Revealing the Blind Spot of Sentence Encoder Evaluation by HEROS

Cheng-Han Chiang[†] **Yung-Sung Chuang**[‡] **James Glass**[‡] **Hung-yi Lee**[†]
dcml0714@gmail.com yungsung@mit.edu
National Taiwan University[†] Massachusetts Institute of Technology[‡]

Abstract

Existing sentence textual similarity benchmark datasets only use a single number to summarize how similar the sentence encoder's decision is to humans'. However, it is unclear what kind of sentence pairs a sentence encoder (SE) would consider similar. Moreover, existing SE benchmarks mainly consider sentence pairs with low lexical overlap, so it is unclear how the SEs behave when two sentences have high lexical overlap. We introduce a high-quality SE diagnostic dataset, HEROS. HEROS is constructed by transforming an original sentence into a new sentence based on certain rules to form a *minimal pair*, and the minimal pair has high lexical overlaps. The rules include replacing a word with a synonym, an antonym, a typo, a random word, and converting the original sentence into its negation. Different rules yield different subsets of HEROS. By systematically comparing the performance of over 60 supervised and unsupervised SEs on HEROS, we reveal that most unsupervised sentence encoders are insensitive to negation. We find the datasets used to train the SE are the main determinants of what kind of sentence pairs an SE considers similar. We also show that even if two SEs have similar performance on STS benchmarks, they can have very different behavior on HEROS. Our result reveals the blind spot of traditional STS benchmarks when evaluating SEs.[1]

1 Introduction

Sentence encoders (SEs) are fundamental building blocks in miscellaneous natural language processing (NLP) tasks, including natural language inference, paraphrase identification, and retrieval (Gillick et al., 2018; Lan and Xu, 2018). SEs are mostly evaluated with the semantic textual similarity (STS) datasets (Agirre et al., 2016; Cer et al., 2017) and SICK-R (Marelli et al., 2014), which consist of sentence pairs with human-labeled similarity scores. The performance of the SEs is summarized using Spearman's correlation coefficient between the human-labeled similarity and the cosine similarity obtained from the SE.

	R1	R2	RL	Lev	Len
STS-b	55.8	32.5	53.2	0.54	12.2
SICK-R	61.2	37.4	56.2	0.53	10.0
HEROS	92.9	84.8	92.9	0.10	13.8

Table 1: We use the ROUGE F1 scores (R1, R2, RL) and the normalized Levenshtein distance (Lev) between sentence pairs to evaluate the degree of lexical overlaps in HEROS and another two widely used STS benchmarks. A higher ROUGE score and a lower normalized Levenshtein distance imply higher lexical overlaps. Len is the average sentence length. Please find details about the metrics used here in Appendix B.

While the STS benchmarks are widely adopted, there are two problems with these benchmarks. First, the performance on the STS dataset does not reveal much about what kind of sentence pairs would the SE deem *similar*. Spearman's correlation coefficient only tells us how correlated the sentence embedding cosine similarity and the ground truth similarity are. However, the idea of what is similar can vary among different people and depend on the task at hand. Therefore, just because the sentence embedding cosine similarity is strongly correlated to the ground truth similarity, it does not provide much information about the specific type of similarity that the SE captures. Prior works mostly resort to a few hand-picked examples to illustrate what kind of sentence pairs an SE would consider similar or dissimilar (Gao et al., 2021; Chuang et al., 2022; Wang et al., 2022). But it is hard to fully understand the traits of an SE by using only a few hand-picked samples.

The second issue is that sentence pairs in the STS-related benchmarks often have low lexical

[1]We release the dataset on https://huggingface.co/datasets/dcml0714/Heros.

Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023), pages 289–302

overlaps, as shown in Table 1, making it unclear how the SEs will perform on sentence pairs with high lexical overlaps, which exist in real-world applications such as adversarial attacks in NLP. Adversarial samples in NLP are constructed by replacing some words in an original sentence with some other words (Alzantot et al., 2018), and the original sentence and the adversarial sample will have high lexical overlaps. SEs are often adopted to check the semantic similarity between the original sentence and the adversarial sample (Garg and Ramakrishnan, 2020; Li et al., 2020b). If we do not know how SEs perform on high lexical overlap sentences, using them to check semantic similarity is meaningless.

To address the above issues, we construct and release a new dataset, HEROS: **H**igh-l**e**xical ove**r**lap diagn**o**stic dataset for **s**entence encoders, for evaluating SEs. HEROS is composed of six subsets, and each subset includes 1000 sentence pairs with very high lexical overlaps. For the two sentences in a sentence pair, one of them is created by modifying the other sentence based on certain rules, and each subset adopts a different rule. These rules are (1) replacing a word with a synonym, (2) replacing a word with an antonym, (3) replacing a word with a random word, (4) replacing a word with its typo, and (5,6) negating the sentence. By comparing the sentence embedding cosine similarity of sentence pairs in different subsets, we can understand what kind of sentence pairs, when they have high lexical overlaps, would be considered similar by an SE. We evaluate 60 sentence embedding models on HEROS and reveal many intriguing and unreported observations on these SEs.

