specifically, for each instance, we compute the arithmetic mean of the original human-annotated similarity rating (y^{human}), the predicted ratings by `GPT-4o` ($y^{\text{GPT-4o}}$), and `Claude-3.7-Sonnet` (y^{Claude}), and round the result to the nearest integer. As shown in Appendix F, combining ratings from both LLMs results in the best performance.

Train	**Test**	**Spearman**
ReVal	ReTest	61.28
ReVal-Mod w/o	ReTest-Mod w/o	64.25
ReVal-Mod w/	ReTest-Mod w/	**66.89**

Table 4: Comparison of condition modification, evaluated using an SNPro model. *w/* and *w/o* denote condition modification with and without stopword removal, respectively.

3 Experiments

For ease of disposition, we define the following dataset naming conventions. **Train-Orig** is the original training set from Deshpande et al. (2023). **Train-Mod** applies condition modifications to Train-Orig, and **Train-Mod-Reanno** further includes our re-annotated ratings. **Val-Orig** denotes the original validation set, and **Val-Reanno** is the *human* re-annotated version introduced by Tu et al. (2024). Val-Reanno is the most accurate human-verified C-STS data to date. We split **Val-Reanno** into **ReVal** (randomly selected 70%) as our validation set and **ReTest** (remaining 30%) as our test set. We construct **ReVal-Mod** and **ReTest-Mod** by applying condition modifications to ReVal and ReTest, respectively.

To evaluate the effectiveness of a particular training dataset, we first use it to train a supervised Non-Linear Projection (**SNPro**) model following Zhang et al. (2025), and then measure the improvement of C-STS task performance on the same human-labelled test data (ReTest). Details of this supervised model architecture are provided in Appendix G. Spearman’s correlation coefficient with human similarity ratings is the standard evaluation metric for C-STS, where a high correlation indicates an accurate C-STS model. We use an NVIDIA RTX A6000 GPU with PyTorch 2.0.1 and CUDA 11.7 for our experiments.

To evaluate the effectiveness of condition modification, we train **SNPro** models on ReVal and evaluate on ReTest as shown in Table 4. Further effect of stopword removal from the modified conditions is also considered. We see that the best performance is reported by the LLM-based condition modification with stopword removal (i.e. ReText-Mod w/).

Stopwords often contribute little or no semantic distinctions to the conditions, and removing them helps the model to attend to content words.

Following these findings, we apply condition modification with stopword removal and follow §2.2 to re-annotate the similarity ratings in the condition-modified C-STS training set. To measure the consistency of LLM-generated annotations, we randomly select 100 instances from Train-Mod and repeat the annotation process five times using `Claude-3.7-Sonnet` with our few-shot prompt. We measured the agreement of the five sets of annotations using the Krippendorff's Alpha (Hayes and Krippendorff, 2007) to be 0.865, indicating a high level of annotation consistency.

To validate the LLM-modified conditions and re-annotated similarity ratings, we randomly selected 300 instances from our dataset to conduct a manual verification. We find that the condition statements are clearer and more specific and in most condition statements, only stopwords are removed. Importantly, we do not find any conditions that degrade in quality or meaning altered significantly. On the other hand, we found that that 23% (69/300) of the original human ratings to be inaccurate. Roughly one-third of these inaccuracies involved serious errors, such as assigning high similarity scores to clearly dissimilar sentence pairs.

In contrast, investigating the re-annotated similarity ratings, we found that the re-annotated similarity ratings to accurately reflect the true conditional semantic textual similarity in most cases. Cases where similar sentence pairs were previously labelled as dissimilar were correctly assigned higher similarity ratings during this re-annotation process. A small proportion of instances (9%, 28/300) deviate slightly from the human ratings, with a difference of only 1 point on the [1, 5] similarity scale. Such minor disagreements are to be expected given the subjectivity involved in both the conditions and the meanings of the sentences.

