ISCTE IUL
REPOSITÓRIO
INSTITUTO UNIVERSITÁRIO DE LISBOA

Repositório ISCTE-IUL

Deposited in *Repositório ISCTE-IUL*:
2026-10-09

Deposited version:
Accepted Version

Peer-review status of attached file:
Peer-reviewed

Citation for published item:
Carneiro, G., Goto, R., Santos, J., Silva, B., Neto, M. & Brito e Abreu, F. (2027). Sentiment analysis using LLMs: A systematic mapping study in software engineering. In Américo Rio, José Pereira dos Reis, Stamatia Bibi, Ricardo Machado (Ed.), Quality of information and communications technology: 18th International Conference, QUATIC 2025, Proceedings (pp. 63?79). Springer. https://doi.org/10.1007/978-3-032-21373-0_5

Further information on publisher's website:
10.1007/978-3-032-21373-0_5

Publisher's copyright statement:
This is the peer reviewed version of the following article: Carneiro, G., Goto, R., Santos, J., Silva, B., Neto, M. & Brito e Abreu, F. (2027). Sentiment analysis using LLMs: A systematic mapping study in software engineering. In Américo Rio, José Pereira dos Reis, Stamatia Bibi, Ricardo Machado (Ed.), Quality of information and communications technology: 18th International Conference, QUATIC 2025, Proceedings (pp. 63?79). Springer. https://doi.org/10.1007/978-3-032-21373-0_5, which has been published in final form at https://dx.doi.org/10.1007/978-3-032-21373-0_5. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

Use policy

Creative Commons CC BY 4.0
The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Serviços de Informação e Documentação, Instituto Universitário de Lisboa (ISCTE-IUL)
Av. das Forças Armadas, Edifício II, 1649-026 Lisboa Portugal
Phone: +(351) 217 903 024 | e-mail: administrador.repositorio@iscte-iul.pt
https://repositorio.iscte-iul.pt

Sentiment Analysis Using LLMs: A Systematic Mapping Study in Software Engineering

Glauco de Figueiredo Carneiro[1], Rafael Takeguma Goto[1], José Amancio Macedo Santos[2], Bruna dos Santos Silva[1], Manoel Gomes de Mendonça Neto[3], and Fernando Brito e Abreu[4]

[1] Federal University of Sergipe (UFS), São Cristóvão, SE 49100-000, Brazil
[2] State Univ. of Feira de Santana (UEFS), Feira de Santana, BA 44036-900, Brazil
[3] Federal University of Bahia (UFBA), Salvador, BA 40170-110, Brazil
[4] University Institute of Lisbon (ISCTE-IUL), 1649-026 Lisbon, Portugal
{glauco.carneiro,rafael.goto,bruna.silva}@dcomp.ufs.br; zeamancio@uefs.br; manoel.mendonca@ufba.br; fba@iscte-iul.pt

Abstract. Assessing programmers' perspectives during software life cycle activities is vital for understanding the reasoning behind software development decisions. Sentiment Analysis (SA) plays a key role in this assessment. Researchers have increasingly leveraged Natural Language Processing (NLP) to conduct such analyses. However, there is a lack of comprehensive and up-to-date studies that explore the challenges and opportunities of applying Large Language Models (LLMs) for SA in Software Engineering (SE). This study systematically maps the topic by reviewing studies published between 2019 and 2025. Following the established protocol, we selected 21 studies that offer a broad perspective on how NLP capabilities within LLMs facilitate SA in software development tasks. Evidence from the selected studies highlights the positive impact of NLP and LLMs on SA in various areas of SE. For instance, some studies apply LLM-driven SA to bug reports and software requirements. Additionally, the mapping study identifies different approaches to utilizing LLMs for SA in SE, including model comparisons, fine-tuning, and prompt engineering.

Keywords: Sentiment Analysis · Systematic Mapping · Large Language Models · Software Engineering

1 Introduction

Programmers share their opinions through written text while executing software life cycle activities in tools and repositories such as version control systems, IDEs, issue trackers, and validation, verification, and testing tools. These texts can be analyzed using natural language processing (NLP) functionalities regarding programmers' sentiments. Understanding the sentiments, opinions, and emotions expressed in these texts can offer insights into the challenges faced in software development [26]. It can also lead to the development of better tools that utilize NLP techniques [19]. The literature provides evidence of the ongoing

progress of Artificial Intelligence (AI). One may highlight the development of Transformer models, which feature self-attention mechanisms and utilize LLMs for enhanced query and content understanding in the context of NLP [35]. Notably, models based on the Bidirectional Encoder Representations from Transformers (BERT) architecture have demonstrated state-of-the-art performance across numerous NLP tasks, including sentiment analysis, establishing expectations for transformer-based approaches in software engineering contexts [10]. For this reason, Large Language Models (LLMs) with transformers as their base architectural component are the cornerstones of generative AI (GenAI). LLMs have improved abilities in comprehending, generating, and enhancing human language [23]. The potential application of LLMs to perform NLP tasks presents an unprecedented frontier in Sentiment Analysis (SA), promising to significantly enhance search functionalities and redefine user interaction with digital information systems. Researchers have investigated the potential use of transformer architecture implemented as LLMs for SA [38].

Regarding SA challenges, there is evidence that general-purpose SA tools are often ineffective in the software domain, as they are not typically trained on datasets related to SE [3], which can produce harmful and biased results [21]. That is why several studies indicated the need for SE-specific tools for SA [34]. The topic should be considered by researchers who are considering LLMs as a support tool for SA. Considering the novelty of NLP in adopting LLMs to support SA in SE, the area lacks syntheses describing the adopted approaches and their effects on software quality. This perception aligns with other authors, who consider the area to lack knowledge about how well LLMs comprehend words and sentences within a SE context and how they advance the state-of-the-art in this field [24]. To address this gap, we conducted a systematic mapping study to provide a comprehensive overview of how NLP capabilities in LLMs are being leveraged to support SA. Our study identifies emerging opportunities, highlights best practices, and outlines key challenges in applying these models within the SE domain. Following the GQM approach [5], the goal of this study is to **analyze software development activities** for the purpose of **characterization**, with respect to **sentiment analysis**, from the point of view of **programmers**, and in the context of **NLP functionalities implemented LLMs**.

