Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks

Nandan Thakur, Nils Reimers, Johannes Daxenberger and Iryna Gurevych

Ubiquitous Knowledge Processing Lab (UKP-TUDA)

Department of Computer Science, Technische Universität Darmstadt

www.ukp.tu-darmstadt.de

Abstract

There are two approaches for pairwise sentence scoring: *Cross-encoders*, which perform full-attention over the input pair, and *Bi-encoders*, which map each input independently to a dense vector space. While cross-encoders often achieve higher performance, they are too slow for many practical use cases. Bi-encoders, on the other hand, require substantial training data and fine-tuning over the target task to achieve competitive performance. We present a simple yet efficient data augmentation strategy called *Augmented SBERT*, where we use the cross-encoder to label a larger set of input pairs to augment the training data for the bi-encoder. We show that, in this process, selecting the sentence pairs is non-trivial and crucial for the success of the method. We evaluate our approach on multiple tasks (in-domain) as well as on a domain adaptation task. *Augmented SBERT* achieves an improvement of up to 6 points for in-domain and of up to 37 points for domain adaptation tasks compared to the original bi-encoder performance.[1]

1 Introduction

Pairwise sentence scoring tasks have wide applications in NLP. They can be used in information retrieval, question answering, duplicate question detection, or clustering. An approach that sets new state-of-the-art performance for many tasks including pairwise sentence scoring is BERT (Devlin et al., 2018). Both sentences are passed to the network and attention is applied across all tokens of the inputs. This approach, where both sentences are simultaneously passed to the network, is called *cross-encoder* (Humeau et al., 2020).

A downside of cross-encoders is the extreme computational overhead for many tasks. For example, clustering of 10,000 sentences has a quadratic
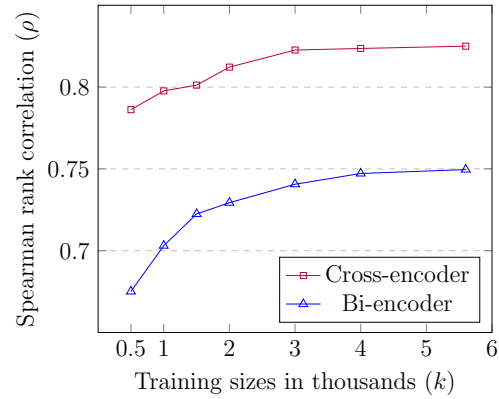

Figure 1: Spearman rank correlation (ρ) test scores for different STS Benchmark (English) training sizes.

complexity with a cross-encoder and would require about 65 hours with BERT (Reimers and Gurevych, 2019). End-to-end information retrieval is also not possible with cross-encoders, as they do not yield independent representations for the inputs that could be indexed. In contrast, *bi-encoders* such as Sentence BERT (SBERT) (Reimers and Gurevych, 2019) encode each sentence independently and map them to a dense vector space. This allows efficient indexing and comparison. For example, the complexity of clustering 10,000 sentences is reduced from 65 hours to about 5 seconds (Reimers and Gurevych, 2019). Many real-world applications hence depend on the quality of bi-encoders.

A drawback of the SBERT bi-encoder is usually a lower performance in comparison with the BERT cross-encoder. We depict this in Figure 1, where we compare a fine-tuned cross-encoder (BERT) and a fine-tuned bi-encoder (SBERT) over the popular English STS Benchmark dataset[2] (Cer et al., 2017)

---

[1]Code available: www.sbert.net

[2]http://ixa2.si.ehu.es/stswiki/index.php/STSbenchmark

for different training sizes and spearman rank correlation (ρ) on the test split.

This performance gap is the largest when little training data is available. The BERT cross-encoder can compare both inputs simultaneously, while the SBERT bi-encoder has to solve the much more challenging task of mapping inputs independently to a meaningful vector space which requires a sufficient amount of training examples for fine-tuning.

In this work, we present a data augmentation method, which we call *Augmented SBERT* (AugSBERT), that uses a BERT cross-encoder to improve the performance for the SBERT bi-encoder. We use the cross-encoder to label new input pairs, which are added to the training set for the bi-encoder. The SBERT bi-encoder is then fine-tuned on this larger augmented training set, which yields a significant performance increase. As we show, selecting the input pairs for soft-labeling with the cross-encoder is non-trivial and crucial for improving performance. Our method is easy to apply to many pair classification and regression problems, as we show in the exhaustive evaluation of our approach.

First, we evaluate the proposed AugSBERT method on four diverse tasks: Argument similarity, semantic textual similarity, duplicate question detection, and news paraphrase identification. We observe consistent performance increases of 1 to 6 percentage points over the state of the art SBERT bi-encoder's performance. Next, we demonstrate the strength of AugSBERT in a domain adaptation scenario. Since the bi-encoder is not able to map the new domain to a sensible vector space, the performance drop on the target domain for SBERT bi-encoders is much higher than for BERT cross-encoders. In this scenario, AugSBERT achieves a performance increase of up to 37 percentage points.

2 Related Work

BERT (Devlin et al., 2018) can be used as a cross-encoder for sentence scoring tasks. Both inputs are separated by a special SEP token and multi-head attention is applied over all input tokens. While the BERT cross-encoder achieves high performances for many sentence pair-tasks, a drawback is that no independent sentence representations are generated. This drawback was addressed by SBERT (Reimers and Gurevych, 2019), which applies BERT independently on the inputs followed by mean pooling on the output to create fixed-sized sentence embeddings.

Humeau et al. (2020) showed that cross-encoders typically outperform bi-encoders on sentence scoring tasks. They proposed a third strategy (poly-encoders), that is in-between cross- and bi-encoders. Poly-encoders utilize two separate transformers, one for the candidate and one for the context. A given candidate is represented by one vector, while the context is jointly encoded with the candidates (similar to cross-encoders). Unlike cross-encoder's full self attention technique, poly-encoders apply attention between two inputs only at the top layer. Poly-encoders have the drawback that they are only practical for certain applications: The score function is not symmetric, i.e., they cannot be applied for tasks with a symmetric similarity relation. Further, poly-encoder representations cannot be efficiently indexed, causing issues for retrieval tasks with large corpora sizes.

