

A Comparison of SVM against Pre-trained Language Models (PLMs) for Text Classification Tasks

Yasmen Wahba¹(✉), Nazim Madhavji¹, and John Steinbacher²

¹ Western University, London ON, Canada

ywahba2@uwo.ca, nmadhavji@uwo.ca

² IBM Canada, Toronto, ON, Canada

jstein@ca.ibm.com

Abstract. The emergence of pre-trained language models (PLMs) has shown great success in many Natural Language Processing (NLP) tasks including text classification. Due to the minimal to no feature engineering required when using these models, PLMs are becoming the de facto choice for any NLP task. However, for domain-specific corpora (e.g., financial, legal, and industrial), fine-tuning a pre-trained model for a specific task has shown to provide a performance improvement. In this paper, we compare the performance of four different PLMs on three public domain-free datasets and a real-world dataset containing domain-specific words, against a simple SVM linear classifier with TFIDF vectorized text. The experimental results on the four datasets show that using PLMs, even fine-tuned, do not provide significant gain over the linear SVM classifier. Hence, we recommend that for text classification tasks, traditional SVM along with careful feature engineering can provide a cheaper and superior performance than PLMs.

Keywords: Text Classification, Pre-trained Language Models, Machine Learning, Domain-specific Datasets, Natural Language Processing

1 Introduction

Text classification is the task of classifying text (e.g., tweets, news, and customer reviews) into different categories (i.e., tags). It is a challenging task especially when the text is ‘technical’. We define ‘technical’ text in terms of the vocabulary used to describe a given document, e.g., classifying health records, human genomics, IT discussion forums, etc. These kinds of documents require special pre-processing since the basic NLP pre-processing steps may remove critical words necessary for correct classification, resulting in a performance drop of the deployed system [1].

Recently, pre-trained language models (PLMs) such as BERT [2] and ELMO [3] have shown promising results in several NLP tasks, including spam filtering, sentiment analysis, and question answering. In comparison to traditional models, PLMs require less feature engineering and minimal effort in data cleaning. Thus becoming the consensus for many NLP tasks [4].

With an enormous number of trainable parameters, these PLMs can encode a substantial amount of linguistic knowledge that is beneficial to contextual representations

[4]. For example, word polysemy (i.e., the coexistence of multiple meanings for a word or a phrase –e.g., ‘bank’ could mean ‘river bank’ or ‘financial bank’) in a domain-free text.

In contrast, in a domain-specific text that contains technical jargon, a word has a more precise meaning (i.e., monosemy) [5]. For example, the word ‘run’ in an IT text would generally only mean ‘execute’ and not ‘rush’. Thus, it appears that domain-specific text classification will likely not benefit from the rich linguistic knowledge encoded in PLMs.

Despite the widespread use of PLMs in a broad range of downstream tasks, their performance is still being evaluated by researchers for their drawbacks [6]. For example: (i) the large gap between the pre-training objectives (e.g., predict target words) and the downstream objectives (e.g., classification) limits the ability to fully utilize the knowledge encoded in PLMs [7], (ii) the high computational cost and the large set of trainable parameters make these models impractical for training from scratch, (iii) dealing with rare words is a challenge for PLMs [8], and (iv) the performance of PLMs may not be generalizable [9].

Thus, this paper evaluates the performance of different pre-trained language models (PLMs) against a linear Support Vector Machine (SVM) classifier. The motivation for this comparative study is rooted in the fact that: (i) while PLMs are being used in text classification tasks [10][11], they are more computationally expensive than the simpler SVMs, and (ii) PLMs have been used predominantly on public or domain-free datasets and it is not clear how they fare against simpler SVMs on domain-specific datasets.

The findings of our study suggest that the problem of classifying domain-specific or generic text can be addressed efficiently using old traditional classifiers such as SVM and a vectorization technique such as TFIDF bag-of-words that do not involve the complexity found in neural network models such as PLMs. To the best of our knowledge, no such comparative analysis has so far been described in the scientific literature.

The rest of the paper is organized as follows. Section 2 describes related work. Section 3 describes the empirical study. Section 4 presents the research results. Section 5 concludes the paper.

