applied sciences

MDPI

*Article*

# Investigating Semantic Differences in User-Generated Content by Cross-Domain Sentiment Analysis Means

**Traian-Radu Ploscă [1,*], Christian-Daniel Curiac [2] and Daniel-Ioan Curiac [1,*]**

[1] Department of Automation and Applied Informatics, Politehnica University of Timisoara, V. Parvan 2, 300223 Timisoara, Romania

[2] Department of Computer and Information Technology, Politehnica University of Timisoara, V. Parvan 2, 300223 Timisoara, Romania; christian.curiac@cs.upt.ro

* Correspondence: traian.plosca@aut.upt.ro (T.-R.P.); daniel.curiac@aut.upt.ro (D.-I.C.)

**Abstract:** Sentiment analysis of domain-specific short messages (DSSMs) raises challenges due to their peculiar nature, which can often include field-specific terminology, jargon, and abbreviations. In this paper, we investigate the distinctive characteristics of user-generated content across multiple domains, with DSSMs serving as the central point. With cross-domain models on the rise, we examine the capability of the models to accurately interpret hidden meanings embedded in domain-specific terminology. For our investigation, we utilize three different community platform datasets: a Jira dataset for DSSMs as it contains particular vocabulary related to software engineering, a Twitter dataset for domain-independent short messages (DISMs) because it holds everyday speech type of language, and a Reddit dataset as an intermediary case. Through machine learning techniques, we thus explore whether software engineering short messages exhibit notable differences compared to regular messages. For this, we utilized the cross-domain knowledge transfer approach and RoBERTa sentiment analysis technique to prove the existence of efficient models in addressing DSSMs challenges across multiple domains. Our study reveals that DSSMs are semantically different from DISMs due to F1 score differences generated by the models.

**Keywords:** sentiment analysis; short messages; cross-domain modeling; RoBERTa transformer

check for updates

**Citation:** Ploscă, T.-R.; Curiac, C.-D.; Curiac, D.-I. Investigating Semantic Differences in User-Generated Content by Cross-Domain Sentiment Analysis Means. *Appl. Sci.* **2024**, *14*, 2421. https://doi.org/10.3390/app14062421

Academic Editor: Luis Javier Garcia Villalba

Received: 7 February 2024
Revised: 11 March 2024
Accepted: 12 March 2024
Published: 13 March 2024

## 1. Introduction

Sentiment analysis is a branch of natural language processing (NLP) that is used for classifying opinions. It is widely used in market research, feedback analysis, and election and social media monitoring [1]. There are two lines of thinking when conducting sentiment analysis: the machine learning-based approach and the lexical dictionary methodology [2]. Through machine learning, the process of training is flexible and can be adjusted according to the task's requirements, but it requires labeled data. The lexicon approach uses a predefined list of words associated with sentiments; however, this procedure is not very adaptable and requires a dictionary based on the task [3].

Studies such as [4–7] focus on how various emotions affect human performance. The researchers stimulate artificial emotions in the participants to comprehend their effects. Contrary to their work, our research focuses on investigating the emotional aspects of communication in domain-specific short messages (DSSMs) and compares them to domain-independent short messages (DISMs). To achieve this objective, we closely examine the messages exchanged between professionals in the software engineering (SE) domain, with the aim of gaining a deeper understanding of the emotions that underlie their interactions.

We have formulated the following research questions that we will investigate:

**RQ1:** *Are DSSMs semantically distant from DISMs?*

**RQ2:** *Are sentiments expressed differently in each domain?*

**RQ3:** *Could cross-domain models accurately capture the meaning of DSSM terminology?*

We aim to compare DSSMs and DISMs by fine-tuning the Roberta transformer model, which will allow us to evaluate sentiments accurately. To make the approach more reliable, we plan to incorporate data from platforms that are known for their communication styles and community dynamics. Furthermore, we will compare the performance of the various models across single and multi-domains, aiming to determine which models perform best.

The rest of the paper is organized as follows. In Section 2, we will present an overview of the related work in this field. In Section 3, we justify the datasets that were used and also introduce the proposed methodology. Section 4 briefly describes various methods in the sentiment analysis approach with a focus on the RoBERTa technique. Section 5 presents discussions related to the obtained results. Finally, we conclude in Section 6.

## 2. Related Work

Different studies suggest that sentiment analysis tools may not produce reliable results when applied to SE texts, primarily due to the lack of annotated datasets for training such a tool [8]. SentiStrength-SE, a dictionary-based tool built on top of SentiStrength, was developed to enhance sentiment analysis in textual contents of SE [9]. The tool's overall performance was evaluated using the F1 score, which measures the balance between precision and recall. The results showed that the tool outperformed SentiStrength, achieving an F1 score of 77.48% compared to SentiStrength's 62.02%.

The experiments conducted in [10] show that the lexicon approach is effective on domain-specific data. These lexicon-based tools utilize three emotional polarities: positivity, negativity, and neutrality. Our goal is to identify and differentiate between DSSMs and DISMs. We believe that a more nuanced approach would be beneficial, including emotions such as joy, love, neutrality, surprise, anger, and fear. By doing so, we hope to gain a deeper understanding of human interactions. In [11], the authors found that sentiments and emotions including joy and love decrease the fixing time of a task, while negative sentiments and emotions, in particular sadness and anger, increase the addressing time. There is a correlation between the level of politeness in a comment of an issue and the duration it takes to resolve that issue. Specifically, tickets with more polite comments tend to be resolved in a shorter amount of time.