Our proposed data augmentation approach is based on semi-supervision (Blum and Mitchell, 1998) for in-domain tasks, which has been applied successfully for a wide range of tasks. Uva et al. (2018) train a SVM model with few gold samples and apply semi-supervision with pre-training neural networks. Another common strategy is to generate paraphrases of existent sentences, for example, by replacing words with synonyms (Wei and Zou, 2019), by using round-trip translation (Yu et al., 2018; Xie et al., 2020), or with seq2seq-models (Kumar et al., 2019). Other approaches generate synthetic data by using generative adversarial networks (Tanaka and Aranha, 2019), by using a language model to replace certain words (Wu et al., 2019) or to generate complete sentences (Anaby-Tavor et al., 2019). These data augmentation approaches have in common that they were applied to single sentence classification tasks. In our work, we focus on *sentence pair tasks*, for which we need to generate suitable sentence pairs. As we show, randomly combining sentences is insufficient. Sampling appropriate pairs has a decisive impact on performance.

3 Methods

In this section we present Augmented SBERT for diverse sentence pair in-domain tasks. We also evaluate our method for domain adaptation tasks.

3.1 Augmented SBERT

Given a pre-trained, well-performing cross-encoder, we sample sentence pairs according to

a certain *sampling strategy* (discussed later) and label these using the cross-encoder. We call these weakly labeled examples the *silver dataset* and they will be merged with the gold training dataset. We then train the bi-encoder on this extended training dataset. We refer to this model as Augmented SBERT (AugSBERT). The process is illustrated in Figure 2.
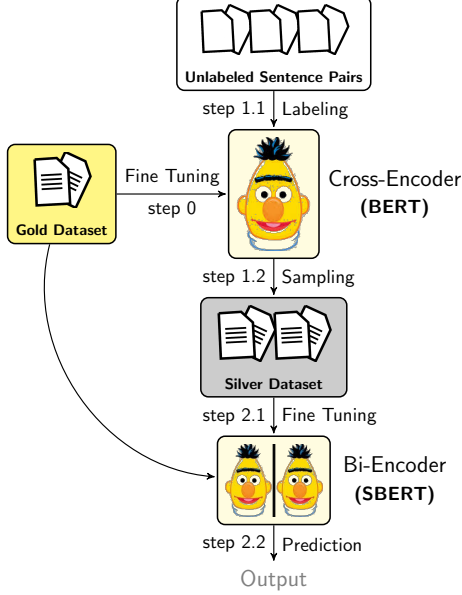

Unlabeled Sentence Pairs

step 1.1 ↓ Labeling

Fine Tuning | Cross-Encoder **(BERT)**

step 0

Gold Dataset

step 1.2 ↓ Sampling

Silver Dataset

step 2.1 ↓ Fine Tuning

Bi-Encoder **(SBERT)**

step 2.2 ↓ Prediction

Output

Figure 2: Augmented SBERT In-domain approach

Pair Sampling Strategies The novel sentence pairs, that are to be labeled with the cross-encoder, can either be new data or we can re-use individual sentences from the gold training set and re-combine pairs. In our in-domain experiments, we re-use the sentences from the gold training set. This is of course only possible if not all combinations have been annotated. However, this is seldom the case as there are $n \times (n - 1)/2$ possible combinations for n sentences. Weakly labeling all possible combinations would create an extreme computational overhead, and, as our experiments show, would likely not lead to a performance improvement. Instead, using the right sampling strategy is crucial to achieve a performance improvement.

Random Sampling (RS): We randomly sample a sentence pair and weakly label it with the cross-encoder. Randomly selecting two sentences usually leads to a dissimilar (negative) pair; positive pairs are extremely rare. This skews the label distribution of the silver dataset heavily towards negative pairs.

Kernel Density Estimation (KDE): We aim to get a similar label distribution for the silver dataset

as for the gold training set. To do so, we weakly label a large set of randomly sampled pairs and then keep only certain pairs. For classification tasks, we keep all the positive pairs. Subsequently we randomly sample out negative pairs from the remaining dominant negative silver-pairs, in a ratio identical to the gold dataset training distribution (positives/negatives). For regression tasks, we use kernel density estimation (KDE) to estimate the continuous density functions $F_{gold}(s)$ and $F_{silver}(s)$ for scores s. We try to minimize KL Divergence (Kullback and Leibler, 1951) between distributions

using a sampling function which retains a sample with score s with probability $Q(s)$:

$$
Q(s) = \begin{cases} 1 & \text{if } F_{gold}(s) \geq F_{silver}(s) \\ \dfrac{F_{gold}(s)}{F_{silver}(s)} & \text{if } F_{gold}(s) < F_{silver}(s) \end{cases}
$$

Note, that the KDE sampling strategy is computational inefficient as it requires labeling many, randomly drawn samples, which are later discarded.

BM25 Sampling (BM25): In information retrieval, the Okapi BM25 (Amati, 2009) algorithm is based on lexical overlap and is commonly used as a scoring function by many search engines. We utilize ElasticSearch[3] for the creation of indexes which helps in fast retrieval of search query results. For our experiments, we index every unique sentence, query for each sentence and retrieve the top k similar sentences. These pairs are then weakly labeled using the cross-encoder. Indexing and retrieving similar sentences is efficient and all weakly labeled pairs will be used in the silver dataset.

Semantic Search Sampling (SS): A drawback of BM25 is that only sentences with lexical overlap can be found. Synonymous sentences with no or little lexical overlap will not be returned, and hence, not be part of the silver dataset. We train a bi-encoder (SBERT) on the gold training set as described in section 5 and use it to sample further, similar sentence pairs. We use cosine-similarity and retrieve for every sentence the top k most similar sentences in our collection. For large collections, approximate nearest neighbour search like Faiss[4] could be used to quickly retrieve the k most similar sentences.

---

[3] https://www.elastic.co/
[4] https://github.com/facebookresearch/faiss

Seed Optimization [Dodge et al. (2020)](#) show a high dependence on the random seed for transformer based models like BERT, as it converges to different minima that generalize differently to unseen data ([LeCun et al., 1998](#); [Erhan et al., 2010](#); [Reimers and Gurevych, 2017](#)). This is especially the case for small training datasets. In our experiments, we apply *seed optimization*: We train with 5 random seeds and select the model that performs best on the development set. In order to speed this up, we apply *early stopping* at 20% of the training steps and only continue training the best performing model until the end. We empirically found that we can predict the final score with high confidence at 20% of the training steps ([Appendix D](#)).
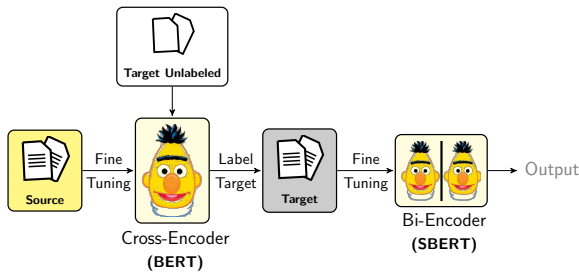

Figure 3: Domain adaptation with AugSBERT.