This paper is organized as follows. Section 2 presents other secondary studies on SA or using LLMs. Section 3 details the research design. Sections 4 and 5 present the results and discuss the main insights and contributions based on evidence from the selected studies. Finally, section 6 draws some conclusions.

2 Related Work

This section presents secondary studies investigating SA in the SE context. We also included secondary studies that discuss the use of LLMs to support SE tasks. We grouped them into the two subsections as follows.

Sentiment analysis in software engineering. Obaidi and colleagues conducted a systematic mapping study to investigate approaches for developing

SA tools for SE in studies published between 2012 and 2021. They concluded that implementing tools based on the BERT transformer model yielded the best results for SA in SE [26]. Wan and colleagues selected 33 primary studies published between 2013 and 2023 to investigate SA, considering the usage of popular datasets, mainstream models, and the latest tools, techniques, and metrics related to the current status of research on techniques specifically for the elicitation of requirements from reviews of applications [36]. Lin and colleagues performed a systematic review of opinion mining in software development, analyzing 185 articles across several key areas. The study categorized articles into well-defined software development activities related to opinion mining, examined available opinion mining approaches, evaluated their adoption and performance in other studies, and reviewed performance assessment and tool customization datasets. Moreover, the review addressed concerns and limitations that SE researchers should consider when applying or adapting these techniques. The findings provide valuable insights into the application of opinion mining tools in SE and suggest directions for future research [22].

Sánchez-Gordón and Colomo-Palacios presented a systematic review of the literature to assess the emotions of software developers. The research reviewed 7.172 articles from five databases, selecting 66 primary studies that met the inclusion criteria. The results indicated that emotions play a significant role in the performance and well-being of developers, but are still relatively little explored in SE. Measurement methods included text-based SA and physiological measures. The study identified gaps in the literature and areas for future research, contributing to a more integrated view of the role of emotions in SE [32]. Lenberg, Feldt, and Wallgren conducted a systematic review to explore the cognitive, behavioral, and social aspects of software development. The study analyzed 10.000 publications and selected 250 studies for review. The research revealed a growing interest in human aspects in SE, but identified gaps, such as a concentration on a few key concepts (personality, communication, and job satisfaction) and a limited focus on specific areas of engineering. Despite this, the research highlighted the importance of integrating psychology and social sciences into SE to improve processes and results [20].

Large language models in software engineering. Hou and colleagues conducted a systematic review to understand the application, effects, and possible limitations of LLMs on SE. They consider 395 papers as the set of primary studies. The studies were published from 2017 to 2025. They considered various dimensions, including the types of models being adopted, dataset domains, techniques used to optimize and evaluate the LLMs, and tasks in SE that have utilized LLMs. They indicated the use of SA to support software maintenance activities [14]. Fan and colleagues investigated how LLMs could support SE activities. They identified an increase in published studies since 2019, suggesting a possible trend in the focus of this research area. The authors highlighted the following open research questions: i) building and adjusting LLM adapted for SE that take advantage of dynamic information; ii) research on dynamic adaptive prompt engineering and model parameter tuning; iii) hybridization of LLMs

in SE frameworks; iv) understanding the programmer-LLM interaction; v) handling longer textual inputs; and, finally, vi) studies of SE subdomains such as requirements engineering, design, and refactoring [13]. These studies synthesize the empirical knowledge in the SA and LLM areas, in most cases, considering these concepts independently. They highlight the importance of the topics and guide future work in the respective areas. Our work deepens knowledge in the area by focusing specifically on the combination of topics, addressing SA with LLMs in the SE context.

3 Research Design

During the planning phase, we followed the systematic mapping guidelines by Petersen et al. [27], defining a comprehensive review protocol that included Research Questions (RQs), study search strategies, selection and exclusion criteria, and data extraction methods. This work is a *Systematic Mapping Study* (SMS) rather than a Systematic Literature Review (SLR). As defined by Petersen et al. [27], an SMS aims to provide a broad classification and overview of the research landscape (e.g., themes, publication trends, research gaps). In contrast, an SLR focuses on deep qualitative synthesis of study findings.

The main Research Question (RQ) of this study is *How do natural language processing functionalities implemented in LLMs support SA in software development activities?* We decomposed the main RQ into three detailed, specific research questions (SRQ) in the context of SA in SE, as follows: *SRQ1: Which strategies have been applied in LLMs that implement natural language processing functionalities to enhance the performance of SA in SE?*; *SRQ2: Is there a relationship between the natural language processing functionalities implemented in LLMs and the quality of the final software product?*;*SRQ3: How does the availability of datasets support the evaluation of natural language processing functionalities implemented in LLMs, promoting the advancement of SA?*.

Search Strategy. The study considers twenty-one primary studies. The main contributions regard the classification of i) LLM enhancements, ii) the relation of LLMs in SA with software product quality, and iii) dataset availability in the area. We defined the search terms based on the RQs, and the criteria were tested and applied during the study's selection phase. The major terms are "sentiment analysis" as one of the main subjects of the study, "large language model" as the addressed tool, and "software engineering" as the context of the study. Each candidate article was first screened by title and abstract to confirm that SA in SE was one of the primary focuses; if this was not clear, we performed a full-text screening, led by four researchers, to reach an agreement. During the process of defining search terms, we included alternative terms. For example, we consider "software development" as an alternative to "software engineering" or "opinion mining" as an alternative term to "sentiment analysis."

