BertSentiment-SO Model: Extracting Cloud Service Information from Stack Overflow's Technical Texts

Aishwarya Sundaram[1], Leow Jun Shou[2], Hema Subramaniam[3]

s2029083@siswa.um.edu.my[1] , 17123313@siswa.um.edu.my[2], hema@um.edu.my[3]

Institute for Advanced Studies, Universiti Malaya, Kuala Lumpur, Malaysia[1], Department of Information Systems, Faculty of Computer Science and Information Technology, Universiti Malaya, Kuala Lumpur, Malaysia[2], Department of Software Engineering, Faculty of Computer Science and Information Technology, Universiti Malaya, Kuala Lumpur, Malaysia[3]

Abstract. The recent surge in sentiment analysis within software engineering texts has gained prominence due to its proven influence on productivity and collaborative work quality. Stack Overflow, a substantial repository of software engineering content, has become a focal point for sentiment analysis research, particularly in software development. Nevertheless, integrating sentiment analysis into software engineering has unveiled potential shortcomings in their efficacy. Researchers have had to adapt these tools to navigate limitations, as exemplified by the cases of SentiCR and Senti4SD. Moreover, natural language processing has made significant strides in recent times, primarily attributed to the emergence of pre-trained transformer-based models. Addressing this research gap, the study introduces a dedicated sentiment analysis model, BertSentiment-SO. This study delves explicitly into sentiment analysis for the three foremost cloud service providers: Amazon, Google, and Microsoft. Extracted from the Big Query public dataset, the data undergoes preprocessing and labelling via a semi-supervised learning approach. Harnessing the capabilities of BERT, our proposed model outperforms its counterparts, enabling a more precise score of 94.06% for the evaluation of sentiment within the domain of cloud services.

Keywords: Technical text, sentiment analysis, cloud service providers, stack overflow.

1 Introduction

In the ever-evolving landscape of technology and information, the importance of technical text mining has emerged as a crucial mechanism for unravelling the wealth of knowledge embedded within the vast expanse of digital content [1]. As industries increasingly digitised, technical texts have become the cornerstone of knowledge dissemination, containing insights, solutions, and expertise that fuel innovation and drive progress. Recently, the prominence of techniques and approaches to automatically extract textual information from software engineering, encompassing coding, frameworks, and cloud services, has grown substantially. These are

IACIDS 2023, November 23-25, Lavasa, India
Copyright © 2024 EAI
DOI 10.4108/eai.23-11-2023.2343182

commonly referred to as "technical texts." Sentiment analyser to mine sentiment automatically for the software engineering (SE) domain has grown in popularity recently. The primary application of these techniques lies in discerning the emotions and sentiments conveyed within textual reviews provided by developers. Understanding the mood of developers holds significant implications for their effectiveness, code quality, and overall job contentment [2]. The underlying aim is to classify the sentiment within programmers' comments as neutral, negative, or positive. This approach has gathered considerable interest and is recognised as sentiment analysis within software engineering. Among its uses in software engineering, sentiment analysis has been used to determine the emotional state of developers [3], to examine the polarity of reviewers' assessments, and to analyse if sentiment analysis tools agree with human reviewers. The research investigated the emotions in commit comments from different open-source projects [4].

Nevertheless, it's important to note that software engineering technical documents are not designed to align seamlessly with general sentiment analysis tools. Examples of such tools include the Stanford CoreNLP [5], which is typically employed for evaluating sentiment in movie reviews, and the SentiStrength tool, which assesses the overall emotional tone of a text by assigning scores to individual words [6]. The use of these sentiment analysis tools in the context of software engineering research has yielded unfavourable results [7].To address this challenge, previous research has made efforts to adapt sentiment analysis tools for the specific domain of software engineering. For instance, some studies have leveraged developers' opinions extracted from Stack Overflow(SO) to create a specialized sentiment analysis tool tailored to the platform, known as Stanford CoreNLP SO-specific sentiment analysis [8]. Similarly, adaptations like SentiStrength-SE have been introduced to enhance the accuracy of sentiment classification in texts related to software engineering [9]. Additionally, a unique sentiment analysis tool called SentiCR has been developed specifically for code reviews [10]. Furthermore, the Senti4SD model has been trained using data from SO to better suit the needs of software development research [11].