While some prior works also crafted sentence pairs to understand the performance of SEs, they either do not make the datasets publicly available (Zhu et al., 2018; Zhu and de Melo, 2020) or do not consider so many SEs as our paper does (Barancikova and Bojar, 2020), especially unsupervised SEs. Our contribution is relevant and significant as it provides a detailed understanding of SEs using a new dataset. The contribution and findings of this paper are summarized as follows:

- We release HEROS, a high-quality dataset consisting of 6000 sentence pairs with high lexical overlaps. HEROS allows researchers to systematically evaluate what sentence pairs would be considered similar by SEs when the lexical overlap is high.
- We evaluate 60 SEs on HEROS and reveal several facts that were never reported before or only studied using a few hand-picked examples.
- We show that supervised SEs trained for different downstream tasks behave differently on different subsets of HEROS, indicating that the SEs for different tasks encode different concepts of similarity.
- We find that all unsupervised SEs are considerably insensitive to negation, and further finetuning on NLI datasets makes them acquire the concept of negation.
- We observe that SEs can have very different performances on different subsets of HEROS even if their average STS benchmark performance difference is less than 0.2 points.

2 HEROS

HEROS consists of six subsets, and each subset consists of 1000 sentence pairs. The six subsets are **Synonym**, **Antonym**, **Typo**, **Random MLM**, and two types of **Negation** subsets. In all subsets, each pair of sentences have high lexical overlap, and the two sentences only differ in at most one content word; we call these paired sentences *minimal pairs*.

The dataset is constructed from the GoEmotions dataset (Demszky et al., 2020), a dataset for emotion classification collected from Reddit comments. We select one thousand sentences from GoEmotions and replace *one word* in the original sentences with its synonym, antonym, a typo of the replaced word, and a random word obtained from BERT (Devlin et al., 2019) mask-infilling. Last, we convert the original sentence into its negation using two different rules. After this process, we obtain six sentences for each of the 1000 selected sentences. We pair the **original sentence** and a **converted sentence** to form a minimal pair, which has high lexical overlaps. We will explain the above process in detail in Section 2.2. Samples from HEROS are shown in Table 2.

2.1 Motivation and Intended Usage

Unlike traditional STS benchmark datasets that asked humans to assign a similarity score as the ground truth similarity, HEROS does not provide a ground truth similarity score for sentence pairs. This is because it is difficult to define how similar two sentences should be for them to be given

a certain similarity score. Moreover, the concept of similarity differs in downstream tasks. For example, in paraphrase tasks, a Negation minimal pair is considered semantically different; but for a retrieval task, we might consider them similar.

Thus, instead of providing a ground truth label for each sentence pair and letting future researchers pursue state-of-the-art results on HEROS, we hope this dataset is used for diagnosing the characteristics of an SE. Specifically, one can compare the average sentence embedding cosine similarity of sentence pairs in different subsets to understand what kind of similarity is captured by the sentence embedding model.

Different subsets in HEROS capture various aspects of semantics. Comparing the average cosine similarity between minimal pairs in Synonym and Antonym allows one to understand whether replacing a word with an antonym is more dissimilar to the original semantics than replacing a word with a synonym. The average cosine similarity between minimal pairs in Negation can tell us how negation affects sentence embedding similarity. Typos are realistic and happen every day. While humans can infer the original word from a typo and get the original meaning of the sentence, it will be interesting to see how the typos affect the sentences' similarity with the original sentences. The Random MLM subset can tell us how similar the sentence embedding can be when two sentences are semantically different but with high lexical overlaps. By comparing the performance of different SEs on different subsets in HEROS, we can further understand the trait of different SEs.

2.2 Dataset construction

2.2.1 Raw dataset preprocessing

HEROS are constructed from GoEmotions. For the sentences in GoEmotions, we only select sentences whose lengths are more than 8 words and less than 25 words. We filter out sentences with cursing, and we use language-tool to filter out sentences that language-tool find ungrammatical. We manually remove the sentences that we find offensive or harmful. The selected sentences are called **original sentences** in our paper. More details on preprocessing are presented in Appendix A.1.

2.2.2 Selecting which word to replace

The next step is to determine which word to replace in the original sentences obtained from preprocessing. The selected word must be (1) semantically significant to the original sentence so that when it is replaced with a non-synonym word, the two sentences will have vastly different meanings and would be considered contradictory in an NLI task. (2) The selected word must have synonyms and antonyms at the same time since it will be replaced with its synonyms and antonyms. We only select verbs and adjectives for replacement because changing them greatly alters the semantics of a sentence. Sentences that do not contain a word that satisfies the two criteria are dropped.

2.2.3 Synonym and Antonym Subsets

The first subset in HEROS includes the minimal pairs formed by replacing a word in the original sentence with its synonym; the second subset includes the minimal pairs formed by replacing a word with its antonym. After selecting the word to be replaced, we determine what synonym and antonym should be used for replacement. There are three principles for replacement: (1) the replacement should fit in the context, (2) the synonym should match the word sense of the original word in the sentence[2], and (3) the collocating words (e.g., prepositions, definite articles) may also need to be modified. The three guiding principles make this process require high proficiency in English and this process is impossible to be done using an automatic process. Thus, this process is performed by the authors ourselves. We use our proficiency in English and four online dictionaries to select the replacement words. The four resources are thesaurus.com, thesaurus of Merriam-Webster, Online Oxford Collocation Dictionary, and Cambridge Dictionary. This step takes 72 hours.

2.2.4 Random MLM

The third subset in HEROS is obtained by replacing the word to be replaced with a random word predicted by a masked language model. We mask the word to be replaced by `[MASK]` token and use `bert-large-uncased` to fill in the masked position. We filter out the synonym, antonym, and their derivational forms[3] from the masked prediction using WordNet and LemmInflect. Additionally, we filter out punctuations and subword tokens that are not complete words. Moreover, we manually filter out mask predictions that are very similar in mean-

[2] A word can have different word senses, and each word sense has its own synonym sets. Synonym sets of different word senses might be different.

[3] For example, different tenses of a verb.