Before executing the whole search analysis, we selected our "gold set", i.e., a curated collection of articles that are expected to be included in the final review. This set served as a benchmark to assess the sensitivity and specificity of our

search strategy. The final search string was then refined to explicitly encompass the three main topics of interest—sentiments, LLM/Transformer models, and the context of SE—and was applied to the title, abstract, and keywords of each document. The search string is as follows: *("sentiment analysis" OR "opinion mining")* ***AND*** *("LLM" OR "large language model" OR "GPT" OR "transformer")* ***AND*** *("software engineering" OR "software development").*

We performed the searches in May 2025 in the following repositories to select the studies: ScienceDirect, IEEE Xplore, ACM Digital Library, and Scopus. The criteria for including or excluding studies, as well as the quality standards for selecting papers, are associated with their respective logical connectors. The exclusion criteria operate independently (*OR*), meaning that meeting any criterion is enough to exclude a paper. In contrast, the inclusion criteria require that all conditions be met (*AND*) for a study to be selected. Additionally, we adopted the quality criteria in the form of questions proposed by Dyba and Dingsoyr [11]. We adapted these questions, applying them to critically examine all other papers that met the criteria (inclusion and exclusion). A study was only selected if it consistently met all criteria with a YES response; otherwise, it was excluded. We adopted the following Exclusion criteria: E1 (Publications that did not answer at), E2 (Publications not written in English), E3 (Publications that are not full papers), E4 (Duplicated studies), E5 (Non-primary studies), E6 (Studies published earlier than 2019); Inclusion Criteria: I1 (Studies published in peer-reviewed conferences or journals), I2 (The studies must address SA on SE activities); Quality criteria: Were the objectives of the study specified? (Q1) Was the context of the study clearly defined? (Q2) Is the way the data is analyzed in line with the study objectives? (Q3).

Study Selection and Quality Assessment. Four researchers were involved in the selection phase, and two pairs of researchers divided the set of returned references. Then, they selected the primary studies, reading titles and abstracts, supported by a third and fourth researcher in case of doubts. The same approach was adopted in the in-depth reading and quality paper analysis. Another action adopted in the selection phase was carried out by the third researcher, reading the title and abstract for all accepted and rejected papers. For the extraction phase, two pairs of researchers independently identified excerpts of the papers. They checked the extraction of the excerpts when needed. Next, they met to reach a consensus in case of disagreement. For example, the quote *"Incorporating such sentiments could aid in the severity prediction of bug reports for mobile apps"* extracted from [1] was mapped to RQ2, as it is related to using LLM in SA to influence the quality of the final software product.

We outlined the steps needed to complete the selection process in Figure 1, focusing on the number of primary studies considered for each phase. The analyzed range of years spans from 2019 to 2025, which was chosen to cover the period following the publication of "Attention Is All You Need" in 2017 [35] and to include the most recent advances in LLM-based sentiment analysis up to now. We performed the searches in May 2025, resulting in 1068 studies retrieved by the search string. From this total, we obtained 211 publications in the ScienceDirect

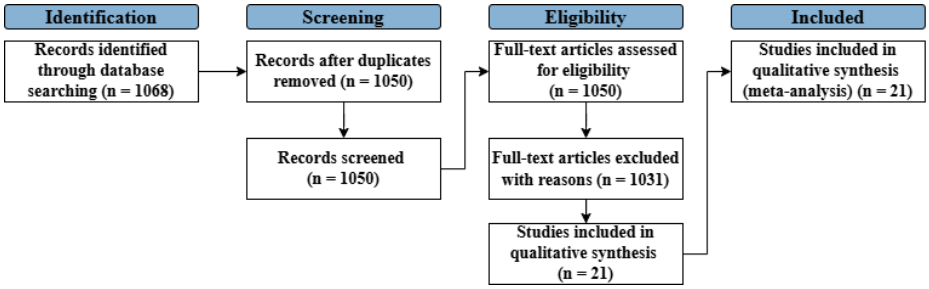


Fig. 1. Phases of the Systematic Mapping.

database, 127 in the ACM Digital Library, 521 in the IEEExplore database, and 209 in Scopus. After applying the inclusion, exclusion, and quality criteria in the 1068 studies, we selected 21 to answer the research questions according to the following distribution per database repository: ScienceDirect (3 studies - [18], [31], [1]), ACM DL (5 studies - [33], [17], [4], [15], [16]), IEEExplore (7 studies - [6], [2], [40], [9], [29], [37], [8]), Scopus (6 studies - [3], [30], [12], [28], [7], [39]).

The effectiveness of the search string after applying all criteria is presented as follows: 3 papers out of 211 from ScienceDirect (1.42%), 5 out of 127 from ACM Digital Library (3.93%), 7 out of 521 from IEEExplore (1.34%), and 6 of 209 from Scopus (2.86%). The steps of the selection process are available online[5].

4 Results

This section discusses the findings of each specific research question (SRQ). This discussion is a reference to answer the primary research question (RQ). It presents an overview of how NLP functionalities implemented in LLMs support SA, highlighting opportunities, best practices, and challenges in SE. Figure 2 shows an overview of evidence keys from the literature to answer the research questions. Each selected paper can answer more than one SRQ.

The **Specific Research Question 1 (SRQ1)** explores strategies to improve the performance of LLMs for conducting SA in SE. Thirteen of the twenty-one studies provide evidence addressing SRQ1, as shown in the *LLM* branch in Figure 2. From this branch, we derived eight sub-branches: *Bigger Open Source LLMs*, *Datasets Increasing*, *Domain Independent Training*, *Fine Tuning*, *Labels Preprocessing*, *LLM Models Comparison*, *LLM vs Other Approaches*, and *Prompt Engineering*. Additionally, the *Fine Tuning* branch led to three further sub-branches: *Bug Classification*, *Open Data*, and *Figurative and Politeness Expressions*, creating a three-level descriptive map, which will be detailed in the following.

