Why only Micro-F_1? Class Weighting of Measures for Relation Classification

David Harbecke♣, Yuxuan Chen♣, Leonhard Hennig♣, Christoph Alt♠♡
♣German Research Center for Artificial Intelligence (DFKI), Berlin
♠Humboldt Universität zu Berlin ♡Science of Intelligence
♣{firstname}.{lastname}@dfki.de
♠christoph.alt@posteo.de

Abstract

Relation classification models are conventionally evaluated using only a single measure, e.g., micro-F_1, macro-F_1 or AUC. In this work, we analyze weighting schemes, such as *micro* and *macro*, for imbalanced datasets. We introduce a framework for weighting schemes, where existing schemes are extremes, and two new intermediate schemes. We show that reporting results of different weighting schemes better highlights strengths and weaknesses of a model.

1 Introduction

Relation classification (RC) models are typically compared with either micro-F_1 or macro-F_1, often without discussing the measure's properties (see e.g. Zhang et al., 2017; Yao et al., 2019). Each measure highlights different aspects of model performance (Sun et al., 2009). However, using an inappropriate measure can lead to the preference of an unsuitable model (Branco et al., 2016), e.g., tasks with an imbalanced or long-tailed class distribution. We argue that model evaluation should better reflect this, particularly as rare phenomena become more important in NLP (Rogers, 2021).

For instance, popular datasets for RC, such as TACRED (Zhang et al., 2017), NYT (Riedel et al., 2010), ChemProt (Kringelum et al., 2016), DocRED (Yao et al., 2019), and SemEval-2010 Task 8 (Hendrickx et al., 2010), often exhibit a highly imbalanced label distribution (see Table 1 and, e.g., the TACRED class distribution[1]). The main reasons are the natural data imbalance, i.e. the occurrence frequency of relation mentions in text, as well as the incompleteness of knowledge graphs like Freebase (Bollacker et al., 2008) used in distantly supervised RC. For example, 58% of the relations in the NYT dataset (Riedel et al., 2010) have fewer than 100 training instances (Han et al., 2018), and the most frequent relation *location/contains* is assigned to 48.3% of the positive test instances. However, for applying RC to real-world problems, it is especially important to discover instances of relations that are not yet covered well in a given knowledge base.

Table 1 lists statistics of the aforementioned RC datasets, including their perplexity and common evaluation measures. TACRED and the original version of NYT contain predominantly negative samples[2]. All datasets, except for undirectional SemEval, exhibit a large ratio between most frequent and least frequent positive class in the test set. The perplexity of test set distributions is also much lower than the relation count for all datasets except SemEval. Reporting only a single measure therefore cannot exhaustively capture model performance on these datasets, especially for the long tail of relation types. For example, Alt et al. (2019) show that on the NYT dataset, AUC scores and P-R-Curves of several state-of-the-art models are heavily skewed towards the two most frequent relation types *location/contains* and *person/nationality*. TACRED, ChemProt, DocRED and SemEval results are usually only reported in micro-F_1, which does not consider class membership.

In this paper, we introduce a framework for weighting schemes of measures to address these evaluation deficits. We present and motivate two new weighting schemes that are in between the extremes of micro- and macro-weighting. We demonstrate these, micro-, class-weighted- and macro-F_1 on TACRED and SemEval with two popular models each. We show that more information about models can be inferred from our results and point out what further steps should be taken to improve evaluation in relation classification.

[1] https://nlp.stanford.edu/projects/tacred/#stats

[2] Negative samples in RC means none of the dataset's relations hold. Depending on the dataset, this class is coined *no-relation*, *NA* or *Other*. We use negative class or *NA*.

Proceedings of NLP Power! The First Workshop on Efficient Benchmarking in NLP, pages 32 - 41
May 26, 2022 ©2022 Association for Computational Linguistics

Dataset	#Rel	#Samples	%NA	Perplexity w NA	 w/o NA	Ratio	Evaluation
TACRED	42	106264	79.5	3.31	23.39	250	micro-F_1
NYT	53	694491	79.4	1.27	7.84	2793	precision at k, AUC
	24	66194	0	6.24	6.24	2485	
ChemProt	13	10065	0	7.23	7.23	314	micro-F_1
DocRED	96	50503	0	33.13	33.13	2837	micro-F_1, AUC
SemEval	19	10717	17.4	14.45	14.37	291	macro-F_1 (official),
	10	10717	17.4	9.61	8.80	2.10	micro-F_1 (popular)

Table 1: Statistics for popular RC datasets. The number of relations, samples and percent of negative samples are for the whole dataset. Perplexity of the classes is given for the test set, with and without negative samples. This value would be equal to #Rel for a fully balanced dataset. Ratio is between the counts of the most and least frequent positive class of the test set. We also list the popular evaluation methods. The upper line for NYT indicates the original dataset by Riedel et al. (2010), the lower line is the frequently used version by Hoffmann et al. (2011). The upper SemEval entry considers the direction between the nominals, the lower one does not.