2 Related Work

In this section, we give an overview of the existing literature on the applications of PLMs and some of the drawbacks reported.

Pre-trained language models (PLMs) are deep neural networks trained on unlabeled large-scale corpora. The motivation behind these models is to capture rich linguistic knowledge that could be further transferred to target tasks with limited training samples (i.e., fine-tuning). BERT [2], XLM [12], RoBERTa [13], and XLNet [14] are examples of PLMs that have achieved significant improvements on a large number of NLP tasks (e.g., question answering, sentiment analysis, text generation).

Nevertheless, the performance of these models on domain-specific tasks was questioned [15] as these models are trained on general domain corpora such as Wikipedia, news websites, and books. Hence, fine-tuning or fully re-training PLMs for downstream

tasks has become a consensus. Beltagi et al. [16] released SciBERT that is fully re-trained on scientific text (i.e., papers). Lee et al. [17] released BioBERT for biological text. Similarly, Clinical BERT [18][19] was released for clinical text, and FinBERT [20] for the financial domain.

Other researchers applied PLMs by fine-tuning the final layers to the downstream task. For example, Elwany et al. [21] report valuable improvements on legal corpora after fine-tuning. Lu [22] fine-tuned RoBERTa for Commonsense Reasoning and Tang et al. [23] fine-tuned BERT for multi-label sentiment analysis in code-switching text. Finally, Yuan et al. [24] fine-tuned BERT and ERNIE [25] for the detection of Alzheimer’s Disease.

However, Gururangan et al. [15] show that simple fine-tuning of PLMs is not always sufficient for domain-specific applications. Their work suggests that the second phase of pre-training can provide significant gains in task performance. Similarly, Kao et al. [26] suggest that duplicating some layers in BERT prior to fine-tuning can lead to better performance on downstream tasks.

Another body of research focuses on understanding the weaknesses of PLMs by either applying them to more challenging datasets or by investigating their underlying mechanisms. For example, McCoy et al. [9] report the failure of BERT when evaluated on the HANS dataset. Their work suggests that evaluation sets should be drawn from a different distribution than the train set. Also, Schick et al. [8] introduce WNLaMPro (WordNet Language ModelProbing) dataset to assess the ability of PLMs to understand rare words. Lastly, Olga et al. [27] show redundancy in the information encoded by different heads in BERT, and manually disabling attention in certain heads will lead to performance improvement.

This paper adds to the growing literature on evaluating PLMs. In particular, our investigative question is: How does a linear classifier such as SVM compare against the state-of-the-art PLMs on both general and technical domains?

3 Empirical Study

In this section, we describe the empirical study that we conducted. In particular, we describe the infrastructure used, the datasets, and the different PLMs used. Finally, we describe the SVM algorithm used, and the pre-processing steps done prior to applying SVM. The experimental algorithms are written in Python 3.8.3. The testing machine is Windows 10 with an Intel Core i7 CPU 2.71 GHz and 32GB of RAM.

3.1 Text Classification Datasets

Our experiments were evaluated on four datasets:

1. BBC News [28]: a public dataset originating from BBC News. It consists of 2,225 documents, categorized into 5 groups, namely: business, entertainment, politics, sport, and tech.
2. 20NewsGroup [29]: a public dataset consisting of 18,846 documents, categorized into 20 groups.

3. Consumer Complaints [30]: a public benchmark dataset published by the Consumer Financial Protection Bureau; it is a collection of complaints about consumer financial products and services. It consists of 570,279 documents categorized into 15 classes.
4. IT Support tickets: a private dataset obtained from a large industrial partner. It is composed of real customer issues related to a cloud-based system. It consists of 194,488 documents categorized into 12 classes.

Table 1 summarizes the properties of the four datasets.

Table 1. Dataset properties

Dataset	# of classes	# of instances	# of features (n-gram=1)	# of features (n-gram=3)
BBC News	5	2,225	26,781	811,112
20NewsGroup	20	18,846	83,667	2,011,358
Consumer Complaints	15	570,279	53,429	6,112,905
IT Support tickets	12	194,488	16,011	3,185,796

The IT Support tickets dataset will be referred to hereon as the ‘domain-specific’ dataset. This dataset suffers from a severe imbalance as seen in Fig. 1. However, we prefer to avoid the drawbacks of sampling techniques [31][32] and keep the distribution as is.

