Are All the Datasets in Benchmark Necessary?
A Pilot Study of Dataset Evaluation for Text Classification

Yang Xiao[1], **Jinlan Fu**[2]*, **See-Kiong Ng**[2], **Pengfei Liu**[3]
[1]Fudan University, [2]National University of Singapore, [3]Carnegie Mellon University
yangxiaocq12@gmail.com, {jinlan,seekiong}@nus.edu.sg, pliu3@cs.cmu.edu

Abstract

In this paper, we ask the research question of whether all the datasets in the benchmark are necessary. We approach this by first characterizing the distinguishability of datasets when comparing different systems. Experiments on 9 datasets and 36 systems show that several existing benchmark datasets contribute little to discriminating top-scoring systems, while those less used datasets exhibit impressive discriminative power. We further, taking the text classification task as a case study, investigate the possibility of predicting dataset discrimination based on its properties (e.g., average sentence length). Our preliminary experiments promisingly show that given a sufficient number of training experimental records, a meaningful predictor can be learned to estimate dataset discrimination over unseen datasets. We released all datasets with features explored in this work on DataLab. [1]

1 Introduction

In natural language processing (NLP) tasks, there are often datasets that we use as benchmarks against which to evaluate machine learning models, either explicitly defined such as GLUE (Wang et al., 2018) and XTREME (Hu et al., 2020) or implicitly bound to the task (e.g., DPedia (Zhang et al., 2015) has become a default dataset for evaluating of text classification systems). Given this mission, one important feature of a good benchmark dataset is the ability to statistically differentiate diverse systems (Bowman and Dahl, 2021). With large pre-trained models consistently improving state-of-the-art performance on NLP tasks (Devlin et al., 2018; Lewis et al., 2019), the performances of many of them have reached a plateau (Zhong et al., 2020; Fu et al., 2020). In other words, it is challenging to discriminate a better model using existing datasets (Wang et al., 2019). In this context, we ask the question:

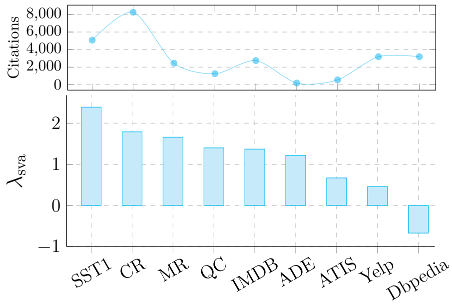


Figure 1: Illustrate different datasets' distinguishing ability w.r.t top-scoring systems characterized by our measure $\log(\lambda_{\text{sva}})$ on text classification and their corresponding citations.

are all benchmark's datasets necessary? We use the text classification task as a case study and try to answer the following two sub-questions:

RQ1: *How can we quantify the distinguishing ability of benchmark datasets?* To answer this question, we first design measures with varying calculation difficulties (§4) to judge datasets' discrimination ability based on top-scoring systems' performances. By exploring correlations among different measures, we then evaluate how reliable a dataset's discrimination is when discrimination is calculated solely based on overall results that top-scoring systems have achieved and generalize this measure to other NLP tasks. Fig. 1 illustrates how different text classification datasets are ranked (the bottom one) based on measures devised in this work (a smaller value suggests lower discrimination) and the corresponding citations of these datasets (the upper one). One can observe that: (i) The highly-cited dataset `DBpedia` (Zhang et al., 2015) (more than 3,000 times since 2015) shows the worst discriminative power. (ii) By contrast, dataset like `ADE` (Gurulingappa et al., 2012) (less than 200 times since 2012) does better in distinguishing top-scoring systems, suggesting that some of the relatively neglected datasets are actually valuable in distinguishing models. This phenomenon

*Corresponding author
[1]https://datalab.nlpedia.ai

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2930 - 2941
July 10-15, 2022

shows the significance of quantifying the discriminative ability of datasets: it can not only help us to **eliminate** those with lower discrimination from *commonly-used datasets* (e.g., DBpedia), but also help us to **recognize** the missing pearl in *seldom used* datasets (e.g., ADE and ATIS (Hemphill et al., 1990)).

RQ2: *Can we try to predict the discriminative power of the dataset?* Given a dataset, we investigate if we can judge its ability to distinguish models based on its characteristics (e.g., average sentence length), which is motivated by the scenario where a new dataset has just been constructed without sufficient top-scoring systems to calculate discrimination defined in RQ1. To answer this question, inspired by recent literature on performance prediction (Domhan et al., 2015; Turchi et al., 2008; Birch et al., 2008; Xia et al., 2020; Ye et al., 2021), we conceptualize this problem as a *discrimination regression task*. We define 11 diverse features to characterize a text classification dataset and regress its discrimination scores using different parameterized models. Preliminary experiments (§5.4) indicate that a meaningful regressor can be learned to estimate the discrimination of unseen datasets without actual training using top-scoring systems.

We brief **takeaways** in this work based on our observations:

(1) Not all datasets in *benchmark* are necessary in terms of model selection[2]: empirical results show that following datasets struggle at discriminating current top-scoring systems: STS-B and SST-2 from GLUE (Wang et al., 2018); BUCC and PAWX-X from XTREME, which is consistent with the concurrent work (Ruder et al., 2021) (§4.3.2).

