- ORIGINAL ARTICLE -

Trust evaluation in virtual software development teams using BERT-based language models

Evaluación de confianza en equipos virtuales de desarrollo de software usando modelos de lenguajes basados en BERT

Sergio Zapata[1], Facundo Gallardo[1], Gustavo Sevilla[1] and Estela Torres[2]
Raymundo Forradellas[3]

[1] *Informatic Institute, Universidad Nacional de San Juan, Argentina*
{szapata, fgallardo, gsevilla}@iinfo.unsj.edu.ar
[2] *Informatic Departament, Universidad Nacional de San Juan, Argentina*
etorres@iinfo.unsj.edu.ar
[3] *Intelligent Systems Laboratory, Universidad Nacional de Cuyo, Argentina*
kike@uncu.edu.ar

Abstract

Nowadays, people from different geographical areas can be closely related thanks to advances in information and communication technologies. This has a greater impact in software development organizations where their members form virtual work teams. In these new co-located work scenarios, the construction of interpersonal trust is more complex and its impact is very relevant in the performance of software development teams. This paper presents the results of the performance evaluation of four pre-trained language models based on BERT applied to trust analysis tasks. For this work, a small dataset of 1453 comments obtained from software projects stored on Github was created. The evaluated language models achieved moderately good values, in the order of 0.84 for the F1-score metric, which augurs that with further research they could be significantly improved.

Keywords: BERT-based language model, Social software engineering, Trust analysis.

Resumen

Actualmente personas de distintas zonas geográficas pueden estar fuertemente relacionadas gracias a los avances en las tecnologías de información y comunicación. Esto tiene un impacto mayor en organizaciones de desarrollo de software en donde sus miembros conforman equipos virtuales de trabajo. En estos nuevos escenarios colocalizados de trabajo la construcción de confianza interpersonal es más compleja y su impacto es muy relevante en el desempeño de los equipos de desarrollo de software. Este trabajo presenta los resultados de la evaluación de desempeño de cuatro modelos de lenguaje pre-entrenados basados en BERT aplicados a tareas de análisis de trust. Para este trabajo se creó un pequeño dataset de 1453 comentarios obtenidos de proyectos de software almacenados en Github. Los modelos de lenguaje evaluados alcanzaron valores moderadamente buenos, del orden de 0.84 para la métrica F1-score, lo que augura que con una mayor investigación podrían mejorarse significativamente.

Palabras claves: Modelo de lenguaje BERT, Ingeniería de software social, Modelo de lenguaje BERT.

1. Introduction

Nowadays people from different parts of the world are more connected thanks to the progress of Information and Communication Technologies (ICT) [1,2]. These new scenarios introduce new technological, cultural and organizational challenges. Thus, in software development organizations have emerged virtual work teams, i.e., groups of software developers that work geographically distributed [3,4]. These work groups are known as virtual software teams (VST).

Understanding how emotions, moods and other human aspects affect the final outcome of technical activities (e.g. software quality) is a topic addressed by research [5-9]. Trust is a crucial social aspect of cooperative work in software engineering [10]. The trust in VST is more important than in collocated software teams [11]. Several researches shown that trust is a key factor in determining the success or failure of virtual work groups [12-15].

Mayer et al. [16] define trust as "the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party". People participate in risky activities, that they cannot monitor, because of the trust they have in others. Even when they can be harmed by the actions of others [17].

Work groups with higher trust are more proactive, more focused on the outcome of the task, more optimistic, initiate interactions more frequently, and provide more productive feedback. [18]. Teams with high trust are correlated to teams with high productivity and member satisfaction. [19]. Teams without trust can be successful, but they tend to pay extra costs such as monitoring teammates and backing up their work [20].

Given that trust is very important in VST, it is necessary to measure, evaluate and control it. Software measurement techniques have the potential to enhance control of the software engineering process, reducing time and costs and producing higher quality software [21]. Software measures have been promoting themselves as essential resources to improve quality and control costs during software development [22].

If a low level of trust is measured, the software project manager could encourage face-to-face meetings [23,24], new leadership styles [25,26], new communication tools [11,27,28], ad-hoc training [28,29], and other initiatives to improve group trust.

In the current digital society, large volumes of data are continuously generated. Thus, the growing use of version control system (VCS) repositories in VST, like GitHub1, and rise of the Mining Software Repositories research [30] could promote the application of indirect and objective measurements of trust in VST.

Natural language processing (NLP) is a subfield of artificial intelligence (AI) that can study human and computer interactions through natural languages, such as the meaning of words, phrases, sentences, and syntactic and semantic processing [31]. In the VCS repositories, comments between team members are recorded, so that by applying NLP techniques to them, we could identify comments that evidence trust.

Deep learning gives us big potential in the NLP field [32]. The machine learning architectures based on transformers have made great progress on many different NLP tasks. There are many transformer-based language models, including BERT [33], RoBERTa [34] GPT-2 [35] and XLNet [36].

Trust analysis could be done using transformers-based NLP models, which are capable of understanding and modeling the context of the text. Several studies showed that, within the various transformers, BERT (Bidirectional Encoder Representations from Transformers) proves to be efficient in analyzing semantic perception or feelings when using data as text [37-39].