One potential solution to improve sentiment analysis in DSSMs is to use cross-domain knowledge transfer by leveraging a dataset from a similar domain to the target domain [12]. By using transformer-based models such as BERT, RoBERTa, or XLNet; we aim to fine-tune models that are capable of interpreting hidden meanings from the domain-specific vocabulary. This type of approach is in line with other works that have adapted the pre-trained BERT model to investigate sentiment analysis inside specific domains [13,14].

In their work, the authors of [15] utilized the pretrained BERT model and customized it for the SE domain. They suggest that the performance of SE models improves because the base BERT vocabulary tokens are substituted with tokens from the field they are trained on. Specifically, half of the base words were replaced with domain-specific vocabulary. Other works compared lexicon-based approaches, such as Stanford CoreNLP, SentiStrength, SentiStrength-SE, SentiCR, and Senti4SD, with transformer models like BERT, RoBERTa, XLNet, and ALBERT, which were trained on six different datasets in SE domain [16]. The findings indicate that transformer-based models show superior performance when compared to all of the lexicon-based tools tested, as measured by the F1 score.

Another interesting approach is to examine the use of emoticons in DSSM. In this context, a deep neural network that uses cross-domain techniques to analyze emoticons used in SE texts, named SEntiMoji, was proposed [17]. It is built on top of DeepMoji, a pretrained model that detects emotions in noisy labels, for instance, the emoticons found in Twitter comments [18].

A comprehensive analysis of DSSMs and DISMs remains absent in the existing literature, with limited attention given to the field of SE. However, exploring DSSMs across various areas could uncover valuable insights into semantics, communication patterns, and practical applications beyond SE contexts.

## 3. Methodology

To achieve our objectives, we have defined the following methodology:

### *3.1. Stage 1: Selection of a DSSM Scope*

We chose SE for studying its DSSMs because it involves a specialized set of jargon, terminology, abbreviations, and communication styles. Short messages within SE may include code comments, commit messages, ticket reports, and other forms of communication exchange among professionals in the field.

### *3.2. Stage 2: Choosing the Right Datasets*

We utilized the emotions dataset provided in [19] for studying DSSMs. This dataset consists of labeled comments extracted from the Jira platform. Jira is an issue-tracking system (ITS) that is used for software task tracking, ticketing systems, and project management [20]. It is commonly used in Agile development with methodologies such as Scrum or Kanban board.

To investigate DISMs, we expanded our scope beyond Jira and included the Twitter dataset detailed in [21]. Twitter, currently named “X”, is a microblogging platform that allows users to communicate by short messages of up to 280 characters [22]. The platform facilitates one-way communication, where the users post content while their followers can engage with the content through comments [23].

We opted to use a Reddit dataset [24] as a middle-ground option to explore both DSSMs and DISMs. Reddit is an online discussion forum where users can create communities, called subreddits, to discuss specific topics [25]. Each subreddit has its unique culture and moderation policies, making them different from one another.

Reddit and Twitter have different features that make them unique. On Reddit, users can follow specific subjects through subreddits, while on Twitter, users primarily follow influencers [26]. On the other hand, Jira is a platform that is used for managing and creating software products, and it hosts developer communities [27]. The level of communication on these platforms also differs. Jira involves technical terminology, in which sprints and Agile methodologies contribute to a specialized language that aids precise communication within development teams. Twitter, with its character limit and fast-paced nature, demands quick and impactful messages, which often leads to a preference for direct, casual, and informal communication styles. Meanwhile, Reddit fosters in-depth discussions through longer-form posts and comments that may vary depending on the specific subreddit community [28]. Some discussions can point towards casual or humorous content, while others maintain a more formal and professional atmosphere.

The Venn diagram shown in Figure 1 displays the semantic distribution of short messages across these platforms. It highlights four intersection regions marked with Roman numerals:

I The semantic intersection of Twitter and Jira datasets can create a blend of professional and casual communication which can uncover trends and correlations among technical events. It is important to note that a negative sentiment on Twitter may express sadness or anger, whereas, on Jira, it can add valuable feedback and criticism.

II Jira and Reddit dataset messages can span various domains, resulting in a mix of domain-specific vocabulary. It may also induce multi-domain topics, with more complex terminology. For instance, the software development jargon could be mixed with marketing or sales-related vocabulary.

III While Reddit and Twitter datasets tend to cater to different user dynamics and discussion formats, some areas overlap in vocabulary. The language might be informal and focused on news, entertainment, humor, travel, and food, but it may also reveal more in-depth terminology for different topics.

IV The semantic intersection of the three datasets may showcase how the technical jargon of Jira joins with social discussions of Twitter and in-depth debates of Reddit.

Examining the intersection zone might reveal a better understanding of SE sentiments in terms of transparency, optimism, seriousness, sarcasm, criticism, etc.

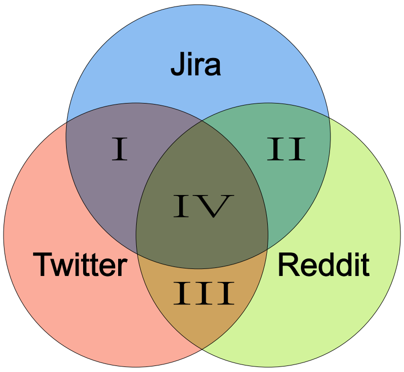

**Figure 1.** Semantic distribution of short messages.

### *3.3. Stage 3: Picking the Sentiment Analysis Algorithm That Best Fits Our Criteria*

Sentiment analysis is a complex task, especially when it comes to DSSMs. We address the challenge of selecting a tool that could accurately analyze sentiments in different fields by means of machine learning. By comparing different sentiment analysis models, as described in Section 4, we selected the RoBERTa transformer as the investigation tool.