(2) In regard to single-task benchmark datasets, for Chinese Word Segmentation task, there are multiple datasets (MSR, CityU, CTB) (Tseng et al., 2005; Jin and Chen, 2008) that exhibit much worse discriminative ability, suggesting that: future works on this task are encouraged to either (i) adopt other datasets to evaluate their systems or (ii) at least make significant test [3] if using these datasets. Similar observations happen in the dataset CoNLL-2003 (Sang and De Meulder, 2003) from Named Entity Recognition task and MultiNLI (Williams et al., 2017) from natural language inference task (§4.3.2).

(3) Some seldom used datasets such as ADE from text classification are actually better at distinguishing top-performing systems, which highlights an interesting and necessary future direction: *how to identify infrequently-used but valuable (better discrimination) datasets for NLP tasks, especially in the age of dataset's proliferation?*[4] (§4.2)

(4) Quantifying a dataset's discrimination (w.r.t top-scoring systems) by calculating the statistical measures (defined in §4.1.2) from leaderboard's results is a straightforward and effective way. But for those datasets without rich leaderboard results,[5] predicting the discrimination based on datasets' characteristics would be an promising direction (§4.3.1).

Our **contributions** can be summarized as:

(1) We try to quantify the discrimination ability for datasets by designing two variance-based measures. (2) We systematically investigate 4 text classification models on 9 datasets, providing the newest baseline performance for those seldom used datasets. All datasets and their features are released on DataLab (Xiao et al., 2022). (3) We study several popular NLP benchmarks, including GLUE, XTREME, NLI, and so on. Some valuable suggestions and observations will make research easier.

2 Related Work

Benchmarks for NLP In order to conveniently keep themselves updated with the research progress, researchers recently are actively building evaluation benchmarks for diverse tasks so that they could make a comprehensive comparison of systems, and use a leaderboard to record the evolving process of the systems of different NLP tasks, such as SQuAD (Rajpurkar et al., 2016), GLUE (Wang et al., 2018), XTREME (Hu et al., 2020), GEM (Gehrmann et al., 2021) and GENIE (Khashabi et al., 2021). Despite their utility, more recently, Bowman and Dahl (2021) highlight that unreliable and biased systems score so highly on standard benchmarks that there is little room for researchers who develop better systems to demonstrate their improvements. In this paper, we make a pilot study on meta-evaluating benchmark evalu-

[2]Caveat: Annotated datasets are always valuable, because the supervision signals provided there can not only help us directly train a system for specific use case, but also provide good supervised transfer for related tasks (Sanh et al., 2021).

[3]We randomly select 10 recently published papers (from ACL/EMNLP) that utilized these datasets and found only 2 of them perform significant test.

[4]https://paperswithcode.com/datasets

[5]The measure can keeps updated as the top-scoring systems of the leaderboard evolves, which can broaden its practical applicability

ation datasets and quantitatively characterize their discrimination in different top-scoring systems.

Performance Prediction Performance prediction is the task of estimating a system's performance without the actual training process. With the recent booming of the number of machine learning models (Goodfellow et al., 2016) and datasets, the technique of performance prediction become rather important when applied to different scenarios ranging from early stopping training iteration (Kolachina et al., 2012), architecture searching (Domhan et al., 2015), and attribution analysis (Birch et al., 2008; Turchi et al., 2008). In this work, we aim to calculate a dataset's discrimination without actual training top-scoring systems on it, which can be formulated as a performance prediction problem.

3 Preliminaries

3.1 Task and Dataset

Text classification aims to assign a label defined beforehand to a given input document. In the experiment, we choose nine datasets, and their statistics can be found in the Appendix A.

- **IMDB** (Maas et al., 2011) consists of movie reviews with binary classes.
- **Yelp** (Zhang et al., 2015) is a part of the Yelp Dataset Challenge 2015 data.
- **CR** (Hu and Liu, 2004) is a product review dataset with binary classes.
- **MR** (Pang and Lee, 2005) is a movie review dataset collected from Rotten Tomatoes.
- **SST1** (Socher et al., 2013) is collected from HTML files of Rotten Tomatoes reviews with fully labeled parse trees.
- **DBpedia14** (Zhang et al., 2015) is a dataset for ontology classification collected from DBpedia.
- **ATIS** (Hemphill et al., 1990) is an intent detection dataset that contains audio recordings of flight reservations.
- **QC** (Li and Roth, 2002) is a question classification dataset.
- **ADE** (Gurulingappa et al., 2012) is a subset of "Adverse Drug Reaction Data".

3.2 Model

We re-implement 4 top-scoring systems with typical neural architectures for each dataset. [6] The brief introduction of the four models is as follows.

- **LSTM** (Hochreiter and Schmidhuber, 1997) is a widely used sentence encoder. Here, we adopt the bidirectional LSTM.
- **LSTMAtt** is proposed by Lin et al. (2017) that designed the self-attention mechanism to extract different aspects of features for a sentence.
- **BERT** (Devlin et al., 2018) was utilized to fine-tuning on our text classification datasets.
- **CNN** is a CNN-based text classification model (Kim, 2014) was expolred in our work.

Except for BERT, the other three models (e.g. `LSTM`) are initialized by GloVe (Pennington et al., 2014) or Word2Vec (Mikolov et al., 2013) pre-trained word embeddings. When the performance on the dev set doesn't improve within 20 epochs, the training will be stopped, and the best performing model will be kept. More detailed model parameter settings can be found in the Appendix B.

4 How to Characterize Discrimination?

To achieve this goal, we design measures based on the performance of different models for a dataset.

4.1 Measures

We design several measures to judge dataset's distinguishing ability based on the performances that top-performing systems have achieved on it.[7] Specifically, given a dataset D together with k top-scoring model *performance list* $\mathbf{v} = [v_1, \cdots, v_k]$, we define the following measures.