BERT is a deep learning architecture that can be used for downstream NLP tasks. BERT takes a distinctive approach to learning. Bidirectional means that BERT learns from the left and right sides of the token (word) during learning [33]. A bidirectional method is essential to understand the meaning of language [31].

There are two main steps in BERT: pre-training and fine-tuning [33]. During pre-training, BERT is trained in a large unlabeled corpus with two unsupervised tasks: masked language model (MLM) and next sentence prediction (NSP) to produce a pre-trained model. For fine-tuning, the model is initialized with the pre-trained parameters, and all the parameters are fine-tuned using a labeled dataset for specific tasks such as text classification for sentiment analysis.

Therefore, we believe that it would be interesting to apply these BERT-based language models to trust classification. That is, to find a BERT-based classifier that predicts whether a comment exchanged between members of a VST contains evidence of trust.

The objective of this work was to evaluate the performance of BERT-based pre-training language models applied to trust classification task of SE comments in VST contexts. The comments were extracted from Github repositories. We evaluated several BERT-based models that were pre-trained with Spanish text corpora.

Our interest is specifically oriented towards software projects of Latin American organizations, since most of the practitioners who interact with our research group come from that region. These organizations predominantly use the Spanish language for their communications.

The rest of this article is structured as follows: Related Work section describes the results from other related studies; Research Method section details the research method applied in this work; Results and Discussion section presents the results, synthesis and discussion of the obtained data. Some proposals for future research and actions to improve the results of this work are presented in the Future work section.

Finally, the Conclusion section presents the conclusions and some proposals for future research.

2. Related works

After conducting searches from various electronic

[1] https://github.com/

data sources (online databases, publisher sites and general search engines) about trust evaluation in software engineering (SE) contexts, we have found some interesting work, but none involving the use of the pre-training language model.

Niazi et al. [40] present a systematic literature review (SLR) aimed at identifying relevant factors for building trust in offshore software outsourcing relationships. In that study, trust is considered for the client vendor's relationship and is defined as clients and vendors having positive expectations of each other's actions. The authors revealed that elements such as: face to face meeting, better communication, contract management between client and vendor, defining process tools, procedures and policies and reliable management, play an important influence in establishing trust between clients and vendors, in the context of offshore software outsourcing. This work only reflects trust between clients and vendors.

Zapata et al. [41] conducted an SLR involving studies through July 2019 to identify, evaluate, and synthesize reported research on the measurement of interpersonal trust in VST. This work shown that most studies use questionnaires or interviews to measure trust, but the authors consider that software repositories mining to obtain trust degrees will be an interesting research trend in the future.

The propose of da Silva et al. paper [42] is to elaborate an evidence based model of distributed software development project management from the research findings about challenges of global software development (GSD). The authors funded the construction of their model on the evidence collected and synthesized by a comprehensive systematic mapping study (SMS), containing 70 research papers published between 1997 and 2009. Specifically, this work identified practices and traditional communication tools that would promote the trust in GSD. The main practices identified were: provision of and training in collaboration and coordination tools, use of a common software process among the several work sites and divide the work into well-defined modules to carrying out progressive integration. The most important communication tools identified were: phone (including teleconference and audio conference), emails and video conference. In all cases, these tools are supported by traditional (non-innovative) technology. This paper does not include discussion about trust evaluation, which is the focus of our current research.

Tyagi et al. [43] present a lightweight SLR to analyze the role of trust in distributed agile software development projects. The paper offers a comprehensive overview about the role of trust in a distributed agile environment and identifies different challenges faced by agile teams that includes lack of face-to-face communication, different cultural background, linguistic barriers, and different time zones. Important issues such as poor socialization among team members, lack of face-to-face interactions and unpredictability in communication are highlighted as causes of lack of trust in VST. The issue of trust assessment is not addressed in this paper. This article also does not cover the issue of trust evaluation.

We have not found studies related to trust detection applying machine learning in SE contexts. However, there are studies related to sentiment polarity classification, a topic that is close to our study of trust.

Uddin et al. [44] report the results of an empirical study that was conducted to determine the feasibility of developing a sentiment detection toolkit for SE by combining the polarity labels of independent SE-specific sentiment detectors. They conclude that transformer-based deep learning models, such as BERT, provide good performance even with small datasets due to their design as pre-trained models. They find that the Sentisead tool combined with RoBERTa offers the best F1 score of 0.805 on multiple datasets, while RoBERTa alone shows an F1 score of 0.801.

Obaidi et al. [45] present the results of a SMS of sentiment analysis tools developed for or applied in the context of SE. The results summarize insights from 106 papers regarding the application domain, the purpose, the used data sets, the approaches for developing sentiment analysis tools, the usage of already existing tools, and the difficulties researchers face. According to this SMS, sentiment analysis is frequently applied to open-source software projects, and most approaches are neural networks or support-vector machines. The best performing approach is neural networks and the best tool is BERT, with 0.94 accuracy and 0.83 F1 score.

Thus, we have not found studies that applied pre-training language models to assess trust in VST, even less when the members of these teams use the Spanish language to interact, this being the purpose of our study.

