On the Cost-Effectiveness of Stacking of Neural and Non-Neural Methods for Text Classification: Scenarios and Performance Prediction

Christian Gomes[1]**, Leonardo Rocha**[2]**, Marcos Gonçalves**[1]

[1] Department of Computer Science – Federal University of Minas Gerais (UFMG)

[2]Department of Computer Science – Federal University of São João del-Rei (UFSJ)

{christianreis,mgoncalv}@dcc.ufmg.br, lcrocha@ufsj.edu.br

Abstract. *Nowadays, neural networks algorithms, such as those based on Attention and Transformers, have excelled on Automatic Text Classification (ATC). However, such enhanced performance comes at high computational costs. Stacking of simpler classifiers that exploit algorithmic and representational complementarity has also been shown to produce superior performance in ATC, enjoying high effectiveness and potentially lower computational costs than complex neural networks. In this master's thesis, we present the first and largest comparative study to exploit the cost-effectiveness of Stacking in ATC, consisting of Transformers and non-neural algorithms. In particular, we are interested in answering the following research question: Is it possible to obtain an effective ensemble with significantly less computational cost than the best learning model for a given dataset? Besides answering that question, another main contribution of this thesis is the proposal of a low-cost oracle-based method that can predict the best ensemble in each scenario using only a fraction of the training data.*

1. Introduction

Natural Language Processing, Machine Learning and Data Mining techniques work together to automate the fundamental task of Automatic Text Classification (ATC). ATC automatically associates documents with classes, providing means to organize information, allowing better comprehension and interpretation of the data. ATC is paramount in applications such as: hate speech and fake news detection; sentiment analysis; content (topic) organization; semantic tagging of products and goods, among many others. Despite advances, ATC is far from being considered a solved problem, with much research being conducted by many groups around the world.

Algorithms based on neural networks have become the highlight in the area, where they are used both to learn features for text representation and as classification algorithms. The main problem of such methods is the very high computational costs needed for training the model [Sun et al. 2019, Cunha et al. 2021b]. Ensemble methods, such as Stacking, are strategies that combine multiple models to improve the prediction generalization in a given task. Such methods have shown promise in ATC [Džeroski and Ženko 2004, Ding and Wu 2020], enjoying high effectiveness and computational costs that depend on the selected learning methods of the ensemble. Among the possible ensemble strategies, Stacking has the characteristic of using a meta-layer capable of combining the prediction outputs of different heterogeneous individual models. The basic premise of Stacking is that different learning models, or different textual representations, can complement each other. The meta-layer can reveal intrinsic information by combining different models, potentially improving the effectiveness of a given task.

However, the benefits of ensemble techniques against a robust classifier are not always clear [Yan-Shi Dong and Ke-Song Han 2004], in part, due to the excellent generalization power of the best classifiers. In fact, previous ensemble works primarily focus on improving the overall classification effectiveness using the results of traditional classification algorithms [Campos et al. 2017, Ding and Wu 2020], paying little or no attention to practical issues such as the execution time or which combination of efficient base algorithms can bring effective results at a lower cost.

Accordingly, in this master's thesis, the main objective is to investigate the cost-effectiveness trade-off that has been vastly ignored up to today in the literature on Stacking in ATC. To this goal, an extensive set of experiments involving supervised text classification algorithms considered state-of-the-art in the field (neural networks and traditional algorithms) is carried out to evaluate the cost-effectiveness trade-offs. In addition, we propose a new algorithm based on a greedy strategy capable of identifying in a short time, without having to train a classifier with all available training data, the Stacking combinations that potentially will produce the best results. The proposed algorithm – **called Oracle** – manages to produce highly effective Stacking combinations using a fraction of the training data.

2. Contributions

The **first contribution of this thesis** is a thorough study of the cost-effectiveness of Stacking for text classification tasks. We study Stacking combinations capable of achieving a better compromise between low cost and high effectiveness when compared to a single individual model (i.e., the single most effective model in a given dataset). We conduct a wide range of comparative experiments with combinations of Stacking and classification algorithms considered state-of-the-art in 8 datasets widely used in ATC. This thesis seeks answers based on empirical evidence for the main research question: Is it possible to obtain an effective ensemble with significantly less computational cost than the best learning model for a given dataset? In order to do this, we divide this question into other three more specific questions, considering the best learning model for each dataset:

- RQ1: Is it possible to obtain an effective ensemble with less computational time than the best individual learning model?
- RQ2: Is it possible to improve the effectiveness of the best learning model using an ensemble without increasing computational time?
- RQ3: Disregarding computational time, is there an ensemble that can improve effectiveness when compared to the best learning model?

