Forum 4.0: An Open-Source User Comment Analysis Framework

Marlo Haering[1] and **Jakob Smedegaard Andersen**[1] and **Chris Biemann**[1]
and **Wiebke Loosen**[2] and **Benjamin Milde**[1] and **Tim Pietz**[1] and **Christian Stoecker**[3]
and **Gregor Wiedemann**[2] and **Olaf Zukunft**[3] and **Walid Maalej**[1]
Universität Hamburg[1], Leibniz Institute for Media Research[2],
Hamburg University of Applied Sciences[3]
{haering,andersen,biemann,milde,5pietz,maalej}
@informatik.uni-hamburg.de
{w.loosen,g.wiedemann}@leibniz-hbi.de
{christian.stoecker,olaf.zukunft}@haw-hamburg.de

Abstract

With the increasing number of user comments in diverse domains, including comments on online journalism and e-commerce websites, the manual content analysis of these comments becomes time-consuming and challenging. However, research showed that user comments contain useful information for different domain experts, which is thus worth finding and utilizing. This paper introduces Forum 4.0, an open-source framework to semi-automatically analyze, aggregate, and visualize user comments based on labels defined by domain experts. We demonstrate the applicability of Forum 4.0 with comments analytics scenarios within the domains of online journalism and app stores. We outline the underlying container architecture, including the web-based user interface, the machine learning component, and the task manager for time-consuming tasks. We finally conduct machine learning experiments with simulated annotations and different sampling strategies on existing datasets from both domains to evaluate Forum 4.0's performance. Forum 4.0 achieves promising classification results (ROC-AUC $\geq$ 0.9 with 100 annotated samples), utilizing transformer-based embeddings with a lightweight logistic regression model. We explain how Forum 4.0's architecture is applicable for millions of user comments in real-time, yet at feasible training and classification costs.

1 Introduction

Comment sections are omnipresent in today's online environments, for example, on news websites, blogs, online shops, or app stores. In these sections, users submit their feedback and opinion, request features and information, or report issues and bugs. Also, in social media such as Twitter or Facebook, users regularly comment on specific topics, events, products, or services. In many domains, including e-commerce and journalism, users discuss with each other, read others' opinions to e.g. assess the quality of the service or the product (Springer et al., 2015; Kümpel and Springer, 2016), and provide feedback to other users and other domain experts like the journalist (Häring et al., 2018), who wrote the article or the developer who created the app (Maalej et al., 2016b).

Even though research has criticized phenomena such as "dark participation" (Frischlich et al., 2019), comments can contain constructive information for different domain experts in different fields (Loosen et al., 2018; Maalej et al., 2016a). For example, in app development, vendors use app reviews in app stores to collect new feature ideas, bug reports, or ideas of additional user scenarios for their app (Stanik et al., 2019). Software vendors consider the reviews to decide which bug or feature request to prioritize in the next development cycle (Martens and Maalej, 2019). In online journalism, media outlets harness user comments to acquire a broader perspective on additional arguments, collect resonance about their articles, or identify and contact experts or persons concerned for follow-up stories (Loosen et al., 2018). However, the quality of the comments varies significantly, and their amount is sometimes overwhelming, which makes manual monitoring and analysis a real challenge (Pagano and Maalej, 2013; Park et al., 2016a).

In this work, we propose Forum 4.0, an open-source user comment analysis framework to semi-automatically analyze a large number of user comments for domain experts from various domains. Forum 4.0 leverages a combination of transfer learning (Howard and Ruder, 2018), human-in-the-loop (Bailey and Chopra, 2018), and active learning (Settles, 2012) strategies to automatically analyze the comments' content. To enable replication and further research, we share Forum 4.0's source code,

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 63–70
April 19 - 23, 2021.

the scripts, and datasets we used for our research[1] and a video, which showcases Forum 4.0[2].

2 Usage of Forum 4.0

We describe exemplary usage scenarios of Forum 4.0 for journalists and product managers in their respective online journalism and app development domains and introduce Forum 4.0's user interface.