4.1.1 Performance Variance

We use the standard deviation to quantify the degree of variation or dispersion of a set of performance values. A larger value of λ_{var} suggests that the discrimination of the given dataset is more significant. λ_{var} can be defined as:

$$\lambda_{\text{var}} = \text{Std}(\mathbf{v}), \tag{1}$$

where $\text{Std}(\cdot)$ is the function to compute the standard deviation. Assume that the performance list ($k = 3$) on dataset D is $\mathbf{v} = [88, 92, 93]$, we can get $\lambda_{\text{var}} = 2.65$.

[6] We mainly focus on neural network-based models, since most top-scoring systems in the leaderboard are based on deep learning.

[7] A dataset's discrimination is defined w.r.t top-scoring models from a leaderboard, keeping itself updated with systems' evolution.

4.1.2 Scaled Performance Variance

For the above measure, it can only reflect the variances of the performance of different models, without considering whether the model's performance is close to the upper limit (e.g., 100% accuracy) on a given data set. To address this problem, we defined a modified variance by scaling λ_{var} with the difference between the upper limit performance u and average performance $\text{Avg}(\mathbf{v})$ of $\mathbf{v}$.

$$\lambda_{\text{sva}} = \lambda_{\text{var}}(u - \text{Avg}(\mathbf{v})). \tag{2}$$

In practice, u can be defined flexibly based on tasks' metrics. For example, in text classification task, u could be 100% (w.r.t F1 or accuracy), while in summarization task, u could be the results of oracle sentences (w.r.t ROUGE). Intuitively, given a performance list on text classification dataset: $\mathbf{v} = [88, 92, 93]$, we can obtain the $\lambda_{\text{sva}} = 23.81$.

4.1.3 Hit Rate

The previous two measures quantify dataset's discriminative ability w.r.t k top-performing systems in an *indirect* way (i.g, solely based on the overall results of different models). However, sometimes, small variance does not necessarily mean that the dataset fail to distinguish models, as long as the difference between models is statistically significant. To overcome this problem, we borrow the idea of bootstrap-based significant test (Koehn, 2004) and define the measure *hit rate*, which quantify the degree to which a given dataset could successfully differentiate k top-scoring systems.

Specifically, we take all $\binom{k}{2}$ pairs of systems (m_i and m_j) and compare their performances on a subset of test samples D_t that is generated using paired bootstrap re-sampling. Let $v_i(D) > v_j(D)$ be the performance of m_1 and m_2 on the full test set, we define $P(m_i, m_j)$ as the frequency of $v_i(D_t) > v_j(D_t)$ over all T times of re-sampling ($t = 1, \cdots, T$). [8] Then we have

$$\lambda_{\text{hit}} = \frac{1}{\binom{k}{2}} \sum P(m_i, m_j) \tag{3}$$

Metric Comparison The first two metrics, performance variance and scaled performance variance, are relative easily to obtain since they only require holistic performances of different top-scoring models on a given dataset, which can be conveniently collected from existing leaderboards. By contrast, although the metric *hit rate* can directly reflect dataset's ability in discriminating diverse systems, its calculation not only require more fine-grained information of system prediction but also complicated bootstrap re-sampling process.

[8] For example, given a test set with 1000 samples, we sample 80% subset from it and repeat this process T times.

4.2 Exp-I: Exploring Correlation Between Variance and Hit Rate

The goal of this experiment is to investigate the reliability of the variance-based discrimination measures (e.g., λ_{sva}), which are easier to obtain, by calculating its correlation with significant test-based measure λ_{hit}, which is costly to get. Since the implementation of λ_{hit} relies on the bootstrap-based significant test, we choose text classification as the tested and re-implement 4 classification models (defined in Sec. 3.2) on 9 datasets. The performance and the distinction degree on the 9 text classification dataset are shown in Tab. 1. λ_{var} and λ_{sva} measures are designed based on performance variance, even if BERT always achieves the best performance on the same dataset, it will not affect the observed results from our experiments.

Correlation measure Here, we adopt the Spearman rank correlation coefficient (Zar, 1972) to describe the correlation between our variance-based measures and the hit rate measure λ_{hit}.

$$S_\lambda = \text{Spearman}(q, \lambda_{\text{hit}}), \tag{4}$$

where the q can be λ_{var} or λ_{sva}.

Result (1) λ_{var} and λ_{sva} are strong correlative (S_λ>0.6) with λ_{hit} respectively, which suggests that variance-based metrics could be a considerably reliable alternatives of significant test-based metric.
(2) $\text{Spearman}(\lambda_{\text{var}}, \lambda_{\text{hit}}) > \text{Spearman}(\lambda_{\text{sva}}, \lambda_{\text{hit}})$, which indicate that comparing with λ_{sva}, dataset discrimination characterized by λ_{var} is more acceptable for λ_{hit}. The reason can be attributed to that the designing of the measure λ_{hit} does not consider the upper limit of the model's performance.
(3) `DPdedia` and `Yelp` are commonly used text classification datasets, while they have the worst ability to discriminate the top-scoring models since they get the lowest value of λ_{var} and λ_{sva}. By contrast, these two seldom used datasets `ADE` and `ATIS` show the better discriminative ability.