### *3.4. Stage 4: Investigating Semantic Differences by Cross-Domain Means*

Priya et al. [28] found that single-domain models perform better on their corresponding domain, but they perform poorly on multi-domains. On the other hand, when it comes to utilizing multi-domain datasets, multi-domain models show superior performance as compared to single-domain models. We decided to compare the performances of multiple models on single and multiple domains.

To extract data from the datasets, we utilized Python's Pandas library for handling CSV format. Single-domain data are defined as J (Jira), T (Twitter), and R (Reddit). To make use of the multi-domain datasets, we combine single-domain data subsets and generate the following sets: J + T (Jira + Twitter), T + R (Twitter + Reddit), J + R (Jira + Reddit), and J + T + R (Jira + Twitter + Reddit).

Once we completed the extraction process, we had to refine the raw data. Both Jira and Twitter datasets have seven basic labels of emotions, which are joy, love, surprise, neutral, sadness, fear, and anger. On the other hand, the Reddit dataset has 28 labeled emotions. To simplify it, we grouped the 28 different labels into an Ekman-style group [29], which consists of six categories as mentioned earlier. This allowed us to better capture the full spectrum of emotional expressions in the comments, including those that did not fit into the initial categories. As fear and surprise data emotions are scarce, we have removed them entirely. Having a severely imbalanced data collection, with some datasets being more dominant than others, biased model predictions and inaccurate results may arise as a consequence. To address the issue, we decided to sample all datasets uniformly. The number of entries for each label was selected based on the Jira dataset's maximum label size. The collected data are described in Figures 2–4 as follows:

- Figure 2 represents the single-domain data distribution for cases J, T, and R; consisting of 5 labels, each with their corresponding size.
- Figure 3 shows the two-domain data distribution for J + T, T + R, and J + R, with 5 labels equally divided between the two domains.
- Figure 4 illustrates a three-domain data distribution for J + T + R, including 5 labels, where every third of the labels corresponds to each domain.

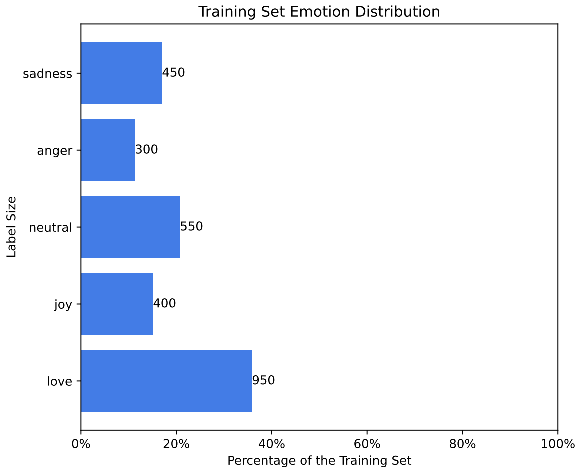

**Figure 2.** Single-domain data distribution.

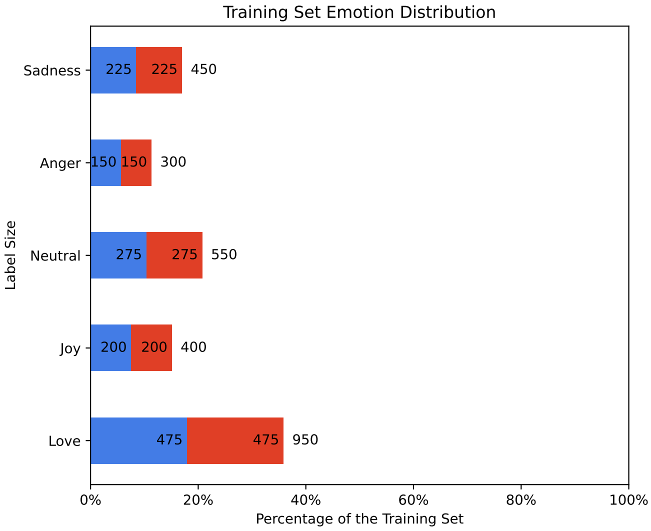

**Figure 3.** Two-domain data distribution.

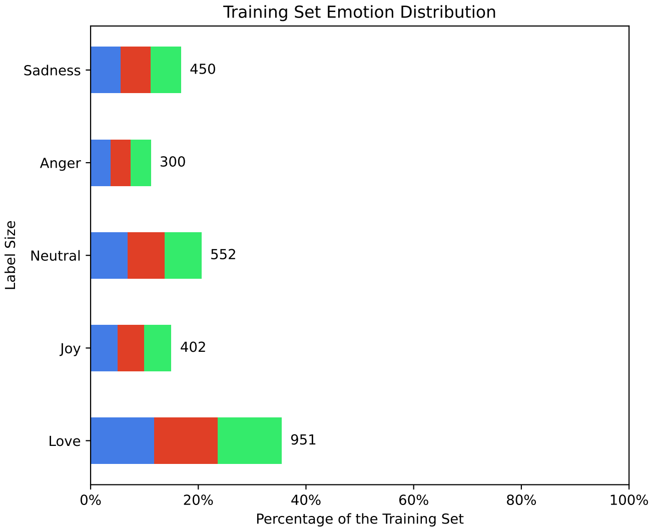

**Figure 4.** Three-domain data distribution.

In the following step, the raw text is transformed into individual tokens using a pre-existing tokenization method, namely RobertaTokenizer. Tokenization is a process of converting text into individual words (tokens), which can then be analyzed and processed further. To specify the maximum number of tokens for encoding, we have first checked the number of tokens per comment. Given that most comments contain fewer than 200 tokens, we set the limit at 128 tokens.