Subset	Example (adjective)	Example (verb)
Original	*And that is why it is (or was)* ***illegal****.*	*You do not know how much that* ***boosted*** *my self-esteem right now.*
Synonym	*And that is why it is (or was)* ***illegitimate****.*	*You do not know how much that* ***increased*** *my self-esteem right now.*
Antonym	*And that is why it is (or was)* ***legal****.*	*You do not know how much that* ***lowered*** *my self-esteem right now.*
Random MLM	*And that is why it is (or was)* ***here****.*	*You do not know how much that* ***affects*** *my self-esteem right now.*
Typo	*And that is why it is (or was)* ***illiegal****.*	*You do not know how much that* ***booste*** *my self-esteem right now.*
Negation (Main)	*And that is* ***not*** *why it is (or was) illegal.*	*You* ***do*** *know how much that boosted my self-esteem right now.*
Negation (Antonym)	*And that is why it is (or was)* ***not*** *illegal.*	*You do not know how much that* ***did not boost*** *my self-esteem right now.*

Table 2: Two examples from HEROS. One example selects a verb while the other selects an adjective for replacement. The first row shows the original sentences in the GoEmotions, and the words highlighted in **blue** are the words to be replaced. Starting from the second rows are the corresponding sentences obtained from the original sentence for different subsets, and the changes compared with the original sentence are highlighted in **green**.

ing to the original word when used in the same context. This is because even if a word is not a synonym of the word to be replaced, it may still express the same meaning when used in the same context. For example, "great" is not a synonym of "good" according to WordNet, but their meaning is very similar. The resulting sentences can be ungrammatical in very few cases, but we leave them as is.

2.2.5 Typos

The fourth subset in HEROS is constructed by swapping the word to be replaced with its typo. Typos are spelling or typing errors that occur in real life. If the word to be typoed is in the Wikipedia lists of common misspellings, we replace the word with its typo in the list. If the word is not in the common misspelling list, we create a typo by randomly deleting or replacing one character or swapping two different characters in the word (He et al., 2021).

2.2.6 Negation

The last subset in HEROS is constructed by negating the original sentence. Negation can happen at different levels in a sentence, and we create two different types of negation datasets based on where the negation happens. The first one is negating the main verb, which is the action performed by the subject, in the sentence.[4] If the main verb is not negated, we negate it by adding appropriate auxiliary verbs and the word "not". If the main verb is already negated, we directly remove the word "not" and do not remove the auxiliary verb. We call this type of negation dataset the **Negation (Main)**.

The other type of negation dataset is related to the Antonym subset. A minimal pair in the Antonym subset is formed by replacing a word in a sentence with its antonym. This *implicitly* negates the meaning of the original sentence. Here, we construct another subset called **Negation (Antonym)**, which *explicitly* negates the word that is replaced with an antonym in the Antonym subset. Given a sentence pair in the Antonym dataset, there is a verb or adjective in the original sentence that is replaced by its antonym in the converted sentence. We directly negate that word in the original sentence by adding "not" in front of an adjective or adding "not" and an auxiliary verb for verbs.[5]. If the word is already negated, we remove the "not". These sentences might sound a bit strange, but

[4]When selecting the original sentences, we do not consider interrogative sentences, so we will not create a negative interrogative sentence. An interrogative sentence and its negation ask the same question and are not semantically different.

[5]If the negation of the minimal pair in the Antonym dataset happens at the main verb, the sentence pair in Negation (Main) and Negation (Antonym) will be the same.

they are still understandable. This type of negation dataset is called Negation (Antonym) because the negation is in the same place as the antonym replacement in the Antonym subset.

3 Comparing 60 Sentence Embedding Models

In this section, we compare the behavior of 60 SEs on HEROS. Detailed information about the SEs, including training data and model size, is listed in Appendix C. We calculate the cosine similarity between each minimal pair in HEROS and normalize it by a baseline cosine similarity to remove the effect of anisotropic embedding space (Ethayarajh, 2019; Li et al., 2020a). The baseline cosine similarity is calculated by averaging the similarity between 250K random sentence pairs (details in Appendix D). We report the average normalized similarity of different subsets in HEROS for each SE. For simplicity, we will use "similarity" to refer to the normalized cosine similarity.

3.1 Supervised SEs

We use 30 supervised transformer-based SEs in the SentenceTransformers toolkit (Reimers and Gurevych, 2019, 2020). These SEs are trained supervisedly using different datasets for specific downstream tasks. The results are presented in Figure 1. In Figure 1, we group SEs into groups based on what training dataset they used. We denote the datasets used to fine-tune the SEs in the parentheses in Figure 1. There are a lot of interesting observations one can obtain from Figure 1, and we are just listing some of those observations.

SEs fine-tuned only on QA datasets are insensitive to negation: The first and second blocks in Figure 1 include different SEs obtained from fine-tuning on QA datasets using contrastive learning (Hadsell et al., 2006). In the fine-tuning stage, a positive pair for contrastive learning is a pair of question and the answer to the question. The high similarity of the two Negation subsets can be explained by the dataset type used for fine-tuning: whether the answer is negated or not, it may still be considered a valid answer to the question. Hence, it is reasonable that a sentence and its negation have high similarity. We also find that replacing a word with a typo will cause the resulting sentence to have lower similarity with the original sentence compared with replacing the word with a synonym. While humans can understand the real meaning of a typo word, this is not the case for the SEs.