4.3 Exp-II: Evaluation of Other Benchmarks

4.3.1 Popular Benchmark Datasets

We also investigate how benchmark datasets from other NLP task perform using two devised mea-

Method	BERT	LSTMAttr	LSTM	CNN	λ_{hit}	λ_{var}	λ_{sva}
SST1	54.12	43.80	47.60	44.80	0.88	4.65	243.56
CR	91.75	83.25	82.50	84.25	0.91	4.27	62.17
MR	85.55	79.92	79.80	82.00	0.86	2.69	48.83
QC	97.19	90.36	89.96	92.17	0.92	3.32	25.18
IMDB	93.34	89.45	89.65	87.81	0.87	2.33	23.18
ADE	93.48	92.90	92.65	89.54	0.78	1.77	13.90
ATIS	97.64	97.42	97.31	94.62	0.78	1.42	4.63
Yelp	97.52	96.60	96.60	95.46	0.81	0.84	2.91
DPedia	99.27	99.01	99.05	98.75	0.68	0.22	0.21
Spearman						0.83	0.73

Table 1: Illustration the 4 models' performance and discrimination degree (characterized by λ_{hit}, λ_{var}, and λ_{sva}) on 9 text classification datasets. The two correlation coefficients pass the significance test ($p < 0.05$). λ_{var} and λ_{sva} measures are designed based on performance variance.

sures. Specifically, we collected three single-task and two multitask benchmarks. For the single-task benchmarks, we collect the top-performing models in a specific period for each dataset, provided by Paperswithcode. [9] For the multitask benchmarks, here, the GLUE [10] and XTREME [11] are considered in this work. Since Paperswithcode provided 5 models for each dataset in most case, for fairness and uniformity, we keep top-5 models for both single-task and multitask benchmark datasets.

Named Entity Recognition (NER) aims to identify named entities of an input text, for which we choose 5 top-scoring systems on 6 datasets and collect results from Paperswithcode.

Chinese Word Segmentation (CWS) aims to detect the boundaries of Chinese words in a sentence. We select 5 top-scoring systems on 8 datasets and collect results from Paperswithcode.

Natural Language Inference (NLI) targets at predicting whether a premise sentence can infer the hypothesis sentence. We select 5 top-performing models on 4 datasets from Paperswithcode.

GLUE (Wang et al., 2018) covers 9 sentence- or sentence-pair tasks with different dataset sizes, text genres, and degrees of difficulty. Fig. 2-(a) shows the tasks/datasets that are considered in GLUE.

XTREME (Hu et al., 2020) is the first benchmark that evaluates models across a wide variety of languages and tasks. The tasks/datasets that are covered by XTREME are shown in Fig. 2-(b).

4.3.2 Results and Analysis

Fig. 2 shows the results of dataset quality measure by λ_{var} and λ_{sva}. We detail several main observations:

- λ_{var} and λ_{sva} have consistent evaluation results for both single-task (CWS, NER, NLI) and multitask (GLUE, XTREME) benchmarks.
- For the XTREME benchmark, `BUCC` and `PAWSX` have lowest λ_{var} and λ_{sva}, which suggest that they are hardly to discriminate the top-performing systems. Moreover, these two data sets will be removed from the new version of the XTREME leaderboard called XTREME-R (Ruder et al., 2021). This consistent observation also shows the effectiveness of our measure.
- For GLUE benchmark, `CoLA`, `QQP`, and `RTE` have the excellent ability to distinguish different top-scoring models (with higher λ_{var} and λ_{sva}), while the `SST-2` and `STS-B` perform worse.
- For CWS benchmarks, there is a larger gap between the value of λ_{var} and λ_{sva}, which indicate that the performance of top-scoring models considered are close to 100%. Furthermore, `MSR`, `CityU` and `CTB` are not suitable as benchmarks since they have poor discrimination ability with $\lambda_{\text{sva}} < 0$. So as `MultiNLI` for NLI task.
- `CoNLL 2003` is a widely used NER dataset, but it is the lowest quality dataset under our dataset quality measure. The reason can be attributed to contain much annotation errors (Fu et al., 2020) in the `CoNLL 2003` dataset, which makes its performance reach the bottleneck.

5 Can we Predict Discrimination?

Although metrics λ_{var}, λ_{sva} ease the burden for us to calculate the datasets' discrimination, one major limitation is: given a new dataset without results from leaderboards, we need to train multiple top-scoring systems and calculate corresponding results on it, which is computationally expensive. To alleviate this problem, in this section, we focus on text

[9] https://paperswithcode.com/
[10] https://gluebenchmark.com/
[11] https://sites.research.google/xtreme