As far as we know, this is the first work to investigate the Stacking cost-effectiveness [Cunha et al. 2021a] of text classifiers based on neural networks and traditional strategies from the perspectives described above.

The **second main contribution** of the thesis is the proposal of an algorithm based on a low-cost greedy strategy that can predict the best stacking ensemble in a given scenario (with and without computational cost limitations) using only a fraction of the available training data. As we named it, the Oracle algorithm predicts efficient ensembles successively, including algorithms that improve its cost using an average meta-layer. This new proposed algorithm is the first known strategy to efficiently predict effective ensembles capable of dealing with practical cost issues related to our research questions. This proposal aims to predict three ensembles corresponding to the time constraints of RQ1, RQ2 and RQ3, respectively, avoiding the potential high computational cost of evaluating expensive base models and their ensembles, especially on large data sets. Oracle's specific research questions are as follows:

• ORQ1: It is possible to predict, using a fraction of the training data, an effective ensemble that will tie or surpass the best learning model when trained with the entire training set available, at a lower cost than the best model?
• ORQ2: Is it feasible to make a prediction similar to ORQ1, but now with a cost less than or at most equal to the best model when trained with all training data?
• ORQ3: Without time constraints, is it possible to predict a combination that will be better than the best learning algorithm in a dataset?

Our experimental evaluation shows affirmative answers to the six research questions in most of the experiments. In most datasets, it is possible to obtain an ensemble of algorithms as good or better than the best individual algorithm at a lower cost. It is possible to obtain an ensemble with statistically significant gains concerning the best algorithm without increasing cost in seven out of the eight experimented datasets. Likewise, in seven of the eight datasets, the Oracle algorithm provides results as good or (statistically significant) better than the best individual algorithm without increasing computational cost, providing empirical evidence for the practical benefits of the proposed Oracle.

Thesis Related Publications. The results of this thesis generated a publication [Gomes et al.] in The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021). ACL is the most important worldwide conference on Natural Language Processing and Computational Linguistics (Qulais A1, h-index: 157). Significant portions of the research conducted in this thesis contributed significantly to a recent publication on the journal Information Processing & Management (IP&M) [Cunha et al. 2021a] (Qualis A1, Impact factor: 6.22; h5-index: 55), one of the leading journals in the field. The article describes the largest known study on the cost-effectiveness tradeoff of recent ATC solutions.

3. Experimental Evaluation

3.1. Configurations

Datasets: We consider the effectiveness and efficiency of the models on eight datasets very known by the ATC community. Of those eight datasets, four are large-scale datasets (more than 100,000 documents) [Zhang et al. 2015, Diao et al. 2014] – AGNews, IMdB Reviews, Sogou and Yelp – and four are mid-sized datasets [Canuto et al. 2014, Canuto et al. 2018] – 20NG, ACM, Reut and WebKB.

Textual Representations and Supervised Classification Algorithms: In terms of representations, beyond the traditional term-weighting alternatives (TFIDF), we consider distributional and other types of word embeddings, such as FastText [Joulin et al. 2016, Bojanowski et al. 2017] and PTE [Tang et al. 2015], as well as recent representations based on MetaFeatures that have obtained state-of-the-art (SOTA) effectiveness in some of the experimented datasets [Canuto et al. 2019, Canuto et al. 2018, Canuto et al. 2016, Cunha et al. 2020, Cunha et al. 2021a]. For those interested, in Table 4.2 in the master's thesis, we have all the settings we used for each chosen textual representation. For supervised classification algorithms, we consider the LinearSVM [Fan et al. 2008], kNN [Altman 1992], LogisticRegression [Fan et al. 2008], XGBoost [Chen and Guestrin 2016], XLNet [Yang et al. 2019] and BERT [Devlin et al. 2018]. The implementations of LinearSVM, kNN and LogisticRegression come from scikit-learn[1] [Pedregosa 2011] while XGBoost [Chen and Guestrin 2016] comes from the authors´ implementation[2].

