Selection Bias Explorations and Debias Methods for Natural Language Sentence Matching Datasets

Guanhua Zhang[1,2*], **Bing Bai**[1*], **Jian Liang**[1], **Kun Bai**[1],
Shiyu Chang[3], **Mo Yu**[3], **Conghui Zhu**[2], **Tiejun Zhao**[2]
[1]Cloud and Smart Industries Group, Tencent, China
[2]Harbin Institute of Technology, China
[3]MIT-IBM Watson AI Lab, IBM Research, USA
{guanhzhang,icebai,joshualiang,kunbai}@tencent.com,
shiyu.chang@ibm.com, yum@us.ibm.com, {chzhu,tjzhao}@hit-mtlab.net

Abstract

Natural Language Sentence Matching (NLSM) has gained substantial attention from both academics and the industry, and rich public datasets contribute a lot to this process. However, biased datasets can also hurt the generalization performance of trained models and give untrustworthy evaluation results. For many NLSM datasets, the providers *select* some pairs of sentences into the datasets, and this sampling procedure can easily bring unintended pattern, *i.e.*, selection bias. One example is the QuoraQP dataset, where some content-independent naïve features are unreasonably predictive. Such features are the reflection of the selection bias and termed as the "*leakage features*." In this paper, we investigate the problem of selection bias on six NLSM datasets and find that four out of them are significantly biased. We further propose a training and evaluation framework to alleviate the bias. Experimental results on QuoraQP suggest that the proposed framework can improve the generalization ability of trained models, and give more trustworthy evaluation results for real-world adoptions.

1 Introduction

Natural Language Sentence Matching (NLSM) aims at comparing two sentences and identifying the relationships (Wang et al., 2017), and serves as the core of many NLP tasks such as question answering and information retrieval (Wang et al., 2016b). Natural Language Inference (NLI) (Bowman et al., 2015) and Semantic Textual Similarity (STS) (Wang et al., 2016b) are both typical NLSM problems. A large number of publicly available datasets have benefited the research to a great extent (Kim et al., 2018; Wang et al., 2017; Tien et al., 2018), including QuoraQP[1], SNLI (Bowman et al., 2015), SICK (Marelli et al., 2014), *etc.* These datasets provide resources for both training and evaluation of different algorithms (Torralba and Efros, 2011).

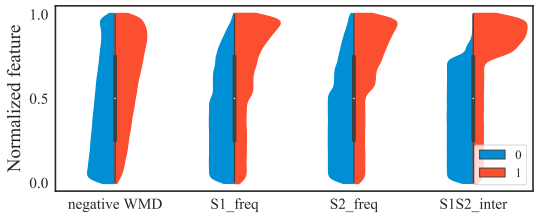

Figure 1: Visualization for the distributions of normalized features versus the label in QuoraQP. The right part (in red) represents the distributions of `duplicated` pairs, and the left part (in blue) represents the distributions of `not_duplicated` pairs. Best viewed in color.

However, most of the datasets are prepared by conducting procedures involving a sampling process, which can easily introduce a *selection bias* (Heckman, 1977; Zadrozny, 2004). It would get even worse when the bias can reveal the label information, resulting in the "*leakage features*," which are irrelevant to the content/semantic of the sentences but are predictive to the label. One example is the QuoraQP, a dataset on classifying whether two sentences are duplicated (labeled as 1) or not (labeled as 0), which has been widely used to evaluate STS models (Gong et al., 2017; Kim et al., 2018; Wang et al., 2017; Devlin et al., 2018). In QuoraQP, three leakage features have been identified, including `S1_freq`, the number of occurrences of the first sentence in the dataset; `S2_freq`, the number of occurrences of the second sentence; and `S1S2_inter`, the number of sentences that are paired with both the first and the

* Equal contributions from both authors. This work was done when Guanhua Zhang was an intern at Tencent.

[1]https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4418–4429
Florence, Italy, July 28 - August 2, 2019.

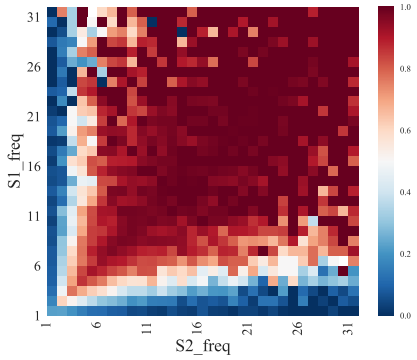

Figure 2: The averages of the labels under different `S1_freq` and `S2_freq`. Red squares indicate that the averages are close to 1, and blue squares indicate that the averages are close to 0. Best viewed in color.

second sentences in the dataset for comparison.

Figure 1 shows the distributions of normalized (negative) Word Mover's Distance (WMD) (Kusner et al., 2015) and normalized leakage features versus the labels in QuoraQP. The features are all normalized to their quantiles. As illustrated, the leakage features are more predictive than the WMD, as the differences between the distributions of positive and negative pairs are more significant. Moreover, combining `S1_freq` and `S2_freq` can make even more accurate predictions as illustrated in Figure 2, where we calculate the averages of the labels under different `S1_freq` and `S2_freq`. We find that when both features' values are large, the pairs tend to be `duplicated` (marked in red), while when one is large and the other is small, the pairs tend to be `not_duplicated` (marked in blue).