The category *Bigger Open Source LLMs* covers work that uses large, open source LLMs (bLLMs) to do sentiment analysis in software engineering, taking

[5] https://doi.org/10.5281/zenodo.13631115

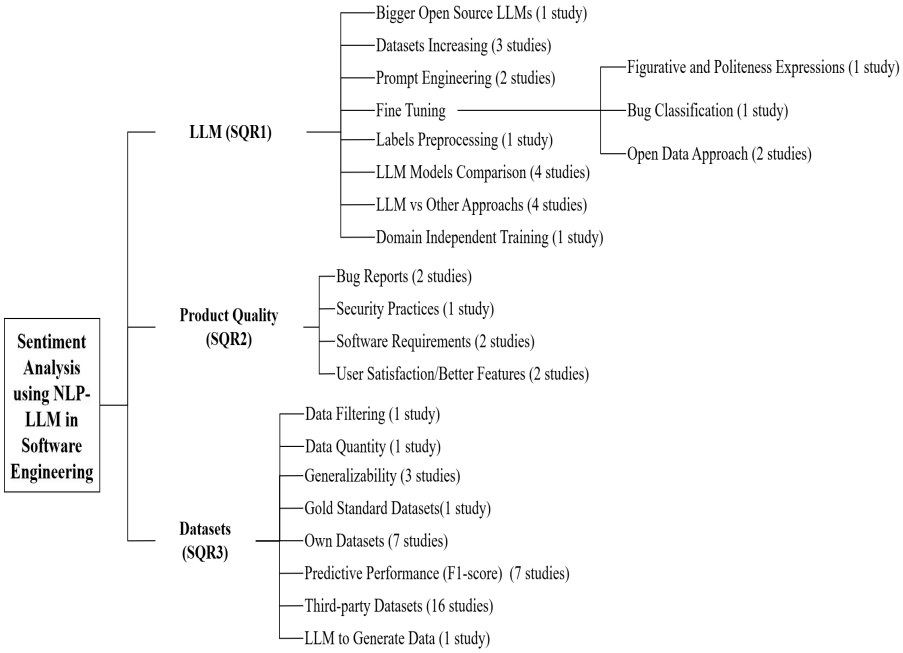


Fig. 2. Literature evidence to answer research questions.

advantage of their ability to learn from examples given right in the prompt and thus needing less labeled data. One study, **SP20** [39], explores this approach, where three bLLMs, Llama2-Chat, Vicuna, and WizardLM, were tested using zero-shot prompting on five software-engineering datasets. This study is also featured in *Prompt Engineering*, as the authors found that simply changing the prompt could alter macro-F1 by up to 22.8% and AUC by up to 33.4%.

The category *Datasets Increasing* contains studies indicating that larger datasets enhance model predictive performance. It also argues that merging multiple datasets improves SA outcomes. Two studies, **SP01** [6] and **SP03** [30], explore the idea that larger datasets lead to better performance, discussing the impact of dataset size in their lessons learned sections. One study highlights the effectiveness of merging datasets in improving the performance of LLMs and predicting emotions expressed in user reports and feedback about software applications **(SP11)** [9].

Domain Independent Training refers to the strategy described in **SP10** [28], where authors discuss the challenges and impact on the model performance when trained on a general dataset not specific to SE. In the experimental results section of the same study, the authors observed that inconsistent labels within the dataset could compromise model performance, highlighting the importance of data cleaning and preprocessing to ensure well-defined labels and avoid excessive flexibility, represented in the subbranch *Labels Preprocessing* in Figure 2.

The category *LLM vs Other Approaches* refers to the studies that compare LLMs with other SA approaches. Four studies address this topic **(SP01)** [6], **(SP04)** [33], **(SP19)** [16] and **(SP18)** [7]. The first study compares a transformer (BERT4SentiSE) with a recurrent neural network (RNN4SentiSE) using sentences from Stack Overflow. The study found substantial improvement in using BERT4SentiSE compared with RNN4SentiSE, obtaining an overall f-measure of 0.87, representing 18% of improvement **(SP01)** [6]. The second study compared the performance of various SA tools specific to SE with an ensemble-based SA tool combining five existing SA tools implemented with RoBERTa. The LLM version achieved an F1 score of 0.805. The authors discovered that combining polarity labels from individual tools with bag-of-words features through a simple supervised approach could enhance the performance (F1-scores) of stand-alone tools by 4% to over 100%. The resulting supervised hybrid tool was named Sentisead. While Sentisead demonstrated improved performance compared to most stand-alone tools, its advantage over the best-performing tool (Senti4SD) did not exceed 4% **(SP04)** [33]. The third study compared the zero-shot LLMs with existing state-of-the-art emotion classification models in SE to predict causes of *frustration* in development activities in a popular open-source project. The results indicated that the zero-shot LLMs effectively recognized the correct emotion cause with a BLEU-2 score of 0.598 **(SP19)** [16], having better performance than reported using traditional approaches (ESEM-E, EMTk, SEntiMoji [25]). Finally, the fourth study benchmarks five SA tools: two pre-trained transformer models and three traditional machine learning tools. The transformer models outperformed the traditional tools on one of the two datasets, with a margin of only a few percentage points.

The category *LLM Model Comparison* refers to three studies that compare LLMs' performance towards SA in SE. One of the studies considers the context of developers performing activities such as fixing bugs in new software versions based on users' feedback, such as ratings and reviews **(SP02)** [18]. The study proposes a method that calculates average ratings based on user review sentiment using BERT-based sentiment annotation and a hybrid RNN-LSTM model. The authors found more positive sentiments comparing their approach, actual ratings, with an average calculated rating of 3.60 stars, to the actual ratings with 3.08 stars. These results are based on reviews from over 250 apps on the Google Play Store. Another study provides evidence of figurative language's impact on SE **(SP16)** [15]. The authors created a dataset of 2000 sentences collected from GitHub issues and pull request comments, resulting in 1661 sentences with figurative expressions. They fine-tuned several state-of-the-art pre-trained LLMs with the annotated dataset. The third study evaluated the performance of fine-tuned models on three publicly available SE-specific datasets. Regarding the LLM Model Comparison, the authors compared the results of the emotion classification task on a multi-label emotion dataset **(SP07)** [17], using BERT, RoBERTa (RBTa), ALBERT (ALBT), CodeBERT (CodBT), and their fine-tuned with figurative language counterparts (BERT-FL, RBTa-FL, ALBT-FL, CodBT-FL). Finally, various BERT-based models were fine-tuned for SA in SE,

along with developing a novel fuzzy ensemble model that leverages the strengths of these individual models **(SP09)** [3]. The authors reported that the ensemble approach enhanced the accuracy and interpretability of sentiment classification within the SE context.