2.1 Online Journalism

The manual effort for comment moderation in online journalism is high (Park et al., 2016b). One the one hand, media outlets filter hate speech (Gao and Huang, 2017), as it might negatively affect their credibility (Naab et al., 2020). On the other hand, user comments can also be useful for different journalistic purposes (Diakopoulos, 2015). For example, journalists can obtain new perspectives and opinions on an article, learn from users' described personal experiences, or identify potential interview partners among the commenting users (Loosen et al., 2018). Journalists can also aggregate user comments to identify and visualize their audience's opinion on current news topics (Wang et al., 2013). Users can also point out errors in reporting, contribute additional or missing sources and information, provide new ideas for further news, or even address the editorial team or authors directly, for example, by criticizing the article's quality (Häring et al., 2018).

Journalists first define a useful user comment label in Forum 4.0. Examples for such labels could be: "criticism towards corona measures," or "pros/cons regarding a legislative proposal". Journalists or forum moderators annotate user comments regarding these labels, gradually increasing the number of training samples. Forum 4.0 trains a machine learning model using the annotated comments and classifies all other user comments. The automatic classification will improve with more annotations until it reaches sufficient precision so that journalists can conduct quantitative and qualitative analyses with the comments.

2.2 App Development

In app stores, product managers utilize user comments for multiple purposes: users report crashes and bugs in app reviews with valuable context information (e.g., device or app version), helping developers identifying and fixing them (Pagano and Maalej, 2013). This is particularly helpful to acquire immediate feedback after a new major release or update (Guzman and Maalej, 2014). Additionally, users suggest desired and useful app feature ideas (Maalej et al., 2016a). Thereby, the product managers get an overview of current app issues, which they can consider for their further development. In the field of mobile learning, the product manager can utilize comments for the automatic evaluation of education apps (Haering et al., 2021).

Similar to the online journalism domain, the product manager can use Forum 4.0 to first create labels for constructive app reviews. In the app development domain, useful labels include "problems since the last app update", "positive/negative feedback on a certain app feature", or "missing or requested features". The domain expert further annotates app reviews, compiling a training set. Forum 4.0 trains a model and classifies the other app reviews for the domain expert to analyze.

2.3 User Interface

Figure 1 shows Forum 4.0's user interface. The domain expert can log in to create a new label or annotate user comments. Below the title bar, the expert can select a data source containing the comments to analyze. In the figure, we selected comments from the Austrian newspaper DER STANDARD[3], which contains the comments of the "One Million Posts Corpus" published by Schabus et al. (2017). Next to the data source selector, the domain expert can create a new label or select relevant existing labels to analyze and annotate the comments.

The pie chart shows the comment distribution among the document categories (news article or app categories). The bar chart shows the number of positive classifications for the selected labels over time with different granularity options. We train one classification model for each label and show the accuracy and the development of the F1-scores with an increasing number of training samples.

The lower part of the Forum 4.0 interface lists the actual user comments for exploration and annotation. With a full-text search, the domain expert can further filter the comment results. The list contains the comment text, the timestamp, and a column for each selected label. Each label column