3.2 Domain Adaptation with AugSBERT

Until now we discussed Augmented SBERT for in-domain setups, i.e., when the training and test data are from the same domain. However, we expect an even higher performance gap of SBERT on out-of-domain data. This is because SBERT fails to map sentences with unseen terminology to a sensible vector space. Unfortunately, annotated data for new domains is rarely available.

Hence, we evaluate the proposed data augmentation strategy for domain adaptation: We first fine-tune a cross-encoder (BERT) over the source domain containing pairwise annotations. After fine-tuning, we use this fine-tuned cross-encoder to label the target domain. Once labeling is complete, we train the bi-encoder (SBERT) over the labeled target domain sentence pairs ([Figure 3](#)).

4 Datasets

Sentence pair scoring can be differentiated in regression and classification tasks. Regression tasks assign a score to indicate the similarity between the inputs. For classification tasks, we have distinct labels, for example, *paraphrase* vs. *non-paraphrase*.

4.1 Single-Domain Datasets

In our single-domain (i.e in-domain) experiments, we use two sentence pair regression tasks: semantic textual similarity and argument similarity. Furthermore, we use two binary sentence pair classification tasks: Duplicate questions detection and news paraphrase identification. Examples for all datasets are given in [Table 2](#).

SemEval Spanish STS: Semantic Textual Similarity (STS)[5] is the task of assessing the degree of similarity between two sentences over a scale ranging from $[0, 5]$ with 0 indicating no semantic overlap and 5 indicating identical content ([Agirre et al., 2016](#)). We choose Spanish STS data to test our methods for a different language than English. For our training and development dataset, we use the datasets provided by SemEval STS 2014 ([Agirre et al., 2014](#)) and SemEval STS 2015 ([Agirre et al., 2015](#)). These consist of annotated sentence pairs from news articles and from Wikipedia. As test set, we use SemEval STS 2017 ([Cer et al., 2017](#)), which annotated image caption pairs from SNLI ([Bowman et al., 2015](#)). For all our experiments, we normalise the original similarity scores to $[0, 1]$ by dividing the score by 5.

BWS Argument Similarity Dataset (BWS): Existing similarity datasets have the disadvantage that the sentence pair selection/sampling process is not always comprehensible. To overcome this limitation, we create and publicly release a novel dataset[6] for argument similarity.

We annotate sentential arguments on controversial topics on a continuous scale. We use the dataset by [Stab et al. (2018)](#), which contains pro and con stance arguments for eight controversial topics (T_1 - T_8) ("cloning", "abortion", "minimum wage", "marijuana legalization", "nuclear energy", "death penalty", "gun control", "school uniforms") retrieved from heterogeneous web sources.

Previous work addressing argument similarity ([Misra et al., 2016](#); [Reimers et al., 2019](#)) used discrete scales. However, expressing an inherently continuous property in this way is counter-intuitive and potentially unreliable due to different assumptions made when binning a range of values into a discrete class ([Kingsley and Brown, 2010](#)).

Collecting continuous annotations is complex due to selection bias and due to a lack of consis-

---

[5]https://ixa2.si.ehu.es/stswiki
[6]Public Data Release (BWS Argument Similarity Corpus): https://tudatalib.ulb.tu-darmstadt.de/handle/tudalib/2496

Dataset	Spanish-STS	BWS (cross-topic)	BWS (in-topic)	Quora-QP	MRPC
# training-samples	1,400	2125	2471	10,000	4,340
# development-samples	220	425	478	3,000	731
# testing-samples	250	850	451	3,000	730
# total-samples	**1,870**	**3,400**	**3,400**	**16,000**	**5,801**

Table 1: Summary of all datasets being used for diverse in-domain sentence pair tasks in this paper.

Dataset	Sentence 1	Sentence 2	Score
BWS	Cloning treats children as objects.	It encourages parents to regard their children as property.	0.89
Quora-QP	How does one cook broccoli?	What are the best ways to cook broccoli?	1
Spanish-STS	Dos hombres en trajes rojos practicando artes marciales.	Dos hombre en uniformes de artes marciales entrenando.	0.80
MRPC	The DVD-CCA then appealed to the state Supreme Court.	DVD CCA appealed that decision to the U.S. Supreme Court.	1

Table 2: Dataset examples for our in-domain tasks. We report the normalized similarity score $[0, 1]$ for regression tasks and the binary label $\{0, 1\}$ for classification tasks.

tency for a single annotator (Kendall, 1948). To solve the consistency problem, we apply a comparative approach, which converts the annotation into a preference problem: the annotators stated their preference on pairs of sentential arguments. We utilized the Best-Worst Scaling (*BWS*) method (Kiritchenko and Mohammad, 2016) to reduce the number of required annotations and used crowdsourcing via the Amazon Mechanical Turk Platform to annotate the data. More details regarding the formation and worker agreement scores are explained in detail in Appendix B.

We use the resulting *BWS Argument Similarity Dataset* with different splitting strategies in our paper. In *cross-topic* tasks, we fix topics (T_1 - T_5) for training, T_6 for development and (T_7 and T_8) for test sets. This is a difficult task, as models are evaluated on completely unseen topics.

Note that the cross-topic experiments on this dataset are quite different from cross-domain tasks (subsection 3.2): the model fine-tunes in-domain on fixed topics (T_1 - T_5 in our case) and is evaluated on unseen topics, whereas in the domain adaptation experiments we fine-tune on target domain data. For *in-topic*, we randomly sample fixed and disjoint pairs from each and every topic (T_1 - T_8) and create our train, development and test splits with approx. equal number of pairs from each topic.

Quora Question Pairs (Quora-QP): Duplicate question classification identifies whether two questions are duplicates. Quora released the dataset[7]

containing 404,290 question pairs. We start with the same dataset partitions from Wang et al. (2017)[8]. We remove all overlaps and ensure that a question in one split of the dataset does not appear in any other split to mitigate the transductive classification problem (Ji et al., 2010). As we observe performance differences between cross- and bi-encoders mainly for small datasets, we randomly downsample the training set to 10,000 pairs while preserving the original balance of non-duplicate to duplicate question pairs.

Microsoft Research Paraphrase Corpus (MRPC): Dolan et al. (2004) presented a paraphrase identification dataset consisting of sentence pairs automatically extracted from online news sources. Each pair was manually annotated by two human judges whether they describe the same news event. We use the originally provided train-test splits[9]. We ensured that all splits have disjoint sentences.

4.2 Multi-Domain Datasets

One of the most prominent sentence pair classification tasks with datasets from multiple domains is *duplicate question detection*. Since our focus is on pairwise sentence scoring, we model this task as a question vs. question (title/headline) binary classification task.