These leakage features play a critical role in the QuoraQP competition[2]. As the evaluations are conducted with the same biased datasets, models that fit the bias pattern can take additional advantages over unbiased models, making the benchmark results untrustworthy. On the other hand, the bias pattern doesn't exist in the real-world, so if a model fits the bias pattern (intentionally or unintentionally), the generalization performance will be hurt, limiting the values of these datasets for further applications (Torralba and Efros, 2011).

In this paper, we study this problem and demonstrate the impact of the selection bias by a series of experiments. We focus on the selection bias embodied in the comparing relationships of sentences, and the main contributions of this paper are the answers to the following questions:

- **Does selection bias exist in other NLSM datasets?** We identify four out of six publicly available datasets that suffer from the selection bias.
- **Would Deep Neural Network (DNN)-based methods learn from the bias pattern unintentionally?** We find that Siamese-LSTM models trained on QuoraQP do capture the bias pattern.
- **Can we help the model learn the useful semantic pattern from the content without fitting the bias pattern?** We propose an easy-adopting method to mitigate the bias. Experiments show that this method can improve the generalization performance of the trained models.
- **Can we build an evaluation framework that gives us more reliable results for real-world adoption?** We propose a more trustworthy evaluation method that demonstrates consistent results with unbiased cross-dataset evaluations.

The rest of the paper is organized as follows. Section 2 gives an empirical look at the selection bias on a variety of NLSM datasets and analyzes why the leakage features are effective. Section 3 examines whether DNN-based methods fit the bias pattern unintentionally. Section 4 introduces the training and evaluation framework to alleviate the biasedness. Taking QuoraQP as an example, we report the experimental results in Section 5. Section 6 summarizes related work, and Section 7 draws the conclusion.

2 Empirical Study of the Selection Bias

In this section, we investigate the problem of selection bias on six NLSM datasets and then analyze why the leakage features are effective.

2.1 Quantifying the Biasedness in Datasets

To quantify the severity of the leakage from the selection bias, we formulate a toy problem for NLSM. We predict the *semantic relationship* of two sentences based on the *comparing relationships* between sentences. We refer *semantic relationship* of two sentences as their labels, for example, `duplicated` for STS and `entailment`

[2] https://www.kaggle.com/c/quora-question-pairs/discussion/34355 and https://www.kaggle.com/c/quora-question-pairs/discussion/33168

Method	SNLI	MultiNLI		QuoraQP	MSRP	SICK		ByteDance
		Matched	Mismatched			NLI	STS	
Majority	33.7	35.6	36.5	50.00	66.5	56.7	50.3	68.59
Unlexicalized	47.7	44.9	45.5	68.20	73.9	70.1	70.2	75.23
LSTM	77.6*	66.9†	66.9†	82.58‡	70.6◇	71.3⊤	70.2	86.45
Leakage	36.6	32.1	31.1	79.63	66.7	56.7	55.5	78.24
Advanced	39.1	32.7	33.8	80.47	67.9	57.5	56.3	85.73
Leakage *vs* Majority	+8.61	-9.83	-14.79	+59.26	+0.30	0.00	+10.34	+14.07
Advanced *vs* Majority	+16.02	-8.15	-7.40	+60.94	+2.11	+1.41	+11.93	+24.99

Table 1: The accuracy scores of predicting the label with unlexicalized features, leakage features, and advanced graph-based features and the relative improvements. Result with * is from Bowman et al. (2015). Results with † are from Williams et al. (2018). Result with ‡ is from Wang et al. (2017). Result with ◇ is from Shen et al. (2018). Result with ⊤ is from Baudiš et al. (2016). Other results are based on our implementations. "%" is omitted.

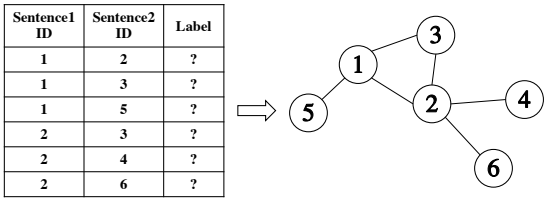

Figure 3: Illustration of the graph built for Problem 1. We only use the comparing relationships to build the graph.

for NLI, and *comparing relationship* as whether they are paired for comparison in the dataset. Here we only consider the index of each sentence, and the actual content is not used. The formal problem definition is as follow:

Problem 1 (Leveraging the Leakage for NLSM)**.** *Given a set of sentence ids* $\mathbb{S}$*, and a set of comparing relationships of the sentences* $\mathbb{C} = \{\langle s_i, s_j \rangle\}, s_i, s_j \in \mathbb{S}$*. The goal is to infer the semantic relationship between given pairs of sentence ids from* $\mathbb{S}$*.*

This toy problem is indeed an edge classification problem (Aggarwal et al., 2016), as we can construct a graph using the comparing relationships as illustrated in Figure 3. In addition, from the graph perspective, `S1_freq` and `S2_freq` are the degrees of nodes, and `S1S2_inter` is the number of 2-hop paths connecting two nodes. Learning on the graph for this toy problem follows a transductive setting (Ji et al., 2010), where the graph is built with the comparing relationships of all the examples.

