PiC: A Phrase-in-Context Dataset for Phrase Understanding and Semantic Search

Thang M. Pham [†]
thangpham@auburn.edu

Seunghyun Yoon [§]
syoon@adobe.com

Trung Bui [§]
bui@adobe.com

Anh Nguyen [†]
anh.ng8@gmail.com

[†]Auburn University   [§]Adobe Research

Abstract

While contextualized word embeddings have been a de-facto standard, learning contextualized *phrase* embeddings is less explored and being hindered by the lack of a human-annotated benchmark that tests machine understanding of phrase semantics given a context sentence or paragraph (instead of phrases alone). To fill this gap, we propose PiC— a dataset of ∼28K of noun phrases accompanied by their contextual Wikipedia pages and a suite of three tasks for training and evaluating phrase embeddings. Training on PiC improves ranking-models' accuracy and remarkably pushes span-selection (SS) models (i.e., predicting the start and end index of the target phrase) near human-accuracy, which is 95% Exact Match (EM) on semantic search given a query phrase and a passage. Interestingly, we find evidence that such impressive performance is because the SS models learn to better capture the common meaning of a phrase *regardless* of its actual context. SotA models perform poorly in distinguishing two senses of the same phrase in two contexts (∼60% EM) and in estimating the similarity between two different phrases in the same context (∼70% EM).

1 Introduction

Understanding phrases in context is a key to learning new vocabularies (Nagy et al., 1985; Fischer, 1994), disambiguation (Pilehvar and Camacho-Collados, 2019), and many downstream tasks, including semantic search (Finkelstein et al., 2001). Yet, the contextualized *phrase* embeddings (Yu and Ettinger, 2020) in existing systems mostly capture the common meaning of a phrase, i.e. without strong dependence on its context (Yu and Ettinger, 2020). While there are *word*-sense disambiguation datasets (Edmonds and Cotton, 2001; Pilehvar and Camacho-Collados, 2019), no such benchmarks exist for *phrases*. Existing phrase-similarity benchmarks (Pavlick et al., 2015; Turney, 2012; Asaadi

et al., 2019; Zhang et al., 2019; Yang et al., 2019) compare phrases alone (without context) and some of them (Pavlick et al., 2015; Zhang et al., 2019) contain a large, undesired amount (∼15% to 99%) of phrase pairs that have lexical overlap (Table 1).

Others generated the context for a phrase by querying GPT-2 (Wang et al., 2021) or by retrieving from Wikipedia (Yu and Ettinger, 2020). Yet, there was no human verification of the realism of generated text (Wang et al., 2021) and no human annotation of how a phrase's meaning changes w.r.t. the context (Yu and Ettinger, 2020). All above drawbacks are limiting the evaluation of phrase understanding.

To advance the development of contextualized phrase embeddings, we propose Phrase-in-Context (PiC), a suite of three tasks: (1) Phrase Similarity (PS), i.e. compare the semantic similarity of two phrases in the same context sentence (Fig. 1b); (2) Phrase Retrieval (PR), which is divided into PR-pass and PR-page (Fig. 1c–d), i.e. from a passage or a Wikipedia page, retrieve a phrase semantically-similar to a given query phrase; and (3) Phrase-Sense Disambiguation (PSD), i.e. find the target phrase p semantically similar to the query phrase from a 2-paragraph document where p appears twice, each time in a different context paragraph that provides a *unique* meaning to p (Fig. 1e). Our ∼28K-example dataset is rigorously (a) *annotated* and *verified* by two groups of annotators: linguistics experts on Upwork.com and non-experts on Amazon Mechanical Turk (MTurk); and then (b) *tested* by models, linguists, and graduate students. Our contributions are:

1. We build PiC[1], the first, human-annotated benchmark for evaluating and training contextualized phrase embeddings (Sec. 4). Compared to existing phrase similarity datasets, PS is the first to require models to rely on context.

---

[1]Dataset, code, and demos are available on https://phrase-in-context.github.io.

2. After training on PR-pass, i.e. finding a phrase from a passage, SS models perform at a near-human accuracy (92–94% vs. 95% EM). They also score high (84–89% EM) on PR-page, i.e. semantic phrase search in a Wikipedia page (Sec. 5.4), suggesting our training set and learned embeddings are useful for real-world semantic search.

3. Interestingly, on PR-pass, harnessing these SS models' phrase embeddings in a ranking approach (i.e. comparing the similarity between the query and *all* candidate phrases) yields poor accuracy of $\leq 59\%$ EM (Sec. H), setting a challenge for future research into learning contextualized phrase embeddings.

4. After training on PR-pass, state-of-the-art (SotA) models perform relatively well on PR-pass and even PR-page but not on PSD (Sec. 5.5). On PS, SotA models perform poorly (below 70% accuracy) in binary classification of phrase similarity given a context sentence (Sec. 5.1).

2 Related Work

Each of our tasks (PS—phrase similarity; PR—phrase retrieval; and PSD—disambiguation) is related to a separate research area discussed below.

Phrase similarity First, most existing phrase similarity datasets—e.g. PPDB-annotated (Wieting et al., 2015), PPDB-filtered (Wang et al., 2021), BiRD (Asaadi et al., 2019), and PAWS-short (Wang et al., 2021; Zhang et al., 2019)—contain a large percent of instances with lexical overlap between two paired phrases while our PS contains the least percent (5.34%; Table 1). Second, PS compares each pair of phrases in a context sentence while existing datasets only compare phrases alone (no context). Third, the phrases in PS are, on average, 2-token long, comparable to that of other datasets (Table 1). Fourth, unlike other datasets, PS contains exclusively *noun-phrases*, the most common phrase type according to Yahoo's search-query statistics (Yahoo, 2022a) (79.54%; Appendix D) and Adobe (internal Acrobat Pro data not shown).

Question answering (QA) Our phrase-retrieval tasks—PR and PSD—follow the format of QA datasets except that our queries are *phrases* instead of questions and hence shorter (Table 2). Like SQuAD 1.1 (Rajpurkar et al., 2016) and HotpotQA

Table 1: Our Phrase Similarity (PS) dataset has a lower percent of lexical-overlap instances and is the *only* human-annotated dataset that provides phrases, each in a context sentence.

	PS (ours)	WiC	PPDB-annotated	PPDB-filtered	BiRD	Turney	PAWS-short
# of All instances	10,004	7,466	3,000	15,532	3,345	2,180	1,214
# of Unique phrases	7,488	2,345	6,000	12,023	2,840	9,776	1,214
Lexical overlap (%)	5.34	100	70.10	97.93	14.98	0	99.42
Mean length (in tokens)							
• phrase$_1$	2.06	1	3.67	2	2	2	9.52
• phrase$_2$	2.46	1	3.73	2	1.49	1	9.42
• context sentence	22.53	8.40	0	0	0	0	0

(Yang et al., 2018), our documents and queries are extracted from Wikipedia articles. While our PR dataset is $\sim 3.5\times$ smaller than those two datasets, the paragraph document length in PR-pass and PSD is $\sim 2\times$ longer than those of SQuAD 1.1 and HotpotQA (Table 2). For our task, intuitively, the longer the document, the harder the task since there would be more candidates a model must compare with the query.

Table 2: Our PR-pass, PR-page and PSD datasets are smaller in size compared to common QA datasets and contain shorter queries that are noun phrases instead of questions. However, our tasks require searching in much longer documents.

	PR-pass	PR-page	PSD	SQuAD 1.1	HotpotQA
All instances	28,147	28,098	4,858	98,169	105,257
Unique queries/questions	27,055	27,016	4,812	97,888	105,249
Unique answers	13,458	13,423	2,314	72,469	57,259
Mean length of					
query (tokens)	2.42	2.42	2.45	11.42	20.03
answer (tokens)	2.17	2.17	2.07	3.46	2.35
sentence (tokens)	23.22	24.08	23.00	27.62	26.77
document (sentences)	10.26	119.32	20.37	5.10	4.14
document (tokens)	238.34	2,872.73	468.48	140.92	110.72

Sense disambiguation While word-sense disambiguation (WSD) is a long-standing problem in NLP, recently, SotA models have reached super-human accuracy (80% F_1) on the common English WSD (Bevilacqua et al., 2021). Interestingly, these high-scoring models still struggle with rare senses that may be outside of the predefined sense inventories or have few training examples (Blevins et al., 2021). Without the need for predefined senses, WiC (Pilehvar and Camacho-Collados, 2019) poses disambiguation as a binary classification task where the goal is to predict whether the same target word in two different sentences carries the same or different meanings.

Compared to WiC PS is also a binary classification task, but with two major differences: (1) in

(a) Q1 & Q2 ask annotators to rephrase "massive figure" in page 1 and page 2. Q3 asks whether this phrase's meaning is the same in both pages.

(b) **PS** positive & negative examples constructed using page 1 context (similarly, we repeat for page 2).

(c) A **PR-pass** example.
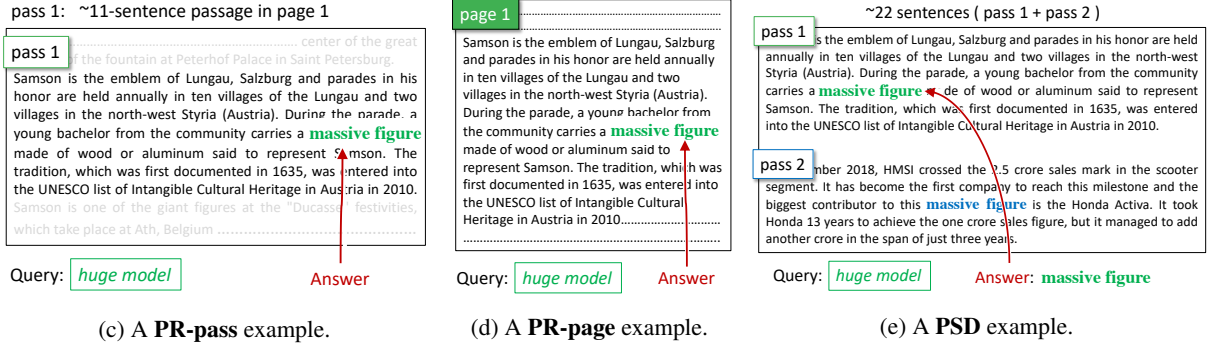
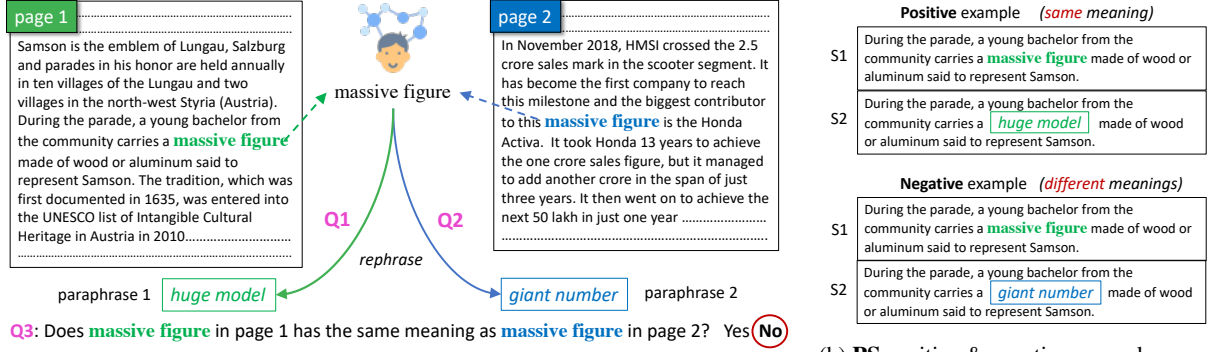(d) A **PR-page** example.

(e) A **PSD** example.

Figure 1: Given a phrase, two associated Wikipedia pages, and expert annotations, i.e. answers to Q1, Q2, and Q3 (a), we are able to construct *two* pairs of positive and negative examples for PS (b), a PR-pass example (c), a PR-page example (d), and a PSD example *only if* the answer to Q3 is No (e).