AskUbuntu, Quora, Sprint, and SuperUser: We replicate the setup of Shah et al. (2018) for

---

Dataset k	Train / Dev / Test (Total Pairs)	Train (Ratio)	Dev / Test (Ratio)
AskUbuntu	919706 / 101k / 101k	1 : 100	1 : 100
Quora	254142 / 10k / 10k	3.71 : 100	1 : 1
Sprint	919100 / 101k / 101k	1 : 100	1 : 100
SuperUser	919706 / 101k / 101k	1 : 100	1 : 100

Table 3: Summary of multi-domain datasets originally proposed by Shah et al. (2018) and used for our domain adaptation experiments. Ratio denotes the duplicate pairs (positives) vs. not duplicate pairs (negatives).

domain adaptation experiments. The AskUbuntu and SuperUser data comes from Stack Exchange, which is a family of technical community support forums. Sprint FAQ is a crawled dataset from the Sprint technical forum website. We exclude Apple and Android datasets due to unavailability of labeled question pairs. The Quora dataset (originally derived from the Quora website) is artificially balanced by removing negative question pairs. The statistics for the datasets can be found in Table 3. Since negative question pairs are not explicitly labeled, Shah et al. (2018) add 100 randomly sampled (presumably) negative question pairs per duplicate question for all datasets except Quora, which has explicit negatives.

5   Experimental Setup

We conduct our experiments using PyTorch Huggingface's transformers (Wolf et al., 2019) and the sentence-transformers framework[10] (Reimers and Gurevych, 2019). The latter showed that BERT outperforms other transformer-like networks when used as bi-encoder. For English datasets, we use *bert-base-uncased* and for the Spanish dataset we use *bert-base-multilingual-cased*. Every AugSBERT model exhibits computational speeds identical to the SBERT model (Reimers and Gurevych, 2019).

Cross-encoders   We fine-tune the BERT-uncased model by optimizing a variety of hyper-parameters: hidden-layer sizes, learning-rates and batch-sizes. We add a linear layer with sigmoid activation on top of the *[CLS]* token to output scores 0 to 1. We achieve optimal results with the combination: learning rate of 1×10^{-5}, hidden-layer sizes in $\{200, 400\}$ and a batch-size of 16. Refer to Table 7 in Appendix C.

Bi-encoders   We fine-tune SBERT with a batch-size of 16, a fixed learning rate of 2×10^{-5},

and AdamW optimizer. Table 8 in Appendix C lists hyper-parameters we initially evaluated.

BM25 and Semantic Search   We evaluate for various top k in $\{3, ..., 18\}$. We conclude the impact of k is not big and overall accomplish best results with $k = 3$ or $k = 5$ for our experiments. More details in Appendix E.

Evaluation   If not otherwise stated, we repeat our in-domain experiments with 10 different random seeds and report mean scores along with standard deviation. For in-domain regression tasks (STS and BWS), we report the Spearman's rank correlation ($\rho \times 100$) between predicted and gold similarity scores and for in-domain classification tasks (Quora-QP, MRPC), we determine the optimal threshold from the development set and use it for the test set. We report the F_1 score of the positive label. For all domain adaptation tasks, we weakly-label the target domain training dataset and measure AUC(0.05) as the metric since it is more robust against false negatives (Shah et al., 2018). AUC(0.05) is the area under the curve of the true positive rate as function of the false positive rate (*fpr*), from *fpr* = 0 to *fpr* = 0.05.

Baselines   For the in-domain regression tasks, we use Jaccard similarity to measure the word overlap of the two input sentences. For the in-domain classification tasks, we use a majority label baseline. Further, we compare our results against Universal Sentence Encoder (USE) (Yang et al., 2019), which is a popular state-of-the-art sentence embedding model trained on a wide rang of training data. We utilise the multilingual model[11]. Fine-tuning code for USE is not available, hence, we utilise USE as a comparison to a large scale, pre-trained sentence embedding method. Further, we compare our data augmentation strategy AugSBERT against a straight forward data augmentation strategy provided by NLPAug,[12] which replaces words in sentences with synonyms utilizing a BERT language model.

6   Results and Discussion

6.1   In-Domain Experiments for AugSBERT

Table 4 summarizes all results for all in-domain datasets. The plain bi-encoder (SBERT w/o Seed Opt.) consistently underperforms (4.5 - 9.1 points) the cross-encoder across all in-domain tasks. Op-

---

[10] https://github.com/UKPLab/sentence-transformers

[11] https://tfhub.dev/google/universal-sentence-encoder-multilingual-large/3

[12] https://github.com/makcedward/nlpaug

Task		Regression ($\rho \times 100$)			Classification (F_1)	
Model / Dataset	(Seed Opt.)	**Spanish-STS**	**BWS (cross-topic)**	**BWS (in-topic)**	**Quora-QP**	**MRPC**
Baseline	-	30.27	5.53	6.98	66.67	80.80
USE (Yang et al., 2019)	-	86.86	53.43	57.23	74.16	81.51
BERT	✗	77.50 ± 1.49	65.06 ± 1.06	65.91 ± 1.20	80.40 ± 1.05	88.95 ± 0.67
SBERT	✗	68.36 ± 5.28	58.04 ± 1.46	61.20 ± 1.66	73.44 ± 0.65	84.44 ± 0.68
BERT (*Upper-bound*)	✓	77.74 ± 1.24	65.78 ± 0.78	66.54 ± 0.94	81.23 ± 0.93	89.00 ± 0.56
SBERT (*Lower-bound*)	✓	72.07 ± 2.05	60.54 ± 0.99	63.77 ± 2.29	74.66 ± 0.31	84.39 ± 0.51
SBERT-NLPAug	✓	74.11 ± 2.58	58.15 ± 1.66	61.15 ± 0.86	73.08 ± 0.42	84.47 ± 0.79
AugSBERT-R.S.	✓	62.05 ± 2.53	59.95 ± 0.70	64.54 ± 1.90	73.42 ± 0.74	82.28 ± 0.38
AugSBERT-KDE	✓	74.67 ± 1.01	**61.49 ± 0.71**	**69.76 ± 0.50**	**79.31 ± 0.46**	84.33 ± 0.27
AugSBERT-BM25	✓	**75.08 ± 1.94**	61.48 ± 0.73	68.63 ± 0.79	79.01 ± 0.45	**85.46 ± 0.52**
AugSBERT-S.S.	✓	74.99 ± 2.30	61.05 ± 1.02	68.06 ± 0.93	77.20 ± 0.41	82.42 ± 0.32

Table 4: Summary of all the datasets being used for the in-domain tasks in this paper. STS and BWS are regression tasks, where we report Spearman's rank correlation $\rho \times 100$. Quora-QP and MRPC are classification tasks, where we report F_1 score of the positive class. Scores with the best AugSBERT strategy are highlighted. Corresponding development set performances can be found in Appendix G, Table 12.

timizing the seed helps SBERT more than BERT, however, the performance gap remains open (2.8 - 8.2 points). Training with multiple random seeds and selecting the best performing model on the development set can significantly improve the performance. For the smallest dataset (STS), we observe large performance differences between different random seeds. The best and worst seed for SBERT have a performance difference of more than 21 points. For larger datasets, the dependence on the random seed decreases. We observe bad training runs can often be identified and stopped early using the early stopping algorithm (Dodge et al., 2020). Detailed results with seed optimization can be found in Appendix D.