The category *Fine Tuning* refers to three studies that focus on data from open-source projects to classify bugs. We derived it into three new categories: *Open Data approach* with two studies, *Bug Classification* and *Figurative and Politeness Expressions* with one study each. The first study involves connecting various transformer models to analyze the sentiment of sentences from Stack Overflow posts. The authors found that, when the training data size is increased from 1,500 to 5,500, the sensitivity of the sentiment classifiers is significantly improved **(SP01)** [6]. The second study is also classified in the category *LLM Model Comparison Study* **(SP16)** [15] as it uses different LLM models to investigate figurative language from GitHub issues and pull request comments. In addition to comparing the LLM models, they fine-tuned the models. They found that fine-tuning LLMs with figurative language improves performance in SA prediction. The selected study, labeled as *Bug Classification* used BERT-SBR, a fine-tuned BERT classifier, to predict bug report severity based on SA of bug reports **(SP05)** [1]. The final study in the *Fine Tuning* category **(SP03)** [30] highlights, as one of its lessons, that curating datasets containing *Figurative and Politeness Expressions* and using them to fine-tune pre-trained models can significantly enhance their performance.

The category *Prompt Engineering* has, apart from **SP20** [39], one selected study that investigates the effectiveness of prompt engineering to analyze sentiments in databases related to SE **(SP19)** [16]. The author conducts a case study investigating the causes of frustration. They consider the last year of development of a popular open-source project, performing the extraction of communications between software developers. Using a prompting-based approach, they use three zero-shot LLMs: ChatGPT, GPT-4, and the open-source Flan-Alpaca.

We answered SRQ2 regarding the relationship between the NLP functionalities implemented in LLMs and the quality of the final software product based on evidence from 21 studies under the label *Product Quality* in Figure 2. The articles that addressed SRQ2 are **(SP10)** [28], **(SP09)** [3], **(SP08)** [12], **(SP05)** [1], **(SP01)** [6], **(SP16)** [15] and **(SP13)** [4]. We derived the previous label in four branches: *Bug Reports*, *Security Practices*, *Software Requirements*, and *User Satisfaction/Better Features*. The first refers to two studies identifying the need for bug reports after the SA activity with LLMs **(SP05)** [1] **(SP16)** [15]. We select a paper that uses BERT to support the severity prediction of bug reports based on the assumption that sentiment significantly influences bug severity prediction **(SP05)** [1]. The authors showed that the use of specific expressions in the bug report, such as *pathetic interface*, *dark*, and *problematic*, may indicate the urgency of a bug report. The other study in the bug report category is classified in the two previous paragraphs as it also addresses the topics *LLM Model Comparison study* and *Fine Tuning* **(SP16)** [15]. The study found that among the

state-of-the-art LLMs investigated, the best fine-tuned versions have an average improvement of 3.71% on a Bugzilla bug report prioritization dataset.

Edbert and colleagues focused on the *Security Practices* category, applying SA to Security-Related technical debt questions on Stack Overflow **(SP08)** [12]. This approach aims to improve software quality by identifying potential security issues. The third category, *Software Requirements*, encompasses studies that use LLM-based SA to enhance software requirements based on user feedback. One of the studies in this branch employs RE-BERT, a fine-tuned version of the BERT model, specifically designed to focus on software requirement tokens **(SP13)** [4]. Another study in this category leverages the results of the Fuzzy Ensemble model to prioritize software requirements **(SP09)** [3].

The final branch within the *Product Quality* category, *User Satisfaction/Better Features*, comprises two studies. In the first study, Plosca and colleagues concluded that recognizing the distinct emotional needs and preferences of users across different contexts enables researchers, teams, and applications to better meet user expectations, enhance satisfaction, and design more tailored features **(SP10)** [28]. The second study emphasizes that BERT's advancements in SA within the SE domain can support developers in downstream tasks, such as recognizing problematic API features by identifying negative sentiment in sentences from online API discussion forums **(SP01)** [6].

We identified 17 studies with evidence to answer SRQ3 regarding dataset support for evaluating NLP functionalities in LLMs, represented under the label *Datasets* in Figure 2. Eight branches derive from this node: *Data filtering*, *Data Quantity*, *Generalizability*, *Gold Standard Datasets*, *Own Datasets*, *Predictive Performance* (F1-score), *Third-party Datasets* and *LLM to Generate Data*.

The *Data Filtering* category relates to a study that evaluated the RoBERTa model **(SP15)** [31]. The authors examined how data filtering supported by LLM models could yield a dataset with more sentiment-rich posts compared to random selection. They concluded that data filtering was more efficient than manual selection. The *Data Quantity* category relates to the study conducted by Shafikuzzaman and colleagues, highlighting that pre-trained language models (PLMs) can deliver reliable results. However, the study emphasizes that the size and label distribution of the dataset significantly influence overall performance. Additionally, the size and diversity of the training data are crucial for PLMs to generalize effectively across various sentiment expressions **(SP03)** [30].

The third category, *Generalizability*, focuses on studies that use generalizability as an evaluation metric for LLM models. According to Anwar and colleagues, incorporating data from multiple websites enhanced the generalizability of classifiers trained on the dataset. The study concludes that exposing models to diverse linguistic styles, topics, and community dynamics improves their adaptability to real-world scenarios in software development **(SP12)** [2]. Another study argues that a fuzzy ensemble model achieved high accuracy across various SE contexts, leveraging diverse datasets such as CodeReview, JavaLib, Jira, and Stack Overflow **(SP09)** [3]. Plosca and colleagues also argue that incorporating data from diverse domains allows multi-domain datasets to integrate various models ef-

fectively. For instance, combining a specialized domain dataset with a general one results in a model that is better equipped to handle more challenging data, thereby enhancing performance across multiple domains **(SP10)** [28].