SEs fine-tuned from T5 are less sensitive to typos when the model size scales up: The GTR models (Ni et al., 2021) in the second block of Figure 1 and the ST5 models (Ni et al., 2022) in the fourth block are SEs fine-tuned from T5 (Raffel et al., 2020). Although the two types of models are trained using different datasets, we find that their performance on the Typo subset shares an interesting trend when the model size scales up from the smallest base-size model to the largest xxl-size model: The similarity on the Typo subset grows higher as the model gets larger and can be as high as or higher than the similarity of the Synonym subset; meanwhile, the similarity on the Synonym subset is almost unchanged when the model size gets larger. This shows that deeper model can better mitigate the negative impact of typos on sentence embeddings.

SEs fine-tuned on paraphrase datasets are extremely sensitive to negations and antonyms: The third block in Figure 1 includes the results of SEs fine-tuned on paraphrase datasets using contrastive learning. Paraphrase datasets include a combination of different datasets such as premise-hypothesis pairs in NLI datasets and duplicate question pairs. Contrary to the previous paragraph which shows fine-tuning only using question-answer pairs makes the model insensitive to negation, we see a completely different result in the third block of Figure 1. We infer that this is mainly due to the NLI datasets used for fine-tuning: negating the original sentence results in a sentence that semantically contradicts the original sentence, and will be considered as a hard negative in contrastive learning. Hence, SEs fine-tuned on NLI will be very sensitive negation. For the same reason, these SEs are also sensitive to replacing words with antonyms. The only exception is the `MiniLM L3 (para)` model (Wang et al., 2020), which has very high similarity on the Negation subsets and is even higher than the Synonym subset. We hypothesize that this is because the number of parameters of the model and the sentence embedding dimension are too small, thus limiting the expressiveness of the sentence embeddings.

SEs fine-tuned on all available sentence pair datasets are again insensitive to negations: The models in the last block in Figure 1 are fine-tuned on all available sentence-pair training data, denoted as *(all)*. The training data consist of 32 datasets

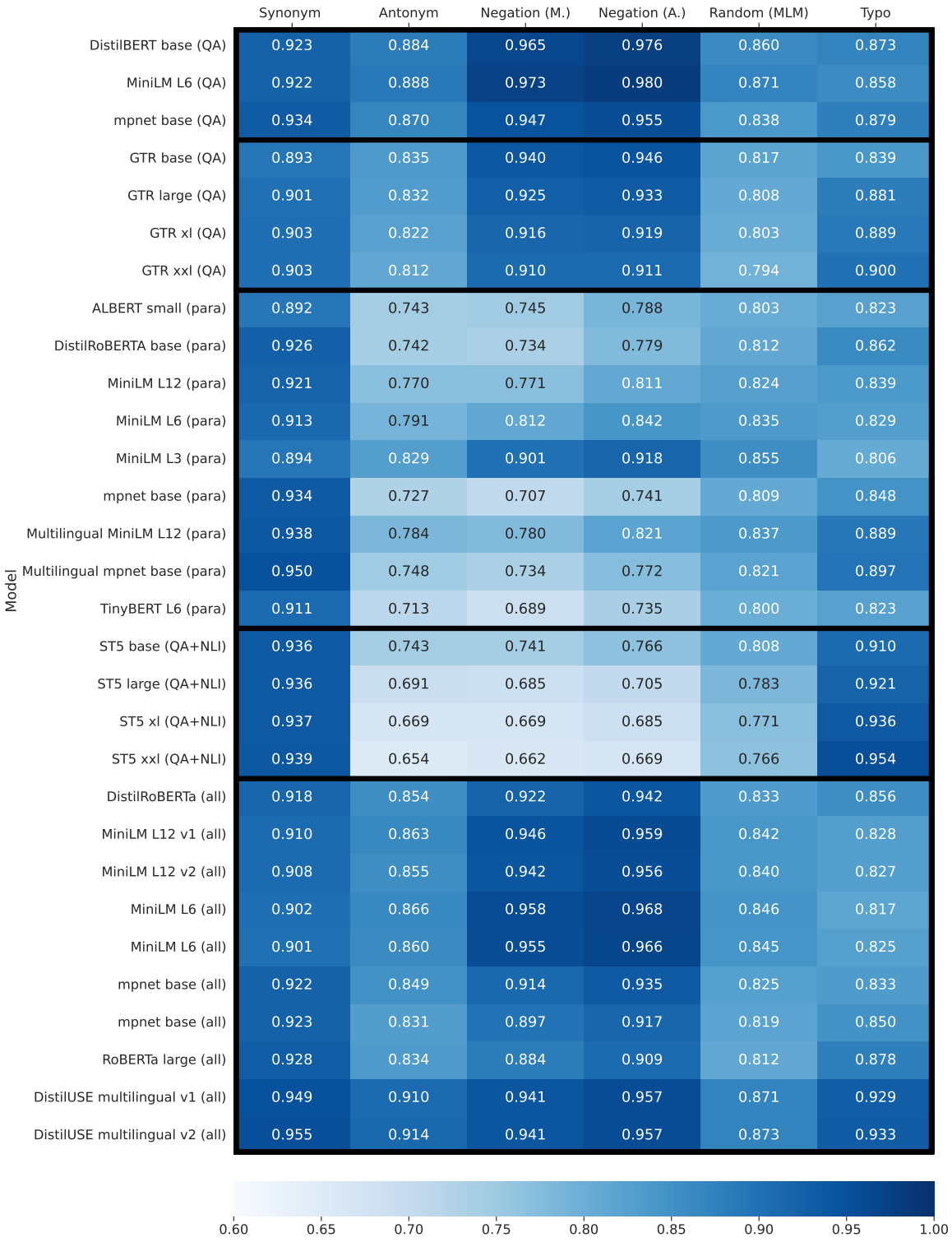

Figure 1: Normalized cosine similarity of supervised SEs. We group SEs that use different training datasets or training procedures together. We denote the datasets used to train the SEs in parentheses.