Our proposed AugSBERT approach improves the performance for all tasks by 1 up to 6 points, significantly outperforming the existing bi-encoder SBERT and reducing the performance difference to the cross-encoder BERT. It outperforms the synonym replacement data augmentation technique (*NLPAug*) for all tasks. Simple word replacement strategies as shown are not helpful for data augmentation in sentence-pair tasks, even leading to worse performances compared to models without augmentation for BWS and Quora-QP. Compared to the off the shelf USE model, we see a significant improvement with AugSBERT for all tasks except Spanish-STS. This is presumably due to the fact that USE was trained on the SNLI corpus (Bowman et al., 2015), which was used as basis for the Spanish STS test set, i.e., USE has seen the test sentence pairs during training.

For the novel BWS argument similarity dataset,

we observe AugSBERT only gives a minor improvement for cross-topic split. We assume this is due to cross-topic setting being a challenging task, mapping sentences of an unseen topic to a vector space such that similar arguments are close. However, on known topics (in-topic), AugSBERT shows its full capabilities and even outperforms the cross-encoder. We think this is due a better generalization of SBERT bi-encoder compared to the BERT cross-encoder. Sentences from known topics (in-topic) are mapped well within a vector space by a bi-encoder.
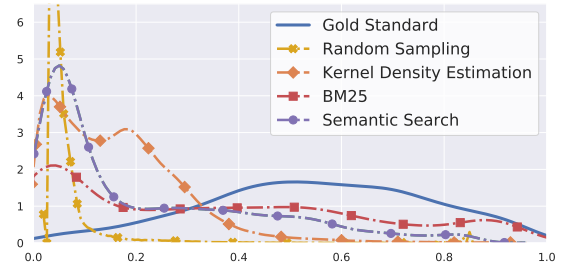

Figure 4: Comparison of the density distributions of gold standard with silver standard for various sampling techniques on Spanish-STS (in-domain) dataset.

Pairwise Sampling We observe that the sampling strategy is critical to achieve an improvement using AugSBERT. Random sampling (R.S.) decreases performance compared to training SBERT without any additional silver data in most cases. BM25 sampling and KDE produces the best AugSBERT results, followed by Semantic Search (S.S.). Figure 4, which shows the score distribution for the gold and silver dataset for Spanish-STS, visual-

		In-Domain		Cross-Domain		
Source (Train)	**Target** (Evaluate)	**SBERT** (*Upper-bound*)	**AugSBERT**	**SBERT** (*Lower-bound*)	**Bi-LSTM** (Direct)	**Bi-LSTM** (Adversarial)
AskUbuntu	Quora	0.504	**0.496**	**0.496**	0.059	0.066
	Sprint	0.869	0.852	0.747	**0.93**	0.923
	SuperUser	0.802	0.779	0.738	**0.806**	0.798
Quora	AskUbuntu	0.715	**0.602**	0.501	0.351	0.328
	Sprint	0.869	**0.875**	0.505	**0.875**	0.867
	SuperUser	0.802	**0.645**	0.504	0.523	0.485
SuperUser	AskUbuntu	0.715	**0.709**	0.637	0.629	0.627
	Quora	0.504	**0.495**	0.495	0.058	0.067
	Sprint	0.869	0.876	0.785	0.936	**0.937**
Sprint	AskUbuntu	0.715	**0.663**	0.613	0.519	0.543
	Quora	0.504	0.495	**0.496**	0.048	0.063
	SuperUser	0.802	**0.769**	0.660	0.658	0.636

Table 5: AUC(0.05) scores for domain adaptation experiments. All except SBERT (in-domain) are evaluated in cross-domain setup with the best transfer strategy highlighted. We adapt (Shah et al., 2018) Bi-LSTM models. Corresponding development set performances can be found in Appendix G, Table 13.

izes the reason for this. With random sampling, we observe an extremely high number of low similarity pairs. This is expected, as randomly sampling two sentences yields in nearly all cases a dissimilar pair. In contrast, *BM25* generates a silver dataset with similar score distribution to the gold training set. It is still skewed towards low similarity pairs, but has the highest percentage of high similarity pairs. *KDE* shows the highest performance in three tasks, but only marginally outperforms BM25 in two of these. Given that BM25 is the most computationally efficient sampling strategy and also creates smaller silver datasets (numbers are given in Appendix F, Table 11), it is likely the best choice for practical applications.

6.2 Domain Adaption with AugSBERT

We evaluate the suitability of AugSBERT for the task of domain adaption. We use duplicate question detection data from different (specialized) online communities. Results are shown in Table 5. We can see in almost all combinations that AugSBERT outperforms SBERT trained on out-of-domain data (cross-domain). On the Sprint dataset (target), the improvement can be as large as 37 points. In few cases, AugSBERT even outperforms SBERT trained on gold in-domain target data.

We observe that AugSBERT benefits a lot when the source domain is rather generic (e.g. Quora) and the target domain is rather specific (e.g. Sprint). We assume this is due to Quora forum covering many different topics including both technical and non-technical questions, transferred well by a cross-encoder to label the specific target domain (thus benefiting AugSBERT). Vice-versa, when we go from a specific domain (Sprint) to a generic target domain (Quora), only a slight performance increase is noted.

For comparison, Table 5 also shows the state-of-the-art results from Shah et al. (2018), who applied direct and adversarial domain adaptation with a Bi-LSTM bi-encoder. With the exception of the Sprint dataset (target), we outperform that system with a substantial improvements for many combinations.

7   Conclusion

We presented a simple, yet effective data augmentation approach called AugSBERT to improve bi-encoders for pairwise sentence scoring tasks. The idea is based on using a more powerful cross-encoder to soft-label new sentence pairs and to include these into the training set.