Based on the new problem definition, we investigate six NLSM datasets, including SNLI, MultiNLI (Williams et al., 2018), QuoraQP, MSRP (Dolan et al., 2004), SICK and ByteDance[3]. We apply two different methods to classify the edges on the graph, including **Leakage** which uses the three leakage features introduced in Section 1 and **Advanced** which uses some more advanced graph-based features (Perozzi et al., 2014; Zhou et al., 2009; Liben-Nowell and Kleinberg, 2007) together with the three leakage features[4]. We also report the results of three baselines, including **Majority** which predicts the most frequent label, **Unlexicalized** which uses 15 handcrafted features from the content of sentences (Bowman et al., 2015) (*e.g.*, the BLEU score (Papineni et al., 2002) of both sentences, the length difference between the two sentences, the percentage of overlap words, and so on) and **LSTM** which is a DNN-based method using sequences of word embeddings. All classifiers are Random Forests if no specific configuration is mentioned. The classifiers are trained with the training set, and we report the results on the testing set. More detailed settings are introduced in Appendix A. The results are reported in Table 1.

Predicting semantic relationships without using sentence contents seems impossible. However, we find that the graph-based features (*Leakage* and *Advanced*) make the problem feasible on a wide range of datasets. Specifically, on the datasets like QuoraQP and ByteDance, the leakage features are even more effective than the unlexicalized features. One exception is that on MultiNLI, *Majority* outperforms *Leakage* and *Advanced* significantly. Another interesting finding is that on

[3]https://www.kaggle.com/c/fake-news-pair-classification-challenge

[4]The features are selected carefully to describe the local structure between two nodes and to prevent the model from remembering the exact ID of sentences to make inferences.

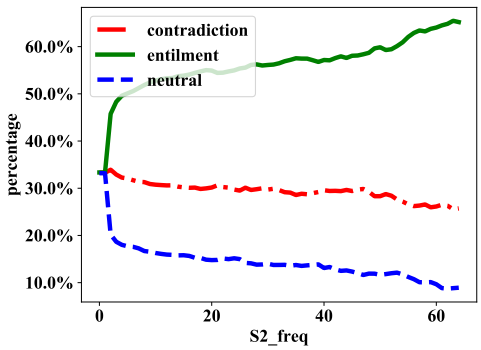

Figure 4: The percentage of each label versus `S2_freq` in SNLI.

Features	SNLI	QuoraQP	SICK_{STS}	ByteDance
`S1_freq`	33.7	65.90	54.5	68.61
`S2_freq`	36.6	69.84	52.5	73.03
`S1S2_inter`	33.7	79.66	50.8	76.63
¬ `S1_freq`	36.6	79.62	53.5	77.17
¬ `S2_freq`	33.7	79.66	53.0	77.44
¬ `S1S2_inter`	36.6	74.75	54.2	74.39
all	36.6	79.63	55.5	78.24
Majority	33.7	50.00	50.3	68.59

Table 2: Ablation experiments of the three leakage features on the datasets. "¬" means without the feature. We report the accuracy scores and "%" is omitted.

SNLI and ByteDance, advanced graph-based features improve a lot over the leakage features, while on QuoraQP, the difference is very small. Among all the tested datasets, only MSRP and SICK_{NLI} are almost neutral to the leakage features. Note that their sizes are relatively small with only less than 10k samples. Results in Table 1 raise concerns about the impact of selection bias on the models and evaluation results.

2.2 Why are the Leakage Features Effective?

As discussed in Section 1, the leakage features are the reflection of selection bias. Intuitively, if we construct a dataset for NLSM by randomly sampling some pairs of sentences, the resulting dataset would be extremely imbalanced, where the most of the pairs are `neutral` for NLI or `not_duplicated` for STS. Thus, to make the dataset relatively balanced, a sampling strategy is often required. If the strategy is not well-designed, it will introduce a bias pattern into the dataset, which can be revealed by leakage features. Here we try to figure out why the leakage features are effective in aforementioned datasets. Since we do not have every detail about how they are constructed, we only analyze based on SNLI and QuoraQP.

During the preparation of SNLI, as introduced in (Bowman et al., 2015), human workers are presented with "premise scene descriptions," and asked to supply "hypotheses" for each of the three labels (*i.e.*, `entailment`, `neutral` and `contradiction`). However, it is found that some workers are "*reusing the same sentence for many different prompts*," which might cause SNLI to suffer from selection bias. To validate, we calculate the percentage of each label versus `S2_freq`, and the results are shown in Figure 4. We see that the percentages of the three labels are similar when `S2_freq` is small, but as `S2_freq` increases, the label is more likely to be an `entailment`.

For QuoraQP dataset, the providers state that "*Our original sampling method returned an imbalanced dataset with many more true examples of duplicate pairs than non-duplicates. Therefore, we supplemented the dataset with negative examples. One source of negative examples were pairs of "related questions" which, although pertaining to similar topics, are not truly semantically equivalent.*" Our hypothesis is that the way in which negative samples were supplemented is the reason why QuoraQP is so biased. For example, the newly added sentences of "related questions" may appear in the dataset for limited times, thus we get the phenomenon in Figure 2, *i.e.*, if two sentences both appear for many times, the pair is likely to be `duplicated`, while if one of them appears for only a few times, the pair is likely to be `not_duplicated`.