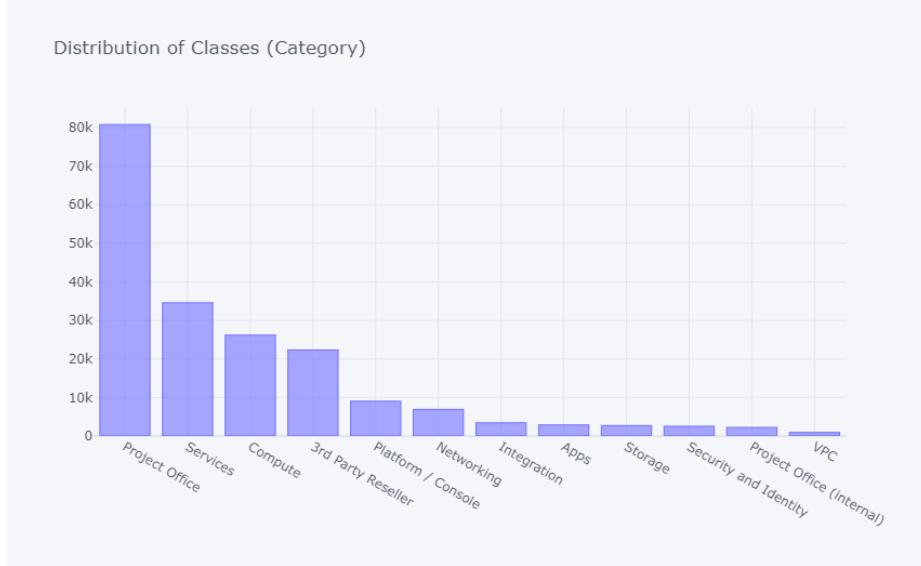


Fig. 1. Class distribution of the domain-specific dataset showing imbalance

Another problem with this dataset is the presence of a large number of technical words (i.e., jargon) related to the Cloud terminologies (e.g., Bluemix, Kubernetes, IaaS, Vmware, etc.). These words are not found in the PLMs vocabulary and hence, they get broken down into subwords using a subword tokenization algorithm. For instance, BERT uses a WordPiece tokenizer [33] which handles non-technical words quite well. However, we notice that it fails to tokenize technical words and domain-specific abbreviations in our domain-specific dataset. For example:

```
"Kubernetes" ⇒ ['ku', '##ber', '##net', '##es']
"configuration" ⇒ "config" ⇒ ['con', '##fi', '##g']
```

3.2 Pre-trained Language Models (PLMs)

The following PLMs were considered for this study:

1. BERT [2]: A widely used pre-training language model that is based on a bidirectional deep Transformer as the main structure. BERT achieved state-of-the-art results on 11 different NLP tasks including question answering and named entity recognition (NER).
2. DistilBERT [34]: A lighter, smaller, and faster version of BERT. By reducing the size of the BERT model by 40%, while keeping 97% of its language understanding capability, it's considered 60% faster than BERT.
3. RoBERTa [13]: One of the successful variants of BERT that achieved impressive results on many NLP tasks. By changing the MASK pattern, discarding the NSP task, and using a larger batch size and longer training sentences.
4. XLM [12]: Designed specifically for cross-lingual classification tasks by leveraging bilingual sentence pairs. XLM uses a known pre-processing technique (BPE) and a dual-language training mechanism.

For this study, we fine-tuned all the PLMs to the domain-specific dataset and the three generic datasets. In all our experiments, we use the following hyperparameters for fine-tuning: maximum sequence length of 256, adam learning rate (lr) of $1e^{-5}$, batch size of 16, and a train-test split ratio of 80:20.

Support Vector Machines (SVM). A Support Vector Machine is a popular supervised margin classifier, reported as one of the best algorithms for text classification [35] [36]. We chose the LinearSVC algorithm in the *Scikit-learn* library [37], which implements a one-versus-all (OVA) multi-class strategy. This algorithm is suitable for high-dimensional datasets and is characterized by a low running time [38].