[1] https://forum40.informatik.uni-hamburg.de/git/

[2] https://forum40.informatik.uni-hamburg.de/demo.mp4

[3] https://www.derstandard.at/

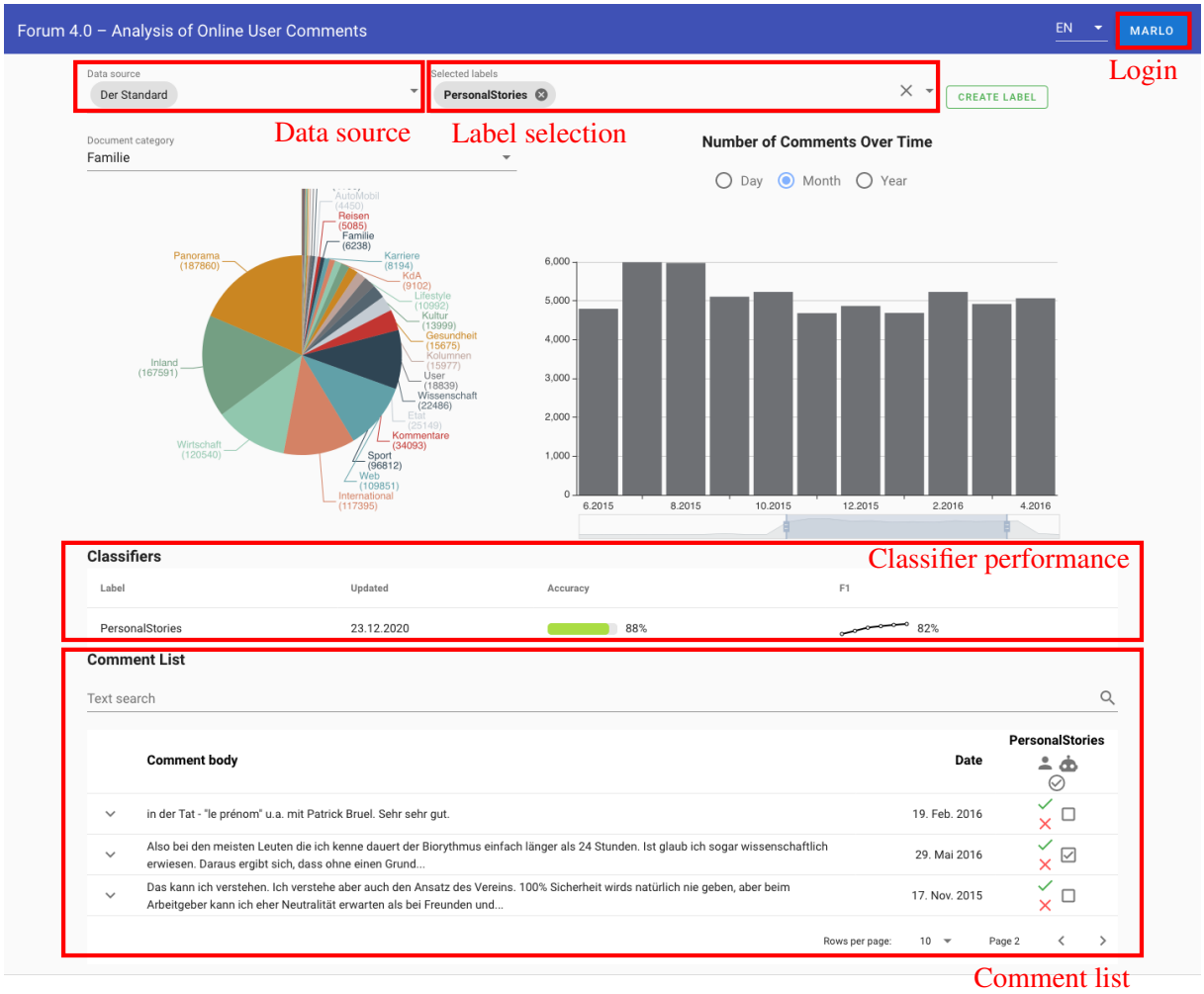


Figure 1: Main user interface of Forum 4.0.

has two sub-columns. The first sub-column with the person symbol shows either existing human annotations when logged out or the own annotations when logged in. A logged-in user can correct the automatic classification or annotate comments as a positive or negative sample for the selected labels. The second sub-column with the robot icon shows binary labels and confidence scores. The domain expert has three sorting options for the classifications: (1) positives first, (2) negatives first, (3) uncertain first (circle with tick mark). Forum 4.0 supports finding positive samples for rare comment labels by suggesting semantically similar user comments. Thereby, Forum 4.0 employs the rapid annotation approach to quickly retrieve additional positive samples for a specific comment label.

3 Architecture

We describe Forum 4.0's container-based architecture and its machine learning pipelines.

3.1 Container-based Architecture

Forum 4.0 is composed of containers, interacting with each other via a restful API. Figure 2 outlines a UML deployment diagram.