WiC, the same target word appears in two different sentences while in PS, two different phrases appear in the same context sentence; (2) PS compares phrases composed of ≥ 2 words instead of a single word as in WiC and WSD. While word senses are defined in WordNet and BabelNet dictionaries (Bevilacqua et al., 2021), there are no English dictionaries of senses for multi-word noun phrases (mNPs). Thus, it is more challenging to acquire and learn the senses of mNPs, hence the importance of our PiC dataset. Like WiC, PSD tests disambiguating the meanings of the same n-gram in two different contexts. Yet, PSD is a phrase search task, which involves many more phrase comparisons per example than PS or WiC.

Before the deep learning era, phrase-sense disambiguation was already proposed (Carpuat and Wu, 2007a,b) but only as an auxiliary task for training machine-translation models. And their phrase senses were not annotated by humans but inferred by performing word-alignment on a bilingual corpus. Here, our PSD is the first phrase-sense disambiguation task annotated by experts and requires understanding of phrase-senses in a passage.

3 PiC Dataset Construction

We first collect a set of phrases with context and human annotations. Then, we derive the examples and labels for three main tasks: PS, PR, and PSD (Fig. 1). Our idea is to mine a set of triplets (p, page$_1$, page$_2$) from Wikipedia where the phrase p is a polysemous mNP that carries two different senses in two Wikipedia pages (e.g., "massive figure" means *a large number* in page$_1$ but *a huge physical shape* in page$_2$; Fig. 1a). Then, we ask experts to rephrase p into two paraphrases q_1 and q_2, maintaining the two original senses of p in page$_1$ and page$_2$, respectively. The resultant set of 5-tuples (p, q_1, q_2, page$_1$, page$_2$) enables the tests for (1) comparing the semantic similarity of two phrases given the same context sentence (PS; Fig. 1b); (2) finding a semantically similar phrase in a document (PR-pass & PR-page; Fig. 1c); (3) disambiguating the senses of the same target mNP given two context paragraphs (PSD; Fig. 1e).

3.1 Data Collection

As there are no English dictionaries that contain sense inventories for mNPs, the key **challenge** to

our data collection is to mine mNPs that have (1) multiple senses; and (2) a Wikipedia context page for each sense. To do that, we take a Wikipedia dump and perform a **6-step** procedure that essentially extracts all the mNPs that occur in more than one Wikipedia page and that contain at least one polysemous word defined in the WiC dataset. From the triplets of (p, page$_1$, page$_2$), we programmatically narrow down to $\sim$600K triplets where the context sentence of the mNP in page$_1$ is the most semantically *dis*similar to the context sentence in page$_2$ (according to SimCSE (Gao et al., 2021)). We continue filtering down to the top 19,500 triplets where page$_1$ and page$_2$ have *the most* semantically dissimilar lists of Wikipedia categories. That is, 19,500 triplets are estimated to yield $\sim$15K annotated triplets (the target size based on our budget) after the human annotation process where annotators are allowed to skip the cases they are not confident labeling. See Appendix C for a detailed description of the data collection and **dataset biases**.

3.2 Data Annotation

Via Upwork, we hire 13 linguistics experts who are native English speakers at a rate of $30/hour to annotate 15,021 out of 19,500 examples. For each phrase, we provide Upworkers with a triplet (p, passage$_1$, passage$_2$) where each passage$_i$ consists of 5 sentences centered at the phrase-containing sentence in the corresponding page$_i$. We ask them to answer the three below questions (Fig. 1a):

Q1 Rephrase the target phrase p to a paraphrase q_1 such that its meaning is constant in passage$_1$.

Q2 Similarly, rephrase p w.r.t. passage$_2$ to obtain a paraphrase q_2.

Q3 Answer Y/N if p has the same meaning in both contextual passage$_1$ and passage$_2$.

Upworkers are asked to provide paraphrases that (1) have at least two words and (2) minimize lexical overlap with each other and the target p. See the annotation guidelines (PiC, 2021a) and a sample annotation assignment (PiC, 2021b) given to Upworkers. After receiving annotations, we use LanguageTool (2022) to automatically find syntactical errors when the paraphrases are replaced by the original target phrase in the original passage and ask Upworkers to fix them. We also have annotators fix the remaining errors that we find via manual inspection.

3.3 Annotation Verification

To verify the annotations obtained in Sec. 3.2 (i.e. $2 \times 15{,}021 = 30{,}042$ paraphrases; and 15,021 Y/N labels), first, we present the same Q1, Q2, and Q3 questions to 1,000 qualified MTurkers and ask whether they agree with the answers by expert annotators in Sec. 3.2. And then, for the cases that the MTurkers disagree with, we seek second opinions from 5 Upwork experts. After these two verification rounds, we discard all the examples where Upwork verifiers reject and arrive at the final 28,325 paraphrases and 13,413 Y/N labels (i.e. those annotations that *either* an MTurk or Upwork verifier endorses). See more details in Appendix J.

The total fee for both MTurk and Upwork combined is around USD 30,000.

4 Three Phrase Understanding Tasks

Using the human-annotated data, we construct three tasks of PS, PR, and PSD (as summarized in Fig. 1) for evaluating contextualized phrase-embeddings and semantic-search models.

4.1 Phrase Similarity (PS)

PS is a binary classification task that asks whether two mNPs are semantically similar or not given *the same context* sentence. The unique challenge of PS is that, *without* context, the two given phrases can be easily interpreted as synonymous. Yet, in our PS context sentence, the two phrases *may* or may *not* carry distinct meanings (Fig. 1b).

Construction From the annotated data, a positive example is a triplet of (an original phrase p, a paraphrase q_1, an original page$_1$'s sentence that contains p). To create a negative example, from the same triplets, we select only those where the paraphrase q_2 holds a *different* meaning than q_1 given the page$_1$ context of q_1 (i.e., when the answer to Q3 is No; see Fig. 1b). For quality assurance, we also hire three extra Upwork experts to double-check PS annotations (see Appendix E), keeping only examples that at least 2 out of 3 experts endorse. In total, we obtain 5,002 *negative* examples. Then, we randomly select 5,002 *positive* examples to form a class-balanced PS dataset.

4.2 Phrase Retrieval (PR)

PR is a task of finding in a given document d a phrase p that is semantically similar to the given query phrase, which is the paraphrase q_1 (the answer by annotators to Q1) or q_2 (the answer to

Q2). We release two versions of PR: **PR-pass** and **PR-page**, i.e. datasets of triplets (query q_1, target phrase p, document d) where d is a random 11-sentence passage that contains p (Fig. 1c) or an entire Wikipedia page (Fig. 1d). While PR-pass contains 28,147 examples, PR-page contains slightly fewer examples (28,098) as we remove those examples whose Wikipedia pages coincidentally also contain exactly the query phrase (in addition to the target phrase). Both datasets are split into ~20K/3K/5K for train/dev/test, respectively.

4.3 Phrase Sense Disambiguation (PSD)

The task is to find the location of the target phrase p where it has a similar meaning to that of the given query q in a 2-paragraph document where, by construction, p appears exactly twice but only one location is the correct answer (Fig. 1e).

Construction From the verified annotations in Sec. 3.3, there are in total 4,938 phrases that both annotators and verifiers agree to hold *different* meanings across the two context Wikipedia pages (i.e., "No" answer to Q3 in Fig. 1a). To create a PSD example, given a phrase p from the above 4,938, we extract two corresponding ~11-sentence paragraphs (from its associated $page_1$ and $page_2$ as in PR-pass) and concatenate them (separated by an empty line) into a single document (Fig. 1e). Since a PSD example shares a pair of phrases (*query* and *answer*) with one PS *positive* example (*phrase$_1$* and *phrase$_2$*), we filter out that PSD example if the corresponding PS example is removed from the additional verification round (Appendix E). As the result, we exclude 80 examples and obtain 4,858 examples in total for PSD.

5 Experiments and Results

We test SotA models on PS, PR-pass, PR-page, and PSD to (1) assess how the models are able to leverage context to improve accuracy; and (2) quantify the headroom for future research.

Phrase embeddings Besides training and testing SotA BERT-based classifiers, we also test a ranking approach that involves computing the cosine similarity between the query's and each candidate's embedding. To compute a *contextualized* phrase embedding, following Yu and Ettinger 2020, we feed the entire phrase-containing sentence (e.g. S_1 in Fig. 1b) into a model, e.g. BERT, and then take the mean pooling of the last-layer embeddings over the words of the given phrase only. For non-

contextualized phrase embeddings, we repeat the same process but input to the model only the phrase (instead of the entire sentence).

Models We choose SotA models in (a) phrase similarity: PhraseBERT (Wang et al., 2021); (b) sentence similarity: USE-v5 (Cer et al., 2018), SentenceBERT (Reimers and Gurevych, 2019), and SimCSE (Gao et al., 2021)); (c) question-answering: Longformer (Beltagy et al., 2020), DensePhrase (Lee et al., 2021); and (d) contextualized embeddings: SpanBERT (Joshi et al., 2020) and BERT (Devlin et al., 2019).

For DensePhrase, we use their Phrase-Encoder (as opposed to the Query-Encoder) to compute phrase embeddings. USE-v5 is only available via public APIs (TensorFlow, 2022) that do not support extraction of contextualized phrase embeddings.

5.1 Phrase Similarity: Contextualized phrase embeddings improve accuracy

Q: *Does incorporating context improve the phrase-similarity accuracy on PS?*

Experiment We split the PS dataset 70/10/20 for train/dev/test and test two approaches: (1) using the cosine similarity score between two pre-trained phrase-embeddings (with and without context) to predict phrase similarity; (2) training BERT-based binary classifiers directly using PS training set. We use 6 backbone BERT models that are all "base" versions unless specified otherwise (Table 3).

Approach 1: Cosine similarity First, we test how *pre-trained* phrase embeddings alone (without finetuning or extra weights) can be leveraged to solve PS. For each PS example of two phrases, we compute their non-contextualized phrase embeddings and compute their cosine similarity score. To evaluate the pre-trained embeddings on PS, we follow Yang (2022) and tune the binary-classification threshold T to maximize the training-set accuracy, and then use the same optimal T to report the test-set accuracy. We repeat the experiment for *contextualized* phrase embeddings.

Approach 2: BERT-based classifiers To complement Approach 1, we test Approach 2, i.e. building a binary classifier by adding two extra MLP layers on top of the pre-trained embeddings used in Approach 1. For a phrase pair, we concatenate the two 768-D phrase embeddings from BERT$_{base}$ into a 1,536-D vector, and then place one ReLU layer (256 units) and a 1-output linear classification layer with sigmoid on top. Following Wang et al.

Table 3: Accuracy (%) of state-of-the-art BERT-based models on the PS test set. Contextualized phrase embeddings ("Phrase + Ctx") yield substantially higher performance on PS than non-contextualized embeddings ("Phrase"). The random baseline is 50%.

Model	Approach 1: Cosine similarity		Approach 2: BERT-based classifiers	
	(a) Phrase	(b) Phrase + Ctx	(c) Phrase	(d) Phrase + Ctx
PhraseBERT	51.75	63.40 (+11.65)	33.60	66.10 (+32.50)
BERT	51.05	64.10 (+13.05)	37.00	68.85 (+31.85)
SpanBERT	49.30	64.00 (+14.70)	40.15	66.85 (+26.70)
SpanBERT$_{Large}$	50.40	**66.30** (+15.90)	35.95	**69.25** (+33.30)
SentenceBERT	50.35	60.30 (+9.95)	31.50	62.55 (+31.05)
SimCSE	52.15	62.50 (+10.35)	34.20	66.65 (+32.45)
mean ± std	50.83 ± 1.04	**63.43 ± 1.98**	35.40 ± 3.01	**66.71 ± 2.40**

(2021), we finetune these models for a maximum of 100 epochs (with early stopping and patience of 10 epochs) on the train set. See Appendix A for more training details.

