Quantifying Train-Evaluation Overlap with Nearest Neighbors

Gauri Kambhatla Thuy Nguyen Eunsol Choi
The University of Texas at Austin
{gkambhat, eunsol}@utexas.edu
n.haithuy1999@gmail.com

Abstract

Characterizing benchmark datasets is crucial to interpreting model performance. In this work, we study train-evaluation overlap as a measure of an individual dataset's adequacy to evaluate model generalization over a wide range of datasets. We quantify the overlap with a simple novel metric based on a nearest neighbors approach between the training and evaluation sets. We identify nearest training examples for each evaluation example by mapping instances with generic and task-specific embedding methods. Our study on eleven classification and extractive QA tasks reveals a wide range of train-evaluation overlap, and we show that the data collection method of the dataset and the difficulty of the task may play a role in the amount of overlap. Lastly, we use our nearest neighbor analysis to identify challenging or potentially mislabeled examples. Our analysis quantifies train-evaluation overlap, providing insights for constructing datasets to study generalization.

1 Introduction

Benchmark datasets in NLP (Rajpurkar et al. 2016; Wang et al. 2018) are invaluable for driving and tracking progress in the field. While evaluating on a held-out set of data ideally tests for generalizability to new data, frequent overlap between training and evaluation sets hinders assessing a model's generalization capacity (Elangovan et al., 2021; Lewis et al., 2021a; Krishna et al., 2021). In this paper, we quantify the overlap between the training and evaluation splits in datasets through a simple metric based on a nearest neighbor approach, and analyze datasets along the axis of dataset collection method.

We categorize data collection methods frequently used in the literature into four categories, based on how naturally the language is captured; some datasets harvest user generated content (e.g., movie reviews paired with their scores), while language in other datasets is written by crowdworkers to fool existing models (Nie et al., 2020) or synthetically generated from templates (Warstadt et al., 2020).

We analyze the train-evaluation overlap in eleven NLP datasets varying in data collection method on two tasks – classification and extractive question answering – through a nearest neighbors approach. To quantify the overlap between training and evaluation datasets, we identify the nearest train neighbor to each evaluation example using cosine similarity between the input representations. We experiment with two types of representations – general sentence embeddings (Gao et al., 2021) and task-specific embeddings (after task-specific training (Devlin et al., 2019)). Then, we copy the label of the nearest training example to each evaluation example, constructing a simple nearest neighbor baseline model. In nearly every setting, we show that copying labels from the nearest train example alone achieve a competitive baseline, indicating overlap in content between the training and evaluation sets without any task specific training. We find that naturally-collected datasets exhibit stronger training and evaluation set overlap compared to more synthetic and adversarially-generated data.

We introduce a new metric, named `InsSim`, which summarizes the distance from each evaluation example to its nearest training examples, indicating the train-evaluation overlap. We use the nearest neighbor classifier and `InsSim` score to estimate the difficulty of *individual* evaluation examples, and suggest splitting evaluation datasets into challenging and easier subsets. Our analysis motivates careful benchmark designs (Koh et al., 2021a) that aims to capture both natural language usage and distributional shifts.

2 Related Work

Representing a sequence of tokens as a single, fixed dimensional vector (Reimers and Gurevych, 2019; Arora et al., 2017; Kiros et al., 2015) has been studied extensively. Such an encoder can act as a dense

Findings of the Association for Computational Linguistics: ACL 2023, pages 2905–2920

passage retriever (Karpukhin et al., 2020), paired with an efficient similarity search method (Qin et al., 2020).

Two prior studies in question answering (Lewis et al., 2021a; Krishna et al., 2021) look in-depth into the overlap between the training and evaluation sets. They identify the most similar training example either by answer string match or comparing the question embedding constructed for passage retrieval. The follow up work further develops the QA model (Lewis et al., 2021b) for copying the answer from the nearest training example, after augmenting training examples with generated question answer pairs. Our study in Section 4.3 extends this setting for a wide range of tasks and different embedding methods. Similar to our work, Elangovan et al. (2021) examine train-test overlap for text classification tasks. They also compute the similarity for each test instance to the entire training set using a similarity function. However, they utilize a bag-of-words approach to represent text (where we use sentence embeddings). In addition, we provide analysis for a broad range of datasets.

Many works have explored whether models simply memorize the training dataset or actually learn the task, thus generalizing to unseen examples. Our nearest-neighbor match classification method resembles ProtoBERT (Tänzer et al., 2022), which shows promising performance in rare classes. The model classifies examples by comparing distance to the centroid of training examples belonging to each class. Our method is simpler, without estimating a probability distribution over the output classes. Tirumala et al. (2022) also study the effect of dataset size and model size on memorization, but look at the dynamics of memorization in language models *during* training, finding that larger language models tend to memorize data faster, and that certain parts of speech are memorized faster than others.

Other work studies different subsets of datasets and how this can change evaluation. Ethayarajh et al. (2022) study dataset difficulty in terms of the lack of usable information to a particular model V, as well as difficulty of data subsets using a measure of pointwise V-information for individual data instances. As in our work, Swayamdipta et al. (2020) study difficulty of individual instances, although they focus on the training rather than evaluation set. Similarly, Godbole and Jia (2022) propose a method for better evaluation of generalization on more difficult examples (those assigned lower likelihood by a pretrained LM), focused on creating the train-eval split. In our work, we introduce a very simple and generalizable method of splitting examples by whether classification with the nearest training example can succeed.

Recent work (Sakaguchi et al., 2021; Malinin et al., 2021, 2022; Koh et al., 2021b) focuses on modeling distributional shifts in carefully constructed real world datasets, such as simulating shifts by having training set from one region and the test set from another region. This can be one path to mitigate frequent train-evaluation overlap in naturally occurring datasets.