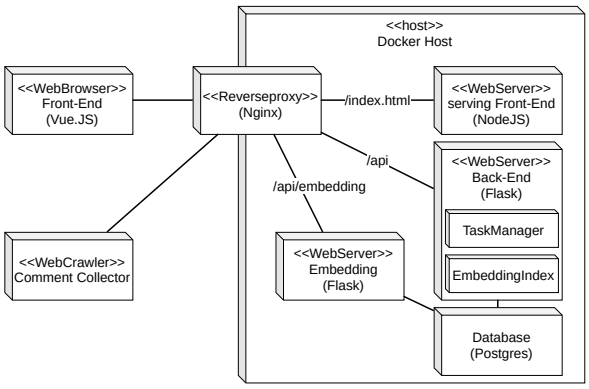


Figure 2: Forum 4.0's container architecture.

The *Comment Collector* aggregates user comments from various sources, including media sites, app stores, and social media. Forum 4.0 currently

contains the "One Million Posts Corpus" and imports comments from the Google Play store and the German news site SPIEGEL Online[4].

The client accessing Forum 4.0's web page requests the *Reverse Proxy*, which forwards the requests depending on the URL path to the responsible container. The first request loads the single page application (Flanagan and Like, 2006) from the *Front-End* web server, which further communicates via a restful API with the Back-End container.

The containers on the Docker host are only accessible from the outside through the reverse proxy for security. The *Back-End* provides the restful API. It invokes all machine learning, NLP, and embedding tasks via a task manager in isolated processes as they are time-consuming and would exceed the HTTP request time out. It further calculates the comment embedding index and queries the database. The *Embedding Container* calculates the embeddings for newly imported user comments. This container can also run on a dedicated host to calculate the embeddings with GPU support.

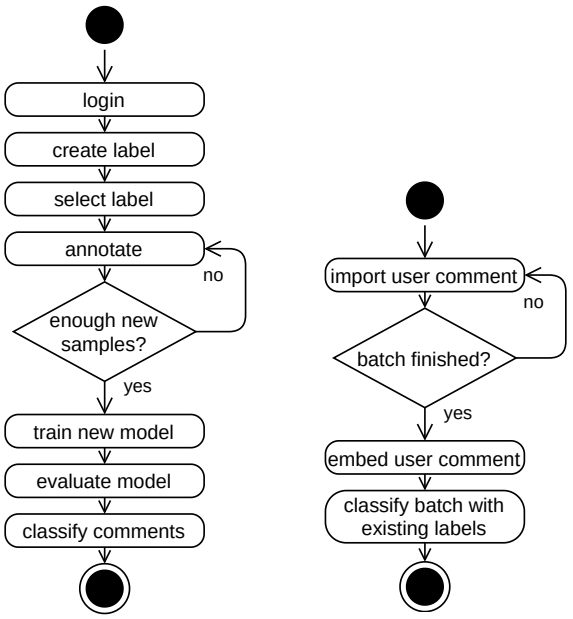
After login, the *Back-End* issues a JSON web token (Jánoky et al., 2018) for the *Front-End*. All sensitive API endpoints of the *Back-End* are protected and require a valid JSON web token in the request's body. Protected actions include the comment and document import, the creation of new labels, and posting annotations.

3.2 Machine Learning Pipelines

Two essential parts of the architecture are the Model Training Pipeline (Figure 3a) and the Comment Import Pipeline (Figure 3b).

The Model Training Pipeline applies supervised machine learning, and active learning strategies (Settles, 2012) to improve the comment classification continuously. To define a label and train a model for the automatic classification, the domain expert must first log in and create a new label. Domain experts can select the new label from the menu and start annotating samples. The domain expert is the human in the loop (Bailey and Chopra, 2018), who annotates and enlarges the training set to improve the automatic classification iteratively.

Annotators can sort the user comments according to the uncertainty score to keep the annotation process most rewarding (Andersen et al., 2020). Forum 4.0 uses the label probability as the uncertainty value. Uncertain instances are those whose classification is the least confident, i.e. $P(c|d) \approx 0.5$ for comment d belonging to class c.



(a) The Model Training Pipeline.

(b) The Comment Import Pipeline.