3. Research method

This section describes the experimental process, see Fig. 1, applied in this work to evaluate the performance of several BERT-based pre-trained language models on an ad-hoc trust dataset of Github comments. We have applied a supervised approach to fine-tuning the BERT-based models. We have prepared a technical data sheet[2] of this article for possible future work by the scientific community.

[2] https://bit.ly/3Fj1vEB

3.1. Trust dataset creation phase

Since we have not found a dataset with labeled comments about trust, we created our own trust dataset to be used in the training phase of the fine-tuning process of the several BERT based languages models to be evaluated.

Given our research interests, we focused on real software projects from well-known Latin American organizations with many messages in Spanish supported on public platforms.

In order to create the trust dataset, we first searched, selected and extracted Spanish comments from three software development public projects registered in Github, a platform containing more than 80 million public software repositories.

We extracted 1453 comments, 288 of them were written in Mexican software projects while the rest in Argentinean software projects.

The comments contained in the Github projects analyzed have special characteristics, such as the following:

- Informality, informal language is used, with idioms, in many cases with poor wording and even spelling mistakes. For example: "*Me late ahora lo actualizo*", "*No, me lo comí, gracias*", "*Uf alto leak! Buen finding!*", "*LGTM!*".
- Task-oriented, most of the comments refer to the technical tasks of software development, there is little social comments. For example: "*No, rollbackeamos el seteo de este color*", "*Hay que eliminar la tarjeta "test"*", "*¿Ya esta listo para merge?*".
- SE Lexico-Oriented, there are many comments that include SE-specific terms. For example: "*Hay que usar el stylesheet*", "*Hago el cambio y aplico otro PR*", "*Creo que utilizar un dropdown sería más escalable*", "*Estos commits se mergearon en Pull Request #259*".

While Github is a great repository of data, most of the projects stored there are not useful for the purposes of our research. We require Spanish language comments from actual and public software projects. Most of the actual Spanish language projects are private, i.e. can only be accessed by obtaining explicit permission from the company or organization that owns the repository, which makes it difficult to access these data.

Many public Spanish projects registered in Github are:

- Personal software development initiatives, no working team.

- Software development initiatives within educational contexts. They are not seen as actual projects of the software industry.

- Software development initiatives with minimal progress, unfinished and without evident commitment from their members. They are not seen as actual projects of the software industry.

Therefore, the task of extracting the appropriate comments from GitHub for our research is not trivial.

The extraction process was carried out using the Github API and the GraphQL query language. The comments were extracted from several pull requests (open, closed or merged) of the three selected software projects.

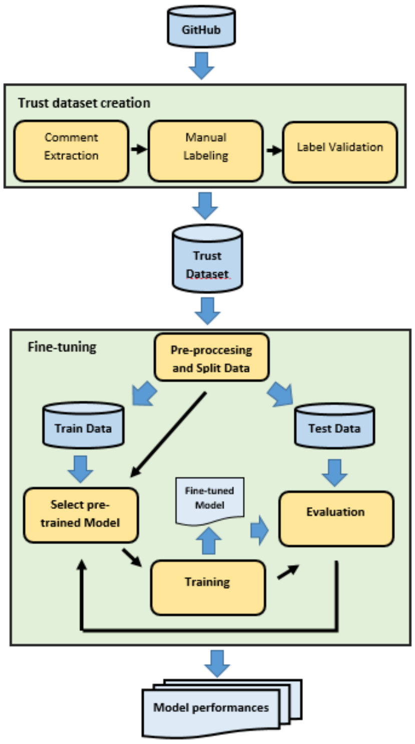

Fig. 1 Experimental process

The labeling task (annotation) was carried out by the first four authors. Given a comment the author (annotator) identify whether the comment has evidence trust. Several studies have shown that positive tone, feedback, integrity, delegation and knowledge sharing are characteristics of trust evidence in virtual work teams. Comments that the annotator identifies as evidence of trust should be labelled as trusting comments, see examples in Table 1.

The greater the trust among participants, the greater is information sharing among them, both in the form of knowledge sharing and acceptance [46]. The trust has strong impact in knowledge transfer [47]. Knowledge sharing is a proxy variable of trust. People share knowledge when they trust that others will use that knowledge in a beneficial way.

Trust and integrity are highly correlated. When the trust in someone is high, the integrity is high as well [48]. Comments that express integrity (honesty, transparency, trustworthiness, sincerity) are comments that indirectly evidence trust.

In high-trust team members tend to show enthusiasm in conversation, praising and encouraging each other. Arguments between members are resolved so delicately that they almost go unnoticed [48]. Positive tone is a proxy variable of trust. Comments that reflect good treatment, no offense, motivating, supportive, grateful, congratulatory, constructive, etc. are considered tone positive comments.

High-trust teams display intense communication and provide feedback about team members' work [47,48,49,50,47]. These comments express the degree of progress/difficulties/status of the tasks of each of the members.

Delegation is behavior that reflects trust [47], it is indirect evidence. These comments mainly express the assignment of tasks to other trusted persons.