In recent years, Natural Language Processing (NLP) has witnessed a transformative shift, primarily driven by the emergence of pre-trained models [12]. Among these groundbreaking models, Bidirectional Encoder Representations from Transformers (BERT), a novel language representation model, has taken center stage. BERT employs a fine-tuning approach to adapt pre-trained transformer models for specific tasks [13]. Remarkably, BERT's pre-trained model, when applied to NLP tasks, demonstrates its versatility by requiring only a single additional output layer to excel in cutting-edge applications such as language inference [13]. Furthermore, Bidirectional Encoder Representations from Transformers (BERT) has been employed in cross-platform assessments to evaluate the performance of SE-specific tools [14]. These evaluations extended to the use of test sets sourced from entirely different data repositories than those used for model training. In benchmark studies, BERT, along with Senti4SD, consistently outperformed other models in each comparison [15]. Notably, BERT models achieved the highest micro-averaged F1-score of 0.90 in sentiment analysis of SO texts, surpassing the performance of the four general sentiment analyzers mentioned.

These technical texts are frequently encountered in discussion forums, API reviews, and similar contexts. However, a notable gap exists in the need for an established model tailored to study technical text related to cloud services. This shortcoming has led to suboptimal performance in this domain. Several studies indicate the presence of sentiments within cloud service providers' discussions on Twitter [16] [17] [18]. However, no existing model or research studies explore sentiment within technical text among cloud service providers. SO, a platform for asking and answering software development queries, is a valuable resource for accumulating specialised datasets. Enhancing the model's effectiveness is imperative as it enables companies to gain deeper insights into developers' genuine sentiments towards specific services. Also, developers can understand the maturity of a development community, which can help improve productivity and quality [19].

2 Related works

Sentiment analysis methods are widely classified into machine-learning and lexical-based approaches [20]. Methods relying on lexical features categorize articles by evaluating the frequency or occurrence of sentiment-related terms. This is done using a predefined collection of words and n-grams, each associated with specific sentiment tags [20]. SentiStrength [6], OpinerDSOSenti [21], and VADER are examples of lexical-based approaches in the table above [22]. Opinion lexicons have been developed via manual labelling, dictionary-based techniques, and corpus-based machine-learning procedures. Approaches rooted in dictionary-based methods create opinion words by establishing the transitive closure of an initial set of words' synonyms and antonyms. This process commonly involves utilizing online lexical databases, like WordNet. On the other hand, corpus-based techniques construct a word collection using a corpus which are particular to the domain and a collection of linguistic restrictions. A comprehensive examination of 23 published sentiment analysers shows that sentiment classifiers perform differently when classifying diverse datasets. Furthermore, it demonstrates that emotion analysers may disagree with one another, implying that the choice of analysers can result in drastically varied classification results [20]. SentiStrength is the most commonly used analyser in sentiment analysis for SE documents [9]. SentiStrength was developed to classify the sentiment of brief, informal messages. It relies on a sentiment dictionary that encompasses terms with non-standard spellings. Utilizing machine learning techniques, the sentiment term weights were optimized by analyzing a corpus of papers from MySpace. Sentiment of documents is evaluated by SentiStrength which assigns positive and negative scores to words associated with various emotions [6]. As of 2017, a minimum of three sentiment analyzers had been created, including SentiStrength-SE [9], SentiCR [10] and Senti4SD [11]. As per the latest benchmark studies, the three SE analyzers mentioned previously outperformed SentiStrength when utilized for sentiment analysis of software engineering texts [23]. [24] introduced Senti-Analyzer, a model that combines Senti4SD, SentiStrength-S, SentiStrength, and TextBlob.