Unlike PLMs, traditional machine learning models require pre-processing data cleaning steps. In our study, we used the following pre-processing steps on the four datasets: (i) removing missing data; (ii) removing numbers and special characters; (iii)

lower casing; (iv) tokenization; (v) lemmatization; and (vi) word vectorization using TFIDF¹.

It is important to note that when applying the TFIDF vectorizer, we tried different N-grams. An ‘N-gram’ is simply a sequence of N words that predicts the occurrence of a word based on the occurrence of its ($N - 1$) previous words. The default setting is Unigrams. In our study, we used trigrams which means that we included feature vectors consisting of all unigrams, bigrams, and trigrams.

4 Results

In this section, we discuss the results of applying four different fine-tuned PLMs (i.e., BERT, DistilBERT, RoBERTa, XLM) and a linear SVM classifier on the four datasets described in Section 3.1.

Table 2 shows the F1-scores obtained when applying the four PLMs and a linear SVM classifier on the four datasets. When evaluating PLMs, we used 3 epochs because we observed that when the number of epochs exceeds 3, the training loss decreases with each epoch and the validation loss increases. This translates to overfitting. Thus, all our experiments are run for 3 epochs only.

For the domain-specific dataset, it is clear how the linear SVM achieves a comparable performance (0.79) as any of the fine-tuned PLMs. Similarly, for the BBC dataset, SVM surprisingly achieves the same F1-score (0.98) as RoBERTa on the third epoch. However, we expected that PLMs would significantly outperform SVM on general domain datasets.

For the 20NewsGroup, SVM outperformed all PLMs with an F1-score of 0.93. This accuracy score was a result of considering the meta-data (i.e., headers, footers, and quotes) as part of the text that is fed to the classifier. However, when we ignored the meta-data, there was a performance drop of 15%.

The last dataset is the Consumer Complaints which is the largest dataset (570,279 instances) as described in Table 1. The accuracy of the linear SVM (0.82) was very close to the highest accuracy of 0.85 obtained by BERT and RoBERTa. While 0.82 is very competitive, we believe there is room for improvement if feature selection techniques were considered as this dataset is characterized by a large feature set.

The accuracy scores of PLMs are generally higher on generic datasets that do not contain domain-specific or rare words. Also, we notice a small gap between the accuracy scores of all PLMs in the third epoch for all datasets.

In summary, the key points are:

- Linear SVM proved to be comparable to PLMs for text classification tasks.
- PLMs accuracy scores are generally higher on generic datasets.

¹ TFIDF stands for Term Frequency-Inverse Document Frequency, which is a combination of two metrics:

1. Term frequency (*tf*): a measure of how frequently a term t , appears in a document d .
2. Inverse document frequency(*idf*): a measure of how important a term is. It is computed by dividing the total number of documents in our corpus by the document frequency for each term and then applying logarithmic scaling on the result.

- The importance of feature engineering for text classification is highlighted by including meta-data.

Table 2. Comparison of four PLMs against SVM Linear classifier in terms of accuracy (F1-score)

Dataset	Model	Epoch 1	Epoch 2	Epoch 3
		Accuracy (F1-score)		
IT Support Tickets	<i>BERT</i>	0.78	0.79	0.79
	<i>DistilBERT</i>	0.77	0.78	0.79
	<i>XLM</i>	0.77	0.79	0.79
	<i>RoBERTa</i>	0.77	0.78	0.79
	<i>LinearSVM(n-gram=3)</i>	0.79		
BBC	<i>BERT</i>	0.97	0.97	0.97
	<i>DistilBERT</i>	0.97	0.97	0.97
	<i>XLM</i>	0.88	0.96	0.97
	<i>RoBERTa</i>	0.97	0.97	0.98
	<i>LinearSVM(n-gram=3)</i>	0.98		
20News-Group	<i>BERT</i>	0.85	0.91	0.92
	<i>DistilBERT</i>	0.82	0.90	0.90
	<i>XLM</i>	0.89	0.91	0.92
	<i>RoBERTa</i>	0.84	0.87	0.90
	<i>LinearSVM</i>	0.93		
Consumer Complaints	<i>BERT</i>	0.83	0.84	0.85
	<i>DistilBERT</i>	0.82	0.84	0.84
	<i>XLM</i>	0.80	0.82	0.83
	<i>RoBERTa</i>	0.83	0.84	0.85
	<i>LinearSVM</i>	0.82		