3 Categorizing Dataset Collection Method

NLP datasets are collected through diverse methods for multiple purposes – some datasets mirror the user-facing applications closely (e.g., question answering datasets and machine translation datasets), while other datasets are carefully designed for diagnostic purposes. With the rise of harder to interpret, high capacity models (Brown et al., 2020; Chowdhery et al., 2022), many datasets are designed to probe model qualities. Would different data collection method yield different level of train evaluation overlap? To investigate this, we first categorize the data collection method of datasets below. We propose a discrete scale of naturalness, from purely synthetic to user-generated, as follows:

- **Synthetic (SYN)**: template-generated or highly-constrained crowd-sourced text. Here, both inputs and outputs are synthetically generated.
- **Crowd-sourced (CWD)**: input text and output labels are both generated by crowdworkers.
- **Artificial labels (LAB)**: input text are collected from real world user interactions, but output labels are annotated by crowdworkers.
- **User-generated (USE)**: input text is collected from user interactions and labels also arise naturally from users.

We note that our definition of synthetic data includes highly-constrained crowd-sourced text, by which we mean that the annotators have limited freedom in the content of their annotations. For example, for the WinoGrande dataset workers are instructed to choose an anchor word to use in the twin sentences, they are given a range for sentence length, and they are asked to maintain 70% overlap

between sentences. This is less natural than what the human might have generated on their own.

We provide examples of the datasets of each type we study here, approximately ordered from the least to most natural datasets.

WinoGrande A crowd-sourced, commonsense reasoning benchmark inspired by the Winograd Schema Challenge, in which twin sentences with a small edit distance each have a missing word and two possible options (Sakaguchi et al., 2021).

CSQA 2.0 (Commonsense Question Answering 2.0) A corpus of crowdsourced yes/no commonsense reasoning questions (e.g., "a playing card is capable of cutting soft cheese?") (Talmor et al., 2021).

ANLI (Adversarial NLI) A natural language inference corpus with data collected "adversarially" in three rounds using a human-in-the-loop approach (Nie et al., 2020).

MNLI (Multi-Genre Natural Language Inference) A corpus of sentence pairs (crowdsourced) with annotations for textual entailment (given a premise and hypothesis, does the first entail, contradict, or is neutral to the other). We conduct experiments using both the matched (in-domain) and mismatched (cross-domain) evaluation sets (Williams et al., 2018).

SQuAD 2.0 (Stanford Question Answering Dataset 2.0) A corpus of crowdsourced questions (along with a Wikipedia context), and annotated answer spans. Unlike SQuAD 1.1, not all questions have answers (Rajpurkar et al., 2018).

MRPC (Microsoft Research Paraphrase Corpus) A corpus of sentence pairs extracted from online news sources, where each pair is annotated for whether the sentences are semantically equivalent (Dolan and Brockett, 2005). The sentences was paired based on heuristics (e.g., "two sentences share at least three common words").

NQ (Natural Questions) A corpus of questions from popular Google search queries, paired with a retrieved Wikipedia document, annotated with an answer. We use simplified MRQA version, which removes unanswerable questions, yes/no questions or questions without a short answer span and considers paragraph containing a short answer span as context instead of the entire document (Kwiatkowski et al., 2019; Fisch et al., 2019).

TweetEval A corpus of tweets containing multiple classification tasks (Barbieri et al., 2020), though we used the subset of the dataset specifically for sentiment analysis. We also pre-process the data to remove examples with the neutral label, making the classification task binary (positive/negative) for out-domain evaluation with SST-2.

SST-2 (Stanford Sentiment Treebank) A corpus of movie review sentences with annotations for sentiment (positive/negative) (Socher et al., 2013).

AG News A corpus of news articles from the web, categorized into four topics (business, sci/tech, sports, world) (Zhang et al., 2015).

IMDb (IMDb Review Dataset) A balanced corpus of movie reviews from IMDb with negative (score ≤ 4 out of 10) and positive reviews (score ≥ 7 out of 10) (Maas et al., 2011).

4 Nearest Neighbor Analysis with Two Types of Encoders

We begin studying overlap with an analysis of nearest neighbor data instances between the train and evaluation datasets. We define the nearest neighbor for each evaluation example x_e in the given training dataset X_{train}. This is dependent on the embedding function $E(x)$, and the training dataset X_{train}. Following prior work (Snell et al., 2017; Tänzer et al., 2022), we define the similarity between two examples x_i and x_j as the cosine similarity between their embeddings, $E(x_i)$ and $E(x_j)$. We describe how to encode each example below.

4.1 Instance Encoder

We consider two types of encoder $E(x)$ for each data instance x – a general sentence embedding function and an embedding function learned while optimizing for the target task. We study two tasks, classification and extractive question answering (Rajpurkar et al.). Classification tasks map input text x to y from pre-defined label set Y, and question answering tasks map an input x consisting of {question q, evidence passage c} to answer string y which is a span in the evidence passage. As the output should be entailed from the input, we only pass in input to the instance encoder. We note that such a nearest neighbors approach to studying overlap of the input could be extended to generation tasks such as translation or summarization, or semantic parsing, although we do not examine these in this work.