2 Methods

We first give background on the F_1-score and existing F_1 weighting schemes. We present our framework of weighting schemes. We introduce two new weighting schemes. Finally, we outline statistical tests.

2.1 Background

The F_β-score (Rijsbergen, 1979; Lewis and Gale, 1994) calculates a score in the interval $[0,1]$ through the formula

$$F_\beta = \frac{(1+\beta^2)\cdot TP}{(1+\beta^2)\cdot TP + \beta^2 \cdot FN + FP} \qquad (1)$$

with the true positives (TP), false negatives (FN) and false positives (FP) of a confusion matrix. This definition is identical to the weighted harmonic mean of precision and recall. The positive coefficient β is used as a trade-off between the error types FN and FP. If there is no preference known or pre-determined, this coefficient is usually set to 1. In multi-class classification the confusion matrix can either be calculated once for the whole dataset, or separately for each class. The former method yields micro-F_1.

Micro weighting does not consider class membership for any test sample. If the predictions and labels of all classes are considered, micro-F_1 is equal to accuracy, as the denominator in Eq. 1 is twice the dataset. In RC, the TP of the negative class are usually not considered, in which case micro-F_1 is not equal to accuracy. For the F-score, *micro* is the only weighting where the impact of a sample on the score is not conditioned on the model performance on the rest of the class (Forman and Scholz, 2010). If the test set is considered to have a representative data distribution, the micro-weighted score is a frequentist evaluation of model performance.

There exist two other ways to calculate and combine F_1-scores for a multi-class problem. First, multi-class F_1-scores can be calculated for each class and then a weighted average class score is taken. Second, precision and recall scores for each class can be calculated and weighted, then the harmonic mean of weighted precision and weighted recall is taken. Opitz and Burst (2019) show that the first method is more robust and less favorable to biased classifiers. We use this method in our proposed framework.

(Class-)weighted-F_1 is similar to micro-F_1. F_1-scores are calculated for each class individually and then weighted by the class count. Thus, both schemes approximately weigh all samples equally.

Macro weighting gives an equal weight for each class with positive sample count regardless of the specific sample count. This gives information about model performance if class imbalance is not considered.

In general, there is a correspondence between training loss and evaluation measure (Li et al., 2020). One disadvantage of multiple weighting schemes is that each weighting scheme can be optimized for. To achieve a better score for a specific weighting, class weights could be set proportional to the weighting of the class during training. How-

Method	Formula	Focus
Micro	-	calculation over dataset, class membership is not considered
Weighted	n_i	weighting all classes by instance count, similar to micro
Dodrans	${n_i}^{3/4}$	evaluating closer to generalization performance
Entropy	$-n_i \cdot \log_2(n_i / \sum_j n_j)$	reducing impact of data distribution on evaluation
Macro	1	equal weighting of all classes

Table 2: Weighting schemes for evaluation of multi-class classification. n_i indicates the count of elements for class i and the Formula column shows the weight the class is assigned before normalization. The metrics are loosely ordered from top to bottom with the higher entries focusing more on instances and the lower entries focusing more on class membership. This usually corresponds to the model score, it is rare that models are better on classes with fewer samples. Methods in bold are proposed by us.

ever, we argue that model results should always be presented with multiple weightings for one dataset. Especially, when comparing different models all weightings should be reported for each model. This can clarify whether a model is good for all weightings or just *micro* or *macro*. Furthermore, with datasets that are currently evaluated with different weightings, it is easier to identify whether a model is specifically good for a dataset or for a weighting.

2.2 Framework for Weighting Schemes

We discuss a framework that summarizes the rules we give to class-weighting schemes. Then we introduce two new class weighting schemes. All discussed weighting schemes can be found in Table 2. They are independent of the measure that is used to calculate a score for each class.