Table 1. Examples of comments for each of the characteristic

Knowledge sharing *"Esto lo que hace es setear las animaciones de entrada/salida del modal, solo en caso de que se necesite. Por default está seteado en true el shouldAnimate, pero lo pueden overraidear para que no se anime."*
Knowledge acceptance *"Me parece muy buena la idea. Ahorita arreglé conflictos en este branch, pero me parece mejor tu solución"*
Integrity *"no sé cómo se hace eso"*
Positive tone *"perfecto ahí entendí, genial entonces! haciendo el otro cambiecito estaríamos"*
Feedback *"Cierro el pr, porque tiene estos commits y algunos fixes."*
Delegation *"Te encargo que resuelvas los conflictos"*

We held an initial joint meeting among the four authors to adjust annotation criteria. During this initial meeting, brief individual labeling sessions of 20 comments were held. Afterwards, a group discussion was held regarding the labeling performed by each of the authors in order to reach agreement on common labeling criteria. These short sections were repeated until a high level of agreement was reached on labelling and a common set of criteria was obtained.

Thus, if a comment expressed any of trust evidence characteristics, the annotator should note as a "trusting comment" and otherwise as a "non-trusting comment", so a two-class trust dataset[3] was created.

As an outcome of the initial meeting, we obtained some trust annotation guidelines[4] to support authors in the labeling process.

Once that a common set of labeling criteria was obtained each trust dataset comment was analyzed to verify whether it evidenced some form of trust between the VST members.

Each comment was analyzed and labeled in individual sessions by the first three authors, see Fig. 2. Then, during the validation stage, each comment was definitively registered with a tag only if there was unanimous agreement of the three authors. Otherwise, a discussion meeting of the first four authors was held to achieve a final decision based on a majority agreement.

After completing the trust dataset creation phase, which involved around 100 man-hours, we obtained a small binary trust dataset of 1435 Github Spanish comments (approximately 40% trusting comments). This dataset was then used as input to the fine-tuning phase.

3.2. Fine-tuning phase

In this phase, we fine-tuned several Bert-based pre-trained language models by training them on the trust dataset created in the previous phase. We then evaluated the performance of each fine-tuned model by applying the accuracy, F1 score, recall and precision metrics.

The text data in the Github comments contained a variety of noises, such as URLs, code, mentions (@), etc. We apply a text preprocessing technique implemented in Python to automatically clean up the comments before feeding them to the BERT-based model training task.

Then, we randomly split the trust dataset into a training dataset and a test dataset (20% of the trust dataset) to evaluate the several language models applied.

According to the NLP-Progress[5], a repository to track the progress in Natural Language Processing (NLP), in trust analysis-like tasks, such as sentiment analysis, BERT-based models are among the highest performance. The BERT-based pre-trained language models used in our work, obtained from Hugging Face[6] AI community, are:

- BETO: a BERT model trained on a big Spanish corpus [51]. BETO is of size similar to a BERT-Base and was trained with the Whole Word

[3] https://bit.ly/3Bfz8DR
[4] https://bit.ly/3BHkI0U
[5] https://nlpprogress.com
[6] https://huggingface.co

Masking technique. BETO used all the data from Wikipedia and all of the sources of the OPUS Project [52] that had text in Spanish. This sources includes United Nations and Government journals, TED Talks, Subtitles, News Stories and more. The corpus for training has about 3 billion words.

- BETO QA: This model is a fine-tuned on SQuAD-es-v2.0 and distilled version of BETO for question-answering tasks. The teacher model for the distillation was bert-base-multilingual-cased [53].
- BERT multilingual: pre-trained model on the top 102 languages with the largest Wikipedia using a masked language modeling objective [33].
- BETO NER: is a fine-tuned model[7] on NER-C version of the Spanish BERT cased (BETO) for name entity recognition (NER) downstream tasks.

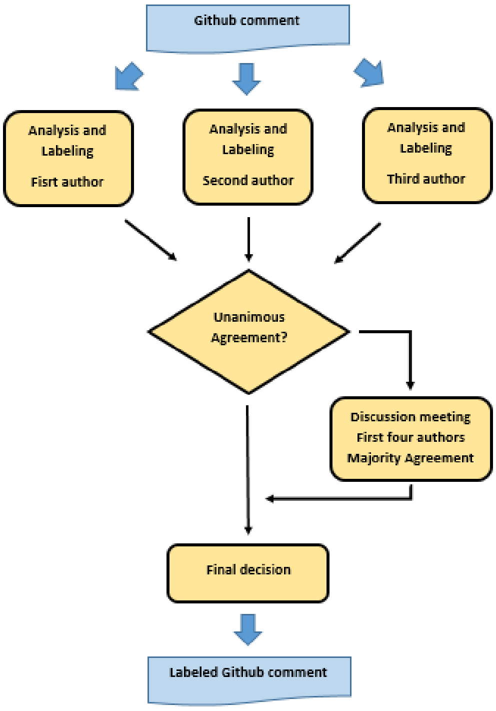
Github comment
Analysis and Labeling Fisrt author
Analysis and Labeling Second author
Analysis and Labeling Third author
Unanimous Agreement?
Discussion meeting First four authors Majority Agreement
Final decision
Labeled Github comment

Fig. 2 Trust labeling procedure

The language models used in this work are the Bert-based models in Spanish most downloaded by the face-hugging community.