We saw a performance improvement of up to 6 points for in-domain experiments. However, selecting the right sentence pairs for soft-labeling is crucial and the naive approach of randomly selecting pairs fails to achieve a performance gain. We compared several sampling strategies and found that BM25 sampling provides the best trade-off between performance gain and computational complexity.

The presented AugSBERT approach can also be used for domain adaptation, by soft-labeling data on the target domain. In that case, we observe an improvement of up to 37 points compared to an SBERT model purely trained on the source domain.

Acknowledgements

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263.

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

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511*. ACL (Association for Computational Linguistics).

Giambattista Amati. 2009. *BM25*, volume 1, pages 257–260. Springer US, Boston, MA.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2019. Not Enough Data? Deep Learning to the Rescue! *arXiv preprint arXiv:1911.03118*.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, page 92–100, New York, NY, USA. Association for Computing Machinery.

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 (EMNLP)*. Association for Computational Linguistics.

John C. Callender and H. G. Osburn. 1979. An empirical comparison of coefficient alpha, guttman's lambda - 2, and msplit maximized split-half reliability estimates. *Journal of Educational Measurement*, 16(2):89–99.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356, Geneva, Switzerland. COLING.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why Does Unsupervised Pre-training Help Deep Learning? *Journal of Machine Learning Research*, 11:625–660.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Ming Ji, Yizhou Sun, Marina Danilevsky, Jiawei Han, and Jing Gao. 2010. Graph regularized transductive classification on heterogeneous information networks. In *Machine Learning and Knowledge Discovery in Databases*, pages 570–586, Berlin, Heidelberg. Springer Berlin Heidelberg.

Maurice Kendall. 1948. *Rank Correlation Methods*. Griffin, Oxford, UK.

David C. Kingsley and Thomas C. Brown. 2010. Preference Uncertainty, Preference Learning, and Paired Comparison Experiments. *Land Economics*, 86(3).

Svetlana Kiritchenko and Saif M. Mohammad. 2016. Capturing reliable fine-grained sentiment associations by crowdsourcing and best–worst scaling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 811–817, San Diego, California. Association for Computational Linguistics.

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. 2019. Submodular Optimization-based Diverse Paraphrasing and its Effectiveness in Data Augmentation. 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 3609–3619, Minneapolis, Minnesota. Association for Computational Linguistics.

Yann LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. 1998. Efficient BackProp. In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, pages 9–50, London, UK, UK. Springer-Verlag.

Amita Misra, Brian Ecker, and Marilyn A. Walker. 2016. Measuring the similarity of sentential arguments in dialogue. In *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*, pages 276–287.

Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019. Classification and Clustering of Arguments with Contextualized Word Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 567–578, Florence, Italy.

Darsh Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. 2018. Adversarial domain adaptation for duplicate question detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1056–1063, Brussels, Belgium. Association for Computational Linguistics.

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018. Cross-topic argument mining from heterogeneous sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, volume Long Papers, pages 3664–3674.

Fabio Henrique Kiyoiti dos Santos Tanaka and Claus Aranha. 2019. Data augmentation using GANs. *arXiv preprint arXiv:1904.09135*.

Antonio Uva, Daniele Bonadiman, and Alessandro Moschitti. 2018. Injecting relational structural representation in neural networks for question similarity. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 285–291, Melbourne, Australia. Association for Computational Linguistics.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4144–4150.

Jason Wei and Kai Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

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

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *Computational Science – ICCS 2019*, pages 84–95, Cham. Springer International Publishing.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. 2020. Unsupervised data augmentation for consistency training.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. 2019. Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. In *International Conference on Learning Representations*.

A Appendices

In this appendix, we mention the following sections in detail: data collection process and guidelines for the BWS argument similarity dataset (B), hyperparameter-tuning (C) and seed-optimization (D); provide analysis of the top-k parameter (E) and computational efficiency (F) for our in-domain sampling strategies; report development set performances for all our tasks (G).

B BWS Argument Similarity Dataset

B.1 Amazon Mechanical Turk Guidelines

The annotations required for the BWS Argument Similarity Corpus were acquired via crowdsourcing on the Amazon Mechanical Turk platform. Workers participating in the study had to be located in the US, with more than 100 HITs approved and an overall acceptance rate of 90% or higher. We paid them at the US federal minimum wage of $7.25/hour. Workers also had to qualify for the study by passing a qualification test consisting of four test questions with argument pairs. Figure 7 exemplifies the instructions given to workers.

B.2 Split-Half Reliability Test

For each topic in BWS Argument Similarity Dataset regardless of stance, all arguments were randomly paired and for ensuring a certain proportion of similar arguments within the pairings, a distant supervision filtering strategy was implemented by labeling pairs with scores between 0 and 1 using the system proposed by Misra et al. (2016).

Afterward, all argument pairs were sampled with a desired similarity distribution, by creating argument pair bins across three categories: top 1%, top 2-50% and remaining pairs. As the final step, we randomly drew pairs from the top 1% with 50% probability, and with each 25% from the two other bins.

The resulting formed sentence pairs were annotated using crowdsourcing via the Amazon Mechanical Turk Platform. For each annotation, workers were given four argument pairs and are asked to vote the most and least similar pair amongst them. Each of these tasks were assigned to four different workers. To assess the quality of the resulting annotations, we used split-half reliability measure (Callender and Osburn, 1979). Workers' votes were split by half and used to independently rank all argument pairs with the BWS method for each half
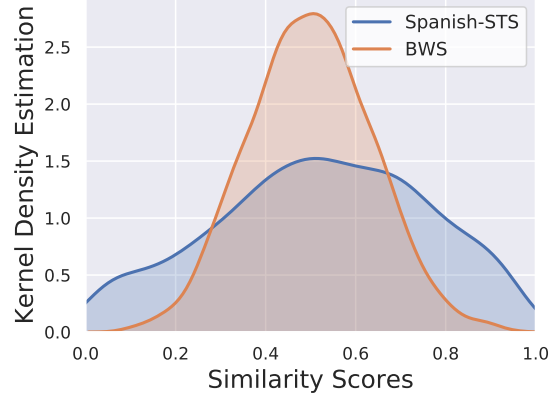

Figure 5: Comparison of density distribution of BWS Argument Similarity dataset with Spanish-STS dataset.

on each task. Finally, the Spearman's rank correlation between the resulting rankings is calculated as a proxy for consistency. The resulting average correlation across all topics in our dataset is 0.66 (random splits are repeated 25 times and final scores averaged), which, given the small number of votes per half (only 2), is in an acceptable range and reflects the difficulty of this task (Kiritchenko and Mohammad, 2016). Table 6 lists the mean split-half reliability estimates for all topics (averages over 25 random splits) in the BWS dataset.

