

“© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Automating the Evaluation of Education Apps with App Store Data

Journal:	<i>Transactions on Learning Technologies</i>
Manuscript ID	TLT-2019-08-0249.R2
Manuscript Type:	Regular paper
Keywords:	I.2.6.g Machine learning < I.2.6 Learning < I.2 Artificial Intelligence < I Computing Methodologies, I.2.7 Natural Language Processing < I.2 Artificial Intelligence < I Computing Methodologies, mobile learning pedagogies, manual content analysis, app store analytics

SCHOLARONE™
Manuscripts

Automating the Evaluation of Education Apps with App Store Data

Marlo Haering, Muneera Bano, Didar Zowghi, *Member, IEEE*, Matthew Kearney, and Walid Maalej

Abstract—With the vast number of apps and the complexity of their features, it is becoming challenging for teachers to select a suitable learning app for their courses. Several evaluation frameworks have been proposed in the literature to assist teachers with this selection. The iPAC framework is a well-established mobile learning framework highlighting the learners’ experience of personalization, authenticity, and collaboration (iPAC). In this paper, we introduce an approach to automate the identification and comparison of iPAC relevant apps. We experiment with natural language processing and machine learning techniques, using data from the app description and app reviews publicly available in app stores. We further empirically validate the keyword base of the iPAC framework based on the app users’ language in app reviews. Our approach automatically identifies iPAC relevant apps with promising results (F1 score ~72%) and evaluates them similarly as domain experts (spearman’s rank correlation 0.54). We discuss how our findings can be useful for teachers, students, and app vendors.

Index Terms—Mobile learning, app store analytics, supervised machine learning, natural language processing.

I. INTRODUCTION

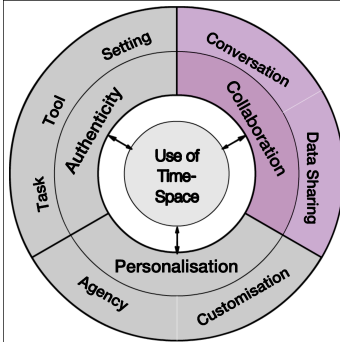
EDUCATIONAL mobile apps offer innovative opportunities for teachers to improve students’ learning [1], [2]. Over the last decade, researchers have investigated the effectiveness of apps in various education domains [3], [4] and proposed new pedagogical frameworks for mobile learning [5], [6]. However, with the vast, continuously increasing number of available apps [7], choosing “the right app” is becoming more and more difficult and time-consuming for teachers and students [8]. One particular challenge is to efficiently select an app that appropriately supports the desired learning activities, assessment strategies, and pedagogical preferences [9].

Teachers may use one of many existing digital frameworks to evaluate educational apps. However, most of them require a manual evaluation regarding different characteristics, which is time-consuming [10], [11]. Numerous frameworks have been proposed in the literature, ranging from complex multi-level

models (e.g., [12]) to smaller frameworks that often omit important socio-cultural characteristics of mobile learning. Common themes include interactivity, control, communication, mobility of learners, and portability of mobile learning devices. The theoretical underpinning for this study is a robust and validated mobile pedagogical framework called iPAC [13], [14]. Grounded in socio-cultural theory [15], it focuses on three distinctive mobile learning evaluation dimensions: *personalization (P)*, *authenticity (A)*, and *collaboration (C)*.

The personalization dimension comprises the sub-dimensions “agency” and “customization.” A high personalization level provides learners with an enhanced degree of agency [16], and the flexibility to customize tools and activities. The authenticity dimension enables in-situ and participatory learning [17], consisting of the sub-dimensions “task,” “tool,” and “setting.” These focus on learners’ involvement in rich, contextualized tasks, realistically making use of tools, and driven by relevant real-life practices and processes [18]. The collaboration dimension captures the conversational, networked features of mobile learning, consisting of the sub-dimensions “conversation” and “data sharing.” Learners engage in joint work, interacting and networking with peers, experts, and the environment [19].

The *iPAC rubric* is a publicly available evaluation scheme for domain experts to evaluate to which degree an app addresses the iPAC dimensions [20]. The rubric consists of three closed questions for each iPAC dimension. A domain expert should answer these closed questions on a scale from one to three. Fig. 1 shows, for instance, the questions for the collaboration dimension. Additionally, Kearney *et al.* [13] collected ~100 *iPAC keywords* based on academic literature in educational research to describe the iPAC dimensions. We refer to these keywords as *iPAC keywords*.