Stacking: We execute the stacking process with the following variants: all combinations of the same individual model with different representations, all combinations of different individual models with their best representations, and a combination that includes all the individual models. To train the Meta-layer, responsible for learning to combine the outputs of the different individual classifiers, we use the following algorithms: Majority Vote (hard/soft), Mean, Median and LinearSVM. An important observation is that we assume that the individual models can be run in parallel to avoid an unfair comparison. Thus, a stacking or oracle combination has the execution time limited by the most costly individual model in the respective combination.

Statistical Techniques and Supervised Classification Metrics: The experiments in the smaller datasets were executed using a 10-fold cross-validation procedure, while in the larger we used 5-fold due to the computational cost. The algorithms parameters were tuned using the Bayesian Optimization [Bergstra et al. 2015] approach with ten iterations, with the 5-fold stratified strategy and the training set (nested cross-validation). To generate the probabilities used as input to the Meta-Layer, we used another internal 5-fold stratified cross-validation to separate only the training set into training/validation [Wolpert 1992, Tang et al. 2014]. The parameters tuned for each model are in the thesis in Tables 4.3 and 4.4. We evaluate all methods, combined with different representations, concerning classification effectiveness and training time. We assess classification effectiveness in the test partitions using MicroF1 and MacroF1 [Sokolova and Lapalme 2009]. In addition to effectiveness, we also assess the cost of each method in terms of training execution time, aiming at analyzing the cost-effectiveness trade-offs for all methods. The metric is the overall time in seconds (average of folds). To compare the average test results on our cross-validation experiments, we assess the statistical significance employing the paired t-test with 95% confidence [Urbano et al. 2019, Hull 1993].

3.2. Results

Here we point out the main results obtained in the studies for Stacking and the proposed Oracle. For more specific details about the results (e.g., which combination or metrics were obtained in each dataset and research question), consult Chapter 4 of the thesis.

Stacking Results - RQ1: Our results show that in 5 out of 8 datasets, it is possible to obtain a stacking combination that is as good as or better (see the *ACM* case, with

[1]`https://scikit-learn.org/stable/index.html`
[2]`https://xgboost.readthedocs.io/en/latest`

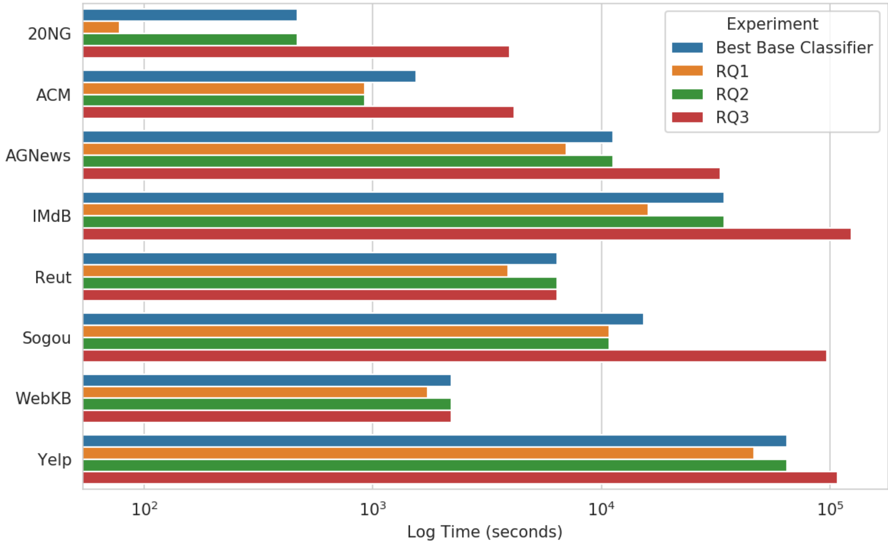

Figure 1. Average time (in log scale) of the Best Base Individual Classifier and Stacking Research Questions for each dataset.

statistically significant gains of 3.1%) than the best individual model, at a lower cost. In fact, the gains in terms of time are very significant (see Figure 1), ranging from 1.87x speedup improvement (in Reuters) to 7.16x (in WebKB)[3]. Even if we consider the two cases in which there were some minimum effectiveness losses (0.39% in AGNews and 2.66% in IMdB), there are some significant speedups: 1.6x in AGNews and 2.15x in IMdB. Some chosen stacked combinations are interesting: in 20NG, the combination contains all versions of kNN; in ACM, the combination contains three versions of Logistic Regression. Both combinations contain classifiers with Metafeatures.