Deep learning involves analysing substantial volumes of intricate data, allowing models to acquire knowledge and formulate predictions or decisions without explicit programming. Deep learning techniques are employed across a broad spectrum of domains, which includes software, speech education, robotics, natural language processing, and various other fields. [25] proposed model RNN4SentiSE to analyse the effectiveness of RNN-based sentiment analyser for the software engineering field. [26] mined the product review data online to create a new dataset named CloudRiviews. Twitter serves as a social media platform and online news source, allowing users to share and engage with concise messages known as "tweets". Developers can check Twitter for user reviews or opinions by searching for hashtags related to cloud products or services. It may also be helpful to check technology websites or forums for more in-depth reviews of cloud products [16] [18]. To summarise, ongoing research investigates technical texts using either lexicon-based or machine-learning methods, often employing technical data and specialised lexicon dictionaries as their foundation. Evidence demonstrates that training models with technical data or technical lexicons can enhance performance in sentiment analysis. However, there needs to be more studies that focus specifically on cloud services' technical data. While some research does exist on cloud services, it primarily revolves around product reviews and Twitter data, needing more perspective from developers. Consequently, this research gap forms the focus of this study objective.

3 Proposed solution

To bridge the existing research gap, we have developed BertSentiment-SO, tailored explicitly for cloud services technology. This endeavour capitalises on the inherent cloud computing expertise in the technical content related to cloud services on SO. Our approach involves fine-tuning the BERT model, which has been proven adaptable for sentiment analysis in spectrum of SE [27]. Illustrated in Figure 1, the process of fine-tuning BERT for sentiment analysis encompasses several vital steps. This entails the selection of a pre-trained BERT model, input data provisioning, and incorporating a sentiment classification layer to facilitate model refinement. Initially developed by Google in 2018, the BERT model has since gained remarkable popularity in natural language processing (NLP) tasks due to its exceptional performance and versatility. Operating on a transformer-based neural network architecture foundation, BERT is trained on an extensive text corpus to generate contextual word representations, also known as embeddings. Distinguishing itself with its bidirectional nature, BERT considers both preceding and following contexts for each word in a sentence, a departure from the unidirectional context consideration in conventional language models. This bidirectional approach significantly augments BERT's performance on NLP tasks, making it a preferred choice in various applications.

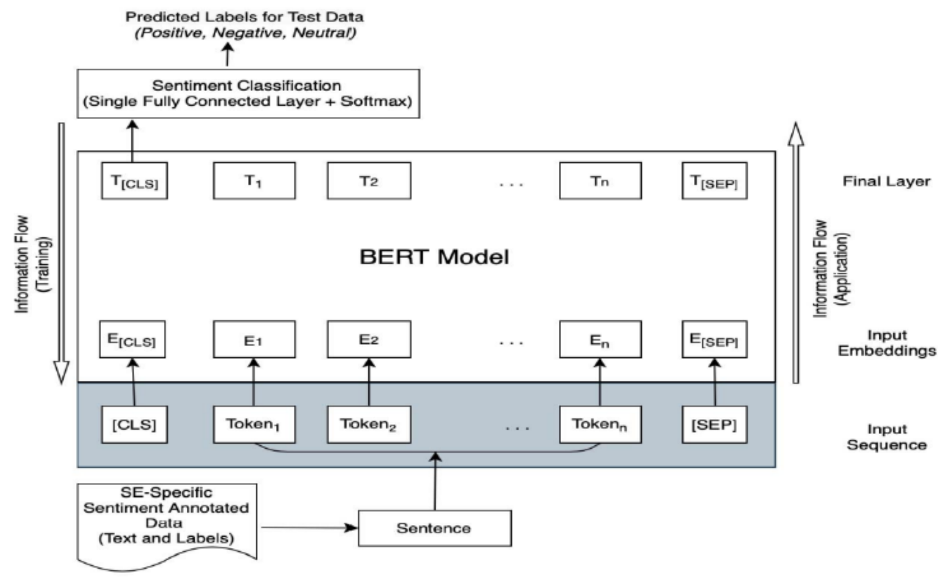
Predicted Labels for Test Data
(Positive, Negative, Neutral)
Sentiment Classification
(Single Fully Connected Layer + Softmax)
Information Flow (Training)
Information Flow (Application)
T[CLS]
T1
T2
...
Tn
T[SEP]
Final Layer
BERT Model
E[CLS]
E1
E2
...
En
E[SEP]
Input Embeddings
[CLS]
Token1
Token2
...
Tokenn
[SEP]
Input Sequence
SE-Specific Sentiment Annotated Data (Text and Labels)
Sentence