(Class-)weighted and *macro* weighting are the extremes of "degressive proportionality"[3] or "allocation functions" (Słomczyński and Życzkowski, 2012). These are, e.g., used by the European Parliament to allocate seats to member nations depending on the population of the nation. They state that allocation should be monotonic increasing (see D1) and proportionally decreasing (see D2). To adopt this to a weighting scheme for multi-class evaluation, we add a normalizing desideratum that determines the sum of weights over all classes to be 1 (see D0).

Let $n_i > 0$ be the count of samples of class i and $w_i \geq 0$ the weight assigned to the score of class i.

We have the following desiderata:

$$\sum_i w_i = 1 \tag{D0}$$

$$n_i \geq n_j \Rightarrow w_i \geq w_j \tag{D1}$$

$$n_i \geq n_j \Rightarrow \frac{w_i}{n_i} \leq \frac{w_j}{n_j} \tag{D2}$$

Note that these desiderata do not restrict the scoring function that assigns scores s_i to class i. The weighted evaluation score is then given by $\sum_i w_i s_i$.

2.3 Weighting Schemes

Macro: Macro weighting is one extreme by setting equality on the weights of desideratum D1. It implies that we do not consider the instance counts per class, but treat all classes equally.

(Class-)weighted: Class-weighted is the other extreme by setting equality on the fraction of weights and counts in desideratum D2. It implies that we do not consider class constituency but weight all samples equally.

Dodrans: Cao et al. (2019) demonstrate that their balanced generalization error bound for binary classifiers in the separable case can be optimized by setting margins proportional to ${n_i}^{-1/4}$. They use this derivation from a limited theoretical scenario to improve the performance of several classifiers on imbalanced multi-class datasets. A term proportional to ${n_i}^{-1/4}$ is added in the loss function. While this added term is not directly transferable, we propose adapting this as a multiplicative factor in weighting classes for multi-class evaluation: $w_i \propto {n_i}^{-1/4} n_i = n_i^{3/4}$. We coin this weighting *dodrans* ("three-quarter").

Entropy: We also want to provide a weighting scheme that takes into consideration how hard a

[3] https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32013D0312&from=EN#d1e114-57-1

class is to predict. To this end, we propose weighting classes proportional to their term in the Shannon entropy formula

$$H(X) = -\sum_i P(x_i)\log(P(x_i)) \tag{2}$$

$$w_i \propto P(x_i)\log(P(x_i)). \tag{3}$$

We interpret $P(x_i)$ for class i to be the probability of it appearing in the dataset, s.t. $P(x_i) = n_i / \sum_j n_j$. Thus, without normalization the model score is now the sum over all classes of the model performance on a class times the difficulty and frequency of the class. Note, that this weighting scheme does not fulfil desideratum D1, since it is decreasing for classes i with $P(x_i) > e^{-1}$. This is related to the fact that classes that are too large become easier to predict for a model, the model can just default to predicting this class. It can also be desirable that a class does not gain relative importance once it contains more than half of the dataset. For RC, this often has little consequence. If we include *NA* in the normalization, it is usually the largest class and other classes are below an e-th of the dataset. Table 2 shows an overview of the mentioned schemes.

Figure 1 displays the weights that these schemes assign to the classes of the TACRED test set. The *weighted* scheme is proportional to class counts and produces the most imbalanced weights. *Dodrans* and *entropy* produce slightly more balanced weights and differ from *weighted* for the most frequent classes. *Macro* considers all classes equally, regardless of class count.

2.4 Statistical Testing

Currently, most RC works report a single score for each dataset. This can be the result from a single run or the median score from multiple runs. However, this does not allow to measure how large the difference between models is. Recently, analysis papers in NLP have recorded mean and standard deviation over multiple runs (Madhyastha and Jain, 2019; Zhou et al., 2020), as this allows for statistical tests.