Figure 3: Machine Learning Pipelines

Forum 4.0 provides rapid annotation techniques to support and accelerate the collection of training samples. Forum 4.0 lists semantically similar comments to an existing comment based on the similarity of the comment embeddings. In case the annotator found a positive training example, chances are higher that semantically similar user comments are also positive user comments, which the annotator can quickly check.

We can adjust the number of required new training samples, which trigger the training of a new model. After each annotation, Forum 4.0 checks whether enough new training samples are available to invoke (re-)training of the model. The task manager executes each model training as a dedicated process, logs its training, and records the evaluation results. Forum 4.0 evaluates each model using ten-fold cross-validation (Stone, 1974) to determine the classification performance. The newly trained model classifies all other user comments, which are not part of the training set, and Forum 4.0 persists its classification scores for that label.

The Data Import Pipeline enables the import and processing of new user comments. After importing a new user comment batch, the task manager triggers the embedding process, which calculates the embeddings for the imported user comments.

[4] https://spiegel.de/

Forum 4.0 employs transfer learning (Howard and Ruder, 2018) by using the embeddings of well-established pre-trained language models, for example, BERT embeddings (Devlin et al., 2019), as machine learning features for the classification model. Subsequently, all existing models classify the new user comment batch.

4 Machine Learning Experiments

To preliminary evaluate the applicability of Forum 4.0 and the performance of its machine learning models, we conducted experiments with comments from news sites and app stores. For the online journalism domain, we used the One Million Post (OMP) corpus (Schabus et al., 2017). It consists of ∼1M German user comments submitted to the Austrian newspaper DER STANDARD, partly annotated by forum moderators. For the app store domain, we used an existing annotated app review dataset (ARD) (Stanik et al., 2019).

We used 9,336 annotated German comments (1,625 positives and 7,711 negatives) regarding OMP's "personal story" label. These user comments share the users' personal stories regarding the respective topic, including experiences and anecdotes. We used 6,406 annotated English app reviews (1,437 positives and 4,969 negatives) regarding the ARD's "bug report" label. In bug reports, users describe problems with the app that should be fixed, such as a crash, an erroneous behavior, or a performance issue.

We simulated the human annotator, who gradually annotates a batch of user comments, triggering a new training and evaluation cycle. We trained the classifier on the training set and evaluated the model on the remaining comments. We started our first training with ten samples and triggered new training for every ten new annotations.

Forum 4.0 allows random sampling and uncertainty sampling for new annotations, which we compared in our experiments. With random sampling, we randomly chose and added ten new samples to our training set. With uncertainty sampling, we added the user comments for which the classifier's output is closest to 0.5. We stopped adding more user comments to the training set as soon as the balanced accuracy score converged.

We evaluated the classification model on the remaining user comments after each training, using the *balanced accuracy*, *F1-score*, and the Receiver Operating Characteristics (*ROC-AUC*) metrics.

For the comment embeddings, we used two different multi-lingual pre-trained language models to embed the comments: (1) BERT (Devlin et al., 2019) is based on a transformer architecture, which learns contextual relations between sub-(word) units in a text. We used an average token embedding of the four last layers of the BERT model as the comment embeddings. (2) Sentence-BERT (S-BERT) (Reimers and Gurevych, 2019) is based on a modification of the BERT network and infers semantically meaningful sentence embeddings. We used a lightweight logistic regression model as a classifier due to performance requirements for quick updates of machine labels during human-in-the-loop coding. To assess the feasibility of our architecture, we further timed the model's training and evaluation. To mitigate the noise of our results, we performed 50 rounds for each experiment. The line plots show the average results of all rounds and the standard deviation.

5 Experiments Results

Figure 4 shows the balanced accuracy, ROC-AUC, and F1-scores for all our classification experiments. Overall, all classification metrics improve with increasing training data. Additionally, the uncertainty sampling strategy outperforms random sampling, and the S-BERT embeddings outperform the BERT embeddings given the same sampling strategy. All evaluation metrics significantly improve within the first 100 training samples and converge afterward.