Results *Without* context, all models perform at $\leq$ 50% accuracy (i.e. the random chance; Table 3a & c). Interestingly, incorporating context information into phrase embeddings substantially improves mean model-accuracy on PS for both Approach 1 (from 50.83% to 63.43%; Table 3b vs. a) and Approach 2 (from 35.40% to 66.71%; Table 3d vs. c), showing evidence that **PS requires models to rely on context**. While starting from the same backbone models, Approach 2 yields higher mean accuracy than Approach 1 (Table 3; 66.71 vs. 63.43), which is expected as Approach 2 models have more capacity and the backbones are allowed to be finetuned on PS. See Figs. A3–A6 for qualitative PS predictions from a PhraseBERT-based classifier.

5.2 Human Baselines and Upperbound (95% Exact Match) on Phrase Retrieval

To interpret the progress of machine phrase-understanding on PR, here, we establish multiple human baselines for both non-experts and linguistics experts (with and without training them).

Experiment We recruit participants and have them perform one or two tests per person. A test consists of 20 PR-pass examples. That is, PR-pass documents are 11-sentence long and are feasible for a person to read in minutes (compared to reading an entire Wikipedia page). We test three groups: (1) 21 graduate students at our institution (1 test per person); (2) five Upwork experts (1 test per person); and (3) another five Upwork experts (2 tests per person, i.e., for a total of $2 \times 5 = 10$ tests).

Table 4: Best SS models reach near the Upperbound (95%) on PR-pass. Yet, ranking models based on phrase embeddings significantly *underperform* SS models.

Accuracy of human groups and models	EM (%)
Group 1: 20 Non-experts (w/o training)	73.60 ± 7.90
Group 2: 05 Experts (w/o training)	82.00 ± 12.00
Group 3: 05 Experts (w/ training)	**90.50 ± 3.70**
Best human accuracy (4 people)—Upperbound	**95.00 ± 0.00**
Best untrained, ranking model (BERT)	47.44
Best PR-trained, ranking model (PhraseBERT)	59.02
Best PR-trained, SS model (Longformer$_{Large}$)	**94.28**

The students in Group 1 volunteer to help our study unpaid while the Upworkers (Group 2 and 3) are hired using the same procedure as in Sec. 3.2.

Results First, we find an unsurprising, large gap between non-experts and experts (Table 4; 73.60% vs. 82.00%). Second, we train experts in Group 3 by having each do a preliminary test and giving them feedback before the real test. We find the training to substantially boost expert accuracy further (from 82.00% to 90.50%). Importantly, we find the Human Exact Match (EM) Upperbound to be 95%, i.e. the highest scores that 4 people (among all groups) make. Upon manual inspection of the submissions of these best performers, we find their incorrect answers sometimes partially overlap with the groundtruth or are sometimes reasonable. In other cases, the best performers find acceptable answers but that do not overlap at all with the groundtruth labels in PR. That is, we estimate a 5% of noise in the annotations of PR.

5.3 Phrase Retrieval: In ranking, context only helps BERT embeddings but not others

One way to evaluate the quality of SotA phrase embeddings is by testing:

Q: *How well do phrase embeddings perform in the **ranking** approach on PR?*

Ranking is a challenging and meaningful phrase-embedding test because the embedding of the query is compared against that of all phrase candidates (extracted by tokenizing the document), which can include syntactically-incorrect phrases, meaningless phrases or rare phrases. Such out-of-distribution challenge appears less often in PS or WiC, i.e. a binary classification setting.

Experiment As described in Sec. 4.2, the PR train/dev/test splits are 20,147/3K/5K examples and we only use the 5K-example test set to test

the models in this ranking experiment (no training). We follow (Lee et al., 2017) for span enumeration to construct a list of candidate phrases, we split each PR document into multiple sentences (using NLTK sentence splitter) and tokenize each sentence into tokens (using NLTK tokenizer) and build an exhaustive list of n-grams (here, $n \in \{2, 3\}$ only for computational tractability). For every example, we add the groundtruth phrase (which can be longer than 3 words) to the list of candidates (since we are only interested in testing phrase embeddings, not the phrase extractor).

Results We report top-k accuracy (for $k = 1, 3, 5$) and top-5 Mean Reciprocal Rank (MRR@5) on the PR-pass test set in Table 5a. First, for most SotA embeddings, incorporating context sentence *hurts* the accuracy (*except for BERT* embeddings). That is, interestingly, for all BERT embeddings (base and large), the accuracy increases substantially (+17.64 and +19.04; Table 5) when the one-sentence context is the input. In contrast, most models that started from BERT but were later fine-tuned lost the capability to leverage the context information (e.g., PhraseBERT, DensePhrase, and SpanBERT in Table 5).

Second, the best top-1 accuracy scores on PR-pass for non-contextualized (USE-v5; 43.36%) and contextualized (BERT; 47.44%) embeddings are substantially lower than the non-expert baselines (73.60%; Table 4) and Human Upperbound (95%). Future work is required to learn more robust, phrase embeddings for ranking. See Figs. A10–A11 for qualitative examples.

5.4 Phrase Retrieval: Span-selection models reach near-human accuracy

Consistent with Yu and Ettinger (2020), our ranking results in Sec. 5.3 reveal that there exists a large headroom for improving both non-contextualized and contextualized phrase embeddings. Yet, because ranking is a naive approach and SS models (Huggingface, 2022b; Devlin et al., 2019) are the SotA approach on many QA tasks (Rajpurkar et al., 2016), here we train SS models on the train set of PR-pass and PR-page in order to test:

Q: *How well do SotA semantic-search models perform on PR-pass and PR-page?*

Experiment We take the SotA embeddings tested in Sec. 5.3 and add a linear classification layer on top and finetune each entire classifier on the train set of PR-pass or PR-page for 2 epochs us-

ing the default HuggingFace hyperparameters (see Appendix B for finetuning details). Following the standard setup of BERT architectures for QA tasks (Devlin et al., 2019), each SS model predicts the start and end index of the target phrase. Additionally, since PR-page documents are much longer than a typical QA paragraph (Table 2), we also test training Longformer (Beltagy et al., 2020), which has a max sequence-length of 4,096, sufficient for an entire Wikipedia page. We take the models of the smallest dev loss and report their test-set performance in Table 6.

Results On PR-pass, in contrast to the poor performance of ranking models (Sec. 5.3), our PR-pass-trained SS models perform impressively at a near-upperbound level (~93–94% EM; Table 6a) surpassing the accuracy of trained experts (90.50% EM). Surprisingly, on PR-page where the documents are substantially longer (around 12×) than the documents of PR-pass, SS models' accuracy only drops slightly (from ~94% to ~85–89% EM; Table 6b). Note that in a full Wikipedia page of PR-page, there might be phrases that can be considered correct but are *not* labeled groundtruth according to our annotations. This remarkable result suggests that training on PR-pass can enable high-performing models on real-world semantic search.

5.5 Phrase Sense Disambiguation: Best models also perform poorly

We find that SotA PR-pass-trained SS models reach superhuman accuracy on PR-pass, i.e. finding a phrase of the same meaning (Sec. 5.4). Yet, PR-pass only tests models' understanding of a *single sense* of the target phrase at a time. It is interesting to study:

Q: *Do PR-pass-trained SS models understand contextualized phrases sufficiently to separate two different senses of the same target phrase?*

Experiment To do that, here we test the best PR-pass-trained SS models on PSD. Note that, PSD has the same task format as PR-pass (see Fig. 1c–e) except that the document is twice as long and contains **two occurrences of the same target phrase**. We do not test the ranking models as they perform much worse than the SS models in Sec. 5.3.

Results Although the PR-pass-trained SS models are never trained on PSD, they interestingly frequently find one occurrence of the target phrase (mean of 94.01% EM; Table 6c). However, they mostly locate the **target phrase in the wrong con-**

Table 5: **Ranking** accuracy (%) on **PR-pass** using the state-of-the-art pretrained phrase embeddings. See Appendix F for the results on PR-page. Δ (e.g. -3.62) denotes the differences between the Top-1 accuracy in the contextualized ("Phrase + Context") vs. the non-contextualized ("Phrase") setting.

Model	Phrase				Phrase + Context			
	Top-1	Top-3	Top-5	MRR@5	Top-1 (Δ)	Top-3	Top-5	MRR@5
PhraseBERT (Wang et al., 2021)	36.62	66.96	75.90	52.20	33.00 (-3.62)	49.60	56.70	41.90
BERT (Devlin et al., 2019)	29.80	47.90	55.40	39.50	**47.44** (+17.64)	**65.78**	**73.30**	**57.30**
BERT$_{Large}$ (Devlin et al., 2019)	23.76	38.52	45.40	31.70	**42.80** (+19.04)	**58.90**	**64.90**	**51.30**
SpanBERT (Joshi et al., 2020)	20.88	31.04	35.20	26.40	14.40 (-6.48)	30.46	39.80	23.40
SentenceBERT (Reimers and Gurevych, 2019)	22.30	50.64	60.60	36.80	**25.14** (+2.84)	39.52	46.20	32.90
SimCSE (Gao et al., 2021)	28.10	53.70	64.60	41.60	**32.40** (+4.30)	53.44	62.80	**43.70**
USE-v5 (Cer et al., 2018)	**43.36**	**70.12**	**78.90**	**57.30**	n/a	n/a	n/a	n/a
DensePhrase (Lee et al., 2021)	32.24	51.30	60.50	42.60	31.50 (-0.74)	46.30	53.80	39.70

Table 6: **Test**-set performance (%) of **SS models** on PR-pass (a), PR-page (b), and PSD (c). When trained on PR-pass (a) and PR-page (b), SotA SS models perform well. However, testing the PR-pass-trained models on PSD shows a significant drop in accuracy (c). That is, SotA SS models tend to understand a *single sense* of a phrase in context well (high PR-pass, PR-page, and PSD EM scores). Yet, they are not able to differentiate two senses of the same phrase (e.g., here, PhraseBERT accuracy drops -41.27 points between EM+loc vs. EM scores on PSD).

Model	(a) PR-pass		(b) PR-page		(c) PSD			
	EM	F$_1$	EM	F$_1$	EM	F$_1$	EM+loc	F$_1$+loc
PhraseBERT (Wang et al., 2021)	93.42	94.97	85.24	87.19	92.98	94.08	51.67 (-41.31)	51.83
BERT (Devlin et al., 2019)	93.26	94.65	85.64	87.77	93.50	94.57	54.84 (-38.66)	55.07
BERT$_{Large}$ (Devlin et al., 2019)	93.64	95.16	87.36	89.52	94.67	95.57	55.43 (-39.24)	55.61
SpanBERT (Joshi et al., 2020)	93.50	95.02	87.28	87.66	92.26	93.30	52.20 (-40.06)	52.34
SentenceBERT (Reimers and Gurevych, 2019)	93.24	94.54	84.66	86.89	93.21	94.15	52.74 (-40.47)	52.85
SimCSE (Gao et al., 2021)	92.90	94.51	85.68	87.66	92.96	94.05	53.83 (-39.13)	53.94
Longformer (Beltagy et al., 2020)	94.26	**95.58**	**89.54**	**91.15**	96.17	96.88	**62.72** (-33.45)	**62.83**
Longformer$_{Large}$ (Beltagy et al., 2020)	**94.28**	95.53	87.58	89.32	**96.32**	**96.91**	59.72 (-36.60)	59.82
mean	93.56	95.00	86.92	88.85	94.01	94.94	55.39 (-38.62)	55.54
± std	0.49	0.42	1.93	1.73	1.54	1.36	3.90	3.88

text passage with high confidence scores. That is, if we consider also the correctness of the location of the predicted phrase, their EM+loc[2] accuracy drops significantly to an average of 55.39%. Also, finetuning on a 2K-example train set of PSD only slightly improves the EM+loc to an average of 64.24% on a 3K-example PSD test set (Appendix G). Note that we estimate the Human Upperbound on PSD to be 95%, i.e. the same as that of PR-pass. See qualitative examples and predictions of Longformer (i.e. the best model tested) in

Figs. A7–A9.