Figure 2 shows how our re-annotated ratings (Train-Mod-Reanno) differ from the original annotations (Train-Orig). Although there is a better agreement for high similarity annotations, we see less agreement for lower similarity ratings. The relatively low Cohen's Kappa (Cohen, 1960) of 0.247 between the two sets of annotations indicates only fair agreement, highlighting that we have made significant revisions to the original C-STS dataset. Importantly, during our first step of modifying the conditions, we deliberately shifted the semantic focus of some sentences to ensure clearer, more consistent criteria.

Train	**Test**	**Spearman**
Train-Orig	ReTest	68.54
Train-Orig	ReTest-Mod	69.68
Train-Mod	ReTest-Mod	69.39
Train-Mod-Reanno	ReTest-Mod	**73.93**

Table 5: Spearman correlation coefficients obtained by training an SNPro model on different training datasets

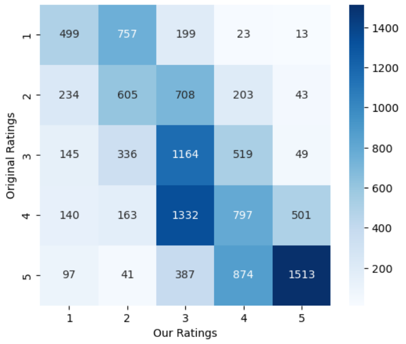


Figure 2: Confusion matrix between the original ratings (Train-Orig) and our re-annotated (Train-Mod-Reanno) ratings.

To evaluate the ability of our LLM modified conditions and the re-annotated similarity ratings for improving C-STS measurement, we train **SNPro** models using different training datasets in Table 5. Compared to training C-STS models on Train-Orig, we see that doing so on Train-Mod-Reanno results in the best performance. This is a 5.4% statistically significant improvement over the best bi-encoder C-STS performance reported by Zhang et al. (2025). This shows that, keeping the model architecture and all other training settings fixed, our re-annotated C-STS training data alone can improve the performance of C-STS. We hope that our re-annotated C-STS training data will expedite the future progress of C-STS research.

4 Conclusion

We identify key issues in the condition definitions and human-annotated similarity ratings in the original C-STS dataset. To address these, we propose an efficient LLM-based data cleansing approach that improves dataset quality through condition modification and re-annotation of similarity scores. By integrating this with human-annotated data, our cleansed dataset significantly advanced the performance of a previously proposed C-STS method.

5 Limitations

There is a large number of LLMs developed and made publicly available. However, it is practically infeasible to use multiple LLMs for the C-STS data re-annotation due to the costs involved. Therefore, we selected two highly popular and accurate models at the time of writing (`GPT-4o` and `Claude-3.7-Sonnet`) to balance performance and cost-effectiveness. Although we modified the conditions, certain stand-alone condition issues such as imbalanced conditions still exist, as the overall distribution of condition types has not changed.

This study was conducted using C-STS datasets for English, which is a morphologically limited language. However, this choice is based on the availability of C-STS datasets. To the best of our knowledge, C-STS datasets are not publicly available for languages other than English. We consider it to be an important task for future work to develop multilingual C-STS datasets to study the language-specific issues pertaining to this task.

6 Ethical Concerns

LLMs have been shown to exhibit social biases, such as those related to age and gender (Gallegos et al., 2024). Such social topics exist in the conditions for the C-STS task. Using LLMs for annotation may further propagate such biases into the dataset. The influence of whether the LLM-based annotation process impacts the data quality with respect to social bias is not evaluated in this study. Additionally, LLM-based condition-aware sentence embeddings could encode unfair social biases. Therefore, it is important to evaluate social bias amplifications (if any) due to training C-STS models on our proposed training dataset before deploying those models in downstream NLP applications.

References

Eneko Agirre, Daniel Matthew Cer, Mona T Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic Textual Similarity. *SemEval*, pages 385–393.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Ameet Deshpande, Carlos E Jimenez, Howard Chen, Vishvak Murahari, Victoria Graf, Tanmay Rajpurohit, Ashwin Kalyan, Danqi Chen, and Karthik Narasimhan. 2023. CSTS: Conditional semantic textual similarity. *Empir Method Nat Lang Process*, pages 5669–5690.

Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179.

Andrew F Hayes and Klaus Krippendorff. 2007. Answering the call for a standard reliability measure for coding data. *Communication methods and measures*, 1(1):77–89.

Baixuan Li, Yunlong Fan, and Zhiqiang Gao. 2024. Seaver: Attention reallocation for mitigating distractions in language models for conditional semantic textual similarity measurement. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 78–95.

Xinyue Liu, Zeyang Qin, Zeyu Wang, Wenxin Liang, Linlin Zong, and Bo Xu. 2025. Conditional semantic textual similarity via conditional contrastive learning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4548–4560.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2023. GPT-4 Technical Report. *arXiv [cs.CL]*.

Ajay Patel, Colin Raffel, and Chris Callison-Burch. 2024. Datadreamer: A tool for synthetic data generation and reproducible llm workflows. *arXiv preprint arXiv:2402.10379*.

Letian Peng, Yuwei Zhang, and Jingbo Shang. 2023. Generating efficient training data via llm-based attribute manipulation. *arXiv preprint arXiv:2307.07099*.

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, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hui Wei, Shenghua He, Tian Xia, Fei Liu, Andy Wong, Jingyang Lin, and Mei Han. 2024. Systematic evaluation of llm-as-a-judge in llm alignment tasks: Explainable metrics and diverse prompt templates. *arXiv preprint arXiv:2408.13006*.

Young Yoo, Jii Cha, Changhyeon Kim, and Taeuk Kim. 2024. Hyper-CL: Conditioning sentence representations with hypernetworks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 700–711, Bangkok, Thailand. Association for Computational Linguistics.

Gaifan Zhang, Yi Zhou, and Danushka Bollegala. 2025. Case – condition-aware sentence embeddings for conditional semantic textual similarity measurement. *Preprint*, arXiv:2503.17279.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Supplementary Materials

A Imbalanced Condition

By analysing the distribution of condition types in the C-STS training dataset, we observe a significant imbalance. As shown in Table 6, two broad condition categories, *number of #* and *type of #*, dominate the dataset, accounting for 16.7% and 16.6% of all conditions, respectively.

With respect to specific conditions, we present the 15 most frequent ones in Table 7. The most common conditions include *The number of people.*, *The type of animal.*, and *The sport.* However, these frequently occurring conditions often introduce problems such as ambiguity and subjectivity in the evaluation process.

Condition Type	Count	Percentage
number of #	1892	16.7%
type of #	1886	16.6%
color of #	664	5.9%
action	357	3.1%
position of #	88	0.8%

Table 6: Counts of general condition types (top 5) in the original C-STS training dataset.

Condition	Count
The number of people.	520
The type of animal.	254
The sport.	249
The name of the place.	162
The animal.	154
The color of the shirts.	123
The number of people visible.	103
The action.	94
The type of food.	87
The number of animals.	85
The type of clothing.	85
The number of people in the image.	72
The location.	65
The color of the clothing.	64
The number of objects.	62

Table 7: Counts of specific conditions (top 15) in the original C-STS training dataset.

B Subjectivity in Human Annotations

Human annotators can give contradictory ratings to some similar instances in the dataset. We show subjectivity in human ratings for the conditions *The number of people*, *Age of person* and *Gender of person* in the original C-STS training dataset as examples. Table 8 lists some examples of instances that show subjectivity. We explain them one by one as follows.

Considering the condition ***The number of people***:

In the instance that Sentence 1: *A man and woman sitting in a booth together and smiling.*, Sentence 2: *Three people sitting at a table at a restaurant.*, Rating: 4, there are 2 people in Sentence 1, and 3 people in Sentence 2. While the number of people differs (2 vs. 3), annotators still rated the pair as highly similar. This suggests that some annotators perceive small differences in number (such as 2 versus 3) as relatively minor.