Stacking Results - RQ2: In 6 out of 8 datasets, it is possible to obtain effectiveness gains with no increase in time. Effectiveness gains vary from 0.4% in AGNews, 1.15% in 20NG[4], 3.1% in ACM, 5.4% in IMdB and 9% in WebKB. Reuters is only considered a tie because of the high variability of the results across folds in this dataset due to class imbalance, which generates large standard deviations/confidence intervals. In absolute terms, there was a positive variation (non-statistically significant gain) of more than 9.7%. Indeed, the MicroF1 stacking results confirm statistically significant gains in Reuters (See Table 1). As expected, to obtain gains in this scenario, it is necessary to include the best individual model in the combination in most datasets, inserting diversity/complementarity into the combination. Only in ACM the individual model is not part of the combination.

Stacking Results - RQ3: Finally, in the scenario with no time constraint, further gains can be obtained with the inclusion of more costly classifiers. There are further gains in AGNews (0.94%), 20NG (2.06%), IMdB (5.8%) and ACM (6.32%). Notice that there is a tendency to include most algorithms in the combinations, like in ACM, WebKB and AGNews, to obtain further improvements in this scenario. This means that most algorithms have complementary information that contributes to the final results. Another interesting aspect to notice is that in some cases, such as in 20NG, a completely different

[3]Speedups in 20NG and ACM were 6x and 6.6x

[4]Improvements in 20NG and AGNews are hard to obtain given the already high effectiveness values.

RQ	MicroF1			MacroF1		
	Win	Tie	Loss	Win	Tie	Loss
RQ1	2	3	3	1	4	3
RQ2	7	1	0	6	2	0
RQ3	8	0	0	7	1	0

Table 1. Win/Tie/Loss summary for Stacking research questions.

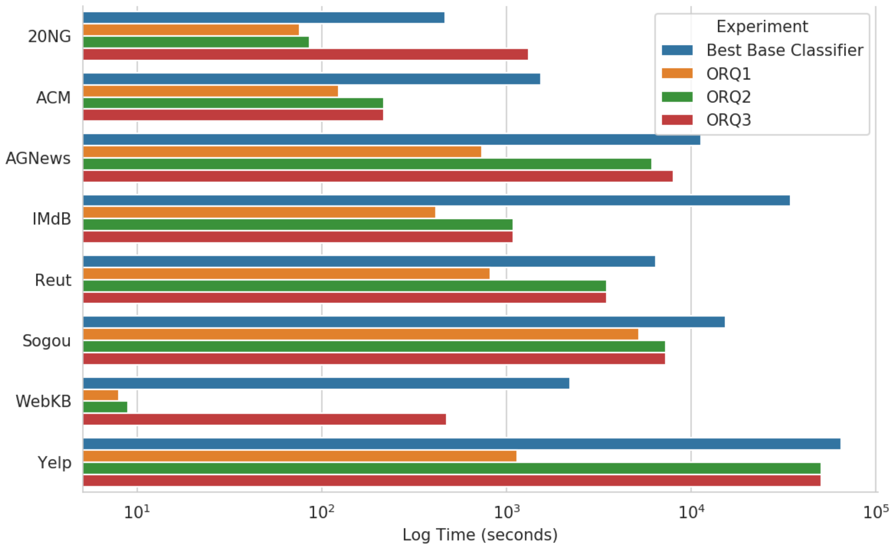

Figure 2. Oracle Times

combination than that chosen in scenario RQ2 was picked. This combination exploits the most effective and complementary algorithms and may not even include the individual model. In other cases, such as in IMdB, a combination of a few of the most effective (and costly) algorithms suffices to obtain larger gains. This means that the meta-layer is really doing a good job learning about the algorithms' individual performance and their complementarity. Finally, these additional effectiveness gains come with potential high increases in time, clearly seen in Figure 1 for the cases of 20NG, ACM and AGNews. In those datasets, the costs have tripled (AGNews), quadrupled (ACM and IMdB), or become 8x more expensive. It is up to the application designer to decide whether this cost-effectiveness trade-off is worth it.