We conduct ablation experiments on the datasets where the leakage features are effective, *i.e.*, SNLI, QuoraQP, SICK_{STS} and ByteDance. The results are reported in Table 2. We can see that `S2_freq` is more effective in SNLI, and `S1_freq` plays a more critical role in SICK_{STS}, while in QuoraQP and ByteDance, `S1S2_inter` is the most predictive.

Based on the experiments and observations, we conclude that existing datasets incline to be biased due to various reasons. More information about dataset preparations and further study are required to understand the problem and prevent bias from being introduced into future datasets.

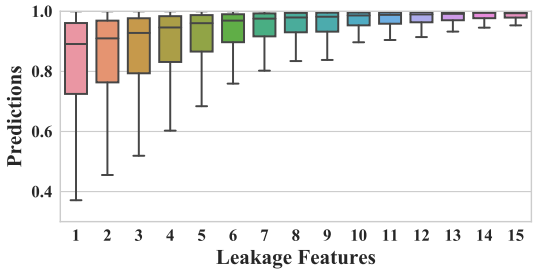

Figure 5: Visualization of predicted scores versus the leakage feature. The boxes represent the upper quartiles to the lower quartiles of predicted scores, and the lowest datum is the 1.5 IQR of the lower quartile.

3 Do NN Models Fit the Bias Pattern Unintentionally?

In this section, we investigate whether DNN models are unintentionally fitting the bias pattern in addition to the semantic pattern. We train a classical Siamese-LSTM model[5] with the training set of QuoraQP, and make predictions on a synthetic dataset. Interestingly, we find that the results are significantly influenced by the bias pattern.

The synthetic dataset is built in the following way. We extract the distinct sentences from the training set of QuoraQP, then compare the sentences with themselves, finally we obtain 517,970 pairs in total. Since the two sentences in the pairs are identical, the labels are all `duplicated`. All three leakage features are the same, *i.e.*, the numbers of occurrences of the sentence in the dataset. If the model can perfectly learn the semantic relationships between sentences, the predictions should be substantially the same for all the pairs.

To illustrate the predicted scores of duplication, we visualize them versus the leakage features in Figure 5, and the boxplot follows the Tukey boxplot style (Frigge et al., 1989). Intriguingly, we find that even though the sentences in pairs are all identical, the model still tends to give lower scores of duplication to the pairs with leakage features equal to 1. This result is consistent with the bias pattern shown in Figure 2, *i.e.*, the data points in the bottom left corner tend to be `not_duplicated`, compared with the data points in the top right corner which represent larger values of `S1_freq` and `S2_freq`.

The results indicate that the model is unintentionally capturing the undesired bias pattern that only exists in the particular dataset. This will make an adverse effect on the generalization performance of the trained models (to be illustrated in Section 5.4).

[5]The detailed setting for the model is introduced in Section 5.2

4 Leakage-Neutral Learning and Evaluation Method

Given a biased dataset, can we eliminate the bias to train completely unbiased models? Unfortunately, this is very difficult due to that the bias is related with the labels, and we cannot have access to the labels of unselected samples (Zadrozny, 2004). In this paper, we propose to take a step back and define a *leakage-neutral* distribution, which is more close to the real-world than the biased one. We make a few reasonable assumptions about it and how the biased dataset is generated from it. We demonstrate that we can train and evaluate models unbiased to the leakage-neutral distribution, with only the biased dataset.

Generation of the biased dataset from leakage-neutral distribution Assuming that there is a *leakage-neutral* distribution $\mathscr{D}$ with domain $\mathcal{X} \times \mathcal{Y} \times \mathcal{L} \times \mathcal{S}$ where $\mathcal{X}$ is the semantic feature space, $\mathcal{Y}$ is the (binary) semantic label space, $\mathcal{L}$ is the sampling strategy feature space and $\mathcal{S}$ is the (binary) sampling intention space. The sampling intentions represent whether dataset providers want to select a positive sample or a negative sample. For example, $S = 1$ means that the providers want to select a positive sample here.

We assume that samples (x, y, l, s) are drawn independently from $\mathscr{D}$, then if $s = y$ (the label matches the sampling intention), the samples are selected into the dataset, otherwise, the samples are discarded. This operation results in the biased distribution $\widehat{\mathscr{D}}$ that are observed from the dataset.

In this section, we use uppercase letters, such as Y and S, to represent random variables, and lowercase letters, such as y and s, to represent specific values for samples. We use $P_{\widehat{\mathscr{D}}}(\cdot)$ to represent the probability on $\widehat{\mathscr{D}}$ and omit the subscripts for $\mathscr{D}$.

Assumptions about the leakage-neutral distribution We make the following assumptions about $\mathscr{D}$. The first one is the *leakage-neutral* assumption defined as follows,

$$P(Y|L) = P(Y),$$

which means that the sampling strategy is independent with the labels, making the leakage-neutral distribution more close to the real-world.