General Sentence Embedding [E_g] We experiment with two types of general sentence embeddings; (1) [CLS] token embeddings from the pre-trained LM before fine-tuning (Liu et al., 2019a)

Dataset	Eval example	Nearest training example (E_g)	Nearest training example (E_t)	Overlap E_g	Overlap E_t
WinoGrande	Megan forgot to buy deodorant at the store so they borrowed Jessica's deodorant and __ hoped they never found out.	Elena asked Erin if she could borrow her deodorant, but __ had forgotten to bring some.	Natalie was having an ant problem and hates bugs so called Elena for help since __ is fearless.	0.330	0.168
MNLI	**Premise:** Most of the dances are suggestive of ancient courtship rituals, with the man being forceful and arrogant, the woman shyly flirtatious. **Hypothesis:** The dances have an equal number of male and female dancers.	**Premise:** In Kerala, try to see the lively kathakali dances, in which men play both male and female parts to enact both divine and heroic Indian legends in the most gorgeous costumes and elaborate makeup. **Hypothesis:** The lively kathakali dances in Kerala feature men who play the role of males and females.	**Premise:** Here there are several attractive hotels, including one with tropical gardens, that cater to visitors hoping to catch a glimpse of the Himalayas at sunrise or sunset. **Hypothesis:** All of the hotels here have an indoor heated pool to offer as well.	0.343	0.119
NQ (MRQA)	who heads the executive department of west virginia government	who's the head of the executive branch of the government	who began the reformed movement (a branch of the protestant reformation) in zurich switzerland	0.630	0.367
AG News	Allianz to fight US court ruling on WTC attacks MUNICH - German insurance concern Allianz said on Tuesday it would fight a US jury decision in New York...	Allianz Says Trade Center Ruling May Cost It Up to 80 Mln Euros Allianz AG, Europe's largest insurer, said a New York court ruling that defined...	Developer Wins Victory in WTC Case NEW YORK (Reuters) - A New York developer hoping to rebuild the destroyed World Trade Center...	0.415	0.372

Table 1: Examples of the most similar instances for the evaluation example according to two embedding methods. Unigram overlap of each train instance with the evaluation example is highlighted in blue. Average unigram overlap over the full dataset between evaluation examples and nearest train examples according to the different embedding methods are shown in the last two columns.

and (2) SimCSE embeddings (Gao et al., 2021) which showed strong performance over various benchmark datasets. Gao et al. (2021) first encode input sentence with a pretrained language model and then take the [CLS] representation to get a fixed dimensional representation and improve it with a contrastive learning objective (Chen et al., 2020). Specifically, they construct positive sentence pairs by applying two different standard dropout masks (Gal and Ghahramani, 2016) on the input representation on the same sentence, and construct negative pairs by taking other sentences in the same mini-batch. While we choose these two embeddings for our analysis, other sentence embedding methods (Kiros et al., 2015; Wu et al., 2020) can be used.

Task Specific Learned Embedding [E_t] To construct task specific embedding, we first fine-tune a pre-trained language model to perform our target tasks. Unless otherwise specified, we use the RoBERTa-large model (Liu et al., 2019b). We use standard recipes for using pre-trained LMs. For classification, we take the [CLS] representation through a fully-connected layer to predict the correct label from a label set (classification task). For extractive QA, we encode concatenation of question and context tokens and take the final representations of the context tokens through fully-connected layer to predict the answer start and answer end token.

4.2 Nearest Neighbor Analysis

We first provide some manual inspection of similar examples. Table 1 presents a few examples from the evaluation set from various datasets, along with their most similar training examples for each embedding function. We observe that the two embedding functions capture different views, and that the general embedding (E_g) captures more lexical similarity. This reiterates prior work showing that task-specific embeddings (such as averaging token representations or using the CLS token) performs poorly on semantic similarity tasks (Reimers and Gurevych, 2019). We report the average unigram overlap between evaluation examples and their nearest train neighbor with both general and task-specific representations in Table 1. We provide examples for additional datasets in Appendix D (see Table 11 for qualitative examples) and quantitative unigram overlap in Appendix A (Table 7).

Dataset	Random	Finetune (FULL)	E_g SimCSE (500)	E_g SimCSE (FULL)	E_g CLS (FULL)	E_t CLS (500)	E_t CLS (FULL)
WinoGrande	49.57	78.37	50.67 (+0.24)	51.78 (+1.82)	49.88 (+0.31)	50.43 (+1.42)	49.80 (+2.37)
ANLI	33.94	57.37	33.38 (+0.57)	35.34 (+1.78)	39.03 (+5.12)	36.03 (+5.34)	56.28 (+45.22)
MNLI	33.15	89.94	37.96 (+9.91)	45.93 (+19.19)	37.23 (+4.08)	73.95 (+66.21)	89.12 (+86.42)
MRPC	56.45	92.12	58.18 (+0.14)	62.68 (+6.23)	63.06 (-4.03)	79.79 (+61.65)	88.42 (+75.43)
TweetEval	45.45	94.60	76.54 (+54.64)	81.11 (+60.30)	72.59 (+34.21)	88.94 (+78.87)	93.72 (+85.94)
SST-2	48.85	96.84	74.31 (+47.93)	78.90 (+53.79)	65.14 (+16.06)	92.09 (+83.83)	95.07 (+90.94)
AG News	25.68	95.47	79.67 (+72.66)	89.83 (+81.32)	84.20 (+59.25)	90.63 (+90.30)	93.75 (+93.55)
IMDb	50.16	95.17	70.94 (+26.48)	72.75 (+32.70)	64.06 (+14.06)	93.34 (+85.94)	94.84 (+89.22)

Table 2: Nearest neighbor classification results by copying the gold label from the nearest training example with different embedding methods. 500 and FULL represent the size of the training data set. The number in parenthesis represents the gap from copying the label from the farthest training example. Random performance and the RoBERTa-large fine-tuned performance are shown for lower and upper bound comparisons.

In every dataset, there is more lexical overlap when nearest neighbor was found using general representations, supporting our qualitative observations.