Cassee and colleagues conducted a study that prioritized the selection of gold-standard SA datasets. These datasets are recognized as the most extensive and comprehensive in the field of SA for SE. In practice, this refers to large, balanced datasets labeled using theoretical models of effect, with annotations provided by raters who achieve high inter-rater agreement **(SP18)** [7]. This study represents the category *Gold Standard Datasets*.

The category *Predictive Performance (F1-score)* refers to studies that used the F1-score metric to evaluate the efficiency of LLM models in SA. Fourteen primary studies used this metric. According to Sarker and colleagues, the fine-tuned RoBERTa model delivered the best performance, achieving an F1 score of 0.88 for detecting toxic class tokens **(SP17)** [29]. Zhang and colleagues evaluated five existing SA4SE tools and variants against four state-of-the-art pre-trained Transformers models (e.g., BERT, XLNet) across six SE datasets and argued that fine-tuned models surpass existing SA4SE tools, outperforming them by 6.5-35.6% in macro/micro-averaged F1 scores across all datasets **(SP14)** [40]. Ali and colleagues introduced BERT-SBR, a BERT-based model designed for forecasting the severity of bug reports in mobile app maintenance **(SP05)** [1]. Other studies belonging to this category are **(SP01)** [6], **(SP16)** [15], **(SP19)** [16], **(SP07)** [17], **(SP02)** [18], **(SP04)** [33], **(SP03)** [30], **(SP08)** [12], **(SP10)** [28], **(SP18)** [7] and **(SP20)** [39]. The results from all these studies demonstrated that BERT-SBR significantly outperforms other deep learning classifiers, achieving improvements of 58.14% in the F-measure metric. This discussion highlights the importance of available SE datasets to effective NLP solutions to SA in SE.

The other two categories, *Own Datasets* and *Third-party Datasets*, represent distinct aspects of dataset availability in the related studies. Seven studies used *Own Datasets*, developed by researchers, to address specific research questions or scenarios. In contrast, *Third-party Datasets* employed in 16 studies highlight their effort in benchmarking and generalizing SA models. Studies such as **(SP10)** [28] and **(SP12)** [2] adopted well-established datasets like Stack Overflow and Jira, which provide a standardized framework for validating models against comparable datasets used by other researchers. Lastly, to address the main research question, we provide a current panoramic view in Figure 2, showcasing evidence of how NLP functionalities implemented in LLM support SA in SE.

In the last category, *LLM to Generate Data*, there is a study which Cobos and colleagues [8] (**SP21**) introduce a GPT-Driven dataset creation approach, where they leverage ChatGPT-4o to synthesize a balanced and diverse set of community comments for model evaluation. Specifically, they prompted GPT-4o to generate 100 examples for each of the ten predefined ethical flags (F1–F10) and 100 neutral comments, resulting in 1,100 total entries. After manual validation—adjusting only 3.1% of the generated samples to ensure alignment with their intended flag definitions—the dataset was fed into their Mixtral-8x22B-based sentiment analyzer. This process enabled them to both assess classification performance:

95.55% accuracy on positive/negative/neutral labels, 84.02% on specific flags) and to highlight nuances—for instance, lower accuracy on sexualized language (F6) and harassment (F8). Therefore, demonstrating how GPT can efficiently create specialized evaluation datasets.

5 Discussion

This section provides insights based on the findings discussed earlier, focusing on the main research question (RQ) and its specific sub-questions. As LLMs support various NLP tasks, their application in research on SA in SE has gained significant attention. This is evident in Figure 2, which highlights many topics. The growing number of papers on this subject is illustrated in Table 1.

Table 1. Selected Studies by Year.

Year	**Studies**
2020	2 studies – [40], [6]
2021	3 studies – [37], [9], [4]
2022	1 study – [33]
2023	5 studies – [12], [29], [31], [18], [17]
2024	8 studies – [7], [28], [3], [30], [1], [2], [16], [15]
2025	2 studies – [39], [8]

Despite growing interest in the field, one may observe that many primary studies focus on the BERT model (see Section 4), even as newer models have emerged. Zhang et al. [39] classify BERT as a "smaller" LLM, with fewer than one billion parameters, in contrast to more recent, "bigger" LLMs. The authors argue that the field still lacks comprehensive studies on bigger LLMs. We hypothesize that this reflects a disparity between the pace of technological advancement and the time required to generate and disseminate scientific knowledge. Furthermore, we contend that this gap underscores the substantial opportunity for future research to replicate and extend existing studies, thereby contributing to a more comprehensive and robust body of knowledge. This observation points to significant potential for continued research and innovation in the field.

Regarding strategies applied in LLMs that implement NLP functionalities to enhance the performance of SA in SE **(SRQ1)**, evidence from selected studies shows that researchers used similar strategies in other domains. For example, *prompt engineering* has been adopted in other areas. The same is valid for comparing the efficiency of different LLMs and crowd workers for text-annotation tasks or for supporting deductive analysis coding. These strategies have been adopted in domains such as health and education research. One may argue that once other areas adopt different strategies to enhance the LLM's performance,

research in the SE areas also benefits from the approaches. We consider that there are few studies investigating the relationship between the use of LLMs supporting SA and software quality attributes **(SRQ2)**. We found evidence in studies addressing *bug reports*, *security practices*, *software requirements*, and *user satisfaction* (Figure 2). There is an opportunity to expand research in this perspective, focusing on many other software development activities that might be addressed, according to Fan et al. [13], e.g., code implementation, testing, maintenance, and deployment, among others. Regarding the use of datasets for evaluating NLP functionalities implemented in LLMs, promoting the advancement of SA **(SRQ3)**, studies indicate relevant challenges in the SA **(SP09)** [3] [21]. We highlight the number of studies addressing predictive performance, which is seven. This reinforces the relevance of characterizing datasets, especially regarding the class imbalance for SA in SE.