The second one is that, given L, S is independent with X and Y defined as follows,

$$P(S|X,Y,L) = P(S|L),$$

which means that the sampling strategy features can completely control the sampling intentions.

Leakage-neutral learning and evaluation method Based on the assumptions above, given a biased dataset, the proposed method works in the following way.

Firstly, we estimate $P_{\widehat{\mathscr{D}}}(Y = 1|l)$ from the dataset for all samples. In practice, this can be achieved by training classifiers and making cross-predictions. Since we don't have access to the true sampling strategy features, we use the leakage features from the graph instead, as they are the reflection of the biased sampling strategy.

Then we can get $P(S = 1|l)$, the conditional probability of the sampling intention S on $\mathscr{D}$ given l, using the following equation with $P(Y = 1)$ given.

$$P(S=1|l) = \frac{P(Y=0)P_{\widehat{\mathscr{D}}}(Y=1|l)}{P(Y=0)P_{\widehat{\mathscr{D}}}(Y=1|l) + P(Y=1)P_{\widehat{\mathscr{D}}}(Y=0|l)}. \quad (1)$$

The derivation of Equation (1) is presented in Appendix B.1.

Afterwards, we use $w = \frac{1}{P(S=y|l)}$ as the weights for the samples (note that the labels y are needed here). Training and evaluating with the weights can give us the results unbiased to the leakage-neutral distribution.

The step-by-step procedure for leakage-neutral learning and evaluation is presented in Algorithm 1. Note that our analyses and the proposed method are general enough for a variety of bias, as long as a sampling strategy feature is given, and can be easily extended to multi-class classification problems.

Theoretical guarantee of unbiasedness Assuming that we know $P(S = y|l)$, and they are greater than zero for any l, the following theorem shows that we can obtain the loss unbiased to the leakage neutral distribution after using the sample weights.

Algorithm 1: Leakage-neutral Training and Evaluation

Input: The dataset $\{x, y\}$, the number of fold K for cross prediction, and the prior probability $P(Y = 1)$.

Procedure:

01 Extract the leakage features l from the dataset.
02 Estimate $P_{\widehat{\mathscr{D}}}(Y = 1|l)$ for all samples by training classifiers and using K-fold cross-predicting strategy.
03 Calculate $P(S = 1|l)$ for all samples according to Equation (1).
04 Obtain the weights $w = \frac{1}{P(S=y|l)}$ for all samples and normalize the mean of the weights.
05 Train and validate models with the training set and validation set respectively using w as the sample weights.
06 Evaluate the models with the testing set using w as the sample weights.

Theorem 1 (Unbiased Expectation)**.** *For any classifier $f = f(x, l)$, and for any loss function $\Delta = \Delta(f(x, l), y)$, if we use $w = \frac{P(S=Y)}{P(S=y|l)}$ as weights, then*

$$E_{x,y,l\sim\widehat{\mathscr{D}}}\Big[w\Delta\big(f(x,l),y\big)\Big] = E_{x,y,l\sim\mathscr{D}}\Big[\Delta\big(f(x,l),y\big)\Big].$$

The proof is presented in Appendix B.2. Since $P(S = Y)$ is only a number which does not affect the models, we can concentrate on the denominator, *i.e.*, $P(S = y|l)$ and use $w = \frac{1}{P(S=y|l)}$ as the weights instead. The loss can be used for both training and evaluation unbiased to the leakage neutral distribution.

5 Experimental Results for the Leakage-neutral Method on QuoraQP

In this section, we present the experimental results for leakage-neutral learning on QuoraQP. We demonstrate that the proposed learning framework can mitigate the bias and improve the generalization performance of trained models. Besides, the corresponding evaluation method can serve as a more reliable in-domain benchmark compared with the biased one.

5.1 Dataset Information and Weight Generation

We use QuoraQP as our experimental dataset. We use the same dataset partition as (Wang et al., 2017).

We use the three leakage features for generating the weights. We use Random Forest classifiers to estimate $P_{\widehat{\mathscr{D}}}(Y = 1|l)$, and the 100-fold cross predictions as the estimated values. $P(Y = 1)$ is chosen to keep the proportion of the weights of positive and negative samples unchanged in order to prevent the influence of prior probabilities, and

the mean of the weights is normalized to 1. The minimum weight of all the samples is 0.51, and the maximum weight is 4953.17.

5.2 Experiment Settings

We implement a classical Siamese-LSTM model with Keras and Tensorflow (Abadi et al., 2016) backend. Sequences of the embeddings of word tokens are fed into the LSTM layer with a hidden size of 128. Then the representations of both sentences, as well as the dot-production of the representations, go through a two Layer MLP where Batch Normalization (Ioffe and Szegedy, 2015) is applied after every hidden layer. Dropout (Srivastava et al., 2014) with rate 0.5 is applied after the last hidden layer. We use the RMSProp (Tieleman and Hinton, 2012) optimizer to train all the parameters. The learning rate starts at 1e-3, and decays at a fixed rate of 0.2 when performance does not improve on the validation set. We also use a gradient clipping of 5.0. The batch size is set to 256. All the results reported in this section are the average numbers of ten runs using the same hyper-parameters with different random initializations. Our implementation achieves slightly better performance compared with the results of the original Siamese-LSTM from Wang et al. (2017).