To evaluate the performance of each of the BERT-based models, we calculated model accuracy (A), a traditional metric in text classification. In addition, we use the F1 score (F1) because it is recommended by research when the test dataset is unbalanced [54], as in our case (40% trusting comments and 60% non-trusting comments). Precision (P), Recall (R) and Specificity (S) metrics were also applied. The calculation formulas for each metric are as follows:

$$A = \frac{TP+TN}{TP+FP+TN+FN} \qquad R = \frac{TP}{TP+FN} \qquad P = \frac{TP}{TP+FP}$$

$$F1 = 2 * \frac{P*R}{P+R} \qquad S = \frac{TN}{TN+FP}$$

TP = # of true positives, FN = # of false negatives, TN = # of true negatives, and FP =# of False positives. For our experimental work, trusting comments are the positive cases and non-trusting comments are the negative cases.

The best results during language models fine-tuning were obtained with the following hyperparameter values: batch size=16; epochs=2; learning rate=0.5e-5; dropout=0.1.

We implement the fine-tuning and evaluation algorithms in Python programing language by using pytorch and transformer libraries. We used the Google Colab[8] platform as software development environment.

The results of the evaluation of the language models are shown and discussed in the next section.

4. Results and discussion

Table 2 shows the results obtained from evaluating the four BERT-based fine-tuned language models in trust analysis. For each model we have registered the five performance metrics calculated; accuracy, F1-score, Recall, Precision and Specificity.

Table 2. Performance of BERT-based language models in trust analysis

Models	A	F1	R	P	S
BERTqa	0.8601	0.8333	0.8475	0.8197	0.8690
BERTm	0.8182	0.7869	0.8136	0.7619	0.8214
BETO	0.8601	**0.8437**	**0.9153**	0.7826	0.8214
BERTner	**0.8671**	0.8403	0.8475	**0.8333**	**0.8810**

BERTner is the model with the best accuracy results (0.8671), with very similar results arising from BERTqa and BETO, both with identical metrics (0.8601). BERTm is the lowest performing model (0.8182), with results approximately 5% lower than those of BERTner.

Regarding the F1-score metric, the BETO model performs best (0.8437), very closely followed by BERTner (0.8403) and BERTqa (0.8333). BERTm has the lowest performance (0.7869), about 7% less than BETO.

BETO is the model with the best performance

[7] https://huggingface.co/mrm8488/bert-spanish-cased-finetuned-ner

[8] https://colab.research.google.com/

with respect to recall metric (0.9153), outperforming all other models by at least 8%. While BERTner is the best model in terms of precision (0.8333) and specificity (0.8810) metrics.

The values achieved by the several models are good considering the small size (1435 comments) of the trust dataset used in fine-tuning. In similar models, such as sentiment analysis models, the best performance reaches 0.95 in accuracy metric by using a dataset with 50000 comments [55].

BERTqa, BERTner and BETO perform very similarly in accuracy metrics and F1 score. This is probably because these models share the same pre-training dataset; they differ only in the classification downstream tasks applied by each of them during fine-tuning. They use a big dataset exclusively in Spanish. While BERTm, the lowest performing model, uses a multilingual dataset which may negatively affect its performance. This confirms the impact of the characteristics of the datasets on the quality of the results obtained in NLP classification tasks.

Although the dataset shared by BERTqa, BERTner and BETO contains Spanish text, the sources (news sites, Wikipedia, TED talks, etc.) are formal texts. While in the Github comments analyzed in this work the text is informal, less structured and oriented to a SE lexicon. Therefore, incorporating SE text into the pre-training corpus of the models would possibly improve their performance in trust analysis.

Since we used an unbalanced trust dataset we could consider that BETO model, which obtains highest values for the F1-score metric, is the best alternative. Because BERTner has a very similar F1-score value to BETO but has more balanced values in the other metrics, it is acceptable to consider the BERTner model as a very good alternative of classifier.

The results obtained from the language models used in this article suggest that this supervised BERT-based automatic approach for trust analysis in VST is promising, emerging as a more feasible and less invasive alternative to trust evaluation traditional approaches using questionnaire or interview. In addition, an automatic approach would allow a real-time evaluation of trust levels in VST.

The results obtained in this paper are also of interest to SE practitioners. VST leaders could have a metric, automatically obtained in real time, of the levels of trust among developers, something very useful to make decisions favoring the performance of these teams.

The metrics obtained in this work are strictly limited by the characteristics of the created data set, which is not representative of all software organizations. These metrics are limited to software projects immersed in medium-sized Argentine and Mexican software organizations that use the Github platform, and they may vary depending on the language, organizational culture, and other sociocultural factors of these organizations. However, the trust measurement method in VST, based on a suitable dataset, could be successfully applied to other contexts.

The main threat to validity we identified was researcher bias during the trust labelling procedure. To minimize this threat, we applied a redundant comment review procedure involving at least three authors, and in some cases up to four, in order to make a highly consensual final decision. Also for this purpose, trust annotation guidelines were developed with the aim of unifying criteria among annotators.