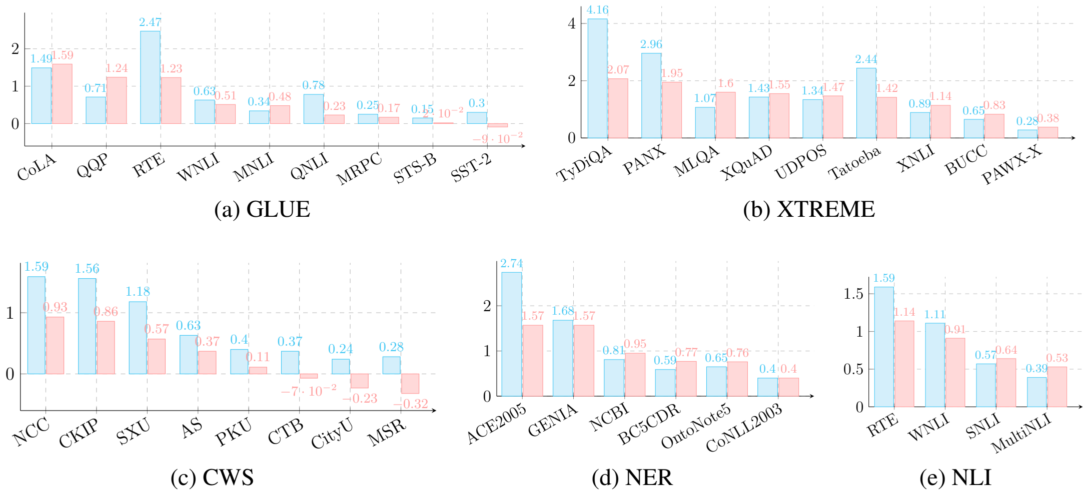


Figure 2: The dataset discrimination characterized by $\log(\lambda_{\text{var}})$ (the logarithm for better visualization) (blue) and $\log(\lambda_{\text{sva}})$ (pink) on five popular NLP benchmarks.

classification task and investigate the possibility of estimating datasets' discrimination solely based on their characteristics without actual training systems on them.

5.1 Task Formulation

5.1.1 Regression-based Task Formulation

We formulate it as a performance prediction problem (Birch et al., 2008; Xia et al., 2020; Ye et al., 2021). Formally, we refer to $\mathcal{M}$, D^{tr} , D^{te}, $\mathcal{S}$ as the machine learning system, training data, test data and training strategy respectively. The goal of performance prediction is to estimate actual performance y without actual training by using features of $\mathcal{M}$, $\mathcal{D}^{tr}$, $\mathcal{D}^{te}$, and $\mathcal{S}$.

$$\hat{y} = \hat{f}(\Phi_{\mathcal{M}}, \Phi_{\mathcal{D}^{tr}}, \Phi_{\mathcal{D}^{te}}, \Phi_{\mathcal{S}}; \hat{\Theta}) \tag{5}$$

where $\hat{y}$ denotes estimated prediction and $\Phi(\cdot)$ is a feature extractor. Following Xia et al. 2020, we only use the features of the datasets as variables and adapt it to our discriminative prediction scenario, we can obtain:

$$\hat{\lambda} = \hat{f}(\Phi_{\mathcal{D}^{tr}}, \Phi_{\mathcal{D}^{te}}; \hat{\Theta}), \tag{6}$$

where $\hat{\lambda}$ denotes predicted variance defined in §4.1.2 such as λ_{var} or λ_{sva}.

5.1.2 Ranking-based Task Formulation

Instead of only regressing one dataset's quality, we also care about the quality ranking of different datasets w.r.t discriminating systems in a task. Therefore, we also formulate it as a listwise LTR(learning to rank) task where a model takes individual lists as instances, to predict the rank of element among the list (Liu, 2011). Given a set of n datasets $d = \{d_1, d_2, \cdots, d_n\}$ ($d \in D = \{D^{tr}, D^{te}\}$), different d construct the dataset of LTR task, the target of the ranker is to predict the dataset quality ranking for each dataset in d according to the datasets' features. The estimated rankings $\overline{\lambda} = \{\lambda_1, \lambda_2, \cdots, \lambda_n\} \in [1, n]$ for set d can be defined as:

$$\overline{\lambda} = \overline{f}(\Phi_{(d)}; \overline{\Theta}), \tag{7}$$

where $\Phi(\cdot)$ is the dataset feature extractor, $\overline{f}$ is the ranking model. $\overline{\lambda} \in [1, n]$ is the estimated rankings of the variance (λ_{var} or λ_{sva}) for datasets in set d.

5.2 Characterization of Datasets

In this section, we will introduce three aspects that characterize datasets: Inherent Feature, Lexical Feature, and Semantic Feature. Due to space limitations, we move a more detailed feature introduction to the Appendix C.

5.2.1 Inherent Feature

Average length (ϕ_{len}): The average sentence length on a dataset, where the number of tokens on a sentence is considered as the sentence length. **Label number (ϕ_{lab}):** The number of labeled classes in a dataset. **Label balance (ϕ_{bal}):** The label balance metric measures the variance between the ideal and the true label distribution.

5.2.2 Lexical Feature

Basic English Words Ratio (ϕ_{basic}): The proportion of words belonging to the 1000 basic English [12] words in the whole dataset. **Type-Token Ratio (ϕ_{ttr}):** We measure the text lexical richness by the type-token ratio (Richards, 1987) based on the lexical richness tool. [13] **Language Mixedness Ratio (ϕ_{lmix}):** To detect the ratio of other languages mixed in the text, we utilize the models proposed by Joulin et al. (2016b) for language identification from fastText (Joulin et al., 2016a) which can recognize 176 languages. **Pointwise Mutual Information (ϕ_{pmi}):** PMI [14] is a measurement to calculate the correlation between variables.

5.2.3 Semantic Feature

Perplexity (ϕ_{ppl}): We calculate the perplexity [15] based on GPT2 (Radford et al., 2019) to evaluate the quality of the text. **Grammar Errors Ratio (ϕ_{gerr}):** We adopt the detection tool [16] to recognize words with grammatical errors, and then calculate the ratio of grammatical errors. **Flesch Reading Ease [17] (ϕ_{fre}):** To describe the readability of a text, we introduce the ϕ_{fre} achieving by textstat. [18]

For feature ϕ_{len}, ϕ_{ttr},ϕ_{lmix}, ϕ_{gerr}, ϕ_{pmi}, ϕ_{fre}, and $\phi_{\text{r}_{\text{fre}}}$, we individually compute $\phi()$ on the training, test set, as well as their interaction. Take average length (ϕ_{len}) as an example, we compute the average length on training set $\phi_{\text{tr,len}}$, test set $\phi_{\text{te,len}}$, and their interaction $((\phi_{\text{tr,len}} - \phi_{\text{te,len}})/\phi_{\text{tr,len}})^2$.