Fig 1. Proposed BertSentiment-SO model

The BertSentiment-SO model involves the fine-tuning of BERT specifically for the purpose of extracting technical opinions from SO data expressed by software engineers. In the fine-tuning process, we would use the SE-Specific Sentiment Annotated Data [11] from the official SO dump of user-generated content from July 2008 to September 2015. These sentences from this annotated dataset serve as the input data for the fine-tuning process for SO context. SE-Specific Sentiment Annotated Data are then tokenized and converted into BERT word embeddings. These embeddings capture the context and meaning of words within the sentences. The BERT model, as shown in the figure, takes the input embeddings for the sentences from the SE-Specific Sentiment Annotated Data. BERT processes these embeddings to generate contextualized representations for the input sentences. The final layer then maps the BERT representations to sentiment predictions. After passing through the final layer, the output is then passed through a connected layer with softmax activation function. This converts the model's raw scores into probability distributions over the sentiment classes. The output of the layer represents the model's confidence in each sentiment category for the input sentence. The probabilities obtained from the fully connected layer are used to predict the sentiment label for each sentence in the test data. Thus, our proposed model fine tunes BERT for technical data mining from SO.

4 Experimental analysis

Recall, Precision, F1, and Accuracy (ACC) metrics are used to assess our proposed BertSentiment-SO's effectiveness against Senti4SD, BERT, and RoBERTa models. While recall measures the model's capability to correctly identify all pertinent instances, precision measures the correctness of a model's positive predictions. The F1 score, a metric that combines precision and recall, provides a balanced performance evaluation by taking into account both false positives and false negatives. Meanwhile, accuracy reflects the ratio of correctly classified instances to all instances, serving as a general measure to evaluate how accurately the model predicts sentiment labels. Table 1 shows the performance result comparisons between selected models.

Table 1. Performance comparison

Model	Recall	Precision	f1	ACC
Senti4SD	61.35%	66.93%	59.59%	61.35%
BERT	57.75%	57.31%	57.22%	57.75%
RoBERTa	65.92%	66.59%	66.05%	65.92%
BertSentiment-SO	94.03%	94.06%	94.02%	94.03%

Senti4SD achieves a relatively high recall of 61.35% but has a lower precision of 66.93%, resulting in an F1 score of 59.59%. Its accuracy is 61.35%. We can conclude that Senti4SD is good at identifying relevant data points but may have more false positives. BERT has a balanced performance with a recall, precision, and F1 score of around 57%. Its accuracy is 57.75%. While it performs reasonably well, it doesn't excel in either precision or recall. RoBERTa shows improved performance with a recall of 65.92% and a precision of 66.59%, resulting in an F1 score of 66.05%. Its accuracy is 65.92%, making it a better model compared to the Senti4SD and BERT. BertSentiment-SO outperforms other models with a high recall of 94.03%, precision of 94.06%, and an F1 score of 94.02%. Its accuracy is 94.03%, indicating that it excels in both precision and recall, making it the best-performing model among all the considered ones. In summary, BertSentiment-SO, trained using 45K combined data, stands out as the best-performing model, achieving consistently high scores across all metrics. RoBERTa also demonstrates strong performance, while Senti4SD and BERT show relatively lower performance, with Senti4SD having a slightly better balance between recall and precision than BERT. This model is not only able to perform well on cloud technical text but also on general technical text.