In sum, there is a large headroom for future research on PSD. SS models are *not* yet capable of leveraging surrounding words to differentiate between two senses of the same phrase. Interestingly, after training on PR-pass, their contextualized phrase embeddings perform much worse in the ranking experiments on PR-pass (Appendix H).

6 Discussion and Conclusion

Limitations   Our dataset is currently limited to multi-word, English noun-phrases. Furthermore, it is expected to contain around a 5% error on PR-pass (i.e. the best human performance is 95% EM).

---

[2]For a PSD example, if the predicted span does not intersect at all with the groundtruth span, the EM+loc and F$_1$+loc scores would be 0. If they intersect, the two scores would be equal to EM and F$_1$, respectively.

On PR-page, there may be more than one correct target phrase; however, we only label one phrase as the correct answer per document. We use only phrases that contain at least one WiC word.

While WiC and English WSD rely exclusively on dictionaries (Pilehvar and Camacho-Collados, 2019) to obtain word senses and example sentences, our data collection depends on Wikipedia, WiC, & NLP models and our annotation depends on experts. In sum, we present PiC, the first 3-task suite for evaluating phrases in context. SS models can obtain high accuracy on semantic search after training on our PR-pass and PR-page datasets. Yet, their capability is limited to finding a semantically-similar phrase given a single context that contains the target phrase (in PR-pass). The results on PS and PSD show that SotA phrase embeddings are still limited in encoding contextualized phrases. It is interesting future work to improve these models for disambiguating the senses of a phrase in context (PS and PSD).

References

Tarik Arici, Hayreddin Ceker, and Ismail Baha Tutar. 2020. Multi-span question answering using span-image network. *preprint*.

Shima Asaadi, Saif Mohammad, and Svetlana Kiritchenko. 2019. Big BiRD: A large, fine-grained, bigram relatedness dataset for examining semantic composition. 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 505–516, Minneapolis, Minnesota. Association for Computational Linguistics.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv preprint*, abs/2004.05150.

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, Roberto Navigli, et al. 2021. Recent trends in word sense disambiguation: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conference on Artificial Intelligence, Inc.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly, Beijing.

Terra Blevins, Mandar Joshi, and Luke Zettlemoyer. 2021. FEWS: Large-scale, low-shot word sense disambiguation with the dictionary. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 455–465, Online. Association for Computational Linguistics.

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

Marine Carpuat and Dekai Wu. 2007a. How phrase sense disambiguation outperforms word sense disambiguation for statistical machine translation. In *Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages: Papers*, Skövde, Sweden.

Marine Carpuat and Dekai Wu. 2007b. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 61–72, Prague, Czech Republic. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *ArXiv preprint*, abs/1803.11175.

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.

Philip Edmonds and Scott Cotton. 2001. SENSEVAL-2: Overview. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5, Toulouse, France. Association for Computational Linguistics.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: the concept revisited. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, pages 406–414. ACM.

Ute Fischer. 1994. Learning words from context and dictionaries: An experimental comparison. *Applied Psycholinguistics*, 15(4):551–574.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92.

Princeton NLP Group. 2022. princeton-nlp/sup-simcse-roberta-large · hugging face. https://huggingface.co/princeton-nlp/sup-simcse-roberta-large. (Accessed on 06/08/2022).

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. *https://spacy.io/*.

Huggingface. 2022a. super_glue · datasets at hugging face. https://huggingface.co/datasets/super_glue/viewer/wic. (Accessed on 06/08/2022).

Huggingface. 2022b. transformers/examples/pytorch/question-answering at main · huggingface/transformers. https://github.com/huggingface/transformers/tree/main/examples/pytorch/question-answering. (Accessed on 06/09/2022).

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

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.

LanguageTool. 2022. Languagetool - online grammar, style & spell checker. https://languagetool.org/. (Accessed on 06/08/2022).

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning dense representations of phrases at scale. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6634–6647, Online. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

William E Nagy, Patricia A Herman, and Richard C Anderson. 1985. Learning words from context. *Reading research quarterly*, pages 233–253.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China. Association for Computational Linguistics.

PiC. 2021a. upwork_annotation_guidelines.pdf. https://drive.google.com/file/d/1UsWXvJLzWzuhoRuaCsla3Ljcxp-vM2KK/view?usp=sharing. (Accessed on 06/08/2022).

PiC. 2021b. upwork_samples.pdf. https://drive.google.com/file/d/1ume4QeQvEPlwajyeNIQjdH_cCI9w3P6f/view?usp=sharing. (Accessed on 06/08/2022).

PiC. 2022. upwork_samples.pdf. https://drive.google.com/file/d/1Lplc_cRla_hbDFzLiJ7ZnO6GOGnRq1gc/view?usp=sharing. (Accessed on 06/08/2022).

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. 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 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Wikimedia Team. 2021a. Api:categories - mediawiki. https://www.mediawiki.org/wiki/API:Categories. (Accessed on 06/08/2022).

Wikimedia Team. 2021b. Wikimedia downloads. https://dumps.wikimedia.org/. (Downloaded on November 1st, 2021).

TensorFlow. 2022. Universal sentence encoder | tensorflow hub. https://www.tensorflow.org/hub/tutorials/semantic_similarity_with_tf_hub_universal_encoder. (Accessed on 08/11/2022).

Peter D Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of artificial intelligence research*, 44:533–585.

Shufan Wang, Laure Thompson, and Mohit Iyyer. 2021. Phrase-BERT: Improved phrase embeddings from BERT with an application to corpus exploration. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10837–10851, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics*, 3:345–358.

Yahoo. 2022a. Webscope | yahoo labs. https://webscope.sandbox.yahoo.com/catalog.php?datatype=l&did=66. (Accessed on 08/10/2022).

Yahoo. 2022b. Webscope | yahoo labs. https://webscope.sandbox.yahoo.com/catalog.php?datatype=l&did=66. (Accessed on 08/10/2022).

Ronghui Yang. 2022. arcface-pytorch/test.py at master · ronghuaiyang/arcface-pytorch. https://github.com/ronghuaiyang/arcface-pytorch/blob/master/test.py. (Accessed on 06/09/2022).

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. 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 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Lang Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4896–4907, Online. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Make a 1-column title

Appendix for:
PiC: A Phrase-in-Context Dataset for Phrase Understanding and Semantic Search

A   Training models on Phrase Similarity

Hyperparameters   We train each *BERT-based classifier* for a maximum of 100 epochs with early stopping monitored on validation accuracy (patience of 10 epochs). We use a batch size of 200 and Adam optimizer with learning rate $\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$.

Training time   On average, with early stopping, training a single model using one V100 GPU takes $\sim$5 and $\sim$8 mins for non-context and context settings, respectively.

B   Training SS models on Phrase Retrieval

We finetune each SS model that consists of a linear layer on top of a pretrained model selected in Sec. 5 to predict the start and end indices of answers (as the common setup in BERT SS models (Devlin et al., 2019; Arici et al., 2020)). The format of a tokenized input is "[CLS] query [SEP] document [SEP]" with maximum sequence length of 4,096 for Longformer$_\text{Base}$ and Longformer$_\text{Large}$ and 512 for the remaining models. If the document exceeds the maximum sequence length, it is split into smaller features for prediction and thus start and end indices with the highest confidence scores are selected.

Hyperparameters   We follow HuggingFace scheme to finetune the SS models for 2 epochs using Adam optimizer with learning rate $\alpha = 0.00003$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. The batch size varies from 1 to 8 for each model: On one V100 GPU, the "base" models can handle 8 examples while the "large" BERT models can only fit 2–4 examples into 16GB of memory. For Longformer$_\text{Large}$, we use an A100 GPU to feed one PR-page example into the model. We take the smallest dev-loss models from the training and report their test-set results.

Training time   On average, training a single SS model for 2 epochs using one A100 GPU takes $\sim$20 mins for base models and $\sim$9.5 hours for Longformer$_\text{Large}$.

C   Data collection

From a Wikipedia dump, we perform a 6-step procedure (summarized in Table A1) for mining a list of mNPs sorted descendingly by their likelihood of containing multiple senses. The most polysemous 19,500 mNPs are then passed to experts for annotation (Sec. 3.2) and others for verification (Sec. 3.3).

Step 1: Download Wiki articles   We download a Wikipedia dump file (Team, 2021b) that contains $\sim$15.78M Wikipedia articles and filter out all empty pages to arrive at $\sim$6.27M non-empty articles.

Step 2: Extract phrases   We use NLTK sentence splitter (Bird et al., 2009) to split each Wikipedia article into multiple sentences. And then we use SpaCy (Honnibal et al., 2020) to extract noun phrases and proper nouns as we do not collect syntactically strict phrases. For each phrase, we remove all preceding and succeeding stopwords (those among the 179 stopwords in NLTK v3.6.5) and non-alphanumeric characters. We remove stopwords because they tend to create more pairs of phrases with lexical overlap,

rendering the phrase similarity task easier. We then remove *unigram* phrases to arrive at $\sim$286.78M mNPs. For example, from "a massive figure", we changed to "massive figure", which would be our final phrase after this step. For each mNP, we construct a 3-tuple (phrase, sentence, metadata), i.e. the phrase, its container sentence, and metadata for identifying the Wikipedia webpage (hereafter, page).

Step 3: Remove phrases of a single context   We further remove all phrases that (1) contain non-ASCII characters (e.g. "phaenná nâsos", which are non-English); and (2) appear only once, i.e. keeping those that occur in multiple sentences since we look for *polysemous* mNPs, which have multiple senses and contexts. After this step, $\sim$17.96M phrases remain.

While some phrases with non-ASCII characters are also commonly used in English (e.g., "déjà vu"), we find only 2.48% of phrases at this stage contain non-ASCII characters, and 29% of them are common in English. In short, we are removing only 0.72% of the English phrases that contain non-ASCII characters in Step 3.

Step 4: Find phrases of polysemous words   To increase the chance of collecting polysemous mNPs, we only keep mNPs that have at least one word in the list of 2,345 unique multiple-sense words of WiC (Huggingface, 2022a), arriving at $\sim$6.5M mNPs, each appearing in ≥ 2 sentences and in ≥ 1 Wikipedia pages. We empirically find that Step 4 is important and substantially increases our chance of finding polysemous mNPs (compared to skipping Step 4).

Step 5: Find phrases in distinct contexts   We observe that a mNP is likely to be polysemous when (a) its context sentences are semantically different; and (b) its context Wikipedia pages are of dissimilar categories (e.g. "massive figure" in finance vs. history; Fig. A8).

To implement this filter, we form all possible triplets (phrase, sentence$_1$, sentence$_2$) from the list of context sentences of each mNP[3]. We compute the cosine similarity of two sentences at the CLS embedding space of a SimCSE (Gao et al., 2021) provided on HuggingFace (Group, 2022). To find triplets where the two sentences are semantically dissimilar, we keep only the triplets where (sentence$_1$, sentence$_2$) has a low cosine similarity, i.e. $\in [-0.3, 0.2]$ and the length difference of the two sentences is < 4 words (as two sentences of substantially different lengths often have a low cosine similarity regardless of their semantic differences). As the result, there are $\sim$600K triplets remaining after this step.

We further re-rank these $\sim$600K descending by the dissimilarity of the lists of Wikipedia categories[4] of the context pages that contain sentence$_1$ and sentence$_2$. That is, we treat each Wikipedia page's comma-separated list of categories as an input text to SimCSE and sort the $\sim$600K descending by the cosine similarity of the resultant embeddings.

Step 6: Select data for annotation   Before asking annotators to label our sorted phrases we perform final filtering by removing proper nouns and phrases whose Wikipedia documents contain missing words.

We perform final filtering to ensure the data given to annotators is in a proper format. That is, from $\sim$600K phrases, we filter down to $\sim$475K phrases by applying two filters: (1) Remove all phrases that are proper nouns (i.e. POS tagging returns PROPN) since proper nouns often refer to a single identity and thus unambiguous; (2) Remove all phrases that have a newline character and all phrases whose context Wikipedia page contains missing words (i.e. errors in the Wikipedia dump).