5.3 Parameterized Models

The dataset discrimination prediction (ranking) model takes a series of dataset features as the input and then predicts discrimination(rank) based on $\hat{f}(\cdot)$ ($\overline{f}(\cdot)$) defined in Eq. 6 (Eq. 7). We explore the effectiveness of four variations of regression methods and two ranking frameworks.

Regression Models: **LightGBM** (Ke et al., 2017) is a gradient boosting framework with faster training and better performance than XGBoost. **K-nearest Neighbor (KNN)** (Peterson, 2009) is a non-parametric model that makes the prediction by exploring the k neighbors. **Support Vector Machine (SVM)** (Suykens and Vandewalle, 1999) uses kernel trick to solve both linear and non-linear problems. **Decision Tree (DT)** (Quinlan, 1990) is a tree-based algorithm that gives an understandable interpretation of predictions.

Ranking Frameworks: LightGBM with Gradient Boosting Decision Tree (Friedman, 2001) boosting strategy was selected as our ranking model. **XGBoost** (Chen and Guestrin, 2016) with gbtree(Hastie et al., 2009) boosting strategy was another ranking model.

5.4 Experiments

5.4.1 Data Construction

To construct a collection with large amount of discriminative datasets, we randomly select three dataset features (e.g. average sentence length ϕ_{len}) to divide the original dataset into several non-overlapping sub-datasets. As a result, we collect 987 sub-datasets. Then, we train four text classification models (CNN, LSTM, LSTMAtt, BERT) on these sub-dastasets. Next, we calculate the dataset features ϕ (defined in Sec. 5.2) and dataset discrimination ability λ_{sva} and λ_{var} on these sub-datasets. **Regression Task Settings** ϕ and λ_{sva} (λ_{var}) will be the input and target of the regression models, as defined by Eq. 6. For the experiment setting, we randomly select 287 (ϕ, λ_{sva} (λ_{var})) pairs as the test set and the rest as the training set (700). **Ranking Task Settings** We construct datasets for ranking task from the dataset used in regression task. Here, we explored the value of n (defined in §5.1.2) to be 5, 7 and 9 to randomly choose samples from D^{tr} (or D^{te}) to construct the datasets for the ranking task, and kept 4200, 600, 1200 samples for training, development and testing set respectively.

5.4.2 Evaluation Metric

Regression Task We use RMSE (Chai and Draxler, 2014) and Spearman rank correlation coefficient (Zar, 1972) to evaluate how well the regression model predicts the discriminative ability for datasets. The Spearman rank correlation coefficient is used for the correlation between the output of a regression model and the ground truth.

Ranking Task NDCG (Järvelin and Kekäläinen, 2000) and MAP (Yue et al., 2007) are the evalua-

[12] `https://simple.wikipedia.org/wiki/Wikipedia:List_of_1000_basic_words`
[13] `https://github.com/LSYS/lexicalrichness`
[14] `https://en.wikipedia.org/wiki/Pointwise_mutual_information`
[15] `https://en.wikipedia.org/wiki/Perplexity`
[16] `https://github.com/jxmorris12/language_tool_python`
[17] `https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests`
[18] `https://github.com/shivam5992/textstat`

tion metric of our ranking task. For NDCG, it considers the rank of a set of discriminative abilities. In our setting, every dataset has its own real discriminative ability. Here, We transfer the predicted discriminative ability to the rank of the dataset in the NDCG metric, so we can use NDCG to evaluate the model's predicted effect. For MAP, it likes how NDCG works, but it considers a set of binary values. Here, we set a threshold value of $\lambda_{\text{var}} = 3$ ($\lambda_{\text{sva}} = 28$) for λ_{var} (λ_{sva}) to distinguish the dataset discrimination ability from good (relevant) to bad (irrelevant).

Method	RMSE		Spearman			
			λ_{var}		λ_{sva}	
	λ_{var}	λ_{sva}	corr	p	corr	p
KNN	2.42	51.21	0.77	9.75E-40	0.87	1.62E-63
LightGBM	1.53	32.74	0.72	2.23E-33	0.87	7.01E-61
DT	1.73	43.33	0.64	9.25E-25	0.84	1.33E-53
SVM	2.83	62.44	0.68	1.14E-28	0.77	7.26E-40

Table 2: The performance of regressing dataset discrimination for the text classification. "*corr*" denotes the "*correlation*".

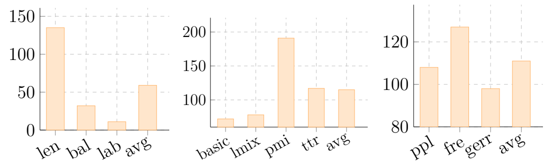
Model	n	NDCG		MAP	
		λ_{var}	λ_{svar}	λ_{var}	λ_{svar}
LightGBM	9	98.20	98.85	97.50	98.27
	7	97.76	98.73	97.01	99.05
	5	96.73	97.08	96.56	98.15
XGBoost	9	96.66	97.13	92.91	93.62
	7	96.74	97.65	94.77	96.11
	5	95.93	97.10	95.49	98.25

Table 3: The performance of ranking dataset discrimination for the text classification task. n is the number of datasets in d defined in §5.1.2

.