We initialize our word embeddings with pre-trained GloVe 840B 300D vectors (Pennington et al., 2014), and the embeddings are kept fixed during training. All the sentences are cut off to have a maximum of 35 word tokens.

Note that the scale of weights of the different samples varies greatly. To prevent the model from jiggling during the mini-batch training, we use a sampling strategy for model training, *i.e.*, we sample examples with probabilities proportional to the weights to get the data for every mini-batch[6].

5.3 Evaluation Scheme

To evaluate the effectiveness of leakage-neutral learning, we use the following strategy in our experiments. Firstly, we train and validate a model using the data from QuoraQP *without* any weights. The model is referred to as **Biased Model**. Then we train and validate a model using the data from QuoraQP *with* the weights, and the model is referred to as **Debiased Model**. These two models are evaluated with the following methods.

[6]Codes and weights are published at https://github.com/arthua196/Leakage-Neutral-Learning-for-QuoraQP

Method	Biased Eva	Debiased Eva
Majority	50.00	51.62
Leakage	79.63	54.40
Biased Model	**83.97**	78.76
Debiased Model	82.90	**80.11**

Table 3: Evaluation Results with the testing set of QuoraQP. We report the accuracy scores and "%" is omitted.

Method	Synthetic	MSRP	SICK_{STS}
Biased Model	89.46	51.94	64.95
Debiased Model	**92.62**	**56.77**	**66.05**

Table 4: Evaluation Results with the synthetic dataset, MSRP and SICK_{STS} dataset. We report the accuracy scores and "%" is omitted.

- **Testing set evaluation**. We evaluate the models with the testing set of QuoraQP. Evaluation without the weights is named as **Biased Eva**, and evaluation with the weights is named as **Debiased Eva**. This can show how the leakage-neutral evaluation proposed in Section 4 affect the evaluation results.

- **Synthetic dataset evaluation**. We evaluate the performance of models with the synthetic dataset introduced in Section 3. Given the prior probabilities of positive/negative classes fixed, a better model is supposed to give higher accuracy, and tended to be less impacted by the bias pattern.

- **Cross-dataset evaluation**. We evaluate that how the models perform on other STS datasets, *i.e.*, MSRP and SICK_{STS}. We use the entire datasets for evaluations. As the preparation strategies of different datasets are different, cross-dataset evaluations will not give additional rewards for the selection bias of QuoraQP. Although different datasets may have different contexts, a better model trained with QuoraQP is still supposed to perform better.

Among all the evaluation methods, using the testing set for evaluation without weights (*Biased Eva*) is biased, and we will show that the *Debiased Eva* is more consistent with the unbiased synthetic dataset evaluation and cross-dataset evaluations.

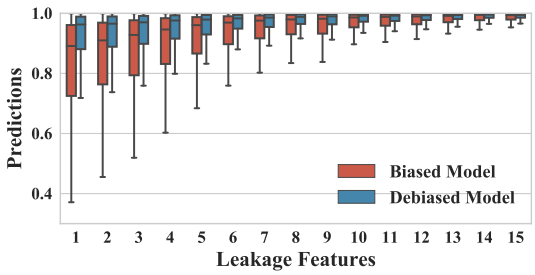

Figure 6: Visualization of predicted scores by the Biased and Debiased Models versus the leakage feature. Red boxes represent the results by the Biased Model, and blue boxes represent the results by the Debiased Model. Best viewed in color.

5.4 Experimental Results

The evaluation results on the testing set of QuoraQP are reported in Table 3. From the accuracy of the method *Leakage*, we can see that although the influence isn't completely eliminated, the evaluation result of Debiased Eva is less impacted by the bias pattern in the original distribution. This makes the results more reliable for evaluations. The reason why in the Leakage method the bias could not be completely eliminated is that we cannot estimate $P(S = y|l)$ perfectly. A minor error of $P(S = y|l)$ may result in a significant difference in the weight especially when the probability is close to zero, since the multiplicative inverse is used.

As for the Biased Model and the Debiased Model, we find that the Biased Model performs significantly better under the Biased Eva. This is the effect of fitting the bias pattern in addition to the semantic pattern, thus taking some extra advantage that cannot be generalized to real-life cases. On the other hand, under the Debiased Eva, we can find that the Debiased Model performs the best.

Table 4 reports the results on the datasets that are not biased to the leakage pattern of QuoraQP. We find that the Debiased Model significantly outperforms the Biased Model on all three datasets. This indicates that the Debiased Model better captures the true semantic similarities of the input sentences. We further visualize the predictions on the synthetic dataset in Figure 6. As illustrated, the predictions are more neutral to the leakage feature.

From the experimental results, we can see that the proposed leakage-neutral training method is effective, as the Debiased Model performs significantly better with Synthetic dataset, MSRP and SICK, showing a better generalization strength. Moreover, the Debiased Eva gives results that are more consistent with the results on unbiased datasets, thus it can serve as a more reliable in-domain way to evaluate models trained with QuoraQP. As a conclusion, our constructed leakage-neutral distribution is more close to the real-world one compared with the biased distribution that is directly observed from the given datasets.