4.3 Classification with the Nearest Neighbor

After identifying the nearest training example for each evaluation example, what can we do with it? Inspired by a recent study in question answering (Lewis et al., 2021a) which copies the answer of the training question that is most similar to the evaluation question (where the evaluation question is a duplicate or paraphrase of the train question), we apply this method widely to all datasets we study to build a non-parametric classification model. This is similar to the protoBERT model (Tänzer et al., 2022) which uses k-nearest neighbor classification algorithms. However, we use the label from the nearest neighbor without constructing an embedding representing each class label. For extractive QA tasks, we use the answer as the label and calculate performance as the exact-match to the nearest neighbor. High performance of this baseline will indicate greater train-evaluation overlap.

Table 2 presents the results for the two embedding types we study, as well as two training data sizes. Here, we look at gold labels, and focus on differences between embedding types and training data sizes. We also report the difference to the classification performance for taking the *farthest* training example in parentheses and a random baseline which assigns labels according to the label distribution. We also show the total RoBERTa-large fine-tuned performance as an upperbound. Fine-tuned performance for all datasets and other models are shown in Appendix B.

How does nearest neighbor classification work with different encoders? Comparing general CLS token embeddings (without fine-tuning) with SimCSE embeddings, we see mixed results – sometimes using SimCSE results in higher performance, sometimes general CLS token embeddings. However, the difference between performance on the nearest neighbor and performance on the farthest neighbor using CLS embeddings without fine-tuning is generally lower than when we use SimCSE embeddings, indicating the nearest semantic neighbor might be more relevant with SimCSE embeddings over CLS tokens, which follows prior work (Reimers and Gurevych, 2019).

After fine-tuning, copying the label of nearest neighbor shows strong performance across all datasets except WinoGrande. We attribute the strong performance to the task-specific nature of CLS embeddings (Reimers and Gurevych, 2019); while they have low semantic similarity, they are close together in terms of *task* similarity (e.g., examples that require the model to do the same type of reasoning are more similar) leading to a high nearest neighbor performance.

How does nearest neighbor classification interact with data collection methods? The nearest neighbor performance roughly corresponds with the degree of naturalness; for all user-generated classification tasks (LAB and USE), copying the label of nearest neighbor shows competitive performance, even without task-specific fine-tuning. On challenging, synthetically and adversarially generated datasets (WinoGrande and ANLI), however, the nearest neighbor approach shows smaller gains. We hypothesize that this is because researchers can control data diversity and task difficulty in the synthetic setting to make a benchmark more challenging, which cannot be done in the natural case. In addition, higher performance with natural data might signify more match with the pre-training data

Figure 1: Nearest neighbor classification performance (%) between the prediction on the evaluation example and their nearest train example for trained models on selected datasets. The x-axis shows the model (DistilBERT, RoBERTA-base, or RoBERTa-large) and the y-axis shows the size of the training data, from 500 examples to the full dataset.

of the model. We also note that the correspondence between performance and data collection method could also be due to task difficulty and types, as the user-generated datasets tend to be easier for models to learn. Label match to the nearest neighbor is nearly always higher than to the farthest neighbor and performs better than the random baseline, showing that a simple nearest neighbor approach corresponds to the overlap between train and evaluation sets.

How does nearest neighbor classification vary with encoder model power and training data size? Figure 1 shows the nearest neighbor classification performance for label predictions of different power models of varying training data sizes for selected user-generated and synthetic/crowdsourced datasets. Here we study predicted labels rather than gold labels, and use RoBERTa-large, RoBERTA-base (Liu et al., 2019b) and DistilBERT (Sanh et al., 2020). As fine-tuned CLS embeddings achieve high performance due to task-specific or reasoning similarity, we use SimCSE representations for more general semantic similarity between nearest neighbors. Across all datasets, the nearest neighbor classification appears to be relatively consistent regardless of the size of the encoder model. For more natural datasets (bottom row of Figure 1), we see a large increase in performance when the training data size increases from 10k to the full dataset; this is less consistent for synthetic and crowdsourced datasets (top row of Figure 1). This could indicate that for more natural datasets, or easier tasks, a larger amount of data leads to a higher comparative overlap, but this is not necessarily the case with synthetic and crowdsourced data.

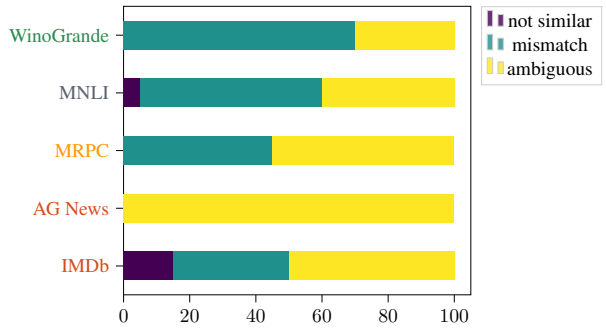

Figure 2: Distribution of label-mismatch examples (%) for selected datasets, from more synthetic (top) to more natural (bottom).