5. Future work

The present article establishes a baseline for future research aimed at solving the problem of trust automatic evaluation among members of VST. The BERT-based language models analyzed have achieved acceptable metric values, which promises that future works will yield interesting results for the SE research.

Some of the future actions that could be undertaken to obtain better performance results are:

  - Increase the size of the trust dataset by manually tagging new Github comments. Considering that the trust dataset was built in approximately 100 man-hours, we believe that manually augmenting the trust dataset size is a feasible task. However, the greatest difficulty lies in obtaining comments from real public software projects in Spanish.
  - Perform a further pre-training of BERT-based language models with a large software engineering domain dataset, which contains informal, unstructured text oriented to the software development lexicon.
- Apply a semi-supervised approach to fine-tuning the BERT-based models. By using feature selection strategies, such as TF-IDF [56] or logistic regression [57], we could obtain a significantly larger trust dataset with comments automatically labeled.
- Analyze the performance of other transformer-based models, such as GPT [35,58].
- Diversify the data sources of the trust dataset by incorporating VST comments from other repositories, such as JIRA[9], and projects from other Spanish-speaking countries. This would improve the generalization of the scope of the language models obtained.

[9] https://www.atlassian.com/es/software/jira

6. Conclusions

This paper presents a performance evaluation of the BERT-based pre-training language models applied to trust classification task of SE comments in VST contexts. We have used a supervised approach using a small trust dataset created especially for this work, which consists of 1435 Spanish comments from Github (approximately 40% trusting comments and the rest non-trusting comments). Four BERT-based language models were evaluated, three of them based on the same Spanish dataset and the other based on a multilingual dataset.

The results obtained show that the evaluated language models obtain moderately good levels of accuracy and F1-score, around 0.86 and 0.84 for the best performing models. The language models based on the Spanish dataset outperform the model based on multilingual dataset by approximately 6%.

This work shows that the BERT-based supervised approach applied trust analysis in VST is promising, emerging as a more feasible and less invasive alternative to traditional trust evaluation using questionnaires or interviews. Nevertheless, future research could be undertaken to obtain yet better performance results.

Applying a semi-supervised approach, increasing the size of the trust dataset and performing a further pre-training of the language models with a dataset specific from the SE domain are some of the research actions that could aim to improve the results.

The repetition of this experience in other software development scenarios, for example in software projects executed in other Spanish-speaking countries, will provide the proposal greater possibilities of generalizing the results.

The contributions of this research can be useful to improve the performance of virtual software development teams by focusing on the human aspects that arise in them. Practitioners using agile software development methods, which place strong emphasis on human relations, could be the main beneficiaries of the advances offered by this work.

Competing interests

The authors have declared that no competing interests exist.

Funding

This work was partially supported by the research project: Human Aspects of the Software Global Development (HuSO, code E21/253) from the Universidad Nacional de San Juan, Argentine.

Authors' contribution

SZ conceived the idea, labeled the trust dataset, developed the classification python code, analyzed the results and wrote the manuscript. FG extracted Github data, labeled the trust dataset and reviewed the manuscript. GS and ET labeled the trust dataset and RF reviewed work idea and manuscript.

References

[1] I. Aza. “Man as Subject of Internet Communication”. In International Conference Communicative Strategies of Information Society (CSIS 2018) (pp. 383-388). Atlantis Press. 2018.

[2] B. H. Malik, S. Faroom, M. N. Ali, N. Shehzad, S. Yousaf, H. Saleem, K. Khan. “Geographical Distance and Communication Challenges in Global Software Development: A Review”. International Journal of Advance Computer Science and Applications. Vol. 9, N° 5, pp. 406-414. 2018.

[3] M. Alsharo, D. Gregg, R. Ramirez. “Virtual team effectiveness: The role of knowledge sharing and trust”. Information & Management. Vol. 54 N° 4, pp. 479-490. 2017.

[4] S. Morrison-Smith, J. Ruiz. “Challenges and barriers in virtual teams: a literature review”. SN Applied Sciences, 2(6), 1-33. 2020.

[5] L. F. Capretz, F. Ahmed. “Making sense of software development and personality types”. IT professional. Vol. 12, N° 1, pp. 6-13.2010.

[6] A. Cockburn, J. Highsmith. “Agile software development, the people factor”. Computer. Vol. N° 11, pp. 131-133. 2001.

[7] D. Graziotin, X. Wang, P. Abrahamsson. “Happy software developers solve problems better: psychological measurements in empirical software engineering”. Peer J. Vol. 2, e289. 2014.

[8] N. Novielli, F. Calefato, F. Lanubile. “Towards discovering the role of emotions in stack overflow”. Proceedings of the 6th international workshop on social software engineering ACM. pp. 32-36. 2014.

[9] M. Ortu, G. Destefanis, S. Counsell, S. Swift, R. Tonell, M. Marchesi. “How diverse is your team? Investigating gender and nationality diversity in GitHub teams”. Journal of Software Engineering Research and Development. Vol. 5 N°1, pp. 1-18. 2017.

[10] M. Hertzum. “The importance of trust in software engineers' assessment and choice of information sources”. Information and Organization. Vol. 12, N° 1, pp 1-18. 2002.