We first test for significance and report p-values. We employ Welch's t-test to test the hypothesis that the models have equal mean. Following Zhu et al. (2020), we also report Cohen's d effect size to determine how large the difference between models is for a specific measure. For two models with the same number $n > 1$ of runs, Cohen's d is given by

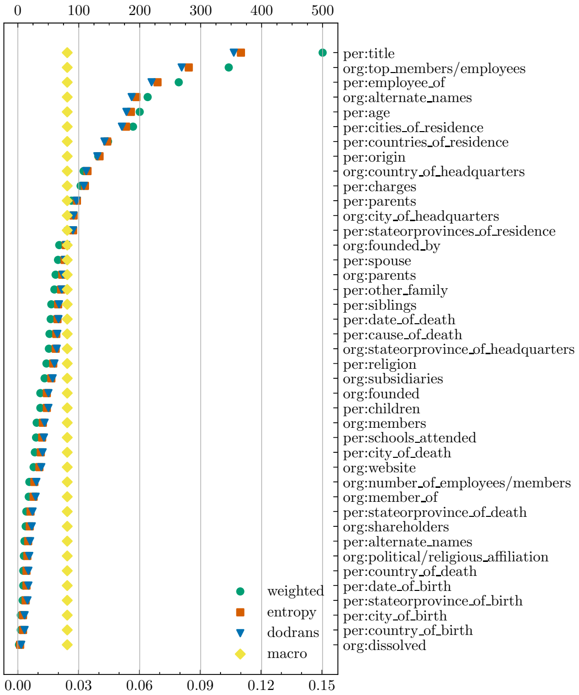

Figure 1: TACRED relations and their respective weights under different weighting schemes. The lower x-axis denotes the normalized weight given to a relation for a scheme. The upper x-axis corresponds to the counts of the relations in the test set for the class-weighted scheme. The y-axis denotes all positive relations. The negative *NA* class is not listed and has 12184 samples. The entropy and dodrans weighting scheme produce similar weights and are between weighted and macro weighting.

$$d = \sqrt{2}\frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2 + \sigma_2^2}} \tag{4}$$

with μ_i and σ_i^2 being mean and variance of model i's scores. We do this, as two different models never perform exactly the same, i.e. significance just depends on the number of runs and we also want to score the difference between the models.

3 Experiments

We evaluate and compare three RC methods with our proposed measures on two datasets. We choose these methods, as RECENT (Lyu and Chen, 2021) and BERT$_{\text{EM}}$ (Baldini Soares et al., 2019) are based on vanilla fine-tuning of a pre-trained language model, with a classification head on top. PTR (Han et al., 2021) is based on prompt-tuning. RECENT and PTR report similar micro-F_1 performance on TACRED, as do BERT$_{\text{EM}}$ and PTR on SemEval. In

Method	Micro	Weighted	Dodrans	Entropy	Macro
RECENT	71.5±0.4	67.8±0.4	62.5±0.4	63.6±0.4	43.1±0.6
PTR	72.5±0.3	72.1±0.5	69.8±0.5	70.3±0.5	60.3±0.8
p-value	$3 \cdot 10^{-3}$	$3 \cdot 10^{-6}$	10^{-8}	$2 \cdot 10^{-8}$	$2 \cdot 10^{-10}$
Cohen's d	2.8	8.7	14.8	13.5	24.2

Table 3: TACRED F_1-scores with different weighting schemes. Positive scores indicate PTR performs better than RECENT for all weighting schemes. The difference is smallest for the micro and largest for the macro weighting scheme. All p-values are smaller than $\alpha = 0.05$. All effect sizes are huge (> 2.0) under Sawilowsky (2009)'s rules of thumb.

Method	Micro	Weighted	Dodrans	Entropy	Macro
BERT$_{\text{EM}}$	89.1±0.3	89.1±0.3	88.7±0.3	88.6±0.3	82.7±0.4
PTR	88.4±0.3	88.3±0.3	88.1±0.3	88.0±0.3	87.8±0.5
p-value	0.005	0.006	0.023	0.023	$7 \cdot 10^{-8}$
Cohen's d	-2.5	-2.4	-1.8	-1.8	11.5

Table 4: SemEval F_1-scores with different weighting schemes. The directionality is of the relations is considered, s.t. there are 19 classes, the negative class is not included in evaluation. Negative scores indicate BERT$_{\text{EM}}$ performs better, positive scores indicate PTR performs better. All p-values are smaller than $\alpha = 0.05$. All absolute effect sizes are very large (> 1.2) or huge (> 2.0).

this way we can compare performance of the two paradigms for other weightings.

RECENT proposes a model-agnostic paradigm that exploits entity types to narrow down the candidate relations. Given an entity-type combination, a separate classifier is trained on the restricted classes. Baldini Soares et al. (2019) compare various strategies that extract relation representation from Transformers and claim ENTITY START (i.e. insert entity markers at the start of two entity mentions) yields the best performance. PTR also takes entity types into consideration and constructs prompts composed of three subprompts, two corresponding to the fill-in of the entity types and one predicting the relation.