In the instance that Sentence 1: *A baseball player swings to hit the ball as another player catches.*, Sentence 2: *A man in a white and black uni- form is attempting to swing a baseball bat.*, Rating: 4, there are 2 people in Sentence 1 and 1 person in Sentence 2. Human annotators give this small difference in number a score of high similarity.

However, in another instance that Sentence 1: *A person is diving into blue water on a rocky coast.*, Sentence 2: *Two males on a rock over water, one in midair about to dive.*, Rating: 1, there are 1 person in Sentence 1 and 2 people in Sentence 2. The number of people is also different, but similar

Sentence 1	Sentence 2	Condition	Rating
A man and woman sitting in a booth together and smiling.	Three people sitting at a table at a restaurant.	The number of people.	4
A baseball player swings to hit the ball as another player catches.	A man in a white and black uniform is attempting to swing a baseball bat.	The number of people.	4
A person is diving into blue water on a rocky coast.	Two males on a rock over water, one in midair about to dive.	The number of people.	1
A person is doing a trick in the air on a bike near some buildings.	Person performing a move on a mountain bike with two people watching.	The number of people.	1
A young girl with a sippy cup swings on a swing.	A child is making a ridiculous face with an open mouth.	The number of people.	4
The boy on the bike is wearing safety glasses and a red helmet.	A man dressed in bicycle gear is riding through a course.	Age of person.	1
Two images show a man reaching out to hit a tennis ball with a racket.	A boy in black shorts jumps and holds his tennis racket out in front of him.	Age of person.	3
A very happy child sits on a chair on top of some rocks.	A child is bouncing on a trampoline that is near a house.	Age of person.	3
A man in a red and yellow outfit is riding a bicycle on one wheel.	A woman is riding a bike with a basket of flowers.	Gender of person.	1
A woman with a red scarf around her neck is smiling.	A man in a black hat looks very happy.	Gender of person.	4
A little girl is brushing her teeth in a bathroom.	A woman is brushing her teeth in a bathroom mirror.	Gender of person.	1
A man is skateboarding on the sidewalk.	A girl is rollerblading on a path.	Gender of person.	4

Table 8: Examples of sentence pairs under the conditions “The number of people”, “Age of person”, and “Gender of person” with subjective similarity ratings by human annotators in the original C-STS training set.

in number (same case as the previous example). Some annotators interpret it as a stronger signal of dissimilarity.

Additionally, in the instance that Sentence 1: *A person is doing a trick in the air on a bike near some buildings.*, Sentence 2: *Person performing a move on a mountain bike with two people watching.*, Rating: 1, there are 1 person in Sentence 1 and 3 people in Sentence 2. Human annotators can regard this mismatch in number as dissimilarity.

Moreover, in the instance that Sentence 1: *A young girl with a sippy cup swings on a swing.*, Sentence 2: *A child is making a ridiculous face with an open mouth.*, Rating: 4, both sentences have 1 person. Human annotators give a high similarity score of 4, even though the numbers are exactly the same.

Considering the condition ***Age of person***:

In the instance that Sentence 1: *The boy on the bike is wearing safety glasses and a red helmet* and Sentence 2 is: *A man dressed in bicycle gear is riding through a course*, the rating is 1. The perceived age difference between “boy” and “man” leads to a low similarity rating. Some annotators may weigh age references heavily when evaluating similarity.

In contrast, in the instance that Sentence 1: *Two images show a man reaching out to hit a tennis ball with a racket* and Sentence 2 is: *A boy in black shorts jumps and holds his tennis racket out in front of him*, the rating is 3. While the age difference between “man” and “boy” still exists, annotators give a moderate similarity score.

In another instance that Sentence 1: *A very happy child sits on a chair on top of some rocks.* and Sentence 2 is: *A child is bouncing on a trampoline that is near a house*, the rating is 3. Both sentences have description about the "child", which should be a higher similarity score of 4. At least, the label should be different with the previous example which compares the age of "man" and "child".