[11] N. B. Moe, D. Šmite. “Understanding a lack of trust in Global Software Teams: a multiple-case study”. Software Process: Improvement and Practice. Vol. 13, N° 3, pp. 217-231. 2008.

[12] M. Grabowski, K. H. Roberts. “Risk mitigation in virtual organizations”. Organization Science. Vol. 10, N° 6, pp. 704-721. 1999.

[13] P. Kanawattanachai, Y. Yoo. “Dynamic nature of trust in virtual teams”. The Journal of Strategic Information Systems. Vol. 11, N° 3-4, pp. 187-213. 2002.

[14] L. L. Martins, L. L.Gilson, M. T. Maynard. “Virtual teams: What do we know and where do we go from

here?". Journal of management. Vol. 30, N° 6, pp. 805-835. 2004.

[15] S. L. Jarvenpaa, T. R. Shaw, D. S. Staples. "Toward contextualized theories of trust: The role of trust in global virtual teams". Information systems research. Vol. 15, N° 3, pp. 250-267. 2004.

[16] R. C. Mayer, J. H. Davis, F. D. Schoorman. "An integrative model of organizational trust". Academy of management review. Vol. 20, N° 3, pp. 709-734. 1995.

[17] J. D. Lewis, A. Weigert. "Trust as a social reality". Social Forces. Vol 63, N°4, pp. 967-985. 1985.

[18] Clark W. R., Clark L. A. and Crossley K.: "Developing multidimensional trust without touch in virtual teams". Marketing Management Journal, 20(1) (2010) 177-193.

[19] J. Wilson, S. Straus, B. McEvily. "All in due time: The development of trust in computer mediated and face-to-face teams". Organizational Behavior and Human Decision Processes. Vol. 99, N° 1, pp. 16-33. 2006.

[20] D. J. McAllister. "Affect and cognition-based trust as foundations for interpersonal cooperation in organisations". Academy of Management J. Vol. 38, N°1, pp. 25-59. 1995.

[21] M. A, Rothenberger, Y. C. Kao, L. N. Van Wassenhove. "Total quality in software development: An empirical study of quality drivers and benefits in Indian software projects". Information & Management. Vol. 47, N° 7-8, pp. 372-379. 2010.

[22] A. Gopal, M. S. Krishnan, T. Mukhopadhyay, D. R. Goldenson. "Measurement programs in software development: determinants of success". IEEE Transactions on software engineering. Vol. 28, N°9, pp. 863-875. 2002.

[23] F. Q. B. da Silva, R. Prikladnicki, A. C. C. França, C. V. F. Monteiro, C. Costa, R. Rocha. "An evidence-based model of distributed software development project management: results from a systematic mapping study". Journal of Software: Evolution and Process. Vol 24, N° 6, pp. 625–642. 2012.

[24] N. B. Moe, D. S. Cruzes, T. Dybå, E. Engebretsen. "Coaching a Global Agile Virtual Team". Proceedings of the 2015 IEEE 10th International Conference on Global Software Engineering. Vol I, Washington, DC, USA, pp. 33-37. 2015.

[25] D. Thomas, R. Bostrom. "Building Trust and Cooperation Through Technology Adaptation in Virtual Teams: Empirical Field Evidence". Information Systems Management. Vol. 25, N °1, pp. 45-56. 2010.

[26] A. L. McNab, K. A. Basoglu, S. Sarker, Y. Yu. "Evolution of cognitive trust in distributed software development teams: A punctuated equilibrium model". Electronic Markets. Vol. 22, N°1, pp. 21-36. 2012.

[27] F. Calefato, D. Gendarmi, F. Lanubile. "Embedding Social Networking Information into Jazz to Foster Group Awareness within Distributed Teams". Proceedings of the 2nd Int Workshop on Social Software Engineering and Applications. Vol I, pp. 23-28. 2009.

[28] V. Casey. "Developing trust in virtual software development teams". Journal of theoretical and applied electronic commerce research. Vol.5 (2) (2010) 41-58.

[29] R. Ocker. "Enhancing Learning Experiences in Partially Distributed Teams: Training Students to Work Effectively Across Distances". Proceedings 42nd Hawaii International Conference on System Sciences. Vol. I, Washington, DC, USA, pp. 1–10. 2009.

[30] L. Tan, A. Hindle. "Guest Editorial: Special Section on Mining Software Repositories". Empirical Software Engineering. Vol. 24, pp. 1-3. 2019.

[31] Nugroho, K. S., Sukmadewa, A. Y., & Yudistira, N. (2021, September). "Large-scale news classification using bert language model: Spark nlp approach". In 6th International Conference on Sustainable Information Engineering and Technology 2021 (pp. 240-246).

[32] Dai, J., & Chen, C. (2020, August). "Text classification system of academic papers based on hybrid Bert-BiGRU model". In 2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC) (Vol. 2, pp. 40-44). IEEE.

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

[34] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). "Roberta: A robustly optimized bert pretraining approach". arXiv preprint arXiv:1907.11692.

[35] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). "Language models are unsupervised multitask learners". OpenAI blog, 1(8), 9.

[36] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). "Xlnet: Generalized autoregressive pretraining for language understanding". Advances in neural information processing systems, 32.