Stacking Results - Summary: Table 1 summarizes the effectiveness results. For RQ1, there are ten win/ties out of 16 possibilities (8 datasets, two metrics). Remind that ties are considered a good result in this scenario due to the reduction in costs. We also have 13 wins for RQ2 and 15 wins for RQ3, only ties in Reut and Sogou with no loss at all. In terms of cost (Figure 1), significant reductions in RQ1 can be obtained in all eight datasets, with minimal losses in terms of effectiveness. For RQ3, effectiveness gains can be obtained in almost all cases with no additional cost compared to the cost of the base classifier. Furthermore, for RQ3, additional effectiveness gains can be obtained, but sometimes with a very high increase in cost.

Oracle Results - ORQ1: In this scenario, in half of the cases, we can perform a good prediction, i.e., one that predicts a combination of methods that will tie or outperform

RQ	MicroF1			MacroF1		
	Win	Tie	Loss	Win	Tie	Loss
ORQ1	2	0	6	2	1	5
ORQ2	3	5	0	3	4	1
ORQ3	8	0	0	6	1	1

Table 2. Win/Tie/Loss summary for Oracle research questions.

the best individual model when trained with all the available training data (100%). It is essential to stress that in an actual situation, we do not really know what will be the best algorithm when using all the training data nor its effectiveness. Indeed, with more data, there is a tendency for some algorithms, such as the transformers, to improve their effectiveness, but their good performance may not be predicted with few training data. Remind also that this is a stringent scenario: even if we can predict which will be the best individual model, we cannot use it in the combination given the time constraints of ORQ1.

Oracle Results - ORQ2: When we are allowed to include the best-predicted algorithm in the stacking (scenario for ORQ2), results are even better – we can make a good prediction in 5 out 6 cases (2 wins and 3 ties). Notice that we consider a tie as a good result in this scenario. We interpret that being able to predict a combination that will tie with the best algorithm with 100% of training in a dataset, without knowing which one will this best, at a very lost cost (Figure 2), as an excellent result. IMdB was the only case in which we could not make a good prediction precisely by the failure in predicting, with 30% of training, that BERT would be the best algorithm when all the training data is used.

Oracle Results - ORQ3: Finally, when no time constraints are imposed, the oracle's prediction results are excellent: 4 wins, 1 tie and only one loss (in IMdB). The same reasons explain this last loss as in the previous scenario: the failure of predicting BERT as the future best algorithm. Nevertheless, even in this case, the prediction produced minimal losses: only 1.05% at a cost much smaller than using BERT.

Oracle Results - Cost: When looking at the costs of making the predictions in each scenario (ORQ1, ORQ2, and ORQ3), shown in Figure 2, we can see that in all cases (but 20NG for ORQ3), the oracle's predictions times are much smaller, in many cases negligible[5], when compared to the time to run the individual model with 100% of training. Given the time constraints imposed by ORQ1 and ORQ2 and the fact that even in the scenario for ORQ3, only a portion of the 18 available algorithms needed to be stacked (in most cases) to produce effectiveness gains, the advantages of running the oracle's predictions in terms of cost stand for themselves.

Oracle Results - Summary: Table 2 summarizes the results in terms of Micro and MacroF1: considering all 36 results (3 RQs, 8 datasets, 2 metrics), the oracle predicted 17 wins, 10 ties (most of them (8) in scenarios ORQ1 and ORQ2, which can be considered good results) and only 9 losses, six of them in a single dataset (IMdB) for the simple reason that we failed in predicting a neural network winner with fewer data. This is certainly a point to be improved in our methodology. One idea is to look not only at the absolute effectiveness values with a single training point (30%) but also look at the tendency of growing considering several points (5%, 10%, ..).

[5]Some differences are in the orders of magnitude.