5 Conclusion

The proposed solution seeks to bridge the research gap by developing BertSentiment-SO for cloud services' technical domain. Leveraging SO's wealth of cloud computing expertise, this approach finetunes the BERT model for sentiment analysis. BERT's bidirectional transformer architecture, combined with domain-specific training data, enables it to capture nuanced sentiment patterns in technical text. This model, when compared to existing sentiment analysers like Senti4SD, BERT, and RoBERTa, outperforms in terms of recall, precision, f1 score, and accuracy, showcasing its ability to analyse sentiment across various technical domains effectively. In conclusion, the rapid evolution of technology has necessitated sophisticated methods for extracting insights from technical texts. Sentiment analysis, particularly in software engineering and cloud services, plays a pivotal role in understanding developers' sentiments and shaping service offerings. Introducing pre-trained models like BERT has revolutionised the field, enabling domain-specific sentiment analysis. However, challenges must still be overcome in adapting these models for specific technical domains like cloud services. The proposed BertSentiment-SO model solves this challenge, offering a novel approach to sentiment analysis in the cloud services technical field. Its fine-tuned BERT model demonstrates exceptional performance in accurately classifying sentiments, thereby bridging the gap in understanding developers' sentiments towards cloud services. This research enhances the quality and efficiency of cloud services by providing insights into the developer community's emotions and perceptions.

References

[1] Y. W. Lai and M. Y. Chen, "Review of Survey Research in Fuzzy Approach for Text Mining," Ieee Access, vol. 11, pp. 39635-39649, 2023, doi: 10.1109/access.2023.3268165.

[2] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson, "What happens when software developers are (un)happy," Journal of Systems and Software, vol. 140, pp. 32-47, 2018, doi: https://doi.org/10.1016/j.jss.2018.02.041.

[3] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," Ain Shams Engineering Journal, vol. 5, no. 4, pp. 1093-1113, 2014, doi: https://doi.org/10.1016/j.asej.2014.04.011.

[4] E. Guzman, D. Azócar, and Y. Li, "Sentiment analysis of commit comments in GitHub: An empirical study," 11th Working Conference on Mining Software Repositories, MSR 2014 - Proceedings, 2014, doi: 10.1145/2597073.2597118.

[5] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, The Stanford CoreNLP Natural Language Processing Toolkit. 2014.

[6] T. Mike, B. Kevan, P. Georgios, C. Di, and K. Arvid, "Sentiment strength detection in short informal text," Journal of the American Society for Information Science and Technology, vol. 61, no. 12, pp. 2544-2558, 2010, doi: DOI: 10.1002/asi.21416.

[7] R. Jongeling, P. Sarkar, S. Datta, and A. Serebrenik, "On negative results when using sentiment analysis tools for software engineering research," Empirical Software Engineering, vol. 22, no. 5, pp. 2543-2584, 2017, doi: 10.1007/s10664-016-9493-x.

[8] B. Lin, F. Zampetti, G. Bavota, M. D. Penta, M. Lanza, and R. Oliveto, "Sentiment Analysis for Software Engineering: How Far Can We Go?," in 2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE), 27 May-3 June 2018 2018, pp. 94-104, doi: 10.1145/3180155.3180195.

[9] M. R. Islam and M. F. Zibran, "Leveraging Automated Sentiment Analysis in Software Engineering," in 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), 20-21 May 2017 2017, pp. 203-214, doi: 10.1109/MSR.2017.9.

[10] T. Ahmed, A. Bosu, A. Iqbal, and S. Rahimi, "SentiCR: A customized sentiment analysis tool for code review interactions," in 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE), 30 Oct.-3 Nov. 2017 2017, pp. 106-111, doi: 10.1109/ASE.2017.8115623.

[11] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, "Sentiment Polarity Detection for Software Development," Empirical Software Engineering, vol. 23, no. 3, pp. 1352-1382, 2018, doi: 10.1007/s10664-017-9546-9.