6 Related Work

In this section, we summarize the related work and distinguish them from our contributions.

Inverse propensity score for debiasing Usually, the Inverse Propensity Score (IPS) is used to reduce the selection bias (Schonlau et al., 2009; d'Agostino, 1998), where the propensity score (Rosenbaum and Rubin, 1983) is the probability that a sample will be selected into the dataset. Zadrozny (2004) studies the learning and evaluating of classifiers under sample selection bias, while his focus was the "missing-at-random" (MAR) (Little and Rubin, 2014) problem where the biasedness only depends on the feature vector x.

For NLSM datasets, the selection bias is "not-missing-at-random" (NMAR) (Little and Rubin, 2014), thus we cannot hope to estimate the true propensity scores directly as it requires the labels of unselected samples (Zadrozny, 2004). In this paper, we propose to fit a constructed leakage-neutral distribution, which could be achieved with only the selected samples that we can access.

Biasedness of datasets Although dataset bias is often mentioned, the research community is not putting sufficient attention to it compared with models and algorithms. Torralba and Efros (2011) studied the dataset bias for image recognition datasets, and categorize the bias into *Selection Bias*, *Capture Bias* and *Negative Set Bias*. Selection bias is widely studied in the search ranking field as position bias (Wang et al., 2016a, 2018; Joachims et al., 2017). Usually the propensity scores are estimated through online Result Randomization (Joachims et al., 2017). Liang et al. (2019) studied the biasedness for authentication, and proposed an additive adversarial learning for unbiased learning.

In the NLP field, Minka and Robertson (2008) studied the selection bias in the LETOR datasets,

and found that Reverse BM25 performs unreasonably well due to the selection procedure. Dixon et al. (2018) studied the potential unfairness for toxic comments classification due to unintended bias, and proposed methods to mitigate it by balancing the training dataset with additional data. Gururangan et al. (2018) and Poliak et al. (2018) found that in some NLI datasets, there is biasedness of specific linguistic phenomena, which makes it possible to classify the relationship of a pair of sentences, by only looking at one of them. Sugawara et al. (2018) investigated what makes questions easier across recent 12 Machine Reading Comprehension (MRC) datasets and the results suggest that one might overestimate recent advances in MRC.

In this paper, we study the selection bias embodied in the comparing relationships in NLSM datasets. To the best of our knowledge, this is the first study on this kind of selection bias.

7 Conclusion

In this paper, we take a close look at the selection bias of NLSM datasets and focus on the selection bias embodied in the comparing relationships of sentences. To mitigate the bias, we propose an easy-adopting method for leakage-neutral learning and evaluations.

However, there is still much to do to form a clearer scope of this problem. For example, we still do not know the details of dataset preparations of many other NLSM datasets, and we can not say to what extent the assumptions in Section 4 hold in QuoraQP and what is the relationship between the leakage-neutral distribution and the real-world distribution. We suggest for future NLSM datasets, the providers should pay more attention to this problem. Furthermore, they could reveal the more detailed strategy of sample selection, and might publish some official weights to eliminate the bias.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283.

Charu Aggarwal, Gewen He, and Peixiang Zhao. 2016. Edge classification in networks. In *Proceddings of the 32nd IEEE International Conference on Data Engineering*, pages 1038–1049.

Petr Baudiš, Jan Pichl, Tomáš Vyskočil, and Jan Šedivỳ. 2016. Sentence pair scoring: Towards unified framework for text comprehension. *arXiv preprint arXiv:1603.06127*.

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.

Ralph B d'Agostino. 1998. Propensity score methods for bias reduction in the comparison of a treatment to a non-randomized control group. *Statistics in medicine*, 17(19):2265–2281.

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*.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73. ACM.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350. Association for Computational Linguistics.

Michael Frigge, David C Hoaglin, and Boris Iglewicz. 1989. Some implementations of the boxplot. *The American Statistician*, 43(1):50–54.

Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural language inference over interaction space. *arXiv preprint arXiv:1709.04348*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 107–112.

James J Heckman. 1977. Sample selection bias as a specification error (with an application to the estimation of labor supply functions).

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456.

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

Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 781–789. ACM.

Seonhoon Kim, Jin-Hyuk Hong, Inho Kang, and Nojun Kwak. 2018. Semantic sentence matching with densely-connected recurrent and co-attentive information. *arXiv preprint arXiv:1805.11360*.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966.

Jian Liang, Yuren Cao, Chenbin Zhang, Shiyu Chang, Kun Bai, and Zenglin Xu. 2019. Additive adversarial learning for unbiased authentication. *arXiv preprint arXiv:1905.06517*.

David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031.

Roderick JA Little and Donald B Rubin. 2014. *Statistical analysis with missing data*, volume 333. John Wiley & Sons.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *LREC*, pages 216–223.

Tom Minka and Stephen Robertson. 2008. Selection bias in the letor datasets. In *SIGIR Workshop on Learning to Rank for Information Retrieval*, pages 48–51. Citeseer.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191.