[37] Arase, Y., & Tsujii, J. (2021). "Transfer fine-tuning of BERT with phrasal paraphrases". Computer Speech & Language, 66, 101164.

[38] Song, H., Wang, Y., Zhang, K., Zhang, W. N., & Liu, T. (2021). "BoB: BERT over BERT for training persona-based dialogue models from limited personalized data". arXiv preprint arXiv:2106.06169.

[39] González-Carvajal, S., & Garrido-Merchán, E. C. (2020). "Comparing BERT against traditional machine learning text classification". arXiv preprint arXiv:2005.13012.

[40] M. Niazi, N. Ikram, M. Bano, S. Imtiaz, S. U. Khan. "Establishing trust in offshore software outsourcing relationships: an exploratory study using a systematic literature review". IET software. Vol. 7, N° 5, pp. 283-293. 2013.

[41] Zapata, S., Barros-Justo, J. L., Matturro, G., & Sepúlveda, S. (2021). "Measurement of interpersonal trust in virtual software teams: A systematic literature review". INGENIARE-Revista Chilena de Ingeniería, 29(4).

[42] F. Q. B. da Silva, R. Prikladnicki, A. C. C. França, C. V. F. Monteiro, C. Costa, R. Rocha. "An evidence-based model of distributed software development project management: results from a systematic mapping study". Journal of Software: Evolution and Process. Vol 24, N° 6, pp. 625–642. 2012.

[43] S. Tyagi, R. Sibal, B. Suri. "Role of trust in distributed agile software development teams- A light

weight systematic literature review". ICTACT Journal on Management Studies. Vol. 4, N° 2, pp. 748-753. 2018.

[44] Uddin, G., Guéhénuc, Y. G., Khomh, F., & Roy, C. K. (2022). "An Empirical Study of the Effectiveness of an Ensemble of Stand-alone Sentiment Detection Tools for Software Engineering Datasets". ACM Transactions on Software Engineering and Methodology (TOSEM), 31(3), 1-38.

[45] Obaidi, M., Nagel, L., Specht, A., & Klünder, J. (2022). "Sentiment analysis tools in software engineering: A systematic mapping study". Information and Software Technology, 107018.

[46] Paul, S., & He, F. (2012, January). "Time pressure, cultural diversity, psychological factors, and information sharing in short duration virtual teams". In 2012 45th Hawaii International Conference on System Sciences (pp. 149-158). IEEE

[47] A. Mitchell, I. Zigurs, Trust in virtual teams: solved or still a mystery? ACM SIGMIS Database 40 (3) (2009) 61–83, http://dx.doi.org/10.1145/1592401.1592407.

[48] Jarvenpaa, S. L., Knoll, K., & Leidner, D. E. (1998). "Is anybody out there? Antecedents of trust in global virtual teams". Journal of management information systems, 14(4), 29-64.

[49]] F.-y. Kuo, C.-p. Yu, An exploratory study of trust dynamics in work-oriented virtual teams, J. Comput.-Mediated Commun. 14 (4) (2009) 823–854, http://dx.doi.org/10.1111/j.1083-6101.2009.01472.x.

[50] Khan, M. S. (2012). Role of trust and relationships in geographically distributed teams: exploratory study on development sector. International Journal of Networking and Virtual Organisations, 10(1), 40-58.

[51] Canete, J., Chaperon, G., Fuentes, R., Ho, J. H., Kang, H., & Pérez, J. (2020). "Spanish pre-trained bert model and evaluation data". Pml4dc at iclr, 2020, 1-10.

[52] Tiedemann, J. (2012, May). "Parallel data, tools and interfaces in OPUS". In Lrec (Vol. 2012, pp. 2214-2218).

[53] Carrino, C. P., Costa-jussà, M. R., & Fonollosa, J. A. (2019). "Automatic spanish translation of the squad dataset for multilingual question answering". arXiv preprint arXiv:1912.05200.

[54] Van Rijsbergen, C. (1979, September). "Information retrieval: theory and practice". In Proceedings of the Joint IBM/University of Newcastle upon Tyne Seminar on Data Base Systems (Vol. 79).

[55] Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019, October). "How to fine-tune bert for text classification?". In China national conference on Chinese computational linguistics (pp. 194-206). Springer, Cham.

[56] Moon, A., & Raju, T. (2013). "A survey on document clustering with similarity measures". International Journal of Advanced Research in Computer Science and Software Engineering, 3(11), 599-601.

[57] Wright, R. E. (1995). "Logistic regression". In L. G. Grimm & P. R. Yarnold (Eds.), Reading and understanding multivariate statistics (pp. 217–244). American Psychological Association.

[58] Radford, A., Wu, J., Amodei, D., Amodei, D., Clark, J., Brundage, M., & Sutskever, I. (2019). Better language models and their implications. OpenAI blog, 1, 2.

Citation: S. Zapata, F. Gallardo, G. Sevilla, E. Torres and R. Forradellas. *Trust evaluation in virtual software development teams using BERT-based language models.* Journal of Computer Science & Technology, vol. 23, no. 1, pp. 45-54, 2023.

DOI: 10.24215/16666038.23.e04

Received: September 26, 2022 **Accepted:** December 22, 2022.