5 Conclusions

The study described in this paper compares the performance of several fine-tuned PLMs (see Section 3.2) against that of a linear SVM classifier (see Section 3.3) for the task of text classification. The datasets used in the study are: a domain-specific dataset of real-world support tickets from a large organization as well as three generic datasets (see Table 1).

To our surprise, we found that a pre-trained language model does not provide significant gains over the linear SVM classifier. We expected PLMs to outperform SVM on the generic datasets, however, our study indicates comparable performance for both models (see Table 2). Also, our study indicates that SVM outperforms PLMs on one of the generic datasets (i.e., 20NewsGroup).

Our finding goes against the trend of using PLMs on any NLP task. Thus, for text classification, we recommend prudence when deciding on the type of algorithms to use. Since our study seems to be the first comparative study of PLMs against SVM on generic datasets as well as on a domain-specific dataset, we encourage replication of this study to create a solid body of knowledge for confident decision-making on the choice of algorithms.

References

1. Brundage, M.P., Sexton, T., Hodkiewicz, M., Dima, A. and Lukens, S.: Technical language processing: Unlocking maintenance knowledge. *Manufacturing Letters*, 27, pp. 42-46 (2021)
2. Devlin, J., Chang, M.W., Lee, K. and Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis. pp. 4171–4186 (2019)
3. Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer.: Deep contextualized word representations. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. New Orleans. pp. 2227–2237 (2018)
4. Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., Zhang, L., Han, W., Huang, M. and Jin, Q.: Pre-trained models: Past, present and future. *AI Open* (2021)
5. Aronoff, M. and Rees-Miller, J. eds: *The handbook of linguistics*. John Wiley & Sons (2020)
6. Acheampong, F.A., Nunoo-Mensah, H. and Chen, W.: Transformer models for text-based emotion detection: a review of BERT-based approaches. *Artificial Intelligence Review*, pp.1-41 (2021)
7. Han, X., Zhao, W., Ding, N., Liu, Z. and Sun, M.: Ptr: Prompt tuning with rules for text classification. *arXiv preprint arXiv:2105.11259* (2021)
8. Schick, T. and Schütze, H.: Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, No. 05, pp. 8766-8774 (2020)
9. McCoy, R. T., Pavlick, E. and Linzen, T.: Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy (2019)
10. Zhao, Z., Zhang, Z. and Hopfgartner, F.: A Comparative Study of Using Pre-trained Language Models for Toxic Comment Classification. In: *Companion Proceedings of the Web Conference*, pp. 500-507 (2021)
11. Zheng, S. and Yang, M.: A new method of improving bert for text classification. In: *International Conference on Intelligent Science and Big Data Engineering*, pp. 442-452 Springer, Cham (2019)
12. Conneau, A. and Lample, G.: Cross-lingual language model pretraining. In: *Proceedings of the Advances in Neural Information Processing Systems*. Vancouver. pp. 7057–7067 (2019)
13. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. *arXiv pre-print arXiv:1907.11692* (2019)

14. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R. and Le, Q.V.: XLNet: Generalized autoregressive pretraining for language understanding. In: Proceedings of the Advances in Neural Information Processing Systems. Vancouver. pp. 5754–5764 (2019)
15. Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D. and Smith, N.A.: Don't stop pretraining: adapt language models to domains and tasks. In: Proceedings of ACL (2020)
16. Beltagy, I., Lo, K. and Cohan, A.: SciBERT: A pretrained language model for scientific text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Hong Kong, pp. 3613– 3618 (2019)
17. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H. and Kang, J.: BioBERT: A pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, pp. 1234–1240 (2020)
18. Huang, K., Altosaar, J. and Ranganath, R.: ClinicalBERT: Modeling clinical notes and predicting hospital readmission. ArXiv: 1904.05342 (2019)
19. Alsentzer, E., Murphy, J.R., Boag, W., Weng, W.H., Jin, D., Naumann, T. and McDermott, M.: Publicly available clinical BERT embeddings. In: Proceedings of the 2nd Clinical Natural Language Processing Workshop, Association for Computational Linguistics, Minneapolis, Minnesota, USA, pp.72-78 (2019)
20. Araci, D.: FinBERT: financial sentiment analysis with pre-trained language models. arXiv preprint. arXiv:1908.10063 (2019)
21. Elwany, E., Moore, D. and Oberoi, G.: Bert goes to law school: Quantifying the competitive advantage of access to large legal corpora in contract understanding. In: Proceedings of NeurIPS Workshop on Document Intelligence (2019)
22. Lu, Daming.: Masked Reasoner at SemEval-2020 Task 4: Fine-Tuning RoBERTa for Commonsense Reasoning. In: Proceedings of the Fourteenth Workshop on Semantic Evaluation. pp. 411-414 (2020)
23. Tang, T., Tang, X. and Yuan, T.: Fine-Tuning BERT for Multi-Label Sentiment Analysis in Unbalanced Code-Switching Text. IEEE Access, vol. 8, pp. 193248-193256 (2020)
24. Yuan, J., Bian, Y., Cai, X., Huang, J., Ye, Z., Church, K.: Disfluencies and fine-tuning pre-trained language models for detection of alzheimer's disease. In: INTER-SPEECH, pp. 2162–2166 (2020)
25. Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H. and Wang, H.: Ernie 2.0: A continual pre-training framework for language understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, No. 05, pp. 8968-8975 (2020)
26. Kao, W.T., Wu, T.H., Chi, P.H., Hsieh, C.C. and Lee, H.Y.: BERT's output layer recognizes all hidden layers? Some Intriguing Phenomena and a simple way to boost BERT. arXiv preprint arXiv:2001.09309 (2020)
27. Kovaleva, O., Romanov, A., Rogers, A., Rumshisky, A.: Revealing the dark secrets of BERT. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China (2019)
28. Greene, D., and Cunningham, P.: Practical solutions to the problem of diagonal dominance in kernel document clustering. In: Proceedings of the 23rd international conference on Machine learning. pp. 377–384 (2006)
29. 20 Newsgroups Data Set Homepage, <http://qwone.com/~jason/20Newsgroups/>. Online, last accessed March 2022.
30. Consumer Complaint Database Homepage, <https://www.consumerfinance.gov/data-research/consumer-complaints/>.Online, last accessed March 2022.

31. Zhou, Z.H. and Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem, *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 63-77 (2006)
32. He, H., Ma, Y.: *Imbalanced Learning: Foundations, Algorithms, and Applications*, 1st edn. Wiley-IEEE Press, New York (2013)
33. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., and Klingner, J.: Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016)
34. Sanh, V., Debut, L., Chaumond, J. and Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In: *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing (EMC2) co-located with the Thirty-third Conference on Neural Information Processing Systems (NeurIPS 2019)*, pp. 1–5 (2019)
35. T. Joachims.: Text categorization with Support Vector Machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds) *Machine Learning: ECML-98*. ECML 1998. Lecture Notes in Computer Science, vol 1398. Springer, Berlin, Heidelberg, pp.137–142 (1998)
36. Telnoni, P.A., Budiawan, R. and Qana'a, M.: Comparison of Machine Learning Classification Method on Text-based Case in Twitter. In: *Proceedings of International Conference on ICT for Smart Society: Innovation and Transformation Toward Smart Region, ICISS* (2019)
37. 1.4. Support Vector Machines — scikit-learn 0.23.1 documentation. <https://scikit-learn.org/stable/modules/svm.html>, last accessed March 2022.
38. V. K. Chauhan, K. Dahiya, and A. Sharma.: Problem formulations and solvers in linear SVM: a review, *Artificial Intelligence Review*, vol. 52, no. 2. Springer Netherlands, pp. 803–855 (2019)