and have a total of 1.17B sentence pairs, including question-answer pairs in QA datasets, premise-hypothesis pairs in NLI dataset, and context-passage pairs in retrieval datasets. In the fifth block, the similarity between sentence pairs from the two Negation subsets is very high and is even higher than the similarity of the Synonym subset for most models. This means that when using these models for retrieval, given a source sentence, it is more likely to retrieve the negation of the source sentence, instead of another sentence that only differs from the source sentence by a synonym. While

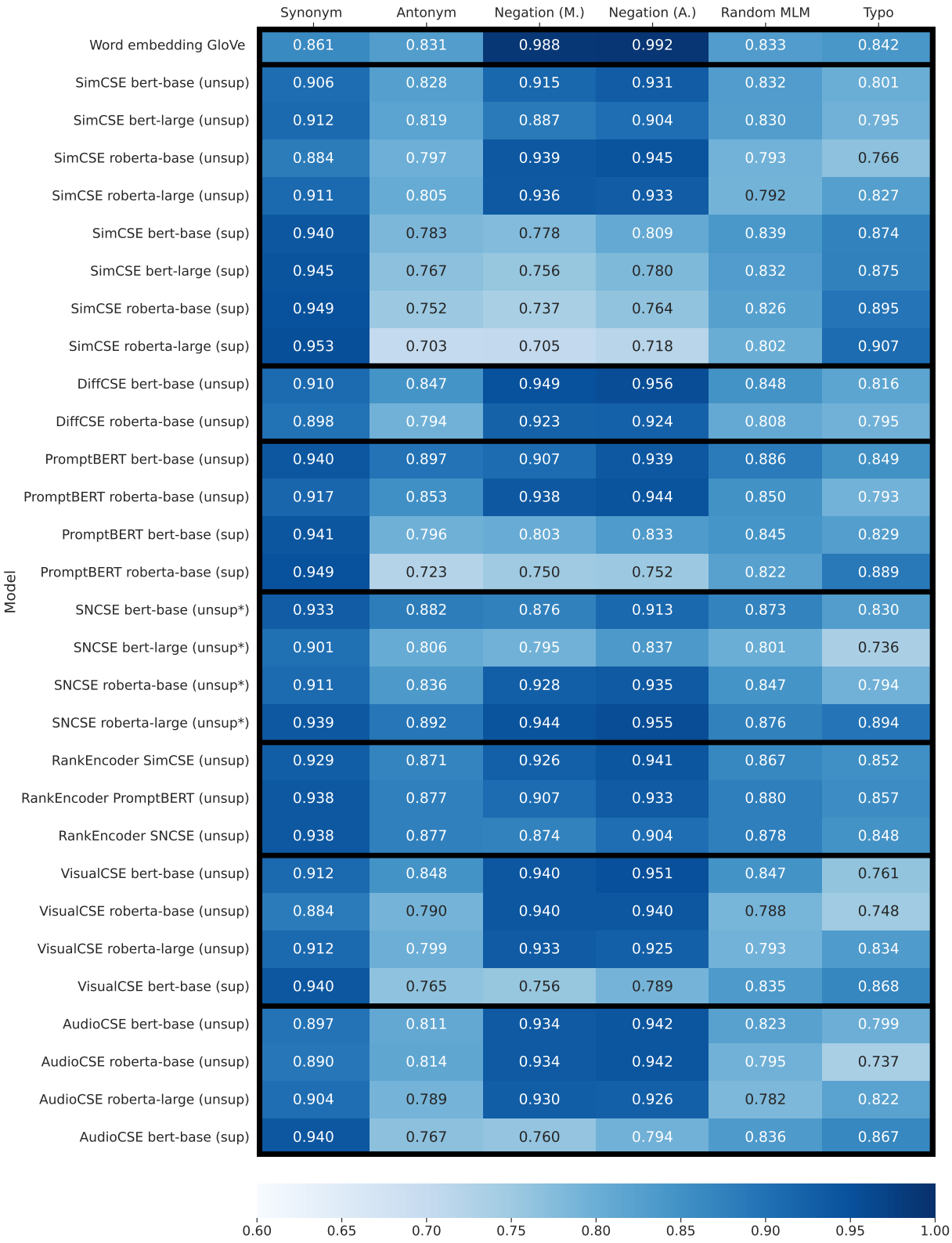

Figure 2: Normalized cosine similarity of unsupervised SEs and their derived supervised SEs. We group the SEs based on the unsupervised fine-tuning method.

these models are also fine-tuned on NLI datasets, the NLI datasets only compose 0.24% of the whole training data. This makes the models in this block much less sensitive to negations, compared with models fine-tuned mainly with NLI datasets (e.g., ST5) and models fine-tuned on paraphrase datasets.

HEROS reveal different characteristics of different SEs: Overall, we see that even if the sen-

tences in HEROS all have high lexical overlaps, the similarity score can still be very different among HEROS subsets for the same SE. HEROS also shows that how the concept of similarity is encoded by an SE is highly related to what the SE is trained on. This further allows us to understand what kind of similarity is required by the task related to the training dataset. For example, NLI tasks consider negation pairs as dissimilar while question-answer pair retrieval task considers negation to be similar. Such interesting observations are not revealed by any prior SE benchmarks, making HEROS very valuable. It will also be interesting to see if there is any correlation between an SE's performance on different subsets in HEROS and different downstream tasks in SentEval (Conneau and Kiela, 2018); we save this in future work.