In our experiments we use RECENT$_{\text{GCN}}$ for RECENT, BERT$_{\text{EM}}$ with ENTITY START, and unreversed prompts for PTR. We use the official repositories for RECENT and PTR, we reimplement BERT$_{\text{EM}}$[4]. We use the hyperparameters proposed in the original papers and conduct five runs for each model. Additional implementation and training details can be found in Appendices A and B.

The main focus is unearthing performance information about these methods that was previously obscured by single score measures. The number of weighting schemes does not influence the computational cost, as each score is determined through the predictions in a run and does not require specific tuning.[5] We acknowledge that each weighting scheme could be optimized for during training which gives additional importance to reporting multiple measures for each model.

3.1 Results

Table 3 shows results for TACRED. PTR significantly outperforms RECENT across all weighting schemes. The difference between the models is smallest for micro-F_1 and increases for all schemes that weigh classes more equally. For macro-F_1 the difference is starkest with effect size 24.2.

Table 4 displays results for SemEval. BERT$_{\text{EM}}$ significantly outperforms PTR in the micro-F_1 measure and all other weightings except for macro-F_1. All effect sizes are either large or huge, by far the largest effect size is between PTR and BERT$_{\text{EM}}$ regarding macro-F_1 though. The SemEval test set contains a single sample of the *Entity-*

[4] Our reimplementation is available at https://github.com/dfki-nlp/mtb-bert-em.

[5] We provide a package to add these scores to a Scikit-learn (Pedregosa et al., 2011) classification report at https://github.com/DFKI-NLP/weighting-schemes-report.

Destination(e2,e1) class which is quite impactful for the macro-F_1 of the models but has negligible impact on all other weighting schemes. The scores from *dodrans* and *entropy* indicate that only if all classes are considered equally important the PTR model should be preferred. This indicates that either the PTR model learns almost regardless of class frequency or BERT_{EM} has a class preference that is only discoverable with macro-F_1.

We demonstrate that evaluation on micro-F_1 does not give adequate information about model performance on long-tail classes. In Tables 3 and 4 we see that the model which performs better under micro-F_1 can either be significantly better or worse for classes with few samples. The weighted-F_1 produces similar results to micro-F_1 except for RECENT. Macro-F_1 on the other hand is very sensitive to model performance on single samples, e.g. the *Entity-Destination(e2,e1)* class in SemEval.

The scores of our proposed schemes are in between the existing measures and might be the best indicators for robust generalization performance. For all experiments, they produce similar results to each other. This could just be a coincidence of the datasets, and is also indicated by Figure 1. Overall, it might be fair to say that one of the former and latter measures is enough. It would mean one measure that does weigh proportional to sample count (micro- or weighted-F_1), an intermediary measure (dodrans-F_1 or entropy-F_1) and macro-F_1.

PTR performs better for macro-F_1 on both datasets. Its scores decrease less when classes are weighted more equally. This suggests that it is a better model for classes with low sample counts. Le Scao and Rush (2021) show that prompts can be worth hundreds of data points which would explain why the macro- and micro-F_1 scores are much closer together than for RECENT and BERT_{EM}.

4 Related Work

Chauhan et al. (2019) do a thorough evaluation of their model and notice the significantly different performance measured by *micro* and *macro* statistics due to the class imbalance, suggesting that the choice of evaluation measure is crucial. Huang and Wong (2020) further use the closeness between micro- and macro-F_1 scores to claim the stable performance of their model.

Mille et al. (2021) point out that evaluating with a single score favors overfitting. They show different evaluation suites that can be created for a dataset. Bragg et al. (2021) address the disjoint evaluation settings across recent research threads in (few-shot) NLP and propose a unified evaluation benchmark which regulates dataset, sample size etc., but fail to take the evaluation measure into consideration, reporting only mean accuracy instead. Post (2018) criticises the inconsistency and under-specification in reporting scores. This problem is also prevalent in RC where the F_1 weighting scheme is often not specified.

Zhang et al. (2020) show that bias from corpora persists for fine-tuned pre-trained language models. These models struggle with rare phenomena. For better performance debiasing with weighting is performed. Søgaard et al. (2021) argue against using random splits. They show that evaluating models with random splits is not a realistic setting but makes tasks easier by fixing the test data distribution to the train data distribution.