Topic T	Score	Topic T	Score
Cloning	0.84	Nuclear energy	0.64
Abortion	0.79	Death penalty	0.58
Minimum wage	0.50	Gun control	0.59
Marijuana legal.	0.57	School uniforms	0.64
Whole dataset = 0.66			

Table 6: Mean split-half reliability estimate is calculated using Spearman's rank correlation ρ per topic T and over the whole BWS dataset.

B.3 Density Distribution Analysis

Figure 5 compares the density distributions of BWS with Spanish-STS. For the Spanish-STS dataset, the pre-sampling process results in a high amount of pairs towards either ends of the similarity scale—leading to selection bias. The pre-sampling of the creation process of the BWS dataset, in turn, is less biased. There is a much lower number of pairs towards either end of the scale, which is in accordance with data from the wild, i.e. randomly paired arguments.

C Hyperparameter Tuning

We implement *coarse to fine* random search to find the optimal combination of hyperparameters for both cross-encoders (BERT) and bi-encoders (SBERT). We choose the optimal combination based on the development dataset performance keeping random seed value fixed.[13]

Cross-Encoder (BERT): For all fine-tuning experiments, we optimize a variety of hyperparameters: hidden-layer sizes, learning-rates and batch-sizes. We first evaluate over a wide range of parameters and later conduct a deeper fine search of these optimal parameters. Experimental setup can be found in Table 7.

BERT model	bert-base (uncased/multi.-cased)
hidden layer sizes	{100, 200, 400, 800, 1600, 3200}
Learning rates	{1e-4, 1e-5, 1e-6}
Batch sizes	{8, 16}

Table 7: Experimental setup for hyperparameter tuning of cross-encoder (BERT).

Bi-Encoder (SBERT): For all fine-tuning experiments, we utilize *bert-base* models, and implement coarse to fine random search with various learning-rates and batch-sizes. Since changing the learning rate scheduler did not contribute to significant improvement, we kept it constant for all our experiments. Experimental setup can be found in Table 8.

BERT model	bert-base (uncased/ multi.-cased)
Learning rates	{2e-5, 1e-6, 1e-7}
Learning rate scheduler	constant
Batch sizes	{8, 16}

Table 8: Experimental setup for hyperparameter tuning of bi-encoder (SBERT).

D Seed Optimization

For our in-domain tasks, we apply seed optimization i.e. we train our models with 5 random seeds and select the model that performs best on the development set, and repeat this complete setup 10 times. Testing various seeds can be computationally expensive. In order to reduce the computational overhead, we evaluate whether bad runs can

be identified and stopped early. At x% of the overall training steps we evaluate the model on the development set and compare the rank with the final ranking of the models on the development set. The results are depicted in Figure 6. We observe a Spearman's rank correlation of over 0.8 at about 20% of the training steps. We conclude, that bad training runs can often be identified and stopped early.
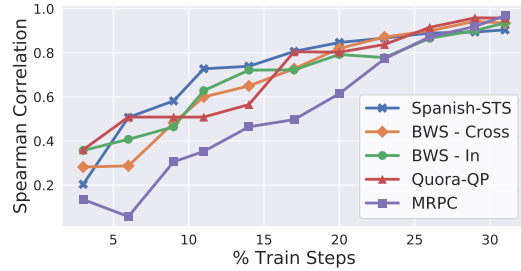

Figure 6: Spearman's rank correlation for SBERT bi-encoder between development scores at x% of the training steps with final development score for in-domain datasets.

E Impact of Top K in Sampling Strategy

In sampling strategies, such as BM25 and semantic search, we are required to pick the top k values returned by the retrieval engine. Typically for small k values, positive-pairs are dominant and with increase in k, negative-pairs start becoming dominant.

We chose a top k value within {3,5,7,9,12,18} and evaluated the final scores retrieved from our experiments, to measure an impact of k. Overall, we find the impact of k to be rather small and $k = 3$ or $k = 5$ producing optimal scores for most of the experiments. Top-k mean test scores for our in-domain datasets are reported in Table 9 for BM25 and Table 10 for semantic search sampling strategies respectively.

F Computational Efficiency vs. Size of Silver Datasets

The augmented SBERT strategy requires to weakly label a large set of sentence pairs with the cross-encoder. The larger the set of silver pairs, the bigger is the overhead for labeling with the cross-encoder and subsequent training the bi-encoder. Hence, for reasons of efficiency, smaller silver dataset sizes are preferable. Table 11 summarizes the performance of each sampling technique versus the size

---

[13]Random seed value = 42 during hyperparameter tuning experiments.

Dataset/Top k	Measure	Top 3	Top 5	Top 7	Top 9	Top 12	Top 18
Spanish-STS	$\rho \times 100$	73.67	**75.08**	74.83	74.71	73.89	72.82
BWS (cross)	$\rho \times 100$	60.02	60.23	**61.48**	60.65	60.89	61.47
BWS (in)	$\rho \times 100$	66.26	**68.63**	67.49	67.38	67.74	68.08
Quora-QP	F_1	**79.01**	78.68	78.49	78.40	78.46	77.75
MRPC	F_1	**85.46**	85.17	85.03	84.15	84.24	84.27

Table 9: Summary of In-domain BM25 Sampling Strategy: Top k mean test scores. We report Spearman's rank correlation $\rho \times 100$ for regression tasks and F_1 score for classification tasks.

Dataset/Top k	Measure	Top 3	Top 5	Top 7	Top 9	Top 12	Top 18
Spanish-STS	$\rho \times 100$	73.84	74.22	**74.99**	74.31	74.22	73.61
BWS (cross)	$\rho \times 100$	60.57	60.51	60.76	**61.04**	60.74	60.87
BWS (in)	$\rho \times 100$	65.39	**68.06**	67.01	66.78	66.95	65.93
Quora-QP	F_1	**77.20**	76.65	76.41	76.68	76.33	76.32
MRPC	F_1	**82.42**	82.18	82.20	81.86	81.81	81.91

Table 10: Summary of In-domain Semantic Search Sampling Strategy: Top k mean test scores. We report Spearman's rank correlation $\rho \times 100$ for regression tasks and F_1 score for classification tasks.

of sampled silver pairs.

Different sampling strategies create vastly different amounts of sentence pairs. Randomly sampling (R.S.) a large number of sentence pairs is not efficient and often leads to worse performances. KDE with large silver datasets produce optimal scores, but is less computationally efficient. Semantic Search (S.S.) requires the bi-encoder to be additionally trained, which causes computational overhead. Finally, BM25 overall on an average performs best for all tasks given computational efficiency, by sampling out the smallest silver dataset sizes for all tasks in Table 11.