What can we learn from examples where nearest neighbor classification fails? We seek to understand cases in which the evaluation label does not match the nearest train label for classification tasks. We randomly sample 100 examples (20 from each of WinoGrande, MNLI, MRPC, AGNews and IMDb datasets) where nearest neighbor classifica-

	Setting	Random	E_g (SimCSE)	E_g (CLS)	E_t
MNLI	in-domain	33.15	45.93 (+19.19)	37.23 (+4.08)	89.12 (+86.42)
	out-domain (ANLI)	33.39	37.84 (+9.20)	34.37 (+2.64)	84.81 (+81.70)
	out-domain (MNLI-mm)	33.06	32.57 (+0.16)	36.36 (+3.59)	88.90 (+85.97)
SST-2	in-domain	48.85	78.90 (+53.79)	65.14 (+16.06)	95.07 (+90.94)
	out-domain (IMDb)	50.02	69.88 (+37.66)	51.13 (+1.18)	88.69 (+76.75)
	out-domain (TweetEval)	27.59	44.98 (+22.70)	49.06 (-6.90)	87.55 (+78.78)

Table 3: Nearest neighbor classification results under domain shift. The E_g (CLS) embeddings are the token embeddings from the pre-trained LM without fine-tuning; the E_t embeddings are after fine-tuning on the full training data set. The number in parenthesis represent the gap from copying the label from the farthest training example. The in-domain performance values are presented for comparison.

tion fails, and manually categorize them into three types:

- *not similar*: Failure at general semantic similarity
- *mismatch*: Semantic / task similarity mismatch
- *ambiguous*: The label for either the evaluation or train example is ambiguous (or incorrect)

We note that the first two categories, *not similar* and *mismatch* are failures due to the nearest neighbors approach, while the last category, *ambiguous*, is relevant to the dataset itself. Table 12 in Appendix E provides examples. We show the percentage of annotated examples in each category for each dataset, in Figure 2. The majority of manually annotated examples were ambiguous, which is a possible reason for why the model performs worse on instances without label match.

How does nearest neighbor classification perform under domain shift? We perform analysis on distribution shifts on two classification tasks – sentiment classification and natural language inference. We report the classification results from copying the nearest neighbor in the training set (parallel to Section 4.3) in Table 3. We find that the most similar example in the train set is less likely to have the same label as the evaluation example when the evaluation example is taken from different distribution. Yet, the nearest neighbor classification almost always outperforms the baseline, sometimes strongly.

5 Quantifying Overlap with Instance Similarity

In this section, we introduce a new metric, Instance Similarity (`InsSim`), and use it to identify easy and challenging instances in the evaluation dataset.

	1k	full
WinoGrande	0.458 / 0.900	0.594 / 0.878
CSQA 2.0	0.399 / 0.900	0.520 / 0.900
ANLI	0.505 / 0.912	0.658 / 0.962
MNLI	0.384 / 0.900	0.622 / 0.900
SQuAD 2.0	0.466 / 0.899	0.636 / 0.900
MRPC	0.525 / 0.841	0.579 / 0.881
TweetEval	0.469 / 0.903	0.561 / 0.939
NQ	0.481 / **0.927**	0.717 / **0.981**
SST-2	0.489 / 0.835	0.608 / 0.900
AG News	0.546 / 0.864	**0.751** / 0.906
IMDb	**0.648** / 0.894	0.709 / 0.959

Table 4: Average `InsSim` score of evaluation subset on each dataset. The first column is computed against a randomly sampled 1K training examples, the second column against the full training portion of each dataset. The first number in each cell represents using general sentence embeddings and the second number represents using task specific embeddings.

Defining `InsSim` We define a metric, $\texttt{InsSim}(x_e)$, for each individual evaluation example x_e based on its nearest neighbors in the provided training dataset. We notate $\texttt{topN}(x_e, X_{\text{train}}, k)$ as set of k nearest examples in the total training dataset X_{train} of x_e according to the similarity function described in Section 4.

$$\texttt{InsSim}(x_e) = \frac{\sum_{x_i \in \texttt{topN}(x_e, X_{\text{train}}, k)} \texttt{sim}(x_e, x_i)}{k}$$

We conduct our analysis with a default setting of $k = 5$.

Interpreting `InsSim` The higher $\texttt{InsSim}(x_e)$, the easier for a machine learning model to estimate $P(y_e|x_e)$, **if** the label of the example matches its nearest train neighbors (we study this further in this section). An alternative metric would be estimating the input distribution $P(x)$ based on the training

Dataset	Total	Performance (MISMATCH)			Performance (MATCH)			M/MM Δ
		All (MM)	Low	High	All (M)	Low	High	
WinoGrande	78.31 (48.22%)	78.56	**79.23**	77.17	78.20	73.47	**80.20**	-0.36
ANLI	57.34 (64.53%)	54.48	**60.26**	49.19	62.55	63.53	**67.16**	+8.08
MNLI	89.94 (54.14%)	88.35	**89.40**	86.39	91.85	92.52	**93.26**	+3.49
MRPC	88.61 (37.32%)	83.79	**84.45**	77.82	91.55	90.52	**92.79**	+7.75
TweetEval	94.50 (18.89%)	83.24	81.89	**83.89**	97.14	96.31	**98.12**	+13.91
SST-2	96.45 (21.10%)	88.59	87.27	**87.5**	98.69	97.57	**99.52**	+10.10
AG News	95.42 (10.17%)	69.47	**73.16**	65.52	98.37	98.19	**98.54**	+28.90
IMDb	95.07 (27.25%)	90.21	86.78	**92.32**	96.89	95.69	**97.93**	+6.68

Table 5: RoBERTa-large performance on MATCH (gold eval label is equivalent to the nearest gold train label) and MISMATCH (the rest) subsets of the full evaluation data. We use SimCSE embeddings for similarity. Performance on the eval examples with the highest (High) or lowest (Low) 30% of `InsSim` are shown with bolded values indicating whether performance is higher on the low or high subset. We compare to performance on the full MISMATCH or MATCH subsets in the All column (MM or M respectively). The difference between MATCH and MISMATCH values is shown in the Δ column.

data and evaluate the likelihood of x_e according to this distribution. While $P(x)$ will estimate how likely x_e is with respect to the entire training set X_{train}, `InsSim` will only consider the k closest elements in the training dataset. Given strong few-shot learning ability of recent pre-trained models (Liu et al., 2019b; Brown et al., 2020), we anticipate this metric can more effectively capture the predicted performance on example x_e.