Key Findings to RQ1. We identified various strategies to enhance SA performance in SE using LLMs. Among the findings, dataset size and merging (*Datasets Increasing*) are essential to improve predictive performance, as reported by studies utilizing datasets from diverse SE tasks. *Domain Independent Training* highlighted improved model performance with general datasets, complemented by the importance of *Labels Preprocessing*. Comparisons between LLMs and alternative methods (*LLM vs Other Approaches*) and among LLMs themselves (*LLM Models Comparison*) showed better results for transformer-based models. Studies in the *Fine Tuning* category emphasized applications such as bug report severity classification and SA on open-source data, further supported by the effectiveness of *Prompt Engineering* for zero-shot sentiment classification tasks. These findings indicate the versatility and potential of LLM enhancements across diverse SE contexts.

Key Findings to RQ2. While the mapped studies report various SE aspects as being positively affected by LLM-based SA, the evidence for each is limited to at most one or two publications, so these observations must be considered preliminary. In *Bug Reports*, two studies suggest improved severity prediction and task prioritization, but further validation is needed. In *Security Practices*, a single study indicates that applying SA to technical debt discussions may help identify potential vulnerabilities, although this finding remains exploratory. In *Software Requirements*, two studies report enhanced extraction and prioritization of user needs. In *User Satisfaction and Feature Design*, two papers show that sentiment-driven insights can guide the development of tailored features and highlight problematic API functionalities, but the sample is too small for strong generalization. Given the low number of studies per aspect, these results serve as initial indicators rather than conclusive evidence and point to the need for more extensive empirical investigations.

Key Findings to RQ3. The selected studies confirmed the importance of dataset characteristics in advancing SA with LLMs. In *Data Filtering*, LLM techniques enhance the creation of sentiment-rich datasets, surpassing manual methods in efficiency. Regarding *Data Quantity*, the size and label distribution of training data seem to significantly influence LLM performance. The *Gen-*

eralizability of models improves through exposure to diverse linguistic styles and datasets, enabling better adaptation to real-world software development scenarios. *Gold Standard Datasets* ensure rigorous benchmarking by providing well-annotated and balanced data. The *Predictive Performance (F1-score)* metric is widely employed, with fine-tuned models consistently surpassing traditional tools, particularly when robust datasets are available. Finally, while *Own Datasets* support specific research objectives, *Third-party Datasets* enable and promote standardization and benchmarking.

6 Conclusions

We conducted a systematic mapping study to investigate how NLP functionalities implemented in LLMs support SA in software development activities. We analyzed evidence from 21 primary studies, offering a comprehensive overview of the achievements, challenges, and opportunities in leveraging NLP, implemented through LLMs, for SA in SE. The results show strategies to use LLMs in SA in SE, such as merging datasets, comparing LLM with other approaches, and approaches for LLM fine-tuning. The results highlight the research community's efforts to explore how NLP capabilities within LLMs can enhance the effectiveness of SA in SE. Finally, we found studies considering dataset availability supporting data filtering, generalizability, and predictive performance. The findings indicate possibilities for future research. We highlight conducting exploratory studies with LLM in experimental settings for SE SA purposes, as all primary studies indicate better results than traditional SA approaches. Another topic is the development of studies focusing on comparing LLM models and varying their architectures. Finally, characterizing and evaluating guidelines for the effectiveness of the public repository of datasets for the SE area and respective LLMS to transform original SE data into tokens, embeddings, and their manipulation in vectorial databases and toolkits to implement generative AI applications. This systematic mapping showed that few studies focus on this relevant direction.

References

1. Ali, A., Xia, Y., Umer, Q., Osman, M.: Bert based severity prediction of bug reports for the maintenance of mobile applications **(SP05)**. JSS **208**, 111898 (2024)
2. Anwar, Z., Afzal, H., Al-Shehari, T., Al-Razgan, M., Alfakih, T., Nawaz, R.: Mining the Opinions of Software Developers for Improved Project Insights: Harnessing the Power of Transfer Learning **(SP12)**. IEEE Access (2024)
3. Anwar, Z., Afzal, H., Altaf, N., Kadry, S., Kim, J.: Fuzzy ensemble of fined tuned BERT models for domain-specific sentiment analysis of software engineering dataset **(SP09)**. Plos one **19**(5), e0300279 (2024)
4. de Araújo, A.F., Marcacini, R.M.: RE-BERT: automatic extraction of software requirements from app reviews using bert language model **(SP13)**. In: Proceedings of the 36th annual ACM symposium on applied computing. pp. 1321–1327 (2021)
5. Basili, V.R., Rombach, H.D.: The TAME project: Towards improvement-oriented software environments. IEEE Trans. on Software Engineering pp. 758–773 (1988)