As the result, we obtain a list of $\sim$475K phrases sorted by their estimate chance of carrying two different senses. After manual inspection, we take the top 19,500 triplets of the format (phrase, page$_1$, page$_2$)—i.e. a phrase p and its two context Wikipedia pages where p is the most likely to have two different senses (e.g., see "massive figure" in Fig. 1a)—and hire linguistic experts to annotate them.

Our manual inspection involves taking 1,000 random triplets and manually reading them. We find that at least $\sim$30% of the 1,000-triplet subset contain a polysemous target phrase p and two Wikipedia pages that give p two unique meanings. We perform this manual inspection repeatedly throughout the

---

[3]For computational tractability, we only keep at most 32 context sentences per mNP where each sentence's length in words is $\in [5, 25]$.

[4]We use the provided Wikipedia API (Team, 2021a) to obtain the categories for each article as the dump file has no category-related information.

process of inventing and refining the data collection process in order to arrive at the final list of steps as presented in this paper.

C.1 Biases in the data collection

While there are many filtering steps in our data collection above, most of them are data cleaning filters that are typically needed in a regular NLP dataset construction.

We recognize that there are *three key filters* in our system that impose strong biases:

1. In Step 4, we use only phrases that contain one word in the WiC. That is, we find Step 4 to substantially increase our chance of finding triplets with a polysemous target phrase. We have added this note in the Data Collection description. It is possible to remove Step 4, but that would require a larger human annotation effort to reach the same 15K labeled triplets.

2. In Step 5, we rely on SimCSE to find target phrases that are placed in two sentences of dissimilar meanings.

3. In Step 5, we rely on SimCSE to find target phrases that are placed in two Wikipedia pages of distinct topics.

D Statistics for search queries in Yahoo Search Query dataset

We analyze 4,496 user queries released in the Yahoo Search Query Log To Entities dataset (Yahoo, 2022b) and use SpaCy tokenizer (Honnibal et al., 2020) to classify them into 4 main categories: Noun phrases, verb phrases, URLs and others. As a result, noun phrases are the most common query type from users with 3,576 queries ($\sim$79.54%) followed by URLs with 675 queries ($\sim$15.01%) while verb phrases and other types are less preferred by users. Moreover, the average length of the real user queries is $\sim$1.60 which is quite close to our PS task with $\sim$2.27.

Table A2: Statistics of Yahoo queries across different query types.

Query type	# queries	Percentage (%)
Noun phrases	3,576	79.54
Verb phrases	148	3.29
URLs	675	15.01
Others	97	2.16
Total	4,496	100.00

E Verification of Phrase Similarity

To enhance the quality of the proposed PiC benchmark, we hire three additional Upwork experts to verify the correctness of PS examples where two phrases are supposed to be *non-equivalent* for negative examples (e.g. massive figure and giant number in Fig. 1b) or *equivalent* for positive examples (e.g. massive figure and huge model in Fig. 1b), and keep an example if it is endorsed by at least two experts (the rest is discarded from PS).

Two Upwork verifiers A_1 and A_2 start checking 5,104 *negative* examples and the third verifier A_3 is responsible for breaking the ties if A_1 and A_2 disagree with each other (see Fig. A1). Both A_1 and A_2 are asked to provide corrections when they do not agree with the labels. As a result, 4,935 out of 5,104 examples are accepted by pairs of (A_1, A_2), (A_1, A_3) or (A_2, A_3), 68 examples incorrect at first but are modified by either A_1 or A_2 and endorsed by A_3. In total, we reject 101 negative examples because there are not at least two experts agreeing with the annotations.

We repeat the same procedure to verify 5,104 *positive* examples. In sum, we retain 5,002 examples including 4,904 examples accepted by pairs of two verifiers and 98 examples incorrect at first but are

modified by either A_1 or A_2 and endorsed by A_3. There are 102 positive examples rejected because there are not at least two experts agreeing with the annotations (Fig. A2).

After this verification round, we collect 5,003 *negative* examples and 5,002 *positive* examples. and randomly exclude 1 negative example to make the dataset balance which results in 10,004 examples in total for PS.

Figure A1: A decision tree describing our verification process for PS that involves three experts. Red, green and blue cells represent Reject, Accept decision and Questions. The numbers of examples for each branch are shown in parentheses.
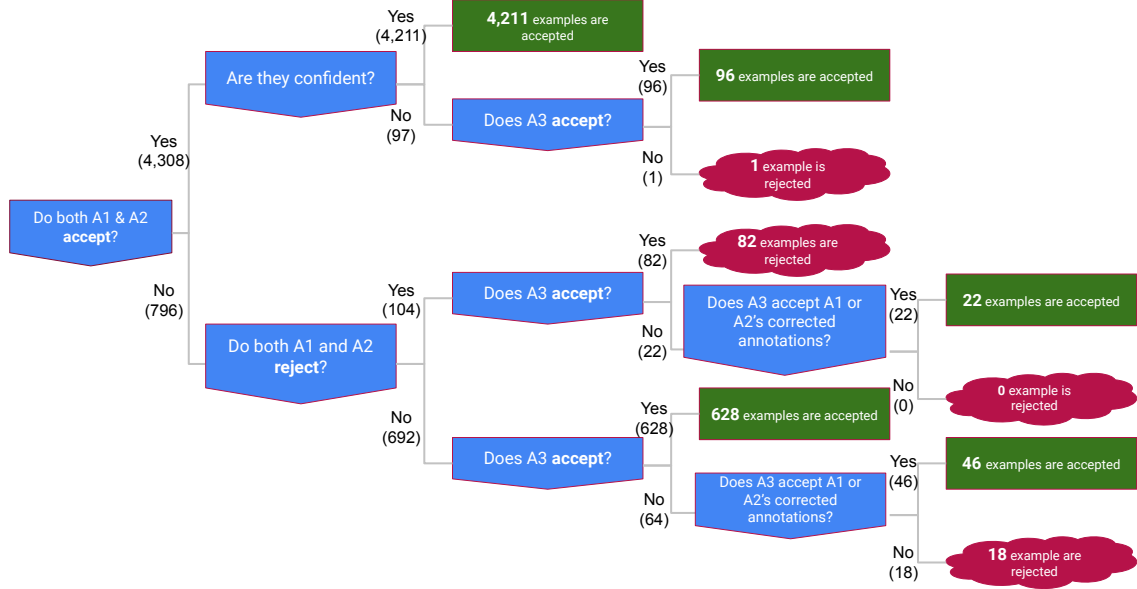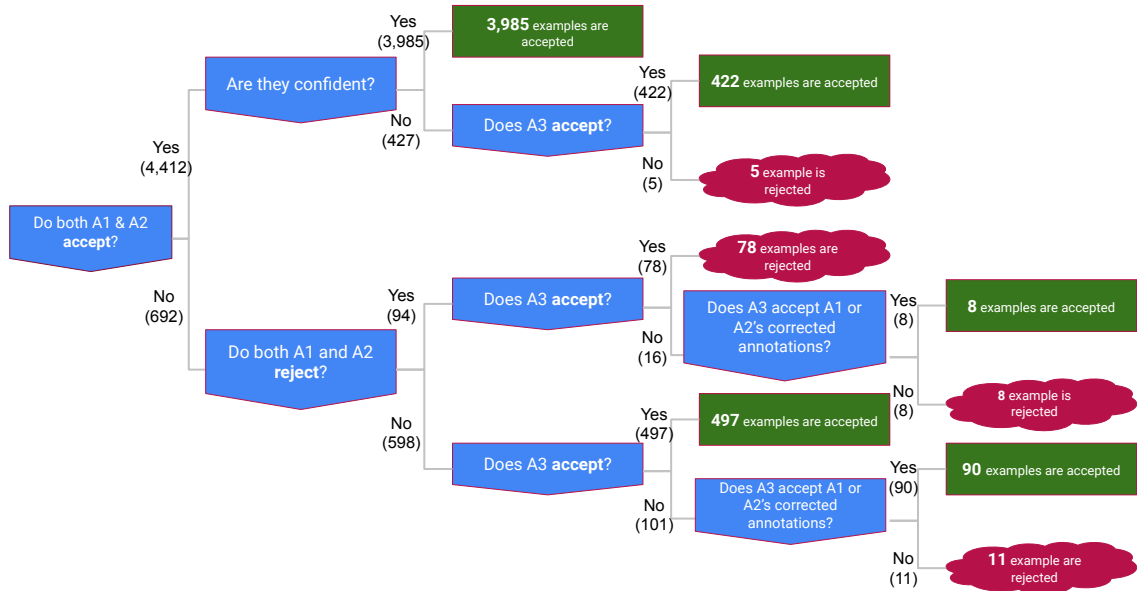
Figure A2: A decision tree describing our verification process for PS that involves three experts. Red, green and blue cells represent Reject, Accept decision and Questions. The numbers of examples for each branch are shown in parentheses.

Table A1: Summary of our 3-stage data construction. $\boldsymbol{p}$, s, m d, q, l denote target phrase, sentence, metadata, document, query, and label, respectively.

	Remaining #	Data type	Description
Sec. 3.1 Data Collection			
Step 1: Download Wiki articles	∼6.27M	articles	Remove ∼9.51M empty articles.
Step 2: Extract phrases	∼286.78M	$(\boldsymbol{p}, s, m)$	Extract noun phrases and proper nouns along with their context sentences from Wikipedia articles.
Step 3: Remove phrases of a single context	∼17.96M	$(\boldsymbol{p}, [s_1, ..., s_n], m)$	For each phrase, gather all sentences where that phrase is used.
Step 4: Find phrases of polysemous words	∼6.5M	$(\boldsymbol{p}, [s_1, ..., s_n], m)$	Filter those phrases that do not contain WiC words.
Step 5: Find phrases in distinct contexts			Sort by X_i and apply filters to find pairs of sentences where their phrase potentially has different meanings.
- Sort and filter by semantic dissimilarity	∼600K	$(\boldsymbol{p}, s_1, s_2, m)$	X_1 : cosine similarity scores of sentences embeddings.
- Sort by domain dissimilarity	∼600K	$(\boldsymbol{p}, s_1, s_2, m)$	X_2 : cosine similarity scores of domain embeddings i.e., use categories of each article to get embeddings.
Step 6: Select data for expert annotation	19,500	$(\boldsymbol{p}, d_1, d_2)$	Remove proper nouns and phrases with missing information and select top 19,500 examples for annotation.
Sec. 3.2 Data Annotations	30,042	$(\boldsymbol{p}, d, q)$	Create a query i.e., paraphrase from the given phrase in each context document.
	15,021	$(\boldsymbol{p}, d_1, d_2, l)$	Create a Yes/No label for each pair of documents.
Sec. 3.3 Verifying Annotations			
Round 1: MTurk verifier	22,496	$(\boldsymbol{p}, d, q)$	Verify queries and Yes/No label by MTurkers.
	10,043	$(\boldsymbol{p}, d_1, d_2, l)$	
Round 2: Upwork verifiers	28,325	$(\boldsymbol{p}, d, q)$	Verify instances rejected in Round 1.
	13,413	$(\boldsymbol{p}, d_1, d_2, l)$	

F Quantitative results on PR-page

Table A3: **Ranking** accuracy (%) on **PR-page** using the state-of-the-art pretrained phrase embeddings (a) and those finetuned on PR-pass via SS-style training (b).