For the training process, we initialized the "roberta-base" model and specified its layers using PyTorch Lightning. We accomplished this by creating the LightningModule class. To enable data forwarding, we redefined the class methods training_step, validation_step, and test_step for our case. The final steps of the training process were concluded by configuring the Adam optimizer and setting up training configurations such as batch size, number of epochs, and learning rate.

Figure 5 illustrates that the transformer models were trained on 70% of the dataset, validated on 20%, and tested on 10% of the data for both self-domain and cross-domain scenarios.

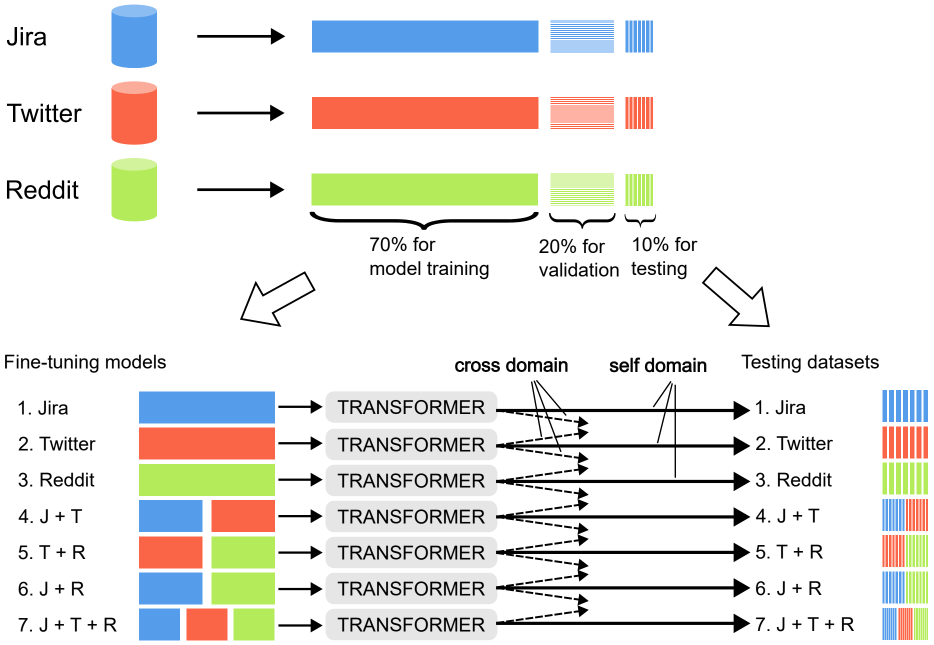

**Figure 5.** Models and data distribution.

## 4. Sentiment Analysis Methods

We fitted various models to the Jira dataset to determine the most appropriate model to be employed. Initially, we trained a basic transformer model, BERT, to obtain results. Following the training, we obtained an average F1 score whose success surpassed SentiStrength-SE's F1 score [9].

With the achieved results, we investigated deeper and searched for a better transformer model. In [30], the authors discovered that, in terms of F1 score, RoBERTa performed the best, closely followed by XLNet and DistilBERT. Considering those results, we compared the performance of four different models: BERT, RoBERTa, XLNet, and ALBERT. We aimed to gain a comprehensive understanding of the strengths and weaknesses of each of these models, to make a decision on the most suitable one for our needs.

RoBERTa shares the architecture with BERT but differs in dynamic masking, next sentence prediction, minibatches, and larger byte pair encoding. Furthermore, RoBERTa was also trained on a larger dataset and for a longer period [31].

Unlike the previously presented architecture, XLNet uses autoregressive language modeling (AR) and autoencoders (AE). To capture bidirectional contexts, AR introduces permutation language modeling instead of dynamic masking. In this architecture, AE is used to reconstruct the original data from a potentially corrupted input. Different from the previously presented architecture, XLNet does not make use of segment embeddings [32].

ALBERT, a lightweight variant of BERT, achieves model efficiency through cross-layer parameter sharing and factorized embedding parameterization [33]. Those techniques can reduce the size of the model and enable faster training without decreasing its performance.

While BERT, XLNet, ALBERT, and RoBERTa are all powerful transformer-based models, we evaluated their performance on the Jira dataset, with the results presented in Table 1 showing that RoBERTa performed slightly better. To further validate this conclusion, we also compared our results from Table 1 with the ones presented in the articles [34,35], concluding that RoBERTa outperforms the other models in terms of precision, recall, and F1-scores. Due to its performance, we decided to use the "roberta-base" model for our experiments.