5.4.3 Results and Analysis

Tab. 2 and Tab. 3 show the results of four regression models and two ranking models that characterize the dataset discrimination ability, respectively. We can observe that: Both the regression models and the ranking models can well describe the discrimination ability of different datasets. For these four regression models, the prediction is highly correlated with the ground truth (with a correlation value larger than 0.6), passing the significance testing ($p < 0.05$). This suggests that the dataset discrimination can be successfully predicted. For these two ranking models, their performance on NDCG and MAP is greater than 95%, which indicates that the discriminative ability of the data set can be easily ranked.



(a) Inherent (b) Lexical (c) Semantic

Figure 3: Feature importance for the text classification measured by LGBoost with the target of λ_{sva}.

Feature Importance Analysis Fig. 3 illustrates the feature importance characterized by LightGBM. For a given feature, the number of times that is chosen as the splitting feature in the node of the decision trees is defined as its importance degree. We observe that: (1) The most influential features are ϕ_{pmi}, ϕ_{len}, and ϕ_{fre}, which come from the lexical, inherent, and semantic features, respectively. This indicated that the LightGBM can extract features from different aspects to make predictions. (2) In the perspective of feature groups, the semantic features are more influential than the inherent features and lexical features.

6 Discussion & Implications

Discussion Given a leaderboard of a dataset, metrics explored in this paper can be easily used to calculate its discrimination, while some limitations still exist. We make some discussion below to encourage more explorations on new measures: (a) **Interpretability**: current metrics can only identify which datasets are of lower indiscriminability while don't present more explanation why it is the case. (b) **Functionality**: a dataset with lower discrimination doesn't mean it's useless since the supervision signals provided there can not only help us directly train a system for the specific use case but also provide good supervised transfer for related tasks. Metrics designed in this work focus on the role of discriminating models.

Calls Based on observations obtained from this paper, we make the following calls for future research: (1) Datasets' discrimination ability w.r.t top-scoring systems could be included in the dataset schema (such as dataset statement (Bender and Friedman, 2018)), which would allow researchers to gain a saturated understanding of the dataset. (2) Leaderboard constructors could also report the discriminative ability of the datasets they aim to include. (3) Seldom used datasets are also valuable for model selection, and a more fair dataset searching system should be investigated, for example, relevance- and scientifically meaningful first, instead of other biases, like popularity.

Acknowledgements

We would like to thank Graham Neubig and the anonymous reviewers for their valuable comments. This work was supported by the National Research Foundation of Singapore under its Industry Alignment Fund – Pre-positioning (IAF-PP) Funding Initiative. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not reflect the views of the National Research Foundation of Singapore.

References

Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Alexandra Birch, Miles Osborne, and Philipp Koehn. 2008. Predicting success in machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 745–754, Honolulu, Hawaii. Association for Computational Linguistics.

Samuel R. Bowman and George E. Dahl. 2021. What will it take to fix benchmarking in natural language understanding? *CoRR*, abs/2104.02145.

Tianfeng Chai and Roland R Draxler. 2014. Root mean square error (rmse) or mean absolute error (mae)?– arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. 2015. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Jinlan Fu, Pengfei Liu, and Qi Zhang. 2020. Rethinking generalization of neural models: A named entity recognition case study. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7732–7739. AAAI Press.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna Clinciu, Dipanjan Das, Kaustubh D Dhole, et al. 2021. The gem benchmark: Natural language generation, its evaluation and metrics. *arXiv preprint arXiv:2102.01672*.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, 45(5):885–892. Text Mining and Natural Language Processing in Pharmacogenomics.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *Boosting and Additive Trees*, pages 337–387. Springer New York, New York, NY.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *CoRR*, abs/2003.11080.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 168–177, New York, NY, USA. Association for Computing Machinery.

Kalervo Järvelin and Jaana Kekäläinen. 2000. IR evaluation methods for retrieving highly relevant documents. In *SIGIR 2000: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 24-28, 2000, Athens, Greece*, pages 41–48. ACM.

Guangjin Jin and Xiao Chen. 2008. The fourth international chinese language processing bakeoff: Chinese word segmentation, named entity recognition and chinese pos tagging. In *Proceedings of the sixth SIGHAN workshop on Chinese language processing*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3146–3154.

Daniel Khashabi, Gabriel Stanovsky, Jonathan Bragg, Nicholas Lourie, Jungo Kasai, Yejin Choi, Noah A Smith, and Daniel S Weld. 2021. Genie: A leaderboard for human-in-the-loop evaluation of text generation. *arXiv preprint arXiv:2101.06561*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Prasanth Kolachina, Nicola Cancedda, Marc Dymetman, and Sriram Venkatapathy. 2012. Prediction of learning curves in machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22–30, Jeju Island, Korea. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ArXiv*, abs/1910.13461.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130.

Tie-Yan Liu. 2011. Learning to rank for information retrieval.

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

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *CoRR*, abs/cs/0506075.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Leif E Peterson. 2009. K-nearest neighbor. *Scholarpedia*, 4(2):1883.

John Ross Quinlan. 1990. Probabilistic decision trees. In *Machine Learning*, pages 140–152. Elsevier.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

Brian Richards. 1987. Type/token ratios: what do they really tell us? *Journal of Child Language*, 14(2):201–209.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Graham Neubig, and Melvin Johnson. 2021. XTREME-R: towards more challenging and nuanced multilingual evaluation. *CoRR*, abs/2104.07412.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun

Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

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

Johan AK Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter for sighan bakeoff 2005. In *Proceedings of the fourth SIGHAN workshop on Chinese language Processing*, volume 171.