Considering the condition ***Gnender of person***:

In the instance that Sentence 1: *A man in a red and yellow outfit is riding a bicycle on one wheel* and Sentence 2: *A woman is riding a bike with a basket of flowers*, the rating is 1. Some annotators view gender as a central feature for this condition, leading to a low similarity rating despite shared activity.

However, in the instance that Sentence 1: *A woman with a red scarf around her neck is smiling* and Sentence 2: *A man in a black hat looks very happy*, the rating is 4. Even though the genders differ, the facial expressions and emotional tone are similar, suggesting that some annotators focus more on affective similarity than gender cues, which is inaccurate.

In the instance that Sentence 1: *A little girl is brushing her teeth in a bathroom.* and Sentence 2: *A woman is brushing her teeth in a bathroom mirror.*, the rating is 1. The gender is both sentences is female. Human annotators should not give a dissimilar score based on gender. When gender information matches across two sentences, it should not contribute to a higher dissimilarity rating.

In the instance that Sentence 1: *A man is skateboarding on the sidewalk.* and Sentence 2: *A girl is rollerblading on a path.* , the rating is 4. The gender is male in Sentence 1, but the gender is female in Sentence 2. Humman annotators should not give a high similarity score of 4 to this mismatching gender information.

C Overlapping Statistics between original training and test sets

The overlaps between the original training and test sets by Deshpande et al. (2023) are counted across the following five types:

- Sentence only
  The same sentence appears, but possibly with different conditions.
  Overlap count: 1,196
  Test side: 27.08% of sentences overlap

- Condition only
  The same condition text appears, but possibly paired with different sentences.
  Overlap count: 804
  Test side: 40.75% of conditions overlap

- Single Sentence with Condition
  A single sentence–condition pair is repeated.
  Overlap count: 185
  Test side: 1.96% overlap

- Sentence pair (order-insensitive)
  The same pair of sentences appears (regardless of order).
  Overlap count: 9
  Test side: 0.38% overlap

- Sentence pair with Condition
  A full instance (two sentences with a condition) is duplicated.
  Overlap count: 2
  Test side: 0.042% overlap

Over one quarter of test sentences and over two-fifths of test conditions are also seen in the training set. Such overlaps may lead to overestimated performance for language models.

D Prompt Used for Modifying the Conditions

Figure 3 shows the full prompt for condition modification. Table 9 provides examples of how our prompt effectively refines various types of problematic conditions.

E Prompt Used for Similarity Annotations

Figure 4 shows the complete prompt for assigning similarity ratings using LLMs. Table 10 provides examples of the original and our re-annotated ratings, showing the improvement in the accuracy of C-STS scores. Selected examples are based on the conditions of the same semantic focus (conditions modified only with stopword removal).

F Evaluating the Averaging Method

Table 11 reports the average performance across different rating aggregation strategies. We use Train-Mod training set with ratings as shown in the table. We use NV-Embed-v2 (NV) to first generate condition-aware sentence embeddings and then train the supervised multi-head non-linear projection as described in §3. Embeddings are evaluated on the ReTest-Mod test set. The projection model is fixed with a hidden dimensionality of 1024, output dimensionality of 512, and a dropout rate of 0.1. Results show that combining human ratings with annotations from both LLMs yields the highest performance.

Condition issue	Before	After
Ambiguous Condition	The animal. The sport.	type of animal presence of vehicles
Unrelated Condition	The name of the game.	type of sport
Inconsistent Phrasing Style	What the person is holding.	object being held
Varying Granularity	The setting. Specific areas of the home.	urban environment areas of home
Verbose Expression	If a tv is present.	presence of television
Grammatical Issue	The food with plate. The the size of the room.	food on plate size of room

Table 9: Examples of conditions before and after using our condition modification prompt.