We report the average `InsSim` score on each dataset in Table 4. A higher score will imply heavier train-evaluation dataset overlap. Using task-specific embeddings brings examples closer together significantly across all datasets. The number of total training instances varies significantly across datasets (see Table 6 in the Appendix A), so larger datasets tend to exhibit higher `InsSim`. We find that the average `InsSim` tends to be higher for tasks that are more naturally generated, indicating less data diversity between training and evaluation sets. Our metric is coarse in that it does not specify whether the similarity between instances are caused by lexical or topical overlap (e.g., containing the same entity) or syntactic overlap (e.g., similar sentence structure).

To better evaluate model generalization, we propose to divide evaluation examples into two subsets – (1) MATCH: examples where the evaluation label equals the nearest gold train label, and (2) MISMATCH: examples where the evaluation label does not match the nearest gold train label. We use general sentence embeddings (SimCSE) for the representations for better generalizability. We hypothesize that the MATCH subset is easier for models.

How does model performance differ between MATCH and MISMATCH subsets? We show RoBERTa-large performance on each of these subsets, along with the difference between them, in Table 5. As expected, performance is generally higher when labels match, confirming our hypothesis. However, this is not the case for WinoGrande. We conjecture this is because semantic similarity is not as relevant to the WinoGrande reasoning task. This is further shown by a high difference between performance on the two subsets for the AG News dataset, for which semantic similarity is more strongly relevant. In addition, Table 5 shows the percent of total examples in the MISMATCH subset; we see that overall performance on the dataset loosely *inversely* correlates with the proportion of MISMATCH examples; further illustrating that these examples are more difficult.

Can we use the `InsSim` score to identify difficult evaluation examples? We further split our MATCH and MISMATCH data subsets by their `InsSim` score: we report performance breakdown on highest and lowest 30% of the data sorted by `InsSim`. RoBERTa-large performance on these sets is also shown in Table 5. Our results indicate that a higher `InsSim` leads to higher performance on examples where the evaluation labels match the nearest train example label, but not necessarily when they do not match. In challenging datasets (WinoGrande, ANLI, MNLI and MRPC), when the label of the evaluation example does not match the la-

bel of the nearest training example, being closer to the nearest neighbor actually hurts the model performance, suggesting over-generalization from the nearest training example. These results emphasize that in addition to evaluating model performance on a full dataset, it could be useful to evaluate models on these subsets individually to better assess model generalization; performance can be significantly different on more challenging subsets. We will publicly release our code for splitting datasets into MATCH and MISMATCH subsets at https://github.com/GauriKambhatla/train_eval_overlap.

6 Conclusion

In this paper, we analyze eleven downstream NLP datasets for train-evaluation overlap using a nearest neighbors approach, quantified with a simple measure of instance similarity. We categorize datasets according to their data collection method, and find that more naturally-collected data and easier tasks tend to demonstrate higher train-eval overlap than more synthetically-generated data and difficult tasks. Lastly, we suggest using nearest neighbor analysis to split the evaluation data into more easy and challenging subsets, determined by the overlap with the training set, and advocate studying model performance on these subsets as well as the full dataset for a more comprehensive evaluation of model generalizability.

Limitations

Our study is limited in scope, studying only classification and extractive QA tasks in English; the trends we highlight in this work might not generalize to different tasks or other languages. We also acknowledge that we only use BERT-based models for our analysis, so it is uncertain whether these findings are applicable to other models. In addition, the overlap we describe in this paper is defined by semantic similarity rather than literal overlap between sentences and phrases. We are not claiming that this overlap is good or bad, rather we show that when the overlap is large, it is more difficult to evaluate model generalization.

We note that there are multiple confounding factors in our results. First, while we highlight the role of dataset collection method in our analysis, the naturalness of data collection method is negatively correlated with task difficulty (i.e., the more natural datasets we study are also the least difficult). As a result, differences in performance can be attributed to task difficulty as well as data collection method. Second, our study is limited in scope of similarity metrics (only cosine similarity) and embeddings used to compute similarity. Using different embedding or metric can change the results.

Acknowledgements

We thank the ACL reviewers and meta-reviewer for thoughtful comments and suggestions to improve the paper.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira,

Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *naacl*, abs/1810.04805.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Aparna Elangovan, Jiayuan He, and Karin Verspoor. 2021. Memorization vs. generalization : Quantifying data leakage in NLP performance evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1325–1335, Online. Association for Computational Linguistics.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. *ArXiv*, abs/1910.09753.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *ArXiv*, abs/1506.02142.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings.

Ameya Godbole and Robin Jia. 2022. Benchmarking long-tail generalization with likelihood splits.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *ArXiv*, abs/2004.04906.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. *ArXiv*, abs/1506.06726.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard L. Phillips, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. 2021a. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. 2021b. WILDS: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664. PMLR.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In *NAACL*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021a. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. RoBERTa: A Robustly Optimized BERT Pretraining Approach. Technical Report arXiv:1907.11692, arXiv. ArXiv:1907.11692 [cs] type: article.

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

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Andrey Malinin, Andreas Athanasopoulos, Muhamed Barakovic, Meritxell Bach Cuadra, Mark J F Gales,

Cristina Granziera, Mara Graziani, Nikolay Kartashev, Konstantinos Kyriakopoulos, Po-Jui Lu, Nataliia Molchanova, Antonis Nikitakis, Vatsal Raina, Francesco La Rosa, Eli Sivena, Vasileios Tsarsitalidis, Efi Tsompopoulou, and Elena Volf. 2022. Shifts 2.0: Extending the dataset of real distributional shifts.