3.2 Unsupervised SEs

Next, we turn our attention to unsupervised SEs. Unlike supervised SEs that are fine-tuned on labeled pairs of sentences, unsupervised SEs are trained using specially designed methods that do not use labeled sentence pairs. Most of these unsupervised SEs can be further fine-tuned on NLI datasets to further improve the performance on the STS benchmarks (Gao et al., 2021; Jiang et al., 2022; Jian et al., 2022). We show the result on HEROS of 7 different types of unsupervised SEs and their derived supervised SEs in Figure 2.

For the completeness of the result, we also report the performance of sentence embeddings calculated by averaging the GLoVe embeddings (Pennington et al., 2014) in the sentence. The result is presented in the first row in Figure 2. We observe that the sentence before and after negation have very high similarity, and the similarity is much higher than replacing one word with its synonym or antonym. This shows that negation words have a very small contribution to the sentence embedding obtained from averaging the GLoVe embeddings.

Unsupervised SEs are insensitive to negation: Unsupervised SEs, denoted with *unsup* in Figure 2, have high similarity on Negation subsets, sometimes even higher than Synonym subsets. SNCSE (Wang et al., 2022) models are an exception, where Negation subsets may have a lower similarity. SNCSE uses the dependency tree of a sentence to convert it into its negation as a "soft-negative" in contrastive learning, but it needs a dependency parser, making it not truly unsupervised. Hence, we use *unsup** to denote SNCSE models in Figure 2. The lower similarity on Negation datasets is not consistent for different SNCSE models, possibly due to a poor negation method in the implementation of SNCSE that does not consider negative contractions, resulting in low-quality augmented data.

Further supervised fine-tuning on NLI datasets significantly change the model's behavior on HEROS: Fine-tuning unsupervised SEs on NLI datasets (denoted with *sup* in Figure 2) leads to a significant drop in similarity on Negation and Antonym and an increase in similarity on the Synonym subset. This show that supervised fine-tuning greatly changes how SEs encode similarity. An interesting trend is that after fine-tuning, similarity on the Typo subset increases for most models, likely because the SE better captures semantic similarity and pays less attention to superficial lexical form.

Almost all SEs rate Negation (Main) to be less similar compared with Negation (Antonym) Recall that the Negation (Main) subsets are created by negating the **main** verb while the Negation (Antonym) subset does not always negate the main verb. The lower similarity on the Negation (Main) subset shows that SEs consider negating the main verb to be less similar to the original sentence, compared with negating other positions in the original sentence. This implies that the SEs can capture the level of the verb in the dependency tree of the sentence, and it considers negating the main verb to be more influential to sentence embeddings.

Close performance on STS benchmarks can have different behaviors on HEROS: We find that two SEs that achieve similar average performance on STS benchmarks (STS 12-17, STS-b, and SICK-R) can perform very differently on HEROS. For example, `RoBERTa large (all)` and `DistilRoBERTa base (para)` in Figure 1 have similar average STS scores (81.07 and 81.12, respectively), but the former have very high similarity on the Negation subsets while the latter does not. This is also the case for `SNCSE roberta-large` in Figure 2 and `mpnet base (para)` in Figure 1, which have average scores of 81.77 and 81.57 on the STS benchmarks, respectively. This shows that HEROS can reveal some traits of the SEs that the traditional STS benchmarks cannot identify.

4 Conclusion

We introduce HEROS, a new dataset of 6000 human-constructed sentence pairs with high lexical overlaps. It is composed of 6 subsets that capture different linguistic phenomena. Evaluating an SE on HEROS can reveal what kind of sentence pairs the SE considers similar. HEROS fills a void in current SE evaluation methods, which only use correlation coefficients with human ratings or performance on downstream tasks to summarize an SE, and mainly use sentence pairs with low lexical overlaps. We use HEROS to evaluate 60 models and reveal numerous new observations. We believe that HEROS can aid in interpreting SE behavior and comparing the performance of different SEs.

Limitations

The SEs in this paper are mainly transformer-based SEs, and we are not sure whether the observations hold for other SEs. However, considering that transformer-based SEs dominate the current NLP community, we think it is fine to only evaluate 59 transformer-based SEs. Another limitation is that the sentences in HEROS are converted from Reddit, which is an online forum and the texts on Reddit may be more casual and informal. This makes the sentence pairs in HEROS tend to be more informal. Users should note such a characteristic of the sentence pairs of HEROS and beware that the results obtained using HEROS may be different from the results obtained using more formal texts. An additional limitation is that there can be more diverse rules to create different sentence pairs other than the six subsets included in HEROS, and our paper cannot include them all. As a last limitation, during the construction of HEROS, we remove sentences that are ungrammatical based on language-tool, so our results may not generalize to ungrammatical sentences.

Ethics Statement

The main ethical concern in this paper is our dataset, HEROS. HEROS is constructed from an existing dataset, GoEmotion. As listed in the model card of GoEmotion, GoEmotion contains biases in Reddit and some offensive contents. As stated in Section 2.2.1, the authors have tried our best to remove all content that we find to be possibly offensive to users. We cannot guarantee that our standard of unbiased and unharmful fits everyone. Thus, we also remind future users of HEROS to be aware of such possible harms. To make sure the accessibility of our paper, we have used an online resource to carefully check that the figures in the paper are interpretable for readers of different backgrounds.

References

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.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Petra Barancikova and Ondřej Bojar. 2020. COSTRA 1.0: A dataset of complex sentence transformations. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3535–3541, Marseille, France. European Language Resources Association.