**Table 1.** BERT, RoBERTa, XLNet, and ALBERT results.

| Model | Metrics |
|---|---|
| BERT | Accuracy: 0.8704705057<br>F1 score: 0.8724603052<br>Loss: 1.291070104 |
| **RoBERTa** | **Accuracy: 0.9221942921**<br>**F1 score: 0.9234281278**<br>**Loss: 0.3483341634** |
| XLNet | Accuracy: 0.9169938469<br>F1 score: 0.9174526834<br>Loss: 0.490663442 |
| ALBERT | Accuracy: 0.9171428680<br>F1 score: 0.9157220125<br>Loss: 1.1986452341 |

*RoBERTa Transformer*

The robustly optimized BERT approach (RoBERTa) is a transformer-based model built upon the BERT model. Based on the architecture illustrated in Figure 6, the prediction process starts by feeding the input text into the tokenizer [31]. Each sequence starts with the [CLS] token, which represents the special classification token. The input embedding, denoted by E in the figure, refers to the initial transformation of input tokens into numerical representations known as vector embeddings. These embeddings capture the semantic meaning of each token.

The final hidden vector of the model begins with the final special [CLS] token, marked with C in the figure. This token outputs the prediction after the normalization by the softmax layer. The vector's other elements are marked with T corresponding to the *i*th input token, representing the output of the last transformer layer. This layer encapsulates the token's contextual representation from all surrounding tokens in the sequence [36].

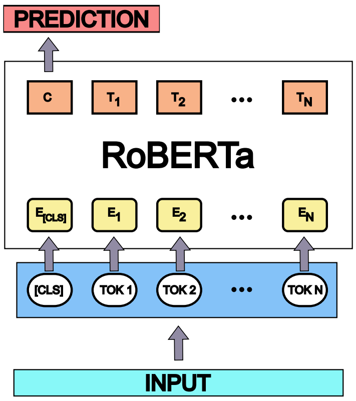

**Figure 6.** RoBERTa architecture.

## 5. Experimental Results

In this section, we present the experimental setup, describe the used datasets, and discuss the obtained results.

### *5.1. Experimental Environment*

We fine-tuned the "roberta-base" models on the seven datasets represented in Figure 5 as part of our experiment. Each model was trained for fifteen epochs using early stopping techniques with the patience of five epochs. The training process was carried out using NVIDIA GeForce GTX 1650 Ti (Santa Clara, CA, USA). Table 2 presents the tools and technologies that were utilized.

**Table 2.** Tools and technologies.

| Environment | Description |
|---|---|
| Model | roberta-base |
| Transformer | Hugging Face |
| Programming language & tools | PyTorch, PyTorch Lightning, Anaconda Distribution, Python 3.10 |
| GPU | NVIDIA GeForce GTX 1650 Ti |

### *5.2. Datasets*

The Jira comments dataset that we utilized [19] contains six emotions as follows: joy, love, sadness, anger, fear, and surprise. In our study, we used Group 2 and Group 3 as described in [19]. To fit our model, we excluded fear and surprise from the dataset as they have insufficient data to be used in fine-tuning.

We used the Twitter dataset as detailed in [21]. The dataset includes six basic emotions: joy, love, sadness, anger, fear, and surprise. However, to match the Jira dataset, we excluded the emotions of fear and surprise.

The authors of [24] created a large dataset of comments taken from the Reddit platform. This dataset was manually annotated, and each comment was labeled according to one of 28 distinct classes. We used Ekman grouping to classify the labels into basic categories to align the dataset with Jira and Twitter datasets.

Table 3 shows the total number of labeled comments for each dataset and their corresponding labels.

**Table 3.** Dataset specifications.

| Dataset | Labeled Comments | Labels |
|---|---|---|
| Jira | 5.6 K | joy, love, surprise, neutral, sadness, anger, fear |
| Twitter | 16 K | joy, love, surprise, neutral, sadness, anger, fear |
| Reddit | 58 K | admiration, amusement, anger, annoyance, approval, caring, confusion, curiosity, desire, disappointment, disapproval, disgust, embarrassment, excitement, fear, gratitude, grief, joy, love, nervousness, optimism, pride, realization, relief, remorse, sadness, surprise, neutral |

### *5.3. Discussions*

Table 4 displays the outcomes of testing the RoBERTa models, trained on five classes of sentiments, against each dataset. Table 5 is similar, but the models in this case were trained on four sentiment classes. The performance is evaluated using the F1 score. Every row represents the model's evaluation on all datasets. Each column displays the dataset evaluated on all models. The "Average" rows and columns symbolize the average F1 score on that particular row or column. A higher F1 score indicates a better-performing model. We consider a low-performing model if its F1 score is below the score of SentiStrength-SE (0.774), underlining the value in the tables. The highest performance achieved on each column is represented in bold font.

**Table 4.** RoBERTa models with 5-class inference.

| **Model** | **Evaluation on Test Sets: (F1 Score: Larger is Better)** | | | | | | | **Average** |
|---|---|---|---|---|---|---|---|---|
| | **Jira** | **Twitter** | **Reddit** | **J + T** | **T + R** | **J + R** | **J + T + R** | |
| Jira | **0.923** | 0.704 | 0.780 | 0.801 | 0.731 | 0.834 | 0.821 | 0.799 |
| Twitter | 0.662 | **0.931** | 0.771 | 0.832 | 0.869 | 0.726 | 0.808 | 0.799 |
| Reddit | 0.704 | 0.789 | **0.863** | 0.757 | 0.816 | 0.833 | 0.804 | 0.795 |
| J + T | 0.918 | 0.921 | 0.788 | **0.924** | 0.858 | 0.843 | 0.883 | 0.876 |
| T + R | 0.742 | 0.918 | 0.842 | 0.868 | **0.909** | 0.804 | 0.871 | 0.850 |
| J + R | 0.899 | 0.753 | 0.816 | 0.813 | 0.797 | **0.871** | 0.819 | 0.824 |
| J + T + R | 0.887 | 0.901 | 0.833 | 0.911 | 0.905 | 0.863 | **0.886** | **0.883** |
| **Average** | 0.819 | 0.845 | 0.813 | 0.844 | 0.841 | 0.825 | 0.842 | |