[12] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," Science China Technological Sciences, vol. 63, no. 10, pp. 1872-1897, 2020, doi 10.1007/s11431-020-1647-3.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2018.

[14] N. Novielli, F. Calefato, D. Dongiovanni, D. Girardi, and F. Lanubile, "Can We Use SE-specific Sentiment Analysis Tools in a Cross-Platform Setting?," in 2020 IEEE/ACM 17th International Conference on Mining Software Repositories (MSR), 25-26 May 2020 2020, pp. 158-168, doi: 10.1145/3379597.3387446. [Online]. Available:
http://doi.ieeecomputersociety.org/10.1145/3379597.3387446

[15] T. Zhang, B. Xu, F. Thung, S. A. Haryono, D. Lo, and L. Jiang, "Sentiment Analysis for Software Engineering: How Far Can Pre-trained Transformer Models Go?," in 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME), 28 Sept.-2 Oct. 2020 2020, pp. 70-80, doi: 10.1109/ICSME46990.2020.00017.

[16] I. Karamitsos, S. Thabit Albarhami, and C. Apostolopoulos, "Tweet Sentiment Analysis (TSA) for Cloud Providers Using Classification Algorithms and Latent Semantic Analysis," Journal of Data Analysis and Information Processing, vol. 07, pp. 276-294, 2019, doi: 10.4236/jdaip.2019.74016.

[17] L. M. Qaisi and I. Aljarah, "A Twitter sentiment analysis for cloud providers: A case study of Azure vs. AWS," in 2016 7th International Conference on Computer Science and Information Technology (CSIT), 13-14 July 2016 2016, pp. 1-6, doi: 10.1109/CSIT.2016.7549473.

[18] A. Koneru, N. B. N. S. R. Bhavani, K. P. Rao, G. S. Prakash, I. P. Kumar, and V. V. Kumar, "Sentiment Analysis on Top Five Cloud Service Providers in the Market," in 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), 11-12 May 2018 2018, pp. 293-297, doi: 10.1109/ICOEI.2018.8553970.

[19] E. Guzman and B. Bruegge, Towards emotional awareness in software development teams. 2013, pp. 671-674.

[20] F. N. Ribeiro, M. Araújo, P. Gonçalves, M. André Gonçalves, and F. Benevenuto, "SentiBench - a benchmark comparison of state-of-the-practice sentiment analysis methods," EPJ Data Science, vol. 5, no. 1, p. 23, 2016, doi: 10.1140/epjds/s13688-016-0085-1.

[21] G. Uddin, F. Khomh, and C. K. Roy, "Mining API usage scenarios from stack overflow," Information and Software Technology, vol. 122, p. 106277, 2020, doi: https://doi.org/10.1016/j.infsof.2020.106277.

[22] N. Kozanidis, R. Verdecchia, and E. Guzmán, Asking about Technical Debt: Characteristics and Automatic Identification of Technical Debt Questions on Stack Overflow. 2022.

[23] N. Novielli, D. Girardi, and F. Lanubile, "A Benchmark Study on Sentiment Analysis for Software Engineering Research," 2018, doi: 10.1145/3196398.3196403.

[24] M. Herrmann and J. Klünder, "From Textual to Verbal Communication: Towards Applying Sentiment Analysis to a Software Project Meeting," in 2021 IEEE 29th International Requirements Engineering Conference Workshops (REW), 20-24 Sept. 2021 2021, pp. 371-376, doi: 10.1109/REW53955.2021.00065.

[25] E. Biswas, K. Vijay-Shanker, and L. Pollock, Exploring Word Embedding Techniques to Improve Sentiment Analysis of Software Engineering Texts. 2019.

[26] M. R. Raza, "DEEP LEARNING-BASED SENTIMENT ANALYSIS FOR CLOUD PROVIDER SELECTION," 2022, doi: 10.13140/RG.2.2.11479.55202.

[27] M. Koroteev, BERT: A Review of Applications in Natural Language Processing and Understanding. 2021.