On the OMP dataset, we achieved a balanced accuracy of 0.86 with 100 training samples using uncertainty sampling and S-BERT embeddings. With 500 training samples, we reached 0.91. Within the first 100 training samples, S-BERT embeddings outperformed the BERT embeddings. We achieved a similar F1-score as Schabus et al. (2017) with ∼50 training samples (0.70) and outperformed their model using 500 training samples with an F1-score of 0.82. On the app review dataset, we achieved a balanced accuracy of 0.92, a ROC-AUC of 0.96, and an F1-score of 0.85 using 500 training samples.

Figure 5 shows the time measurements for training the logistic regression model. In all cases, the training size has a linear increase. Overall, the training time with the S-BERT embeddings (0.1s for 500 samples) takes a shorter time than training with the BERT embeddings (0.4s for 500 samples) on both datasets. We also measured the classification time on the remaining test set, which takes less

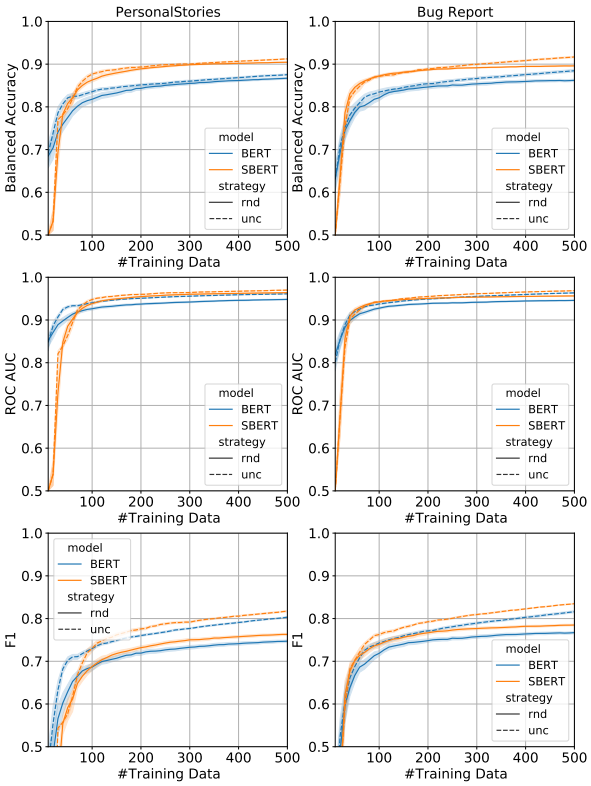


Figure 4: Balanced accuracy (top), ROC-AUC (center), and F1-scores (bottom) for all classification experiments on the OMP (left column) and the ARD (right column).

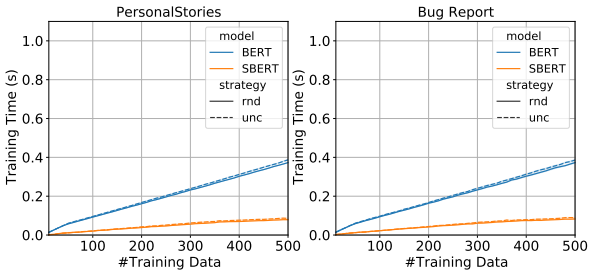


Figure 5: Training time of the logistic regression model.

than ∼3ms on the OMP (∼8,000 test samples) and the ARD (∼6,000 test samples) dataset.

6 Related Work

Previous work in the app development domain automatically analyzed comments on apps including, app reviews (Guzman and Maalej, 2014; Dhinakaran et al., 2018; Harman et al., 2012) and tweets (Guzman et al., 2016; Williams and Mahmoud, 2017), to understand and summarize users' needs and support development decisions (Stanik and Maalej, 2019). A typical analysis goal is to reduce the noisy user feedback and classify the remaining ones into bug reports, feature requests, and experience reports (Maalej et al., 2016a).

Similarly, in online journalism, previous work aimed to reduce noise and hate speech (Gao and Huang, 2017), identify high-quality contributions (Park et al., 2016a; Diakopoulos, 2015; Wang and Diakopoulos, 2021), summarize the audiences' sentiment (Wang et al., 2013), or identify comments, which address journalistic aspects (Häring et al., 2018). Park et al. (2018) and Fast et al. (2016) developed a prototype, which supports the analysis of documents and comments regarding a custom concept based on seed terms.