Model	Phrase				Phrase + Context			
	Top-1	Top-3	Top-5	MRR@5	Top-1	Top-3	Top-5	MRR@5
(a) Pre-trained embeddings								
BERT (Devlin et al., 2019)	20.70	34.30	41.00	28.20	**35.40** (+14.70)	**52.10**	**59.10**	**44.50**
USE-v5 (Cer et al., 2018)	**32.20**	**52.70**	**60.80**	**43.20**	n/a	n/a	n/a	n/a
(b) PR-pass-trained SS models' phrase embeddings								
PhraseBERT (Wang et al., 2021)	**49.40**	**69.40**	**76.70**	**60.10**	14.70	21.60	26.10	18.70
SimCSE (Gao et al., 2021)	44.20	66.60	73.50	55.70	24.60	37.80	43.20	31.70

G Finetuning on PSD does not substantially improve accuracy

As PSD has only 4,858 examples, we use all examples for testing in Sec. 5.5 and find the best PR-trained SS models to perform poorly. To further understand the challenge of PSD, here, we ask:

Q: *How much does training on PR-pass and finetuning on PSD improve accuracy on PSD?*

Experiment We take the PR-pass-trained SS models and further finetune them on a subset of PSD to measure how training directly on PSD improves SS models. We split PSD into 1,438/500/3,000 examples for train/dev/test sets, respectively, and finetune the PR-pass-trained SS models on this PSD train set. For comparison with the results in Sec. 5.4, we use the same set of hyperparameters as when finetuning on PR-pass in Sec. 5.4. Below, we report the test-set results of the lowest dev-loss models.

Results On the PSD-3K test set, all models perform poorly at a mean EM score of 55.14% (Table A4a; mean). Interestingly, finetuning the original models using the 1,938 examples (hereafter, PSD-2K) instead of PR-pass decreases accuracy, on average by -6.51 points. An explanation is that 1,438 PSD training examples are too few for the finetuning to be effective. Indeed, finetuning the PR-pass-trained SS models further on PSD-2K increases the scores for all models by +9.10 on average (Table A4c; mean). The best model is Longformer$_{Base}$ (Beltagy et al., 2020) (Table A4; 71.10 EM), which is still substantially lower than the human upperbound of 95%.

Table A4: Performance of **SS models** on 3,000 PSD **test** examples. (a) and (b) models are **finetuned** only on PR-pass and 1,938 PSD examples (PSD-2K), respectively. (c) models are finetuned on PR-pass first and then finetuned on PSD-2K. All models are "base" unless otherwise specified. The definitions of EM+loc and F$_1$+loc are in Table 6's caption.

Models finetuned on	(a) PR-pass		(b) PSD-2K		(c) PR-pass + PSD-2K	
	EM+loc	F$_1$+loc	EM+loc	F$_1$+loc	EM+loc	F$_1$+loc
PhraseBERT (Wang et al., 2021)	51.00	51.15	35.43 (-15.57)	36.02	56.53 (+5.53)	56.81
BERT (Devlin et al., 2019)	54.53	54.75	44.33 (-10.20)	45.28	63.83 (+9.30)	64.14
BERT$_{Large}$ (Devlin et al., 2019)	54.77	54.99	54.07 (-0.70)	54.82	67.13 (+12.36)	67.36
SpanBERT (Joshi et al., 2020)	52.27	52.37	44.67 (-7.60)	45.35	69.93 (+17.66)	70.14
SentenceBERT (Reimers and Gurevych, 2019)	52.27	52.41	38.63 (-13.64)	39.31	58.93 (+6.66)	59.21
SimCSE (Gao et al., 2021)	53.47	53.59	43.67 (-9.80)	44.38	60.60 (+7.13)	60.80
Longformer (Beltagy et al., 2020)	**62.47**	**62.58**	61.97 (-0.50)	62.69	**71.10** (+8.63)	**71.30**
Longformer$_{Large}$ (Beltagy et al., 2020)	60.33	60.42	**66.27** (+5.94)	**67.10**	65.87 (+5.54)	66.10
mean	55.14	55.28	48.63 (-6.51)	49.37	64.24 (+9.10)	64.48
± std	4.10	4.08	11.03	11.08	5.23	4.13

H SS-style training improves *non*-contextualized but not contextualized phrase embeddings

As the SS models trained on PR-pass and PR-page perform impressively (Sec. 5.4), almost $1.5\times$ better than the ranking models that are based on pre-trained embeddings, an interesting question is:

Q: *Does SS training also improve contextualized phrase embeddings?*

This is important to understand because the impressive SS-models' performance gain may come from the extra linear-classification layer (not necessarily from the finetuned embeddings).

Experiment We extract the phrase embeddings (both non-contextualized and contextualized) from the PR-pass-trained SS models from Sec. 5.4 (i.e. discarding the classification layer) and test them in the PR-pass ranking experiments (as in Sec. 5.3).

Results After finetuning on PR-pass, the *non*-contextualized phrase embeddings improve substantially for most models at an average gain of +16.61 in top-1 accuracy (e.g., PhraseBERT top-1 accuracy increases from 36.62% to 59.02%; Table A5b). This result shows that training on PR-pass improves non-contextualized phrase embeddings. In stark contrast, the ranking scores of *contextualized* phrase embeddings drop significantly, -11.95 points on average (Table A5c), compared to before finetuning on PR-pass.

In sum, we are observing a consistent trend that the contextualized phrase embeddings of the original pre-trained BERT (both "base" and "large") are remarkably beneficial for retrieval (i.e. PR). However, after finetuning, e.g. on PR-pass or using other techniques (e.g. in PhraseBERT or SentenceBERT), such benefits of leveraging context disappear. Aligned with Yu and Ettinger (2020), we find that incorporating context effectively into phrase embeddings is an open research challenge.

Table A5: **Ranking** accuracy (%) on **PR-pass** using the state-of-the-art pretrained phrase embeddings (a) and those finetuned on PR-pass via SS-style training (b). See Appendix F for the results on PR-page. Δ (e.g. -3.62) denotes the differences between the Top-1 accuracy in the contextualized ("Phrase + Context") vs. the non-contextualized ("Phrase") setting.

Model	Phrase				Phrase + Context			
	Top-1	Top-3	Top-5	MRR@5	Top-1 (Δ)	Top-3	Top-5	MRR@5
(a) Pre-trained embeddings								
PhraseBERT (Wang et al., 2021)	36.62	66.96	75.90	52.20	33.00 (-3.62)	49.60	56.70	41.90
BERT (Devlin et al., 2019)	29.80	47.90	55.40	39.50	**47.44** (+17.64)	**65.78**	**73.30**	**57.30**
BERT$_{Large}$ (Devlin et al., 2019)	23.76	38.52	45.40	31.70	**42.80** (+19.04)	**58.90**	**64.90**	**51.30**
SpanBERT (Joshi et al., 2020)	20.88	31.04	35.20	26.40	14.40 (-6.48)	30.46	39.80	23.40
SentenceBERT (Reimers and Gurevych, 2019)	22.30	50.64	60.60	36.80	**25.14** (+2.84)	39.52	46.20	32.90
SimCSE (Gao et al., 2021)	28.10	53.70	64.60	41.60	**32.40** (+4.30)	53.44	62.80	**43.70**
USE-v5 (Cer et al., 2018)	**43.36**	**70.12**	**78.90**	**57.30**	n/a	n/a	n/a	n/a
DensePhrase (Lee et al., 2021)	32.24	51.30	60.50	42.60	31.50 (-0.74)	46.30	53.80	39.70
(b) PR-pass-trained SS models' phrase embeddings								
PhraseBERT (Wang et al., 2021)	**59.02**	**81.58**	**87.90**	**70.60**	24.98 (-34.04)	37.78	43.90	32.00
BERT (Devlin et al., 2019)	50.10	66.16	71.40	58.60	20.34 (-29.76)	31.40	37.10	26.50
BERT$_{Large}$ (Devlin et al., 2019)	32.70	42.40	45.90	37.80	11.40 (-21.30)	17.00	20.50	14.60
SpanBERT (Joshi et al., 2020)	15.22	22.88	26.60	19.40	8.92 (-6.30)	13.56	16.60	11.60
SentenceBERT (Reimers and Gurevych, 2019)	53.14	74.86	80.70	64.20	20.12 (-33.02)	30.04	34.90	25.60
SimCSE (Gao et al., 2021)	50.96	76.70	83.40	64.00	37.70 (-13.26)	52.38	58.90	45.60
(c) Differences between after vs. before finetuning, i.e. the 6 models in (b) vs. those in (a)								
mean differences	+16.61				-11.95			

I Qualitative examples for PS, PR-pass, PR-page and PSD

PS example. Groundtruth: "positive"	
P_1	moderate speed
P_2	steady pace
S_1	Deforestation due to logging and land conversion has likely caused the population to decline at a <mark>moderate speed</mark>.
S_2	Deforestation due to logging and land conversion has likely caused the population to decline at a <mark>steady pace</mark>.

Figure A3: PhraseBERT-based classifier **correctly** predicts "positive" given two phrases P_1 and P_2 with and without the presence of context S_1 and S_2. Here, to humans, the phrases are non-polysemous and have the same meaning.

PS example. Groundtruth: "negative"	
P_1	greatest emphasis
P_2	highest stress
S_1	However, the rock art had the <mark>greatest emphasis</mark> on domesticated cattle.
S_2	However, the rock art had the <mark>highest stress</mark> on domesticated cattle.

Figure A4: PhraseBERT-based classifier **correctly** predicts "negative" given two phrases P_1 and P_2 with and without the presence of context S_1 and S_2. Here, to humans, the two phrases are non-ambiguously carrying different meanings.

PS example. Groundtruth: "positive"	
P_1	unique image
P_2	uncommon style
S_1	Bayliss has been praised for her <mark>unique image</mark> and tendency to change up songs.
S_2	Bayliss has been praised for her <mark>uncommon style</mark> and tendency to change up songs.

Figure A5: PS case that requires context to determine similarity. Without context, a PhraseBERT-based classifier incorrectly thinks P_1 and P_2 are different. Yet, it changes the prediction to "positive", i.e. thinking two phrases have the same meaning, when the context is taken into account.

PS example. Groundtruth: "negative"	
P$_1$	permanent post
P$_2$	stable location
S$_1$	His assistant, John Carver took over as caretaker manager, managing one win, but was not considered for the permanent post, and left in September 2004.
S$_2$	His assistant, John Carver took over as caretaker manager, managing one win, but was not considered for the stable location, and left in September 2004.

Figure A6: PS case that requires context to determine similarity. Without context, PhraseBERT-based classifier incorrectly thinks P$_1$ and P$_2$ carry the same meaning. Yet, it correctly changes the prediction to "negative" when the context is taken into account.

PSD example.	
d	Bubble memory is a type of non-volatile computer memory that uses a thin film of a magnetic material to hold small magnetized areas, known as "bubbles" or "domains", each storing one bit of data. The material is arranged to form a series of parallel tracks that the bubbles can move along under the action of an external magnetic field. The bubbles are read by moving them to the edge of the material where they can be read by a conventional magnetic pickup, and then rewritten on the far edge to keep the memory cycling through the material. In operation, bubble memories are similar to delay line memory systems. Bubble memory started out as a promising technology in the 1970s, offering memory density of an order similar to hard drives but performance more comparable to core memory while lacking any moving parts. This led many to consider it a contender for a "universal memory" that could be used for all storage needs. The introduction of dramatically faster semiconductor memory chips pushed bubble into the slow end of the scale, and equally dramatic improvements in hard drive capacity made it uncompetitive in price terms. Bubble memory was used for some time in the 1970s and 80s where its non-moving nature was desirable for maintenance or shock-proofing reasons. The introduction of Flash RAM and similar technologies rendered even this niche uncompetitive, and bubble disappeared entirely by the late 1980s. History. Precursors. The Inkerman stone, of which the building is made, was mined near Sevastopol and transported by barges. No convenient mooring facilities existed at that time, so the barges had to anchor in the harbor and the load was moved to the shore by boats and then transported to the construction site across the steppe. During the first year of construction, the builders concentrated on the basic structure at the expense of various facilities and decorations. At the end of 1816, the lighthouse looked like a conic 36-metre-high stone tower with a wooden 3.3-metre-high decagonal lantern. The lighthouse became operational in 1817 after its lighting system had been repaired. Three houses were built next to the tower to accommodate the lighthouse personnel and for storage needs. However, cold and humid winters of the Tarkhanut Peninsula, however, made these houses nearly unsuitable for living. In 1862, the lighting system was upgraded, and the spread of light reached 12.4 miles. In 1873, the construction resumed along with cleaning efforts of the surrounding areas. The building was finished and painted white. In 1876, an additional telegraph spot was built near the tower.
q_1	storehouse purposes Groundtruth: storage needs & Prediction: storage needs (confidence: 0.99)
q_2	data caching Groundtruth: storage needs & Prediction: storage needs (confidence: 0.99)

Figure A7: Given document d, our Longformer$_{Large}$ SS model trained on PR-pass correctly retrieves storage needs in the second paragraph for the query q_1 "storehouse purposes" but *fails* to retrieve the answer when the query q_2 is "data caching". The predicted answer for q_2 should be storage needs (i.e. in the first passage) since this phrase relates to caching data digitally in computers while storage needs refers to physically storing objects.