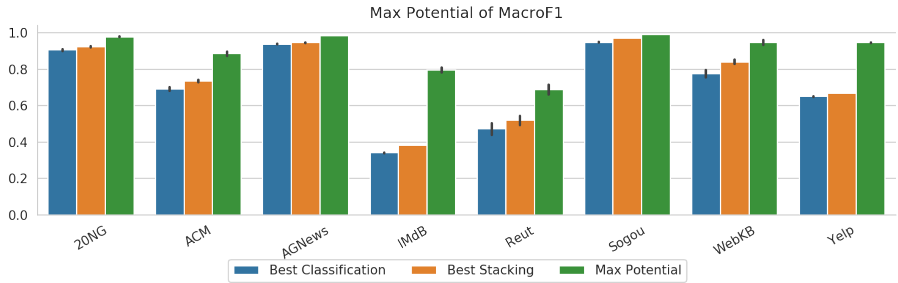

Figure 3. Maximum potential gain considering the 18 models *versus* the results of the best individual model (Best Classification) and the best stacking combination (Best Stacking) for each dataset.

Impact of the Datasets´ Characteristics: In the traditional stacking results (RQ3 - no time constraints) it is possible to notice that not all datasets include all algorithms in the ensemble. From the characteristics of each dataset, described in Table 4.1 in the dissertation, and observing the stacking results, we can observe some dataset´s characteristics (e.g., in IMDB and Reuters) that may influence the choice of the algorithms in the ensemble such as the high class imbalance and the large number of features. However, to infer a general rule is hard as there are imbalanced datasets with a high number of features that include, for instance, all algorithms in the ensemble. On the other hand, all datasets with fewer attributes and low class imbalance tend to have better overall stacking performance. We leave for future work exploring further those issues.

Stacking Maximal Potential: In our last set of experiments, we show how far the stacking strategies presented in this work are from the effectiveness upper bound considering the 18 models adopted in all the experiments. To measure the maximal potential, for each test instance of each dataset, if one of the 18 models correctly predicts its class, we assume that the ensemble strategy also correctly predicts its correct class. We observe these results in Figure 3 for MacroF1. We also present the results achieved by the best stacking and the best individual algorithm. For all evaluated collections, we can observe that there is still large potential to be explored - in some collections (e.g., IMdB and Reut) more than others (e.g., Sogou). Despite being unrealistic to assume that the meta-layer will always make the best choice, comparing all these results show us how much we can still engage efforts in this line of research, trying to improve the results obtained in this thesis.

4. Conclusion

We presented two important contributions to the application of Stacking in ATC: a thorough study of cost-effectiveness trade-offs and a new oracle method to predict the best ensemble combination for a dataset at a low cost. Our extensive experiments, composed of 4 textual representation methods, 8 datasets, 4 non-neural-based and 2 neural-based algorithms, provided us with answers to questions that had not yet been explored in the literature. By performing stacking with different time constraints, we showed that it was possible to obtain combinations that positively answered the posed questions regarding the time-constrained stacking and the oracle predictions in terms of both effectiveness and efficiency. Our proposed Oracle efficiently predicts effective best base models on time-constrained scenarios, allowing adaptable solutions that automatically optimize the choice of base learners for each specific dataset.

As future work, we will add new classifiers to our Stacking, such as different variations of the BERT algorithm (e.g., RoBERTa and DistilBERT [Liu et al. 2019]). We will also apply recent approaches of classification interpretability [Lundberg and Lee 2017] to extract explanations related to their predictions, evaluating how much each classifier contributes to the final prediction in the meta-layer. We did not find studies in literature that explore interpretability models for stacking strategies. We also aim to explore multi-objective feature selection [Viegas et al. 2018] in the stacking meta-layer to optimize both effectiveness and computational cost. There is also room for improvements regarding the Oracle strategy. For example, we can apply selective sampling [Silva et al. 2016] to reduce the total training data used for the individual algorithms, decreasing the Oracle's computational training cost and improving effectiveness. Finally, our Oracle proposal can be used in other applications such as Recommender Systems.

References

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185.

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., and Cox, D. D. (2015). Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8(1):014008.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of ACL*, 5:135–146.

Campos, R., Canuto, S., Salles, T., de Sá, C. C., and Gonçalves, M. A. (2017). Stacking bagged and boosted forests for effective automated classification. In *SIGIR*, page 105–114.

Canuto, S., Gonçalves, M. A., and Benevenuto, F. (2016). Exploiting new sentiment-based meta-level features for effective sentiment analysis. In *WSDM*, pages 53–62.