Sentence 1	Sentence 2	Condition	Before	After
A room that has white walls and a window shade up has a double unmade bed on the floor.	A bed appears to have nothing else on it except two pillow in a bedroom.	type of room	2	4
A deep dish pizza in a metal pan topped with several kinds of toppings.	The margarita pizza is on a plate, and ready to be cut and served.	type of pizza	5	3
Older men sitting on wooden benches on a sidewalk together, with scooters parked in the street and stores across the street.	There are people looking at a booth and a woman and man in a wheelchair on the sidewalk.	gender of people	5	3
a man sitting on a couch with a silver laptop in a living room	A computer desk topped with a monitor and a keyboard next to a mouse.	number of people	4	1
A person flying a kite at the beach while two others walk past him	Three people standing on the shore of a sandy beach in front of waves	action of people	5	3
A colorful purple airplane sits on the runway with a darkened sky in the background.	A white and gray passenger plane has just landed or is about to take off.	type of vehicle	2	4
Two elephants are bathing in deep water as a person sits on one of their backs.	A group of people stand on the shore while watching an elephant in the water.	name of animal	2	4

Table 10: Examples of ratings with modified condition before and after using our re-annotation prompt.

G Supervised Non-Linear Projection

The supervised non-linear projections are proposed by Zhang et al. (2025). These supervised models are Siamese bi-encoders tailored for the C-STS task which have proven high performance (Deshpande et al., 2023; Yoo et al., 2024). Each model takes as input two condition-aware embeddings corresponding to sentence 1 and sentence 2 with the condition, respectively.

Zhang et al. (2025) propose that input condition-aware sentence embeddings are generated from LLM-based models, using the prompt “Retrieve semantically similar texts to the [CONDITION], given the Sentence: [SENTENCE].” They show that the LLM-based embeddings work better than

Rating Data	Spearman
$y^{\text{GPT-4o}}$	70.88
y^{Claude}	71.95
$\mathrm{V}(y^{\text{GPT-4o}} + y^{\text{Claude}})$	72.21
$\mathrm{V}(y^{\text{GPT-4o}} + y^{\text{human}})$	71.11
$\mathrm{V}(y^{\text{Claude}} + y^{\text{human}})$	72.74
$\mathrm{V}(y^{\text{human}} + y^{\text{GPT-4o}} + y^{\text{Claude}})$	**73.10**

Table 11: Average Spearman Correlation based on rating data across different aggregation strategies. V() denotes taking the arithmetic mean and rounding to the nearest integer.

the Masked Language Model (MLM)-based embeddings. To improve the condition-specific relevance, a post-processing step of subtracting the corresponding embeddings of the conditions is applied after generating the condition-aware sentence embeddings. Here, the embeddings of the conditions are generated using the prompt "Retrieve semantically similar texts to a given Sentence: [CONDITION]."

Denote the resulting LLM-generated condition-aware sentence embeddings by $\mathbf{e}_1, \mathbf{e}_2$ for each instance. The **Supervised Non-Linear Projection** (SNPro) is defined as $f(\cdot)$, a two-layer feed-forward network with ReLU activations and dropout. The final projected embeddings are obtained as

$$\mathbf{z}_i = f(\mathbf{e}_i), \quad i \in \{1, 2\}.$$

Hyperparameters are tuned on our validation set ReVal-Mod. We fix the batch size to 512, the dropout rate to 0.15 and the learning rate to 10^{-3}. We select the output dimensionality of 512.

Model	Non-linear Feed Forward Network (FFN)	Linear FFN
NV	69.30	**69.95**
SFR	**62.85**	59.22
GTE	**64.16**	56.10
E5	**62.12**	47.03
SimCSE_large	**56.67**	45.96
SimCSE_base	**56.60**	39.54

Table 12: Spearman correlation of embedding models based on supervised FFNs with reduced dimensionality 512.