Forum 4.0 builds upon this previous work and features a domain-independent comment analysis framework for domain experts. Domain experts can create or reuse useful labels, annotate user comments regarding these labels, and train machine-learning models, which automatically classify the comments for further utilization.

7 Conclusion

We presented Forum 4.0, an open-source framework to semi-automatically analyze user comments in various domains including, online journalism and app store. Domain experts can flexibly define or reuse comment analysis dimensions as classification labels in our framework. Forum 4.0's architecture leverages state-of-the-art semantic text embeddings with a lightweight logistic regression model to address the labeling flexibility and the scalability requirements for an application to millions of user comments. Forum 4.0 starts a new model training after the domain expert annotated additional comments for the concerned label. Forum 4.0 evaluates each new model and classifies the remaining user comments for further analysis.

We achieved promising results with our machine learning experiments in both domains with different semantic embedding and sampling strategies already after n $\geq$ 100 annotations with a low training time ($t = 0.1s$). Our evaluation suggests that Forum 4.0 can also be applied at a larger scale with millions of user comments.

Acknowledgments

This work is partly funded by the Hamburg's *ahoi.digital* program in the Forum 4.0 project.

References

Jakob Smedegaard Andersen, Tom Schöner, and Walid Maalej. 2020. Word-Level Uncertainty Estimation for Black-Box Text Classifiers using RNNs. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5541–5546, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Katherine Bailey and Sunny Chopra. 2018. Few-shot text classification with pre-trained word embeddings and a human in the loop. *arXiv preprint arXiv:1804.02063*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, MN, USA. Association for Computational Linguistics.

Venkatesh T. Dhinakaran, Raseshwari Pulle, Nirav Ajmeri, and Pradeep K. Murukannaiah. 2018. App Review Analysis Via Active Learning: Reducing Supervision Effort without Compromising Classification Accuracy. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pages 170–181, Banff, AB. IEEE.

Nicholas Diakopoulos. 2015. Picking the NYT picks: Editorial criteria and automation in the curation of online news comments. *#ISOJ*, page 147.

Ethan Fast, Binbin Chen, and Michael S. Bernstein. 2016. Empath: Understanding topic signals in large-scale text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4647–4657, San Jose, CA, USA. ACM.

David Flanagan and Will Sell Like. 2006. Javascript: The definitive guide, 5th.

Lena Frischlich, Svenja Boberg, and Thorsten Quandt. 2019. Comment Sections as Targets of Dark Participation? Journalists' Evaluation and Moderation of Deviant User Comments. *Journalism Studies*, 20(14):2014–2033.

Lei Gao and Ruihong Huang. 2017. Detecting online hate speech using context aware models. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 260–266, Varna, Bulgaria. INCOMA Ltd.

Emitza Guzman, Rana Alkadhi, and Norbert Seyff. 2016. A Needle in a Haystack: What Do Twitter Users Say about Software? In *2016 IEEE 24th International Requirements Engineering Conference (RE)*, pages 96–105, Beijing, China.

Emitza Guzman and Walid Maalej. 2014. How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews. In *2014 IEEE 22nd Int. Requirements Engineering Conf. (RE)*, pages 153–162, Karlskrona, Sweden.

Marlo Haering, Muneera Bano, Didar Zowghi, Matthew Kearney, and Walid Maalej. 2021. Automating the evaluation of education apps with app store data. *IEEE Transactions on Learning Technologies (TLT)*, pages 1–12.

Marlo Häring, Wiebke Loosen, and Walid Maalej. 2018. Who is Addressed in This Comment?: Automatically Classifying Meta-Comments in News Comments. *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW):67:1–67:20.

Mark Harman, Yue Jia, and Yuanyuan Zhang. 2012. App Store Mining and Analysis: MSR for App Stores. In *Proceedings of the 9th IEEE Working Conference on Mining Software Repositories*, MSR '12, pages 108–111, Piscataway, NJ, USA. IEEE, IEEE Press.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

László Viktor Jánoky, János Levendovszky, and Péter Ekler. 2018. An analysis on the revoking mechanisms for JSON Web Tokens. *International Journal of Distributed Sensor Networks*, 14(9):155014771880153.