	PSD example.
d	In the libretto, Delilah is portrayed as a seductive "femme fatale", but the music played during her parts invokes sympathy for her. The 1949 biblical drama "Samson and Delilah", directed by Cecil B. DeMille and starring Victor Mature and Hedy Lamarr in the titular roles, was widely praised by critics for its cinematography, lead performances, costumes, sets, and innovative special effects. It became the highest-grossing film of 1950, and was nominated for five Academy Awards, winning two. According to "Variety", the film portrays Samson as a stereotypical "handsome but dumb hulk of muscle". Samson has been especially honored in Russian artwork because the Russians defeated the Swedes in the Battle of Poltava on the feast day of St. Sampson, whose name is homophonous with Samson's. The lion slain by Samson was interpreted to represent Sweden, as a result of the lion's placement on the Swedish coat of arms. In 1735, C. B. Rastrelli's bronze statue of Samson slaying the lion was placed in the center of the great cascade of the fountain at Peterhof Palace in Saint Petersburg. Samson is the emblem of Lungau, Salzburg and parades in his honor are held annually in ten villages of the Lungau and two villages in the north-west Styria (Austria). During the parade, a young bachelor from the community carries a `massive figure` made of wood or aluminum said to represent Samson. The tradition, which was first documented in 1635, was entered into the UNESCO list of Intangible Cultural Heritage in Austria in 2010. Samson is one of the giant figures at the "Ducasse" festivities, which take place at Ath, Belgium. On September 22, 2015, Honda announced that they had sold over 1 million Activas in five months in the Indian market, from April to August. Honda launched their 5th generation of Honda Activa in 2018, and the sixth-generation Honda Activa 6G have been launched in India with prices starting at 63,912 (ex-showroom, Delhi). Milestones. In April, 2014, "The Economic Times" reported the Honda Activa to be the best selling two wheeler in India, outselling the Hero Splendor. During the month of September 2013, 141,996 Honda Activa scooters were sold, nearly equal to Honda's entire annual sales in North America. The 110cc Activa is the company's biggest seller, by far. It is responsible for over 2,00,000 sales units each month. In November 2018, HMSI crossed the 2.5 crore sales mark in the scooter segment. It has become the first company to reach this milestone and the biggest contributor to this `massive figure` is the Honda Activa. It took Honda 13 years to achieve the one crore sales figure, but it managed to add another crore in the span of just three years. It then went on to achieve the next 50 lakh in just one year.
q_1	huge model Groundtruth: `massive figure` & Prediction: `massive figure` (confidence: 0.99)
q_2	giant number Groundtruth: `massive figure` & Prediction: `massive figure` (confidence: 0.99)

Figure A8: Given document *d*, Longformer$_{Large}$ model trained with SS approach on PR-pass correctly retrieves `massive figure` in the second paragraph for the query q_2 "giant number" but *fails* to retrieve the answer when the query q_1 is "huge model". The predicted answer for q_1 should be `massive figure` in the first passage since this phrase relates to a physical shape instead of a number.

	PSD example.
d	Eva held ambitions to replace Hortensio Quijano for the 1951 election, although her poor health kept her from this. Nonetheless many were concerned that her agenda would be pushed through. In march of 1951 the government arrested several retired army officers due to their dissent and disapproval of Perón's administration. This raised tensions among the rest of the army, although action did not occur. By September tensions had risen among the military due to the `unrivalled power` of the Peronist regime. On September 28, 1951, during the election, Menéndez led the military uprising in an attempt to overthrow the government. He led a core of officers, commanding a division, and left Campo de Mayo bound for the Casa Rosada. Resolve for the uprising, especially among the non-commissioned officers and enlisted men, was not strong enough. They were not prepared to fight their own countrymen. The uprising was over as soon as opposition was encountered, almost completely bloodless. Perón admired the loyalty of the troops and pardoned all those involved. The design uses a similar standard to the JVX in terms of distortion reduction with crossbraces and 27 cells but that's where the similarity ends. Petra was built from the ground up with entirely new panel shaping and trim. Petra has a highly elliptical planform and very high sweep. NZ Aerosports say she has a high roll rate, a long recovery arc and high maximal glide ratio. She is said to deliver `unrivalled power` in the turn, plane out and flare. Petra has a long list of World Records, National and International titles to back that up. She had an impressive debut at the PD Big Boy Pants event in July 2011, with Nick Batsch setting a new distance world record of 222.45m (729ft). One month later Nick took out the Pink Open in Klatovy and the FAI World Cup also; first in distance, speed and overall. He also won the 2011 US CP nationals on Petra. Patrick Boulongne came 2nd in the European Championships and 6th overall at the World Cup with Petra in his first competition with her. He went on to win the 2011 French Canopy Piloting Nationals.
q_1	incomparable energy Groundtruth: `unrivalled power` & Prediction: `unrivalled power` (confidence: 0.99)
q_2	indomitable strength Groundtruth: `unrivalled power` & Prediction: `unrivalled power` (confidence: 0.99)

Figure A9: Given document *d*, Longformer$_{Large}$ model trained via the SS approach on PR-pass correctly retrieves `unrivalled power` in the first paragraph for the query q_2 "indomitable strength" but *fails* to retrieve the answer when the query q_1 is "incomparable energy". The predicted answer for q_1 should be `unrivalled power` in the second passage since the second passage changes "unrivalled power" meaning to a competition strength instead of military power.

	PR-pass example. Groundtruth: `common thought`
d	As the medical corps grew in size there was also specialization evolving. Physicians surfaced that specialized in disease, surgery, wound dressing and even veterinary medicine. Veterinary physicians were there to tend to livestock for agricultural purposes as well as combat purposes. The Cavalry was known for their use of horses in combat and scouting purposes. Because of the type of injuries that would have been commonly seen, surgery was a somewhat common occurrence. Tools such as scissors, knives and arrow extractors have been found in remains. In fact, Roman surgery was quite intuitive, in contrast to `common thought` of ancient surgery. The Roman military surgeons used a cocktail of plants, which created a sedative similar to modern anesthesia. Written documentation also showed surgeons would use oxidation from a metal such as copper and scrape it into wounds, which provided an antibacterial effect; however, this method was most likely more toxic than providing an actual benefit. Doctors had the knowledge to clean their surgical instruments with hot water after each use. Wounds were dressed, and dead tissue was removed when bandages were changed.
q	prevalent theory
R	0.882 `common thought` 0.855 common thought of 0.702 fact 0.698 to common thought 0.675 common occurrence

Figure A10: A **ranking** model based on the phrase embeddings of the PR-pass-trained PhraseBERT SS model correctly ranks and retrieves the most semantically relevant answer "common thought" as the top-1 prediction in the retrieval list R for the query "prevalent theory" in a PR-pass example (which contains a document *d* and a query *q*).

PR-page example.	Groundtruth: continued risk

d	... Following a United Nations agreement between Indonesia and Portugal, a UN-supervised referendum held on 30 August 1999 offered a choice between autonomy within Indonesia and full independence. The people of East Timor voted overwhelmingly for independence. An Australian-led and Indonesian-sanctioned peacekeeping force, INTERFET, was sent into the territory to restore order following a violent 'scorched-earth' policy carried out by pro-integration militia and supported by elements of the Indonesian military. In response to Australia's involvement, Indonesia abrogated the 1995 security pact, asserting that Australia's actions in East Timor were inconsistent with 'both the letter and spirit of the agreement'. Official meetings were cancelled or delayed, including the Indonesia-Australia Ministerial Dialogue, which would not reconvene until March 2003. INTERFET was later replaced by a UN force of international police, UNTAET, which formed a detachment to investigate alleged atrocities. "Tampa" affair and the War on Terror. The relationship came under strain in August 2001 during the "Tampa" affair, when Australia refused permission for the Norwegian freighter ship MV "Tampa" to enter Australian waters while carrying Afghan asylum seekers that it had rescued from a distressed fishing vessel in international waters. The Indonesian Search and Rescue Agency did not immediately respond to requests from Australia to receive the vessel. When the ship entered Australian territorial waters after being refused permission, Australia attempted without success to persuade Indonesia to accept the asylum seekers. Norway also refused to accept the asylum seekers and reported Australia to international maritime authorities. The incident prompted closer coordination between Indonesian and Australian authorities, including regional conferences on people smuggling, trafficking in persons and other transnational crime. In 2002, a terrorist attack in Kuta, Bali killed 202 people, including 88 Australians, and injured a further 240. Jemaah Islamiyah, a violent Islamist group, claimed responsibility for the attack, allegedly in retaliation for Australia's support for East Timorese independence and the War on Terror. A subsequent attack in 2005 resulted in the deaths of a further 20 people, including 15 Indonesians and 4 Australians. The 2003 Marriott Hotel bombing was also perceived as targeted at Western interests in Indonesia; Al Qaeda claimed the attack was carried out by a Jemaah Islamiyah suicide bomber in response to actions of the United States and its allies, including Australia. A 2004 attack on the Australian embassy in Jakarta by Jemaah Islamiyah resulted in the deaths of nine Indonesians. The following year, Indonesian diplomatic and consular premises in Australia received a number of hoax and threat messages. Since then, both the United States and Australian governments have issued warnings against travel to Indonesia, advising their citizens of a continued risk of attacks. These incidents prompted greater cooperation between law enforcement agencies in the two countries, building on a 1999 agreement on drug trafficking and money laundering. The Australian Federal Police's Jakarta Regional Cooperation Team provided assistance to the Indonesian National Police, and has contributed to the Jakarta Centre for Law Enforcement Cooperation. This relationship has attracted criticism, particularly following the arrest and sentencing of the Bali Nine, a group of nine Australians arrested in Denpasar while attempting to smuggle heroin from Indonesia to Australia. The 2005 conviction of Schapelle Corby for attempting to smuggle drugs to Bali also attracted significant attention in the Australian media. The 2004 Indian Ocean earthquake prompted a significant humanitarian response from Australia, including a $1 billion aid package from the federal government, a further $17.45 million contribution from state and territory governments, and the commitment of 900 Australian Defence Force personnel to relief efforts in northern Sumatra and Aceh. A telethon broadcast on Australia's three major commercial television networks called "" generated pledges of more than $10 million, contributing to total private aid of $140 million. The Eighth "Australia-Indonesia Ministerial Forum" (AIMF) was held in Bali on 29 June 2006 and was attended by five Australian and eleven Indonesian ministers. A key outcome was support for the conclusion of a security agreement, later realised as the Lombok Agreement, providing a framework for the development of the security relationship by the end of 2006 on defence, law enforcement, counter-terrorism, intelligence, maritime security, aviation safety, WMD non-proliferation, and bilateral nuclear cooperation for peaceful purposes. Australia-Indonesia-East Timor Trilateral Ministerial Meetings occurred three times to September 2006. Recent relations. 2010 President Susilo Bambang Yudhoyono visited Australia in April 2010, and became the second Indonesian leader to address federal parliament: Finally, I look forward to a day in the near future. The day when policy makers, academicians, journalists and other opinion leaders all over the world take a good look at the things we are doing so well together. And they will say: these two used to be worlds apart. But they now have a fair dinkum of a partnership. ...
q	sustained threat
R	0.830 threat .
	0.802 potential threat
	0.800 threat reached
	0.787 threat as
	0.787 threat to

Figure A11: A **ranking** model based on the non-contextualized embeddings of USE-v5 fails to retrieve the correct answer "continued risk" for the query "sustained threat" in the PR-page example (which contains a document *d* and a query *q*). The top-5 phrases retrieved (R) contains the word "threat" but have no identifier conveying the "continued" or 'sustained' sense. Here, the Wikipedia page is truncated to fit into a single manuscript page.