Zhang et al. (2025) found that LLM-based models work better than MLM-based models such as SimCSE for the C-STS task. Although a direct comparison with prior C-STS methods is challenging due to issues in the test sets and lack of implementation details (e.g., Tu et al. (2024) do not release their hyperparameters or test/validation splits), we include a comparison table to highlight the performance improvements achieved using the method proposed by Zhang et al. (2025). Table 12 shows the performance of different embedding models. Three are LLM-based: *NV-Embed-v2* (**NV**), *SFR-Embedding-Mistral* (**SFR**), *gte-Qwen2-7B-instruct* (**GTE**). Three are MLM-based: *Multilingual-E5-large-instruct* (**E5**), *sup-simcse-roberta-large* (**SimCSE_large**), and *sup-simcse-bert-base-uncased* (**SimCSE_base**). [2] NV achieves the highest Spearman correlation, significantly outperforming all other models. Therefore, we select NV as the base model for evaluating dataset cleansing effectiveness in our study.

[2] All models are available at `https://huggingface.co/spaces/mteb/leaderboard` and `https://huggingface.co/princeton-nlp`

```
This is a Conditional STS task: Evaluate the similarity between the two sentences, with respect to
the condition.
Sentence pair has a label (score) between 1 and 5 as follows: 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.

Check and modify the provided condition if it is inaccurate or ambiguous, following these guidelines
strictly:
* Conditions must be clear and specific. (e.g., instead of ”animal”, specify clearly such as
”species of animal”.)
* Remove stopword from conditions (e.g., ”the”).
* Conditions must accurately match human-annotated labels.
* Provide conditions concisely, without context-specific details. Good examples: color of clothing,
type of event, intention of travel.
* Do NOT overly specify the condition more narrowly than the original meaning.

Return a JSON object with two fields:
improved_condition: the improved condition,
justification: a single sentence explaining why you update the condition.
Give empty str this if only stopword ’the’ is removed.
```

Figure 3: Prompt for modifying conditions

```
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.

Evaluate the similarity for condition type ”number of”, following these guidelines strictly:
* Numbers need to be counted explicitly (e.g., “a man and a woman” → 2 people)
* If the two sentences mention the same number of entities → Label = 5
* If the numbers differ → Label = 1
* If no explicit number, follow the definition above and judge based on approximate quantity (e.g.,
”many” vs ”a few”).

Return a JSON object with two fields:
”rating”: the similarity rating (between 1 to 5 as defined above),
”justification”: a single sentence explaining why you gave that similarity rating.

Do not return anything else other than this JSON object.
Do not use code blocks.

## Example 1
Sentence1: A close up of a giraffe laying on a ground near many large rocks.
Sentence2: A giraffe reaches up his head on a ledge high up on a rock.
Condition: animal’s posture
{”rating”: 1, ”justification”: ”In Sentence1 the giraffe is lying down, while in Sentence2 the
giraffe is stretching its head upward.”}

## Example 2
Sentence1: This bathroom stall has toilet tissue on the floor while the toilet is raised.
Sentence2: A full trashcan is beside the commode in a public restroom toilet that needs to be
cleaned.
Condition: location of trash
{”rating”: 2, ”justification”: ”Sentence2 does not clearly state that there is any trash outside
the trashcan.”}

## Example 3
Sentence1: A large red and blue boat sitting on top of a lake next to other boats.
Sentence2: Part of a ship sits in the shallow end of the bay next to a city.
Condition: body of water type
{”rating”: 3, ”justification”: ”The two sentences mention lake and bay and are roughly equivalent,
but Sentence2 does not clarify whether it is a bay within a lake.”}

## Example 4
Sentence1: A monkey mug in front of a computer with a stuffed penguin beside it.
Sentence2: A laptop computer sitting on top of a table next to two computer monitors.
Condition: name of the device
{”rating”: 4, ”justification”: ”Both sentences mention computers, but Sentence1 does not specify
the type, while Sentence2 explicitly mentions a laptop.”}

## Example 5
Sentence1: This bathroom stall has toilet tissue on the floor while the toilet is raised.
Sentence2: A full trashcan is beside the commode in a public restroom toilet that needs to be
cleaned.
Condition: room function
{”rating”: 5, ”justification”: ”Both sentences describe a room functioning as a restroom or toilet.”}
```

Figure 4: Few-shot prompt for conditional sentence similarity annotation