Anna Sophie Kümpel and Nina Springer. 2016. Qualität kommentieren. Die Wirkung von Nutzerkommentaren auf die Wahrnehmung journalistischer Qualität. *Studies in Communication — Media*, 5(3):353–366.

Wiebke Loosen, Marlo Häring, Zijad Kurtanović, Lisa Merten, Julius Reimer, Lies van Roessel, and Walid Maalej. 2018. Making sense of user comments: Identifying journalists' requirements for a comment analysis framework. *SCM Studies in Communication and Media*, 6(4):333–364.

Walid Maalej, Zijad Kurtanović, Hadeer Nabil, and Christoph Stanik. 2016a. On the automatic classification of app reviews. *Requirements Engineering*, 21(3):311–331.

Walid Maalej, Maleknaz Nayebi, Timo Johann, and Guenther Ruhe. 2016b. Toward data-driven requirements engineering. *IEEE Software*, 33(1):48–54.

Daniel Martens and Walid Maalej. 2019. Release early, release often, and watch your users' emotions: Lessons from emotional patterns. *IEEE Software*, 36(5):32–37.

Teresa K. Naab, Dominique Heinbach, Marc Ziegele, and Marie-Theres Grasberger. 2020. Comments and Credibility: How Critical User Comments Decrease Perceived News Article Credibility. *Journalism Studies*, 21(6):783–801.

Dennis Pagano and Walid Maalej. 2013. User feedback in the appstore: An empirical study. In *2013 21st IEEE International Requirements Engineering Conference (RE)*, pages 125–134, Rio de Janeiro, Brasil.

Deokgun Park, Seungyeon Kim, Jurim Lee, Jaegul Choo, Nicholas Diakopoulos, and Niklas Elmqvist. 2018. ConceptVector: Text visual analytics via interactive lexicon building using word embedding. *IEEE transactions on visualization and computer graphics*, 24(1):361–370.

Deokgun Park, Simranjit Sachar, Nicholas Diakopoulos, and Niklas Elmqvist. 2016a. Supporting comment moderators in identifying high quality online news comments. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 1114–1125, New York, NY, USA. Association for Computing Machinery.

Deokgun Park, Simranjit Sachar, Nicholas Diakopoulos, and Niklas Elmqvist. 2016b. Supporting comment moderators in identifying high quality online news comments. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1114–1125, San Jose, CA, USA. ACM.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Dietmar Schabus, Marcin Skowron, and Martin Trapp. 2017. One million posts: A data set of German online discussions. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1241–1244, Miyazaki, Japan. ACM Press.

Burr Settles. 2012. *Active Learning*. Morgan & Claypool Publishers.

Nina Springer, Ines Engelmann, and Christian Pfaffinger. 2015. User comments: Motives and inhibitors to write and read. *Information, Communication & Society*, 18(7):798–815.

Christoph Stanik, Marlo Haering, and Walid Maalej. 2019. Classifying multilingual user feedback using traditional machine learning and deep learning. In *IEEE 27th International Requirements Engineering Conference Workshops*, pages 220–226, Jeju Island, South Korea.

Christoph Stanik and Walid Maalej. 2019. Requirements intelligence with OpenReq analytics. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 482–483, Jeju Island, South Korea.

Mervyn Stone. 1974. Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133.

Changbo Wang, Zhao Xiao, Yuhua Liu, Yanru Xu, Aoying Zhou, and Kang Zhang. 2013. SentiView: Sentiment Analysis and Visualization for Internet Popular Topics. *IEEE Transactions on Human-Machine Systems*, 43(6):620–630.

Yixue Wang and Nicholas Diakopoulos. 2021. The Role of New York Times Picks in Comment Quality and Engagement. In *Hawaii International Conference on System Sciences*, page to appear, Hawaii.

Grant Williams and Anas Mahmoud. 2017. Mining Twitter feeds for software user requirements. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 1–10, Lisbon, Portugal.