G Development Set Performances

The development set performances for all sentence pair in-domain and domain adaptation tasks can be referred in Table 12 and Table 13 respectively.

Sampling Tech.	**None**	**BM25**		**Sem. Search**		**KDE**		**Random Samp.**	
Dataset	Score	(#Silver)	(Score)	#Silver	(Score)	(#Silver)	(Score)	(#Silver)	(Score)
Spanish-STS	72.07	**3,964**	**75.08**	12,715	74.99	230,364	74.67	911,072	62.05
BWS (cross-topic)	60.54	**11,694**	61.48	18,824	61.05	559,630	**61.49**	72,3540	59.95
BWS (in-topic)	63.77	**9,236**	68.63	40,816	68.06	394,252	**69.76**	565,820	64.54
Quora-QP	74.66	**28,014**	79.01	47,055	77.20	50,147	**79.31**	1,000,000	73.42
MRPC	84.39	**10,637**	**85.46**	18,292	82.39	32,353	84.33	1,000,000	82.28

Table 11: Summary of (*#silver dataset samples*, *mean score*) for each sampling technique across all in-domain datasets. For STS and BWS datasets, we report the Spearman's rank correlation $\rho \times 100$ and the F_1 score for Quora-QP and MRPC datasets. None represents plain bi-encoder i.e. SBERT. Scores with best sampling strategy and smallest silver dataset size across each dataset are highlighted.

Task		Regression ($\rho \times 100$)			Classification (F_1)	
Model / Dataset	(Seed-Opt.)	**Spanish-STS**	**BWS (cross-topic)**	**BWS (in-topic)**	**Quora-QP**	**MRPC**
Baseline	-	16.98	6.31	5.06	66.67	80.75
BERT	✗	89.10 ± 0.69	60.97 ± 1.35	64.89 ± 1.41	81.87 ± 1.07	89.71 ± 0.54
SBERT	✗	82.15 ± 0.95	54.77 ± 0.68	62.66 ± 1.06	76.69 ± 0.34	87.30 ± 0.45
BERT (*Upper-bound*)	✓	88.86 ± 0.74	62.17 ± 0.83	66.23 ± 0.96	81.64 ± 0.99	89.75 ± 0.46
SBERT (*Lower-bound*)	✓	82.30 ± 1.11	54.90 ± 0.88	62.75 ± 1.16	76.73 ± 0.39	87.37 ± 0.52
SBERT-NLPAug	✓	84.63 ± 0.77	58.66 ± 0.49	66.16 ± 0.41	79.72 ± 0.41	87.24 ± 0.52
AugSBERT-R.S.	✓	75.90 ± 1.89	57.36 ± 0.80	**67.83 ± 0.43**	73.42 ± 0.74	84.12 ± 0.64
AugSBERT-KDE	✓	**85.69 ± 0.54**	57.71 ± 0.85	66.14 ± 0.72	**80.00 ± 0.31**	87.04 ± 0.29
AugSBERT-BM25	✓	85.10 ± 1.11	**57.95 ± 0.98**	65.37 ± 1.18	77.73 ± 0.47	**88.04 ± 0.51**
AugSBERT-S.S.	✓	85.28 ± 0.85	56.84 ± 1.22	66.21 ± 0.92	79.29 ± 0.31	83.52 ± 0.27

Table 12: Summary of development set scores for the in-domain tasks in this paper. STS and BWS are regression tasks, where we report Spearman's rank correlation $\rho \times 100$. Quora-QP and MRPC are classification tasks, where we report F_1 score of the positive class. For Baselines, we use a simple Jaccard similarity for regression tasks and a majority label baseline for classification tasks. Scores with the best augmented SBERT strategy are highlighted.

		In-Domain	Cross-Domain			
Source (Train)	**Target** (Evaluate)	**SBERT** (*Upper-bound*)	**AugSBERT**	**SBERT** (*Lower-bound*)	**Bi-LSTM** (Direct)	**Bi-LSTM** (Adversarial)
AskUbuntu	Quora	0.675	**0.638**	0.496	0.062	0.071
	Sprint	0.989	0.773	0.670	**0.921**	0.917
	SuperUser	0.908	**0.801**	0.586	0.797	0.782
Quora	AskUbuntu	0.844	**0.512**	0.511	0.328	0.309
	Sprint	0.989	0.675	0.524	0.639	**0.848**
	SuperUser	0.908	0.509	0.512	**0.529**	0.473
SuperUser	AskUbuntu	0.844	0.612	0.564	0.607	**0.620**
	Quora	0.675	**0.672**	0.496	0.066	0.077
	Sprint	0.989	**0.848**	0.650	0.936	0.933
Sprint	AskUbuntu	0.844	**0.724**	0.601	0.521	0.532
	Quora	0.511	**0.668**	0.497	0.049	0.063
	SuperUser	0.908	**0.748**	0.620	0.652	0.631

Table 13: AUC(0.05) development scores for domain adaptation experiments. All except SBERT (in-domain) are evaluated in cross-domain setup with the best transfer strategy highlighted. We adapt (Shah et al., 2018) Bi-LSTM models.

Arguments are *similar* if -

- They say exactly the same thing in different words. Example for topic "Fracking",

  Argument A: "*And the toxic chemicals associated with fracking operations can contaminate the soil, air and water, and leach into crops*".

  Argument B: "*The chemicals used in fracking are toxic and threaten to poison and pollute our air, ground, water and food supplies - basic necessities for life*".

- They cover the same aspect and only differ in minor details. Example for topic "Electric Cars",

  Argument A: "*With literally hundreds of moving parts, a petro-fired automobile requires considerably more maintenance than an electric car*".

  Argument B: "*Electric cars are much more reliable and require less maintenance than gas-powered cars*".

- They talk about the same general aspect but differ in important details. Example for topic "Electric Cars",

  Argument A: "*Electric cars are environmentally friendly as it reduces air pollution*".

  Argument B: "*Many people think that electric cars are better than gasoline models, not only because of lower operating costs, but because of quicker acceleration and cleaner air*".

Arguments are *not similar* if

- They have the same topic but do not cover the same aspect. Example for topic "Electric cars",

  Argument A: "*Electric cars are environmentally friendly as it reduces air pollution*".

  Argument B: "*Generally electric motors for automobiles are much easier to maintain*".

- They have different topics. Example for topic "Robotic Surgery",

  Argument A: "*Opponents argue that more drilling offshore could damage sensitive ecosystems*".

  Argument B: "*Robotic surgery offers patients less pain, fewer complications, and a faster return to normal daily activities*".

Figure 7: Amazon Mechanical Turk HIT Guidelines used in the annotation study for the BWS Argument Similarity Corpus.