Andrey Malinin, Neil Band, Ganshin, Alexander, German Chesnokov, Yarin Gal, Mark J F Gales, Alexey Noskov, Andrey Ploskonosov, Liudmila Prokhorenkova, Ivan Provilkov, Vatsal Raina, Vyas Raina, Roginskiy, Denis, Mariya Shmatova, Panos Tigas, and Boris Yangel. 2021. Shifts: A dataset of real distributional shift across multiple large-scale tasks.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Chunyuan Qin, Chuan Deng, Jiashun Huang, Kun xian Shu, and Mingze Bai. 2020. An efficient faiss-based search method for mass spectral library searching. *2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, pages 513–518.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text.

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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. *ArXiv*, abs/1703.05175.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2021. Commonsenseqa 2.0: Exposing the limits of ai through gamification. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Michael Tänzer, Sebastian Ruder, and Marek Rei. 2022. Memorisation versus generalisation in pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7564–7578, Dublin, Ireland. Association for Computational Linguistics.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

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.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *ArXiv*, abs/2012.15466.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

A Dataset Statistics

We provide additional statistics about the datasets we studied, including licensing and data split sizes (Table 6). The WinoGrande and CSQA 2.0 datasets are licensed with CC-BY, ANLI is licensed with Creative Commons-Non Commercial 4.0, MNLI, the TweetEval sentiment task, and NQ (MRQA version) are licensed with MIT. All the datasets we study are in English.

Dataset	Train	Dev	Collection	Task
WinoGrande	40k	1.2k	SYN	Classfication
CSQA 2	9.2k	2.5k	SYN	Classfication
ANLI	163k	3.2k	CWD	Classfication
MNLI	392k	9.8k	CWD	Classfication
SQuAD 2	131k	11.8k	CWD	ExtractiveQA
MRPC	3.6k	2.1k	LAB	Classfication
TweetEval	24.9k	6.3k	LAB	Classfication
NQ	104k	12.8k	LAB	ExtractiveQA
SST-2	67k	872	USE	Classfication
AG News	12k	7.6k	USE	Classfication
IMDb	25k	25k	USE	Classfication

Table 6: Dataset statistics. For Natural Questions (NQ), we use the MRQA subset, and for TweetEval, we use the sentiment split, with neutral label examples filtered out.

B Model Performance & Compute

Here we list the total fine-tuned model performance for each model on each validation dataset for varying amounts of training data. DistilBERT (66M parameters) performance is listed in Table 10, RoBERTa-base (123M parameters) performance in Table 9, and RoBERTa-large (354M parameters) performance in Table 8. We take the average of three runs to get the numbers listed in these tables. We run all experiments on RTX 8000 GPUs.

Dataset	E_g	E_t
WinoGrande	0.330	0.168
CSQA 2.0	0.284	0.105
ANLI	0.951	0.207
MNLI	0.343	0.119
SQuAD 2.0	0.319	0.124
MRPC	0.343	0.131
TweetEval	0.125	0.062
NQ	0.630	0.367
SST-2	0.296	0.103
AG News	0.492	0.127
IMDb	0.415	0.372

Table 7: Average unigram overlap between evaluation examples and nearest train examples according to different representation types. General embeddings are notated E_g and task-specific embeddings as E_t.

Dataset	Training Size			
	500	1k	10k	Full
WinoGrande	55.33	62.19	75.93	78.37
CSQA 2.0	51.87	54.54	–	54.66
ANLI	34.56	35.18	43.28	57.34
MNLI	76.00	84.40	86.81	89.94
SQuAD 2.0	59.12	69.29	80.44	87.49
MRPC	81.62	86.52	–	92.12
NQ	64.46	68.04	76.48	80.33
TweetEval	90.45	92.58	93.95	94.60
SST-2	92.89	93.23	95.41	96.84
AG News	90.58	90.66	93.66	95.47
IMDb	93.53	93.81	95.04	95.17

Table 8: Performance (RoBERTa-large) for each training setting. F1 scores are shown for MRPC, SQuAD 2.0, and NQ, accuracy scores shown for all other datasets. MRPC and CSQA 2.0 have training set sizes less than 10k.

C Hyperparameters

We use the hyperparameters from existing work when listed, otherwise we perform hyperparameter tuning through a grid search over learning rate (LR), number of epochs, batch size, and max sequence length. For classification tasks, these are: LR $\{2e-7, 2e-5, 2e-3\}$, epochs (full dataset) $\{3, 5, 7\}$, epochs (10k) $\{5, 7, 9, 10\}$, epochs (1k, 500) $\{7, 11, 15, 20\}$, batch size $\{32, 64, 128\}$, sequence length $\{128, 256, 512\}$. For the extractive QA tasks, these are: LR $\{3e-7, 3e-5, 3e-3\}$, epochs (full dataset) $\{2, 3\}$, epochs (10k) $\{3, 4, 5\}$, epochs (1k, 500) $\{5, 7, 10\}$, batch size $\{8, 12\}$, max length $\{384\}$.

Dataset	Training Size			
	500	1k	10k	Full
WinoGrande	53.51	56.35	61.09	66.14
CSQA 2.0	51.79	51.79	–	54.02
ANLI	35.94	36.72	42.00	51.75
MNLI	65.92	73.14	81.62	87.56
SQuAD 2.0	50.83	56.21	72.94	83.43
MRPC	81.62	86.52	–	91.50
NQ	45.77	57.40	71.96	78.92
TweetEval	89.38	91.02	93.16	93.28
SST-2	88.99	92.09	93.35	94.5
AG News	88.93	89.36	92.71	95.21
IMDb	92.86	92.71	94.86	95.54

Table 9: Performance (RoBERTa-base) for each training setting. F1 scores are shown for MRPC, SQuAD 2.0, and NQ, accuracy scores shown for all other datasets. MRPC and CSQA 2.0 have training set sizes less than 10k.