J   Verifying annotations

There are two common methods for evaluation of dataset quality: (1) Verify only a small, random subset (Pilehvar and Camacho-Collados, 2019) to estimate the quality of the full dataset or (2) verifying the entire dataset with multiple annotators and use the inter-annotator agreement (IAA) to control quality (Bowman et al., 2015; Kwiatkowski et al., 2019). The first approach for approximation is budget-friendly but it remains unknown whether the rest of examples are at high quality, while IAA is more desired but annotating thousands of instances can be prohibitively slow and costly.

We propose a **hybrid approach** to evaluate (leveraging both linguistic experts and non-experts) and ensure high quality for 30,042 queries and 15,021 Yes/No answers at lower cost compared to IAA via two rounds:

1. First, we ask around 1,000 highly qualified freelancers on Amazon Mechanical Turk (MTurk verifiers) to verify whether the *query* annotated by our Upwork annotators is interchangeable i.e. has the same meaning with the given *phrase* in *paragraph*. To verify Yes/No answers, MTurk verifiers need to read two short paragraphs containing the same phrase like Upwork annotators to make decisions. We do not show answers to the MTurk verifiers to avoid biases.

2. Second, we continue hiring 5 Upwork verifiers who are writing experts to double-check those instances rejected by MTurk verifiers from the previous round and only discard an example if the Upwork verifiers agree with MTurk verifiers.

J.1   Round 1: Verification by MTurk non-experts

We use AMT platform to recruit more than 1,000 MTurk verifiers. Also, we use Gorrila (`gorilla.sc`) to develop user interface to collect answers from participants because (1) Gorilla provides easy-to-use tools to build graphical interface, (2) it is straightforward to monitor and discard results from unqualified participants and (3) we can easily share the experiment with MTurk verifiers via a link. Per 30 verified answers in around ∼20 minutes, the verification process costs us $5.6 (AMT fees included) and 1 token to Gorilla to a single MTurk verifier.

Participants are given detailed instructions along with 5 practice samples to get familiar with the task (Fig. A12). They need to pass an evaluation checkpoint including 6 questions randomly sampled from our verified question bank in order to start working with sets of 30 questions. With this approach, all examples in the dataset are verified once and as a result, 22,496/30,042 queries (∼74.88%) and 10,043/15,021 Yes/No answers (∼66.86%) accepted by MTurkers are considered high quality since they are annotated by a writing expert and confirmed by a qualified English native speaker. The remaining 7,546 queries and 4,978 Yes/No answers rejected that are passed to another group of 5 writing experts for confirmation.

J.2   Round 2: Verification by Upwork experts

We hired another set of 5 writing experts from Upwork (Upwork verifiers) with an hourly rate of $25-40/hour to verify 12,524 examples rejected by MTurk verifiers, i.e., at an average cost of approximately $0.26 per example. See a sample assignment given to an Upwork expert in (PiC, 2022).

We rely on IAA to decide whether to accept or reject an example. Specifically, we use the *same question types* as shown to MTurk verifiers in the previous step and see whether these Upwork verifiers agree with the Upwork annotators to keep this example or with MTurk verifiers to reject it. We find that the agreement between the first- and third-round annotators are 5,829 (out of 7,546) paraphrases and 3,370 (out of 4,978) Yes/No answers in total and thus the total high-quality queries and Yes/No answers we achieve are 28,325 and 13,413, respectively.

Instructions

In this study, you will work with 2 types of Yes/No questions:

Type 1: Same paragraphs, different phrases

> **P1**: 2020. The cancellation of the conference in 2020 due to the COVID-19 pandemic led to an online series of lectures entitled Skeptical Inquirer Presents. These sessions included presentations by well known figures in the skeptical community and opportunities for viewers to ask questions. Conference details.
>
> **P2**: 2020. The cancellation of the conference in 2020 due to the COVID-19 pandemic led to an online series of lectures entitled Skeptical Inquirer Presents. These sessions included presentations by Omicron variant in the skeptical community and opportunities for viewers to ask questions. Conference details.
>
> **Question**: In both passages, are well known figures and Omicron variant interchangeable (i.e., having the same meaning)?

Type 2: Different paragraphs, same phrases

> **P1**: In 1949, a memorial for the 442 Regimental Combat Unit was incorporated and remembered for the Japanese-American soldiers who had fallen during World War II. Every year during the Obon festival, families gather to upkeep their relatives tombstones and to visit the spirits. Biddy Mason, nurse and philanthropist, was one of the well known figures to be buried at the cemetery in 1891. There is a section called the "Showmen's Rest" in which 400 carnival workers and circus performers are buried by a memorial that is decorated with a lion.
>
> **P2**: 2020. The cancellation of the conference in 2020 due to the COVID-19 pandemic led to an online series of lectures entitled Skeptical Inquirer Presents. These sessions included presentations by well known figures in the skeptical community and opportunities for viewers to ask questions. Conference details.
>
> **Question**: In both passages, does well known figures have the same meaning?

Continue

(a) Detailed instructions given to MTurkers

Your progress: 0/36 trials completed

P1: HotDocs transforms documents and graphical (PDF) forms into document-generation templates and deploys these templates to various server environments. Document modeling in HotDocs can range from variable insertions to the formation and insertions of complex, computed variables. Business logic consisting of IF/THEN statements and REPEAT loops can be built into the template to control the inclusion or exclusion of language blocks. HotDocs includes a variety of other scripting instructions and sets of pre-packaged functions using boolean logic.

P2: HotDocs transforms documents and graphical (PDF) forms into document-generation templates and deploys these templates to various server environments. Document modeling in HotDocs can range from variable insertions to the formation and insertions of complex, computed variables. Business logic consisting of IF/THEN statements and REPEAT loops can be built into the template to control the inclusion or exclusion of linguistic sections. HotDocs includes a variety of other scripting instructions and sets of pre-packaged functions using boolean logic.

Q: In both passages, are language blocks and linguistic sections interchangeable (i.e., having the same meaning)?

No   Yes

Note: Yes/No buttons will be displayed in 20 seconds. Please read the given contents carefully before answering question.

(b) Upon completion of training stage, MTurkers need to correctly answer the first 5 out of 6 questions to be invited to verify annotations from Upwork experts.

Training

P1: This library was supposedly founded in 1945, but has started work in current object in 1947. During 1953-1956 it has played the role of the national library since the Kosovo National Library was closed. Academy of Sciences and Art is a necessary institution for the education system that is placed in Pristina. This institution was founded in 1975 as the Association of Science and Arts of Kosovo.

P2: This library was supposedly founded in 1945, but has started work in current object in 1947. During 1953-1956 it has played the role of the national library since the Kosovo National Library was closed. Academy of Sciences and Art is a big supermarket for the education system that is placed in Pristina. This institution was founded in 1975 as the Association of Science and Arts of Kosovo.

Question 3: In both passages, are necessary institution and big supermarket interchangeable (i.e., having the same meaning)?

No   Yes

(c) Verification of paraphrases via type-1 question.

Training

P1: Mercury is poured over the dirt with bare hands. The method leaves much gold undetected, and therefore some miners are using metal detectors. The mercury pollution in the area is both an environment problem, and a health hazard. Most of the gold gets transported to the North of Paramaribo where the gold buyers are located.

P2: Tools such as KLEE, Cloud9, and Otter take this approach by implementing models for file system operations, sockets, IPC, etc. Forking the entire system state. Symbolic execution tools based on virtual machines solve the environment problem by forking the entire VM state. For example, in S2E each state is an independent VM snapshot that can be executed separately.

Question 1: In both passages, does environment problem have the same meaning?

No   Yes

(d) Verification of Yes/No labels via type-2 question.

Training

P1: This library was supposedly founded in 1945, but has started work in current object in 1947. During 1953-1956 it has played the role of the national library since the Kosovo National Library was closed. Academy of Sciences and Art is a necessary institution for the education system that is placed in Pristina. This institution was founded in 1975 as the Association of Science and Arts of Kosovo.

P2: This library was supposedly founded in 1945, but has started work in current object in 1947. During 1953-1956 it has played the role of the national library since the Kosovo National Library was closed. Academy of Sciences and Art is a big supermarket for the education system that is placed in Pristina. This institution was founded in 1975 as the Association of Science and Arts of Kosovo.

Question 3: In both passages, are necessary institution and big supermarket interchangeable (i.e., having the same meaning)?

NO. The phrase necessary institution means a required establishment of the livery stable which does not totally refer to a big supermarket.

Yes   ❌   Continue

(e) Feedback is given when MTurkers give a wrong answer.

Training

P1: Mercury is poured over the dirt with bare hands. The method leaves much gold undetected, and therefore some miners are using metal detectors. The mercury pollution in the area is both an environment problem, and a health hazard. Most of the gold gets transported to the North of Paramaribo where the gold buyers are located.

P2: Tools such as KLEE, Cloud9, and Otter take this approach by implementing models for file system operations, sockets, IPC, etc. Forking the entire system state. Symbolic execution tools based on virtual machines solve the environment problem by forking the entire VM state. For example, in S2E each state is an independent VM snapshot that can be executed separately.

Question 1: In both passages, does environment problem have the same meaning?

NO. The phrase environment problem in the first passage mentions an issue of the physical environment in which we are living while in the second passage, it means the issue of the digital environment of an operating system. Thus, it should have different meanings.

No   ✓   Continue

(f) or even a right answer.

Figure A12: Gorilla layouts shown to MTurkers to verify annotations in the first round.

K Data Sheet

We follow the documentation template provided by Gebru et al. 2021 (Gebru et al., 2021).

K.1 Motivation

For what purpose was the dataset created? Understanding phrases in context plays a vital role in solving many Natural Language Understanding (NLU) tasks such as question answering or reading comprehension. While there are *word-sense* disambiguation datasets like WiC, no such benchmarks exist for *phrases*. Existing phrase benchmarks compare only phrases without context and some of them contain numerous phrase pairs that have lexical overlap. The major drawback is no human annotation of how a phrase's meaning changes w.r.t the context. This motivates us to construct a Phrase-in-Context benchmark to drive the development of contextualized phrase embeddings in NLU.

Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)? Anonymous research group.

K.2 Composition/collection process/preprocessing/cleaning/labeling and uses

We describe the data construction process, annotation and verification methods in our paper (See Sec. 3 and Sec. 4).

K.3 Distribution

Will the dataset be distributed to third parties outside the entity (e.g., company, institution, organization) on behalf of which the dataset was created? We release three datasets PS, PR (including PR-pass and PR-page) and PSD to the public.

When will the dataset be distributed? It has been released in July 2022.

What is the dataset format and how it can be read? We use JSON - a widely used data format for PiC dataset and follow a scheme of HuggingFace datasets to host it. Three datasets PS, PR and PSD in the PiC dataset are loaded as follows:

```
# The following pip command is to install the HuggingFace library "datasets":
pip3 install datasets

from datasets import load_dataset

ps      = load_dataset("PiC/phrase_similarity")
pr_pass = load_dataset("PiC/phrase_retrieval", "PR-pass")
pr_page = load_dataset("PiC/phrase_retrieval", "PR-page")
psd     = load_dataset("PiC/phrase_sense_disambiguation")
```

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? Our dataset is distributed under the CC-BY-NC 4.0 license.

K.4 Maintenance

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? Yes. If we include more tasks or find any errors, we will correct the dataset. It will be updated on our website and also HuggingFace.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? They can contact us via email for the contribution.