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

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

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.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level representation learning using

citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

William Coster and David Kauchak. 2011. Simple English Wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669, Portland, Oregon, USA. Association for Computational Linguistics.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2021. Ms marco: Benchmarking ranking models in the large-data regime. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1566–1576, New York, NY, USA. Association for Computing Machinery.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. In *58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

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

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong, China. Association for Computational Linguistics.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 1156–1165, New York, NY, USA. Association for Computing Machinery.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. 2018. WikiAtomicEdits: A multilingual corpus of Wikipedia edits for modeling language and discourse. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 305–315, Brussels, Belgium. Association for Computational Linguistics.

Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1481–1491, Seattle, Washington, USA. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018. End-to-end retrieval in continuous space. *arXiv preprint arXiv:1811.08008*.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Xuanli He, Lingjuan Lyu, Lichao Sun, and Qiongkai Xu. 2021. Model extraction and adversarial transferability, your bert is vulnerable! In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2006–2012.

Matthew Henderson, Paweł Budzianowski, Iñigo Casanueva, Sam Coope, Daniela Gerz, Girish Kumar, Nikola Mrkšić, Georgios Spithourakis, Pei-Hao Su, Ivan Vulić, and Tsung-Hsien Wen. 2019. A repository of conversational datasets. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 1–10, Florence, Italy. Association for Computational Linguistics.

Christopher Hidey and Kathy McKeown. 2016. Identifying causal relations using parallel Wikipedia articles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1424–1433, Berlin, Germany. Association for Computational Linguistics.

Yiren Jian, Chongyang Gao, and Soroush Vosoughi. 2022. Non-linguistic supervision for contrastive learning of sentence embeddings. In *Advances in Neural Information Processing Systems*.

Ting Jiang, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Liangjie Zhang, and Qi Zhang. 2022. Promptbert: Improving bert sentence embeddings with prompts. *arXiv preprint arXiv:2201.04337*.

Daniel Khashabi, Amos Ng, Tushar Khot, Ashish Sabharwal, Hannaneh Hajishirzi, and Chris Callison-Burch. 2021. Gooaq: Open question answering with diverse answer types. *arXiv preprint*.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Wuwei Lan and Wei Xu. 2018. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3890–3902.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020a. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020b. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

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 *ECCV (5)*.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

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

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yeon Seonwoo, Guoyin Wang, Sajal Choudhary, Changmin Seo, Jiwei Li, Xiang Li, Puyang Xu, Sunghyun Park, and Alice Oh. 2022. Ranking-enhanced unsupervised sentence representation learning. *arXiv preprint arXiv:2209.04333*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Hao Wang, Yangguang Li, Zhen Huang, Yong Dou, Lingpeng Kong, and Jing Shao. 2022. Sncse: Contrastive learning for unsupervised sentence embedding with soft negative samples. *arXiv preprint arXiv:2201.05979*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.

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

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.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Xunjie Zhu and Gerard de Melo. 2020. Sentence analogies: Linguistic regularities in sentence embeddings. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3389–3400, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xunjie Zhu, Tingfeng Li, and Gerard De Melo. 2018. Exploring semantic properties of sentence embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 632–637.

A Further Details of HEROS

A.1 Dataset Preprocess

The sentences in GoEmotions are already anonymized, where the names of people are replaced with a special `[NAME]` token, so we do not need to further perform anonymization. We filter out all sentences that have more than one `[NAME]` token and replace all `[NAME]` with a gender-neutral name "Jackie".

A.2 Dataset License

HEROS is constructed based on the GoEmotion dataset (Demszky et al., 2020). GoEmotion is released under the Apache 2.0 license, so our modification and redistribution to GoEmotion are granted by the dataset license. **Our dataset, HEROS is also released under the Apache 2.0 license.**

B Comparing the Lexical Overlaps of Different Datasets

In Table 1, we show the basic statistics of three different datasets. We use the ROUGE F1 and Levenshtein distance to quantify the lexical overlap between sentence pairs of a dataset. The statistics of HEROS is averaged over different subsets, and those of STS-b and SICK-R are calculated based on the test set.

R1, R2, and RL: ROUGE F1 score between the sentence pairs. (R1 and R2: unigram and bigram overlap; RL: longest common subsequence.) We use the implementation of python rouge 1.0.1 to calculate the ROUGE score.

Lev is the average normalized **token-level** Levenshtein distance among the sentence pairs, and the normalized Levenshtein distance is the Levenshtein distance between two sentences divided by the length of the longer sequence of the sentence pairs. We first tokenize the sentence using the tokenizer of `bert-base-uncased` and calculate the Levenshtein distance between the token ids of the sentence pairs. We normalize the Levenshtein distance to make it falls in the range of $[0, 1]$.

The average sentence length is the average number of tokens per sentence, and the tokens are obtained by using the tokenizer of `bert-base-uncased`.

C Supplementary Materials for Sentence Encoders

C.1 Supervised Sentence Encoders

Table 3 shows the number of parameters and the sentence embedding dimension of the SEs used in this paper.

C.1.1 Datasets Used to Train Supervised SEs

The datasets indicated in Figure 1 is listed as follows:

all Reddit comments (2015-2018) (Henderson et al., 2019), S2ORC Citation pairs (Abstracts) (Lo et al., 2020), WikiAnswers Duplicate question pairs (Fader et al., 2014), PAQ (Question, Answer) pairs (Lewis et al., 2021), S2ORC Citation pairs (Titles) (Lo et al., 2020), S2ORC (Title, Abstract) (Lo et al., 2020), Stack Exchange (Title, Body) pairs, MS MARCO triplets (Craswell et al., 2021), GOOAQ: Open Question Answering with Diverse Answer Types (Khashabi et al., 2021), Yahoo Answers (Title, Answer) (Zhang et al., 2015), Code Search, COCO Image captions (Lin et al., 2014), SPECTER citation triplets (Cohan et al., 2020), Yahoo Answers (Question, Answer) (Zhang et al., 2015), Yahoo Answers (Title, Question) (Zhang et al., 2015), SearchQA (Dunn et al., 2017), Eli5 (Fan et al., 2019), Flickr 30k (Young et al., 2014), Stack Exchange Duplicate questions (titles), SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), Stack Exchange Duplicate questions (bodies), Stack Exchange Duplicate questions (titles+bodies), Sentence Compression (Filippova and Altun, 2013), Wikihow (Koupaee and Wang, 2018), Altlex (Hidey and McKeown, 2016), Quora Question Triplets (Wang et al., 2019), Simple Wikipedia (Coster and Kauchak, 2011), Natural Questions (NQ) (Kwiatkowski et al., 2019), SQuAD2.0 (Rajpurkar et al., 2016), and TriviaQA.