Canuto, S., Salles, T., Gonçalves, M. A., Rocha, L., Ramos, G., Gonçalves, L., Rosa, T., and Martins, W. (2014). On efficient meta-level features for effective text classification. In *CIKM ´14*, pages 1709–1718.

Canuto, S., Salles, T., Rosa, T. C., and Gonçalves, M. A. (2019). Similarity-based synthetic document representations for meta-feature generation in text classification. In *SIGIR*, pages 355–364.

Canuto, S., Sousa, D. X., Goncalves, M. A., and Rosa, T. C. (2018). A thorough evaluation of distance-based meta-features for automated text classification. *IEEE TKDE*, 30(12):2242–2256.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *SIGKDD conference*, pages 785–794.

Cunha, W., Canuto, S., Viegas, F., Salles, T., Gomes, C., Mangaravite, V., Gonçalves, M. A., and Rocha, L. (2020). Extended pre-processing pipeline for text classification: On the role of meta-feature representations, sparsification and selective sampling. *Inf. Processing & Management*, 57(4):102263.

Cunha, W., Mangaravite, V., Gomes, C., Canuto, S., Resende, E., Nascimento, C., Viegas, F., França, C., Martins, W. S., Almeida, J. M., et al. (2021a). On the cost-effectiveness of neural and non-neural approaches and representations for text classification: A comprehensive comparative study. *Inf. Processing & Management*, 58(3):102481.

Cunha, W., Mangaravite, V., Gomes, C., Canuto, S. D., Resende, E., Nascimento, C., Viegas, F., França, C., Martins, W. S., Almeida, J. M., Rosa, T., da Rocha, L. C., and Gonçalves, M. A. (2021b). On the cost-effectiveness of neural and non-neural approaches and representations for text classification: A comprehensive comparative study. *Inf. Process. Manag.*, 58(3):102481.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Diao, Q., Qiu, M., Wu, C.-Y., Smola, A. J., Jiang, J., and Wang, C. (2014). Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *SIGKDD conference*, KDD '14, page 193–202.

Ding, W. and Wu, S. (2020). A cross-entropy based stacking method in ensemble learning. *Journal of Intelligent & Fuzzy Systems*, pages 1–12.

Džeroski, S. and Ženko, B. (2004). Is combining classifiers with stacking better than selecting the best one? *Machine learning*, 54(3):255–273.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874.

Gomes, C., Gonçalves, M. A., Rocha, L., and Canuto, S. D. On the cost-effectiveness of stacking of neural and non-neural methods for text classification: Scenarios and performance prediction. In *Proc. of the Association for Computational Linguistics: ACL/IJCNLP*, pages 4003–4014.

Hull, D. (1993). Using statistical testing in the evaluation of retrieval experiments. In *SIGIR*, pages 329–338.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

Pedregosa, F. e. (2011). Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830.

Silva, R. M., Gomes, G. C., Alvim, M. S., and Gonçalves, M. A. (2016). Compression-based selective sampling for learning to rank. In *CIKM ´16*, pages 247–256.

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

Sun, C., Qiu, X., Xu, Y., and Huang, X. (2019). How to fine-tune bert for text classification? In *Conference on Chinese Computational Linguistics*, pages 194–206.

Tang, J., Alelyani, S., and Liu, H. (2014). Data classification: algorithms and applications. *Data Mining and Knowledge Discovery Series*, pages 37–64.

Tang, J., Qu, M., and Mei, Q. (2015). Pte: Predictive text embedding through large-scale heterogeneous text networks. In *SIGKDD Conference)*, pages 1165–1174.

Urbano, J., Lima, H., and Hanjalic, A. (2019). Statistical significance testing in information retrieval: an empirical analysis of type i, type ii and type iii errors. In *SIGIR*, pages 505–514.

Viegas, F., Rocha, L., Gonçalves, M., Mourão, F., Sá, G., Salles, T., Andrade, G., and Sandin, I. (2018). A genetic programming approach for feature selection in highly dimensional skewed data. *Neurocomputing*, 273:554–569.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259.

Yan-Shi Dong and Ke-Song Han (2004). A comparison of several ensemble methods for text categorization. In *SCC 2004*, pages 419–422.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *NIPS*, pages 5753–5763.

Zhang, X., Zhao, J. J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626.