**Table 5.** RoBERTa models with 4-class inference.

| **Model** | **Evaluation on Test Sets: (F1 Score: Larger is Better)** | | | | | | | **Average** |
|---|---|---|---|---|---|---|---|---|
| | **Jira** | **Twitter** | **Reddit** | **J + T** | **T + R** | **J + R** | **J + T + R** | |
| Jira | **0.862** | 0.714 | 0.765 | 0.788 | 0.649 | 0.801 | 0.779 | 0.765 |
| Twitter | 0.759 | **0.916** | 0.782 | 0.802 | 0.840 | 0.683 | 0.780 | 0.795 |
| Reddit | 0.739 | 0.756 | **0.835** | 0.708 | 0.818 | 0.799 | 0.767 | 0.775 |
| J + T | 0.844 | 0.898 | 0.775 | **0.916** | 0.832 | 0.818 | 0.846 | 0.847 |
| T + R | 0.764 | 0.884 | 0.832 | 0.794 | **0.902** | 0.800 | 0.839 | 0.831 |
| J + R | 0.832 | 0.749 | 0.821 | 0.797 | 0.782 | **0.862** | 0.822 | 0.809 |
| J + T + R | 0.854 | 0.868 | 0.818 | 0.894 | 0.885 | 0.851 | **0.901** | **0.867** |
| **Average** | 0.808 | 0.826 | 0.804 | 0.814 | 0.815 | 0.802 | 0.819 | |

By investigating the results presented in Tables 4 and 5, the following observations are worth mentioning:

**(1) Models struggle to predict well in the SE domain, possibly due to differences in technical language and content.** Based on the table columns, it is evident that Jira and Reddit datasets show the poorest performance. The Reddit dataset is extensive and has numerous labels. Despite using the Ekman grouping method, we can assume that the diverse data can impact the training of models. Upon analyzing the averages of both columns and rows, it is clear that the Reddit dataset performed the worst. It should be noted that although Reddit had the lowest performance, the Jira dataset produced results that were nearly as poor. This could be because technical communications between software engineers often include industry-specific terminology and jargon, which is not typically present in general conversation or news articles. As a result, the base models, which are trained on more general data, may struggle to accurately interpret these technical messages. On the other hand, Twitter-based models achieved the best performance, maybe due to their common data insight.

**(2) Single-domain DSSM models, tested on general datasets, showed an expected decrease in performance, but the performance can be significantly improved if the model is trained with data from DISMs.** By incorporating a variety of data from different domains, multi-domain datasets can integrate numerous models. For example, combining a rigorous domain dataset with a common domain dataset creates a model familiar with more challenging data, improving the performance over multi-domains. When examining the J + T row, we observe that by merging technical messages from Jira with day-to-day messages from Twitter, we built a model that is nearly as effective as the three-domain model.

**(3) Multi-domain models have a higher average F1 score than single-domain models, indicating a better performance.** Based on the table, the three-domain model has the highest F1 score: 0.883 in Table 4 and 0.867 in Table 5. Moreover, all the multi-domain models achieved an average F1 score over 0.800, whereas the single-domain models scored below 0.800 on average. This is because cross-domain models were trained on data from multiple datasets, while single-domain models were fine-tuned only on their specific data collection. As a consequence, single-domain models were surpassed by multi-domain models due to decreased performance on unfamiliar domains, despite having high accuracy on self-domains.

**(4) Multi-domain models are less efficient than single-domain models when evaluated on single-domain datasets.** Cross-domain models, which are trained on data from multiple domains, might be less efficient when evaluated on single-domain datasets compared to models specifically trained on those individual domains. The reason behind this is that multi-domain models handle a wide range of data, with various linguistic styles, terminologies, or contextual nuances. This diversity is beneficial across different domains but may result in reduced performance when focused on a particular area. On the other hand, single-domain models are fine-tuned specifically for a narrow domain, which can make them yield better results when applied to datasets from that specific domain.

**(5) Labels that do not fit in the dataset can affect the performance of the models.** When comparing Tables 4 and 5, it can be observed that all scores in Table 5 are lower. The reason behind this is that certain data were labeled as neutral because they cannot be classified under a specific emotion. While having a label that can fit anywhere can improve the model's performance, it can also potentially degrade the quality of the data it was trained on.

Using flexible labels can improve a model's ability to handle noisy data that may not strictly adhere to predefined categories or labels [37]. However, if the labels are too broad or vague, it can induce ambiguity and the transformer model might struggle to learn meaningful patterns. This may result in poor performance. Ultimately, by performing data cleaning and preprocessing, we can ensure that the model receives high-quality training data, thus preventing potential issues related to overly flexible labels [38].

### *5.4. Research Questions*

Based on the mentioned results, we provide the answers to the research questions:

**RQ1:** *Are DSSMs semantically distant from DISMs?*

In order to answer this question, we need to refer to Tables 4 and 5. The DSSMs are representative of the Jira single-domain model, while the DISMs are representative of the Twitter single-domain model. The Reddit self-domain model falls somewhere in between. Looking at the F1 scores, we can see that whenever the Jira model performed the best, the Twitter model performed the worst and vice versa. Meanwhile, the Reddit model consistently performed at a moderate level. This means that a particular sentiment is expressed differently across different domains, making it seem distant from each other. In our example, Twitter's casual and informal messages contain domain-independent words to describe emotions like love, joy, surprise, sadness, fear, or anger. On the other hand, the SE field utilizes entirely different syntaxes to express the same emotions. After considering these factors, we deduce that DSSMs and DISMs are semantically distant from each other.