QA All the QA datasets in ***all***.

paraphrase SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), Simple Wikipedia (Coster and Kauchak, 2011), Altlex (Hidey and McKeown, 2016), MS MARCO triplets (Craswell et al., 2021), Quora Question Triplets (Wang et al., 2019), COCO Image captions (Lin et al., 2014), Flickr 30k (Young et al., 2014), Yahoo Answers (Title, Question) (Zhang et al., 2015), Stack Exchange Duplicate questions (titles+bodies) and WikiAtomicEdits (Faruqui et al., 2018).

GTR fine-tuning data: QA+MRC Natural Questions (NQ) (Kwiatkowski et al., 2019), MS MARCO triplets (Craswell et al., 2021), input-response pairs and question-answer pairs from online forums and QA websites including Reddit, Stack-Overflow, etc.[6]

ST5 fine-tuning data: QA+NLI SNLI (Bowman et al., 2015) and question-answer pairs from community QA websites.

[6]Ni et al. (2021) does not specify the exact online forums.

C.2 Unsupervised Sentence Encoders

The full list of unsupervised SEs and their supervised derivations we compared are: SimCSE (Gao et al., 2021), DiffCSE (Chuang et al., 2022), PromptBERT (Jiang et al., 2022), SNCSE (Wang et al., 2022), RankEncoder (Seonwoo et al., 2022), AudioCSE and VisualCSE (Jian et al., 2022). For all the unsupervised SEs shown in Figure 2, if it is a base-size model, its number of parameters is roughly 110M; if it is a large-size model, its number of parameters is roughly 335M. The `bert` models shown in Figure 2 are all `uncased` models.

D Normalization

For each SE, we first calculate the cosine similarity between each minimal pair in HEROS. However, if the embedding space is highly anisotropic (Ethayarajh, 2019; Li et al., 2020a), the cosine similarity between two random sentences is expected to be rather high. To remove the effect of anisotropic embedding space and better interpret the result, we normalize the cosine similarity by a baseline cosine similarity. The baseline cosine similarity is calculated by the following procedure: We split the 1000 original sentences into the first 500 and the last 500 sentences, and calculate the average cosine similarity between the sentence embeddings of these 500×500 random sentence pairs. This average cosine similarity, $\cos_{avg}$, gives us an idea of how similar sentence embedding can be for two randomly selected sentences. Last, we normalize the cosine similarity of the minimal pairs to lessen the effect of anisotropy by the following formula:

$$\cos_{normalized} = \frac{\cos_{orig} - \cos_{avg}}{1 - \cos_{avg}}, \qquad (1)$$

where $\cos_{orig}$ is the original cosine similarity of a sentence pair and $\cos_{normalized}$ is the similarity after normalization.

E Runtime and Computation Resource

The experiments on Section 3, except T5 xl and xxl, are conducted on an NVIDIA 1080 Ti, and it takes less than one hour to run all the experiments. The T5 xxl and xl models cannot be loaded on a 1080 Ti, and we use V100 to conduct the experiment of the SEs whose base models are T5 xl and xxl, which takes less than 15 minutes.

Model and Link	#Param	d_{emb}
Word embedding GloVe	120M	300
DistilBERT base (multi-QA)	66M	768
MiniLM L6 (multi-QA)	22M	384
mpnet base (multiQA)	110M	768
ALBERT small (paraphrase)	11M	768
DistilRoBERTA base v2 (paraphrase)	82M	768
MiniLM L12 v2 (paraphrase)	33M	384
MiniLM L3 v2 (paraphrase)	17M	384
MiniLM L6 v2 (paraphrase)	22M	384
mpnet base v2 (paraphrase)	110M	768
Multilingual MiniLM L12 v2 (paraphrase)	33M	384
Multilingual mpnet base v2 (paraphrase)	110M	768
TinyBERT L6 v2 (paraphrase)	14.5M	768
GTR base	110M	768
GTR large	335M	768
GTR xl	1,24B	768
GTR xxl	4.8B	768
Sentence-T5 base	110M	768
Sentence-T5 large	335M	768
Sentence-t5 xl	1,24B	768
Sentence-T5 xxl	4.8B	768
DistilRoBERTa v1 (all)	82M	768
MiniLM L12 v1 (all)	33M	384
MiniLM L12 v2 (all)	33M	384
MiniLM L6 v1 (all)	22M	384
MiniLM L6 v2 (all)	22M	384
mpnet base v1 (all)	110M	768
mpnet base v2 (all)	110M	768
RoBERTa large v1 (all)	355M	1024
DistilUSE base multilingual v1 (all)	134M	512
DistilUSE base multilingual v2 (all)	134M	512

Table 3: The model sizes and embedding dimensions of the supervised SEs shown in Figure 1. The model names are clickable links. # is the number of parameters of the SE, and d_{emb} is the dimension of the sentence embedding.