Marco Turchi, Tijl De Bie, and Nello Cristianini. 2008. Learning performance of a machine translation system: a statistical and computational analysis. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 35–43.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

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

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Mengzhou Xia, Antonios Anastasopoulos, Ruochen Xu, Yiming Yang, and Graham Neubig. 2020. Predicting performance for natural language processing tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8625–8646, Online. Association for Computational Linguistics.

Yang Xiao, Jinlan Fu, Weizhe Yuan, Vijay Viswanathan, Zhoumianze Liu, Yixin Liu, Graham Neubig, and Pengfei Liu. 2022. Datalab: A platform for data analysis and intervention. *CoRR*, abs/2202.12875.

Zihuiwen Ye, Pengfei Liu, Jinlan Fu, and Graham Neubig. 2021. Towards more fine-grained and reliable NLP performance prediction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3703–3714, Online. Association for Computational Linguistics.

Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. 2007. A support vector method for optimizing average precision. In *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, pages 271–278. ACM.

Jerrold H Zar. 1972. Significance testing of the spearman rank correlation coefficient. *Journal of the American Statistical Association*, 67(339):578–580.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

A Statistics of Datasets

Tab. 4 shows the statistical information of the nine datasets of text classification task used in our work. For those datasets without explicit the development set, we randomly selected 12.5% samples from the training set as the development set.

Dataset	Train	Test	Development
IMDB	25,000	25,000	-
Yelp	560,000	38,000	-
QC	5,452	500	-
DPedia	560,000	70,000	-
CR	3,594	400	-
ATIS	4,978	893	-
SST1	8,544	2,210	1,101
MR	9,596	1,066	-
ADE	23,516	-	-

Table 4: Statistics of datasets.

B Parameter Settings for Text Classification Model

In this section, we will introduce the parameter settings of the neural network-based models explored in Section 3.2. The optimizer is AdamW for the four mdoels. The settings of other parameters are shown in Tab. 5.

Parameter	BERT	CNN	LSTM	LSTMAtt
learning rate	2*e-5	1*e-4	1*e-3	1*e-3
batch size	4	4	32	32
word emb	-	Word2vec	GloVe	GloVe
word emb size	-	300	300	300
hidden size	768	120	256	256
max sent len	512	-	-	-
filter size	-	1,3,5	-	-

Table 5: the parameters of four models.

C Characterization of Datasets

C.1 Inherent Feature

Label balance (ϕ_{bal}): The label balance metric measures the variance between the ideal and the true label distribution: $\phi_{\text{bal}} = (c_t - c_s)/c_s$, where the c_t and c_s are the true and ideal label information entropy (Shannon, 1948), respectively.

C.2 Lexical Feature

Type-Token Ratio (ϕ_{ttr}): TTR (Richards, 1987) is a way to measure the documents lexical richness: $\phi_{\text{ttr}} = n_{\text{type}}/n_{\text{token}}$, where the n_{type} is the number of unique words, and n_{token} is the number of tokens. We use lexical richness [19] to calculate the TTR for each sentence and then average them.

Language Mixedness Ratio (ϕ_{lmix}): The proportion of sentence that contains other languages in the whole dataset. To detect the mixed other languages, we utilize the models proposed by Joulin et al. (2016b) for language identification from fastText (Joulin et al., 2016a) which can recognize 176 languages.

Pointwise Mutual Information (ϕ_{pmi}): is a measurement to calculate the correlation between variables. Specifically, for a word in one class $\phi_{\text{pmi(c,w)}} = \log(\frac{p(c,w)}{p(c)p(w)})$, where $p(c)$ is the proportion of the tokens belonging to label c, $p(w)$ is the proportion of the word w, and $p(c,w)$ is the proportion of the word w which belongs to class c. For every class, all the $\phi_{\text{pmi(c,w)}}$, larger than zero, are added to get the sum, which serve as the dataset's pmi. Finally,ϕ_{pmi} is calculated by dividing the sum by the numbers of pairs(c,w) of the train dataset. We pick up the top-ten words sorted by $\phi_{\text{pmi(c,w)}}$ in all classes, then the ration related to the class-related word($\phi_{\text{r}_{\text{pmi}}}$) is calculated by dividing the number of samples who contain the top-ten words by the total samples in the train set.

C.3 Semantic Feature

Grammar errors ratio (ϕ_{gerr}): The proportion of words with grammatical errors in the whole dataset. We adopt the detection tool [20] to recognize words with grammatical errors. We first compute the grammar errors ratio for each sentence: n/m, where the n and m denote the number of words with grammatical errors and the number of the token for a sentence, averaging them.

Flesch Reading Ease (ϕ_{fre}): Flesch Reading Ease [21] calculated by textstat [22] is a way to describe the simplicity of a reader who can read a text. First, we calculate the ϕ_{fre} for each sample, and then average them as the dataset's feature. Then we pick out the samples whose score below 60, then the ration related to the low score samples($\phi_{\text{r}_{\text{fre}}}$) is calculated by dividing the number of the picked samples by the total samples in the train set.

[19] https://github.com/LSYS/lexicalrichness

[20] https://github.com/jxmorris12/language_tool_python

[21] https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests

[22] https://github.com/shivam5992/textstat