**RQ2:** *Are sentiments expressed differently in each domain?*

In the Twitter dataset, a negative comment will generally express a disagreement between the users. However, in the Jira dataset, such comments could be a form of criticism or feedback within a team. While negative sentiments in the first case can lead to conflicts, in a community of software engineers the sentiments can indicate growth within the team. It is worth mentioning that negative sentiments could be more valuable than positive ones in SE communication. The sentiments might not be expressed differently in all domains as evidenced by the Reddit dataset, which was found to be similar in sentiment expression to both Jira and Twitter.

**RQ3:** *Could cross-domain models accurately capture the meaning of DSSM terminology?*

Our experiment has concluded that cross-domain models outperform single-domain models in terms of average F1 score. Therefore, we may state that cross-domain models are able to successfully capture the meaning of DSSM terminology.

## 6. Conclusions

Our investigation using Jira, Twitter, and Reddit short messages has demonstrated that the user-generated DSSMs are different in terms of lexicon, vocabulary, and terminology, from DISMs. We were able to determine this by using sentiment analysis in a cross-domain environment and relying on the RoBERTa model as the backbone of our methodology. Based on the experimental results, we can observe that while different domains share a common lexicon, they also have unique characteristics specific to each domain. Jira messages, which are typically used by professionals, tend to be more complex and use specific terminology, while Twitter messages are often brief and to the point, with a focus on common vocabulary. Reddit messages fall somewhere in between: they contain various communities with messages from different levels of specificity and independence.

By recognizing the distinct emotional needs and preferences of users in different contexts, researchers, teams, or applications can benefit by meeting user expectations, enhancing satisfaction, or tailoring better features. It can also facilitate cross-communication and collaboration. By understanding how emotions vary, individuals and organizations can adapt their interactions to effectively engage in diverse fields.

In the future, we aim to adjust the models to recognize emoticons within messages. By integrating emoticon recognition capabilities, we anticipate improving the accuracy and performance of sentiment analysis models within DSSMs. Additionally, we may use more advanced models like GPT-4 or T5 to obtain more precise insights.

**Author Contributions:** Conceptualization, T.-R.P.; methodology, T.-R.P., C.-D.C. and D.-I.C.; software, T.-R.P. and C.-D.C.; validation, T.-R.P. and C.-D.C.; formal analysis, T.-R.P., C.-D.C. and D.-I.C.; investigation, T.-R.P., C.-D.C. and D.-I.C.; resources, D.-I.C.; data curation, T.-R.P., C.-D.C. and D.-I.C.; writing—original draft preparation, T.-R.P. and C.-D.C.; writing—review and editing, T.-R.P., C.-D.C. and D.-I.C.; supervision, D.-I.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Devika, M.D.; Sunitha, C.; Ganesh, A. Sentiment analysis: A comparative study on different approaches. *Procedia Comput. Sci.* **2016**, *87*, 44–49. [CrossRef]
2. Kolchyna, O.; Souza, T.T.; Treleaven, P.; Aste, T. Twitter sentiment analysis: Lexicon method, machine learning method and their combination. *arXiv* **2015**, arXiv:1507.00955.
3. Gonçalves, P.; Araújo, M.; Benevenuto, F.; Cha, M. Comparing and combining sentiment analysis methods. In Proceedings of the First ACM Conference on Online Social Networks, Boston, MA, USA, 7–8 October 2013; pp. 27–38.
4. Khan, I.A.; Brinkman, W.P.; Hierons, R.M. Do moods affect programmers' debug performance? *Cogn. Technol. Work.* **2011**, *13*, 245–258. [CrossRef]
5. Lesiuk, T. The effect of music listening on work performance. *Psychol. Music.* **2005**, *33*, 173–191. [CrossRef]
6. Graziotin, D.; Wang, X.; Abrahamsson, P. Are happy developers more productive? The correlation of affective states of software developers and their self-assessed productivity. In Proceedings of the Product-Focused Software Process Improvement: 14th International Conference, PROFES 2013, Paphos, Cyprus, 12–14 June 2013; Proceedings 14; Springer: Berlin/Heidelberg, Germany, 2013; pp. 50–64.