Paul R Rosenbaum and Donald B Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.

Matthias Schonlau, Arthur Van Soest, Arie Kapteyn, and Mick Couper. 2009. Selection bias in web surveys and the use of propensity scores. *Sociological Methods & Research*, 37(3):291–318.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 440–450.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What makes reading comprehension questions easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31.

Huy Nguyen Tien, Minh Nguyen Le, Yamasaki Tomohiro, and Izuha Tatsuya. 2018. Sentence modeling via multiple word embeddings and multi-level comparison for semantic textual similarity. *arXiv preprint arXiv:1805.07882*.

Antonio Torralba and Alexei A Efros. 2011. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE.

Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016a. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 115–124. ACM.

Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position bias estimation for unbiased learning to rank in personal search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 610–618. ACM.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4144–4150. AAAI Press.

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016b. Sentence similarity learning by lexical decomposition and composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1340–1349.

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

Bianca Zadrozny. 2004. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, page 114. ACM.

Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. 2009. Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630.

A Detailed Settings for the Experiments in Section 2.1

A.1 Dataset Description

We summarize the statistics of the datasets used in Section 2 in Table 5.

Dataset	Training	Testing	# classes
SNLI	549k	10k	3
MultiNLI	393k	10k	3
QuoraQP	384k	10k	2
MSRP	4k	2k	2
SICK	5k	5k	2/3
ByteDance	256k	32k	3

Table 5: Information about the datasets.

For SICK, both `entailment_label` and `relatedness_score` are provided. We use the sentence pairs with `relatedness_score` greater than 3.6 as `duplicated`, and otherwise `not_duplicated`. This threshold gives roughly 50% of positive pairs and 50% negative pairs.

For ByteDance, since no existing dataset partition is available, we randomly divide the dataset into a training set, a validation set, and a testing set in a ratio of 8:1:1. We use the sentences in English during our experiments.

A.2 Features Used in Unlexicalized

We list the 15 features we used in method **Unlexicalized** in Section 2.1. We use 3 types of unlexicalized features (Bowman et al., 2015):

- The BLEU score of both sentences, using n-gram length from 1 to 4, which are totally 4 features.
- The length difference between the two sentences, as one real-valued feature.
- The number and percentage of overlap words between both sentences over all words and over just nouns, verbs, adjectives and adverbs, which are totally 10 features.

A.3 Features Used in Advanced

We list the features we used in method **Advanced** in Section 2.1. As mentioned above, if we use a node to represent a sentence and add an undirected edge if two sentences are compared in the dataset, the whole dataset can be viewed as a graph as illustrated in Figure 3. To classify the edges in the graph, we use 3 types of graph-based features:

- The origin and extended leakage features: degrees of both nodes, number of 2-hop and 3-hop paths between the two nodes, number of 2-hop and 3-hop neighbors of both nodes, which are totally 8 features.
- The element-wise product and dot product of Deepwalk (Perozzi et al., 2014) embedding of the two nodes, all together as 65 features.
- The resource allocation index, Jaccard coefficient, preferential attachment score and Adamic-Adar index (Zhou et al., 2009; Liben-Nowell and Kleinberg, 2007) of both two nodes, which are totally 4 features.

B Proof for the Theorems

B.1 Derivation of Equation (1)

Here we present the derivation of Equation (1).

Proof.

$$\begin{aligned}
&P_{\widehat{\mathscr{D}}}(Y=1|l)\\
=&P(Y=1|S=Y,l)\\
=&\frac{P(Y=1,S=1|l)}{P(Y=1,S=1|l)+P(Y=0,S=0|l)}\\
=&\frac{P(Y=1|l)P(S=1|l)}{P(Y=1|l)P(S=1|l)+P(Y=0|l)P(S=0|l)}\\
=&\frac{P(Y=1)P(S=1|l)}{P(Y=1)P(S=1|l)+P(Y=0)P(S=0|l)}.
\end{aligned}$$

By solving the above equation, we have the result in Equation (1). □

B.2 Proof of Theorem 1

Here we present the proof for Theorem 1, *i.e.*, the unbiased expectation theorem.

Proof.

$$
\begin{aligned}
&E_{x,y,l\sim\widehat{\mathscr{D}}}\Big[w\Delta\big(f(x,l),y\big)\Big] \\
=&\int \frac{P(S=Y)}{P(S=y|l)}\Delta(f(x,l),y)dP_{\widehat{\mathscr{D}}}(x,y,l) \\
=&\int \Delta(f(x,l),y)\frac{P(S=Y)}{P(S=y|l)}dP(x,y,l|S=Y) \\
=&\int \Delta(f(x,l),y)\frac{P(S=Y)}{P(S=y|l)}\frac{P(S=y|x,y,l)dP(x,y,l)}{P(S=Y)} \\
=&\int \Delta(f(x,l),y)dP(x,y,l) \\
=&E_{x,y,l\sim\mathscr{D}}\Big[\Delta(f(x,l),y)\Big].
\end{aligned}
$$

□

As illustrated above, by adding specific weights to the samples, we can obtain the loss unbiased to the leakage neutral distribution $\mathscr{D}$. The unbiased loss can be used for both training and evaluation.