Long-tail evaluation is becoming more prominent in NLP research. Models in deep learning tend to show a gap in performance between frequent and infrequent phenomena (Rogers, 2021). Models in NLP have been shown to perform badly on specific subsets of data (Zhang et al., 2020).

Sokolova and Lapalme (2009) analyze measures for multi-class classification and present invariances regarding the confusion matrix. Gösgens et al. (2021) also determine which class measures (including F_1) fulfil specific assumptions. Further evaluation can be based on this. Our weighting schemes for F_1 can be transferred to other measures that calculate a score for each class.

5 Outlook

We suggest creating and using a bidimensional leaderboard like Kasai et al. (2021) where measures and models can be contributed. To this end, benchmarking of RC models could be done on a centralized site where a model or test set predictions are submitted and measures are calculated automatically through a script. For measures that modify weighting of classes and intra-class scoring, this does not require additional training computation.

Due to the reproducibility crisis (Baker, 2016), not all state-of-the-art scores can be replicated. Possible future work includes a comprehensive evaluation study of papers on leaderboards of RC tasks. This would enable an in-depth discussion of strength and weaknesses (including reproducibility) of these models.

The analysis we present can also be extended to other NLP tasks with imbalanced datasets, such as named entity recognition (Tjong Kim Sang and De Meulder, 2003), part-of-speech tagging (Pradhan et al., 2013) and coreference resolution (Pradhan et al., 2012).

6 Conclusion

We criticise the current practice of reporting a single score when evaluating imbalanced RC datasets. We propose a new framework to weight scores for multi-class evaluation of imbalanced datasets. We provide two new weighting schemes, *dodrans* and *entropy*, which are positioned between *class-weighted* and *macro*. In our experiments, we show that model performance on both TACRED and SemEval, especially on the long-tail relations, is not adequately captured by a single score. Thus, we advocate the use of multiple weighing schemes when reporting model performance on imbalanced datasets.

Acknowledgments

We would like to thank Nils Feldhus, Sebastian Möller, Lisa Raithel, Robert Schwarzenberg and the anonymous reviewers for their feedback on the paper. This work was partially supported by the German Federal Ministry of Education and Research as part of the project CORA4NLP (01IW20010) and by the German Federal Ministry for Economic Affairs and Climate Action as part of the project PLASS (01MD19003E). Christoph Alt is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2002/1 "Science of Intelligence" – project number 390523135.

References

Christoph Alt, Marc Hübner, and Leonhard Hennig. 2019. Fine-tuning pre-trained transformer language models to distantly supervised relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1388–1398, Florence, Italy. Association for Computational Linguistics.

Monya Baker. 2016. Reproducibility crisis. *Nature*, 533(26):353–66.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Jonathan Bragg, Arman Cohan, Kyle Lo, and Iz Beltagy. 2021. Flex: Unifying evaluation for few-shot nlp. *Advances in Neural Information Processing Systems*, 34.

Paula Branco, Luís Torgo, and Rita P. Ribeiro. 2016. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2):1–50.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in Neural Information Processing Systems*, 32:1567–1578.

Geeticka Chauhan, Matthew B.A. McDermott, and Peter Szolovits. 2019. REflex: Flexible framework for relation extraction in multiple domains. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 30–47, Florence, Italy. Association for Computational Linguistics.

George Forman and Martin Scholz. 2010. Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. *Acm Sigkdd Explorations Newsletter*, 12(1):49–57.

Martijn Gösgens, Anton Zhiyanov, Aleksey Tikhonov, and Liudmila Prokhorenkova. 2021. Good classification measures and how to find them. *Advances in Neural Information Processing Systems*, 34.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. *CoRR*, arXiv:2105.11259.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38. Association for Computational Linguistics.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550, Portland, Oregon, USA. Association for Computational Linguistics.

Haojie Huang and Raymond Wong. 2020. Deep embedding for relation extraction on insufficient labelled data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Lavinia Dunagan, Jacob Morrison, Alexander R Fabbri, Yejin Choi, and Noah A Smith. 2021. Bidimensional leaderboards: Generate and evaluate language hand in hand. *CoRR*, arXiv:2112.04139.