D Additional Nearest Instance Examples

Table 11 shows additional examples of nearest neighbors for the datasets not shown in Table 1.

E Examples of Nearest Neighbor Classification Failure Categories

Table 12 shows examples of evaluation examples and their nearest train neighbor whose labels do not match.

Dataset	Training Size			
	500	1k	10k	Full
WinoGrande	48.77	48.93	51.22	51.38
CSQA 2.0	51.04	51.71	–	53.99
ANLI	35.63	36.59	41.34	46.25
MNLI	49.13	54.67	68.70	82.00
SQuAD 2.0	44.12	45.14	52.52	69.75
MRPC	77.37	77.74	–	88.70
NQ	29.26	32.78	60.12	74.45
TweetEval	87.62	89.05	90.94	91.51
SST-2	82.80	84.86	89.79	91.06
AG News	87.93	88.89	91.41	94.73
IMDb	88.11	88.91	91.95	93.18

Table 10: Performance (DistilBERT-base) for each training setting. F1 scores are shown for MRPC, SQuAD 2.0, and NQ, accuracy scores shown for all other datasets. MRPC and CSQA 2.0 have training set sizes less than 10k.

Dataset	Eval example	Nearest training example (E_g)	Nearest training example (E_t)
CSQA 2.0	You should always try to phrase your questions with the most double negatives.	Do people always quote facts after being asked a question?	A good reporter always does their best work even when the assignment is underwhelming.
ANLI	**P** The Toffee Crisp bar is a chocolate bar first manufactured in the United Kingdom by Mackintosh's in 1963. It is now produced by Nestlé in the UK. It consists of... **H** The Toffee Crisp bar is not sold in the US.	**P** The Toffee Crisp bar is a chocolate bar first manufactured in the United Kingdom by Mackintosh's in 1963. It is now produced by Nestlé in the UK. It consists of... **H** The company will make a bar with no toffee.	**P** The following is a list of female cabinet ministers of Thailand. Thailand is a country located at the centre of the Indochina peninsula in Southeast Asia... **H** Thailand does not have male cabinet ministers.
SQuAD 2.0	Inter-network routing was what kind of system?	What is defined as a way of filtering network data between a host or network and another network?	In which year did Poland declassify most of its Warsaw Pact-era archives?
MRPC	**Phrase 1** Saddam's other son, Odai, surrendered Friday, but the Americans are keeping it quiet because he's a U.S. agent. **Phrase 2** Hussein's other son, Uday, surrendered yesterday, but the Americans are keeping it quiet because he's a US agent.	**Phrase 1** The only other JI member to reveal similar information is Omar al Faruq , now held at a secret location by the United States. **Phrase 2** The only other JI member to reveal similar information is Omar al Faruq, now held by the United States at a secret location.	**Phrase 1** Initial reports said the attackers fired from a mosque within in the city, 30 miles west of Baghdad. **Phrase 2** The Centcom statement said the gunmen appeared to have fired from a mosque in the city, 50 km (32 miles) west of Baghdad.
SST-2	i just loved every minute of this film.	i loved this film.	gives a superb performance full of deep feeling.
IMDb	Haines is excellent as the brash cadet who thinks West Point will really amount to something now that he has arrived. Haines displays his easy, goofy comic persona as he takes on West Point and Joan Crawford, the local beauty...	One of the biggest hits of 1926, Brown of Harvard is a exciting comedy/drama featuring regatta and football scenes that gave William Haines the role he needed to become a major star. It's patented Haines all the way: brash smart aleck who takes nothing serious until he is rejected by everyone...	As Jack Nicholson's directorial debut, Drive, He Said displays at the least that he is a gifted director of actors. Even when the story might seem to lose its way to the audience (and to a modern audience - if they can find it, which pops up now and again on eBay - it might seem more free formed than they think)...

Table 11: Examples of the most similar instances for the evaluation example according to two embedding methods.

Category	Dataset	Example (eval and most similar train)	Labels
not similar	MNLI	*Eval*: **P** uh i don't know i i have mixed emotions about him uh sometimes i like him but at the same times i love to see somebody beat him **H** I like him for the most part, but would still enjoy seeing someone beat him. *Train*: **P** You can imagine what a thorn in the flesh I am to him! **H** You can imagine how much he is bothered by me, even though I treat him well	*Eval*: Entail *Train*: Neutral
mismatch	WinoGrande	*Eval*: Randy only ever added a little bit of hot sauce to his food, especially compared to Adam, as _ was much more sensitive to spice. *Train*: Randy found it easier to be healthy than Derrick because _ did not eat a wide variety of fruits and vegetables.	*Eval*: Randy *Train*: Derrick
ambiguous	AG News	*Eval*: Intel Doubles Dividend, Boosts Buyback by $11.5 Bln (Update2) Intel Corp., the world's biggest computer-chip maker, doubled its quarterly dividend and boosted its stock buyback program by $11. *Train*: Intel Doubles Dividend, Expands Buyback Chip giant Intel Corp. reported Wednesday that its board doubled the company's quarterly dividend and authorized an expansion of its ongoing stock repurchase program.	*Eval*: Business *Train*: Sci/Tech

Table 12: Examples of label-mismatched eval and nearest train examples for each category.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section after conclusion (6)

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Sections 3, 4

☑ B1. Did you cite the creators of artifacts you used?
Sections 3, 4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix A

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
These are listed in Table 6 (Appendix)

C ☑ Did you run computational experiments?

Sections 4, 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix B

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix C

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Appendix B

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.