6. Biswas, E., Karabulut, M.E., Pollock, L., Vijay-Shanker, K.: Achieving reliable sentiment analysis in the software engineering domain using bert **(SP01)**. In: 2020 IEEE ICSME. pp. 162–173. IEEE (2020)
7. Cassee, N., Agaronian, A., Constantinou, E., Novielli, N., Serebrenik, A.: Transformers and meta-tokenization in sentiment analysis for software engineering **(SP18)**. Empirical Software Engineering **29**(4) (2024)
8. Cobos, S., Izquierdo, J.L.C.: A Bot-based Approach to Manage Codes of Conduct in Open-Source Projects **(SP21)** (2025)
9. De Arriba, A., Oriol, M., Franch, X.: Merging datasets for emotion analysis **(SP11)**. In: Proc. of the 36th ASE Workshops. pp. 227–231. IEEE/ACM (2021)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proc. of the Conf. of the North American Chapter of the Assoc. for Computational Linguistics: Human Language Technologies. vol. 1, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019)
11. Dybå, T., Dingsøyr, T.: Empirical studies of agile software development: A systematic review. Information and software technology **50**(9-10), 833–859 (2008)
12. Edbert, J.A., Oishwee, S.J., Karmakar, S., Codabux, Z., Verdecchia, R.: Exploring Technical Debt in Security Questions on Stack Overflow **(SP08)**. In: International Symposium on Empirical Software Engineering and Measurement (2023)
13. Fan, A., Gokkaya, B., Harman, M., Lyubarskiy, M., Sengupta, S., Yoo, S., Zhang, J.M.: Large language models for software engineering: Survey and open problems. In: 2023 IEEE/ACM Future of Software Engineering (ICSE-FoSE). pp. 31–53. IEEE Computer Society, Los Alamitos, CA, USA (2023)
14. Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., Luo, X., Lo, D., Grundy, J., Wang, H.: Large Language Models for Software Engineering: A Systematic Literature Review. ACM TOSEM **33**(8) (Dec 2024)
15. Imran, M.M., Chatterjee, P., Damevski, K.: Shedding Light on Software Engineering-specific Metaphors and Idioms **(SP16)**. In: Proc. of the 46th ICSE. pp. 1–13. IEEE/ACM (2024)
16. Imran, M.M., Chatterjee, P., Damevski, K.: Uncovering the causes of emotions in software developer communication using zero-shot llms **(SP19)**. In: Proceedings of the IEEE/ACM 46th ICSE. pp. 1–13 (2024)
17. Imran, M.M., Jain, Y., Chatterjee, P., Damevski, K.: Data augmentation for improving emotion recognition in software engineering communication **(SP07)**. In: Proceedings of the 37th IEEE/ACM ASE. pp. 1–13 (2023)
18. Islam, M.J., Datta, R., Iqbal, A.: Actual rating calculation of the zoom cloud meetings app using user reviews on google play store with sentiment annotation of BERT and hybridization of RNN and LSTM **(SP02)**. Expert Systems with Applications **223**, 119919 (2023)
19. Islam, M.R., Rabbi, M.F., Jo, Y., Champa, A., Young, E., Wilson, C., Scott, G., Zibran, M.: A Four-Dimension Gold Standard Dataset for Opinion Mining in Software Engineering. In: 2024 IEEE/ACM 21st MSR. pp. 487–491. IEEE (2024)
20. Lenberg, Feldt, Wallgren: Behavioral software engineering: A definition and systematic literature review. JSS **107**, 15–37 (2015)
21. Lin, B., Zampetti, F., Bavota, G., Di Penta, M., Lanza, M., Oliveto, R.: Sentiment analysis for software engineering: How far can we go? In: Proceedings of the 40th ICSE. pp. 94–104 (2018)
22. Lin, C., Serebrenik, B., Novielli, L.: Opinion Mining for Software Development: A Systematic Literature Review. ACM Transactions on Software Engineering and Methodology **31**(3), 1–41 (2022)

23. Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., Tang, J.: GPT understands, too. AI Open (2023)
24. Von der Mosel, J., Trautsch, A., Herbold, S.: On the validity of pre-trained transformers for natural language processing in the software engineering domain. IEEE Transactions on Software Engineering **49**(4), 1487–1507 (2022)
25. Murgia, A., Ortu, M., Tourani, P., Adams, B., Demeyer, S.: An exploratory qualitative and quantitative analysis of emotions in issue report comments of open source systems. Empirical Software Engineering **23**, 521–564 (2018)
26. Obaidi, M., Nagel, L., Specht, A., Klünder, J.: Sentiment analysis tools in software engineering: A systematic mapping study. Information and Software Technology **151**, 107018 (2022)
27. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: Proc. of the 12th Int. Conf. on Evaluation and Assessment in Software Engineering (EASE). BCS Learning & Development (2008)
28. Plosca, T.R., Curiac, C.D., Curiac, D.I.: Investigating Semantic Differences in User-Generated Content by Cross-Domain Sentiment Analysis Means **(SP10)**. Applied Sciences (Switzerland) **14**(6) (2024)
29. Sarker, J., Sultana, S., Wilson, S.R., Bosu, A.: ToxiSpanSE: An explainable toxicity detection in code review comments **(SP17)**. In: 2023 ACM/IEEE International Symposium on ESEM. pp. 1–12. IEEE (2023)
30. Shafikuzzaman, M., Islam, M.R., Rolli, A.C., Akhter, S., Seliya, N.: An Empirical Evaluation of the Zero-Shot, Few-Shot, and Traditional Fine-Tuning Based Pretrained Language Models for Sentiment Analysis in Software Engineering **(SP03)**. IEEE Access **12**, 109714 – 109734 (2024)
31. Swillus, M., Zaidman, A.: Sentiment overflow in the testing stack: Analyzing software testing posts on Stack Overflow **(SP15)**. JSS **205**, 111804 (2023)
32. Sánchez-Gordón, Colomo-Palacios: Taking the emotional pulse of software engineering — A systematic literature review of empirical studies. Information and Software Technology **115**, 23–43 (2019)
33. Uddin, G., Guéhénuc, Y.G., Khomh, F., Roy, C.K.: An empirical study of the effectiveness of an ensemble of stand-alone sentiment detection tools for software engineering datasets **(SP04)**. ACM TOSEM **31**(3), 1–38 (2022)
34. Umer, Q., Liu, H., Illahi, I.: CNN-based automatic prioritization of bug reports. IEEE Transactions on Reliability **69**(4), 1341–1354 (2019)
35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proc. of the 31st Int. Conf. on Neural Information Processing Systems. p. 6000–6010. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
36. Wan, H., An, Z., Wang, B., Xiong, T.: Sentiment Analysis for Requirements Elicitation from App Reviews: A Systematic Mapping Study. In: Proc. of the 30th Asia-Pacific Software Engineering Conference (APSEC). pp. 201–210. IEEE (2023)
37. Wu, J., Ye, C., Zhou, H.: BERT for sentiment classification in software engineering **(SP06)**. In: 2021 ICSS. pp. 115–121. IEEE (2021)
38. Young, J.C., Arthur, R., Williams, H.T.: CIDER: Context sensitive sentiment analysis for short-form text. PLoS ONE 19(4) (2024)
39. Zhang, T., Irsan, I.C., Thung, F., Lo, D.: Revisiting Sentiment Analysis for Software Engineering in the Era of Large Language Models **(SP20)**. ACM TOSEM **34**(3) (Feb 2025)
40. Zhang, T., Xu, B., Thung, F., Haryono, S.A., Lo, D., Jiang, L.: Sentiment analysis for software engineering: How far can pre-trained transformer models go? **(SP14)**. In: 2020 IEEE ICSME. pp. 70–80. IEEE (2020)