Jens Kringelum, Sonny Kjaerulff, Søren Brunak, Ole Lund, Tudor Oprea, and Olivier Taboureau. 2016. Chemprot-3.0: A global chemical biology diseases mapping. *Database*, 2016:bav123.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice loss for data-imbalanced NLP tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476, Online. Association for Computational Linguistics.

Shengfei Lyu and Huanhuan Chen. 2021. Relation classification with entity type restriction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 390–395, Online. Association for Computational Linguistics.

Pranava Madhyastha and Rishabh Jain. 2019. On model stability as a function of random seed. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 929–939, Hong Kong, China. Association for Computational Linguistics.

Simon Mille, Kaustubh Dhole, Saad Mahamood, Laura Perez-Beltrachini, Varun Gangal, Mihir Kale, Emiel van Miltenburg, and Sebastian Gehrmann. 2021. Automatic construction of evaluation suites for natural language generation datasets. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Juri Opitz and Sebastian Burst. 2019. Macro f1 and macro f1. *CoRR*, arXiv:1911.03347.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

C. J. Van Rijsbergen. 1979. *Information Retrieval*, 2nd edition. Butterworth-Heinemann.

Anna Rogers. 2021. Changing the world by changing the data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2182–2194, Online. Association for Computational Linguistics.

Shlomo S. Sawilowsky. 2009. New effect size rules of thumb. *Journal of modern applied statistical methods*, 8(2):26.

Wojciech Słomczyński and Karol Życzkowski. 2012. Mathematical aspects of degressive proportionality. *Mathematical Social Sciences*, 63(2):94–101.

Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. We need to talk about random splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832.

Marina Sokolova and Guy Lapalme. 2009. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437.

Yanmin Sun, Andrew K. C. Wong, and Mohamed S. Kamel. 2009. Classification of imbalanced data: a review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(04):687–719.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Guanhua Zhang, Bing Bai, Junqi Zhang, Kun Bai, Conghui Zhu, and Tiejun Zhao. 2020. Demographics should not be the reason of toxicity: Mitigating discrimination in text classifications with instance weighting. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4134–4145, Online. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45. Association for Computational Linguistics.

Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. 2020. The curse of performance instability in analysis datasets: Consequences, source, and suggestions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8215–8228, Online. Association for Computational Linguistics.

Haotian Zhu, Denise Mak, Jesse Gioannini, and Fei Xia. 2020. NLPStatTest: A toolkit for comparing NLP system performance. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 40–46, Suzhou, China. Association for Computational Linguistics.

A Implementation Details

To evaluate RECENT and PTR, we use the official code at `https://github.com//Saintfe/RECENT` (last updated on 01.10.2021) and `https://github.com/thunlp/PTR` (last updated on 20.11.2021). Since the official code of BERT$_{\text{EM}}$ is not available, we implement this method using the HuggingFace Transformers library (Wolf et al., 2020) and PyTorch (Paszke et al., 2019), and make our code base available at `https://github.com/dfki-nlp/mtb-bert-em`. To make our results reproducible, we randomly generated seeds $\{9, 148, 378, 459, 687\}$ and employed these for all models in their 5 runs.

B Training Details

B.1 RECENT

We consider GCN as the base model. Following the paper and the official code, we set the batch size to be 50, the optimizer to be SGD with learning rate 0.3, and the number of epochs to be 100. It takes a single RTX-A6000 GPU approximately 10 hours to complete all 5 runs on TACRED.

B.2 BERT$_{\text{EM}}$

We use the pre-trained language model (PLM) `bert-large-uncased` from the HuggingFace model hub and directly fine-tune the model for the RC task, without matching-the-blank pre-training. As the paper suggests, we set the batch size to be 64, the optimizer to be Adam with learning rate $3 \cdot 10^{-5}$, and the number of epochs to be 5. Additionally, we use the max sequence length of 512.

It takes a single RTX-A6000 GPU 30 minutes to complete all 5 runs on SemEval.

B.3 PTR

According to the paper and the official code base, we apply the same settings to evaluate both TACRED and SemEval: We use the PLM `roberta-large` and set the max sequence length to be 512, the batch size to be 64, the optimizer to be Adam with learning rate $3 \cdot 10^{-5}$, the weight decay to be 10^{-2}, and the number of epochs to be 5. It takes 4 Quadro-P5000 GPUs 84 hours to complete 5 runs on TACRED, and it takes 8 Titan-V GPUs 9 hours on SemEval.