7. Wrobel, M.R. Emotions in the software development process. In Proceedings of the 2013 6th International Conference on Human System Interactions (HSI), Sopot, Poland, 6–8 June 2013; pp. 518–523.
8. 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 International Conference on Software Engineering, Gothenburg, Sweden, 27 May–3 June 2018; pp. 94–104.
9. Islam, M.R.; Zibran, M.F. Leveraging automated sentiment analysis in software engineering. In Proceedings of the 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), Buenos Aires, Argentina, 20–21 May 2017; pp. 203–214.
10. Muhammad, A.; Wiratunga, N.; Lothian, R.; Glassey, R. Domain-Based Lexicon Enhancement for Sentiment Analysis. In Proceedings of the SMA@ BCS-SGAI, Cambridge, UK, 10 December 2013; pp. 7–18.
11. Ortu, M.; Adams, B.; Destefanis, G.; Tourani, P.; Marchesi, M.; Tonelli, R. Are bullies more productive? Empirical study of affectiveness vs. issue fixing time. In Proceedings of the 2015 IEEE/ACM 12th Working Conference on Mining Software Repositories, Florence, Italy, 16–17 May 2015; pp. 303–313.
12. Al-Moslmi, T.; Omar, N.; Abdullah, S.; Albared, M. Approaches to cross-domain sentiment analysis: A systematic literature review. *IEEE Access* **2017**, *5*, 16173–16192. [CrossRef]
13. Durairaj, A.K.; Chinnalagu, A. Transformer based Contextual Model for Sentiment Analysis of Customer Reviews: A Fine-tuned BERT. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 474–480. [CrossRef]
14. Lin, C.; Bethard, S.; Dligach, D.; Sadeque, F.; Savova, G.; Miller, T.A. Does BERT need domain adaptation for clinical negation detection? *J. Am. Med. Inform. Assoc.* **2020**, *27*, 584–591. [CrossRef]
15. 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 Trans. Softw. Eng.* **2022**, *49*, 1487–1507. [CrossRef]
16. 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? In Proceedings of the 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME), Adelaide, SA, Australia, 28 September–2 October 2020; pp. 70–80.
17. Chen, Z.; Cao, Y.; Lu, X.; Mei, Q.; Liu, X. Sentimoji: An emoji-powered learning approach for sentiment analysis in software engineering. In Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Tallinn, Estonia, 26–30 August 2019; pp. 841–852.
18. Felbo, B.; Mislove, A.; Søgaard, A.; Rahwan, I.; Lehmann, S. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv* **2017**, arXiv:1708.00524.
19. Ortu, M.; Murgia, A.; Destefanis, G.; Tourani, P.; Tonelli, R.; Marchesi, M.; Adams, B. The emotional side of software developers in JIRA. In Proceedings of the 13th International Conference on Mining Software Repositories, Austin, TX, USA, 14–22 May 2016; pp. 480–483.
20. Ortu, M.; Destefanis, G.; Adams, B.; Murgia, A.; Marchesi, M.; Tonelli, R. The jira repository dataset: Understanding social aspects of software development. In Proceedings of the 11th International Conference on Predictive Models and Data Analytics in Software Engineering, Beijing, China, 21 October 2015; pp. 1–4.
21. Saravia, E.; Liu, H.C.T.; Huang, Y.H.; Wu, J.; Chen, Y.S. Carer: Contextualized affect representations for emotion recognition. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 3687–3697.
22. Chen, G.M. Tweet this: A uses and gratifications perspective on how active Twitter use gratifies a need to connect with others. *Comput. Hum. Behav.* **2011**, *27*, 755–762. [CrossRef]
23. Davenport, S.W.; Bergman, S.M.; Bergman, J.Z.; Fearrington, M.E. Twitter versus Facebook: Exploring the role of narcissism in the motives and usage of different social media platforms. *Comput. Hum. Behav.* **2014**, *32*, 212–220. [CrossRef]
24. Demszky, D.; Movshovitz-Attias, D.; Ko, J.; Cowen, A.; Nemade, G.; Ravi, S. GoEmotions: A dataset of fine-grained emotions. *arXiv* **2020**, arXiv:2005.00547.
25. Proferes, N.; Jones, N.; Gilbert, S.; Fiesler, C.; Zimmer, M. Studying reddit: A systematic overview of disciplines, approaches, methods, and ethics. *Soc. Media+ Soc.* **2021**, *7*, 20563051211019004. [CrossRef]
26. Medvedev, A.N.; Lambiotte, R.; Delvenne, J.C. The anatomy of Reddit: An overview of academic research. In *Dynamics on and of Complex Networks III: Machine Learning and Statistical Physics Approaches 10*; Springer: Cham, Switzerland, 2019; pp. 183–204.
27. Ortu, M.; Destefanis, G.; Kassab, M.; Marchesi, M. Measuring and understanding the effectiveness of jira developers communities. In Proceedings of the 2015 IEEE/ACM 6th International Workshop on Emerging Trends in Software Metrics, Florence, Italy, 17 May 2015; pp. 3–10.
28. Priya, S.; Sequeira, R.; Chandra, J.; Dandapat, S.K. Where should one get news updates: Twitter or Reddit. *Online Soc. Netw. Media* **2019**, *9*, 17–29. [CrossRef]
29. Ekman, P. An argument for basic emotions. *Cogn. Emot.* **1992**, *6*, 169–200. [CrossRef]
30. Cortiz, D. Exploring transformers in emotion recognition: A comparison of bert, distillbert, roberta, xlnet and electra. *arXiv* **2021**, arXiv:2104.02041.
31. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.

32. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–11.
33. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942.
34. Adoma, A.F.; Henry, N.M.; Chen, W. Comparative analyses of bert, roberta, distilbert, and xlnet for text-based emotion recognition. In Proceedings of the 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 18–20 December 2020; pp. 117–121.
35. Qasim, R.; Bangyal, W.H.; Alqarni, M.A.; Ali Almazroi, A. A fine-tuned BERT-based transfer learning approach for text classification. *J. Healthc. Eng.* **2022**, *2022*, 3498123. [CrossRef]
36. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
37. Sheng, V.S.; Provost, F.; Ipeirotis, P.G. Get another label? improving data quality and data mining using multiple, noisy labelers. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 614–622.
38. Munappy, A.R.; Bosch, J.; Olsson, H.H.; Arpteg, A.; Brinne, B. Data management for production quality deep learning models: Challenges and solutions. *J. Syst. Softw.* **2022**, *191*, 111359. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.