	3	2	1
The features of this app are likely to support:	Learners talking with peers online	Limited, online peer discussion	No online peer discussion
The features of this app are likely to support:	Learners working together to create/modify digital content	Limited opportunities for learners to work together to create/modify content	No creation/modification of content together
The features of this app are likely to support:	Learners sharing/exchanging digital content online	Limited opportunities for learners to share/exchange digital content online	No opportunities for learners to share/exchange digital content

Fig. 1. The evaluation questions for the iPAC collaboration dimension [21].

Manuscript received August 29, 2019; revised May 31, 2020; accepted XX XX, XXXX. Date of publication XX XX, XXXX; date of current version May 31, 2020. This work was supported by BWFG Hamburg within the “Forum 4.0” project as part of the ahol.digital funding line.

M. Haering is with the Department of Informatics, University of Hamburg, Germany, (e-mail: haering@informatik.uni-hamburg.de).

M. Bano is with the School of IT, Deakin University, Melbourne, Australia, (e-mail: muneera.bano@deakin.edu.au).

D. Zowghi is with the Faculty of Engineering and IT, University of Technology Sydney, Australia, (e-mail: didar.zowghi@uts.edu.au).

M. Kearney is with the Faculty of Arts and Social Sciences, University of Technology Sydney, Australia, (e-mail: matthew.kearney@uts.edu.au).

W. Maalej is with the Department of Informatics, University of Hamburg, Germany, (e-mail: maalej@informatik.uni-hamburg.de).

Digital Object Identifier XX.XXXX/TLT.XXXX.XXXXXXX

The iPAC framework has recently been used to inform research on mobile learning in school education [22], teacher education [23], [24], indigenous education [25], and other areas of higher education [26]. For example, Viberg and Grönlund [27] used the framework to develop a survey for eliciting students' attitudes toward mobile technology use for foreign language learning in higher education. It is known as a robust framework [28] which is relevant to teachers and education technology researchers [14].

The overarching aim of this study is to support teachers to navigate through the vast selection of education apps by automatically identifying and comparing iPAC-relevant apps. We mine and analyze the app descriptions (written by vendors) and app reviews (written by users) from the Google Play store. App descriptions summarize the characteristics and functionality of the app, including its pedagogical and technical features. Fig. 2 shows an example of the app "Shotclasses" and highlights the iPAC-relevant parts in the app description. In this study, we call this example an *iPAC-based app* as it addresses at least one of the iPAC dimensions (i.e., personalization, authenticity, or collaboration).

Further, app users write app reviews, which consist of free-text comments and star ratings to express their opinions and feedback on an app and its features. The popularity of this feedback mechanism has continuously increased over the past years. Popular apps might get hundreds or even thousands of reviews per day, which makes it laborious and time-consuming to analyze every review manually [29], [30]. Many users read app reviews before deciding to use a specific app [31]. Vendors also might monitor the satisfaction of their customers by analyzing the reviews [32], [33]. Finally, app developers can identify bug reports and feature requests from the reviews [34]. Similar to an iPAC-based app, in this study, an *iPAC-based review* addresses at least one of the iPAC dimensions.

The contribution of this paper is threefold: First, we use a semi-automated approach to extend the literature-based iPAC keyword set [8] with users' vocabulary found in app reviews. Second, we introduce an approach to automatically identify and compare iPAC-based apps by using a combination of the app descriptions and review texts. Third, we discuss how our approach could facilitate the search and navigation for education apps for stakeholders, including teachers, students, and app vendors. Additionally, we share our code and dataset to enable replication [35].

The remainder of the paper is structured as follows. In Section II, we discuss the related work. Section III introduces the research questions, method, and data. Section IV describes in detail how we conducted each of our research phases. In Section V, we summarize the results of our experiments and answer the research questions. We discuss implications, future directions, and threats to validity in Section VI, and conclude the paper in Section VII.

II. RELATED WORK

We focus the related work discussion on the areas of automation in mobile learning and app feedback analytics.

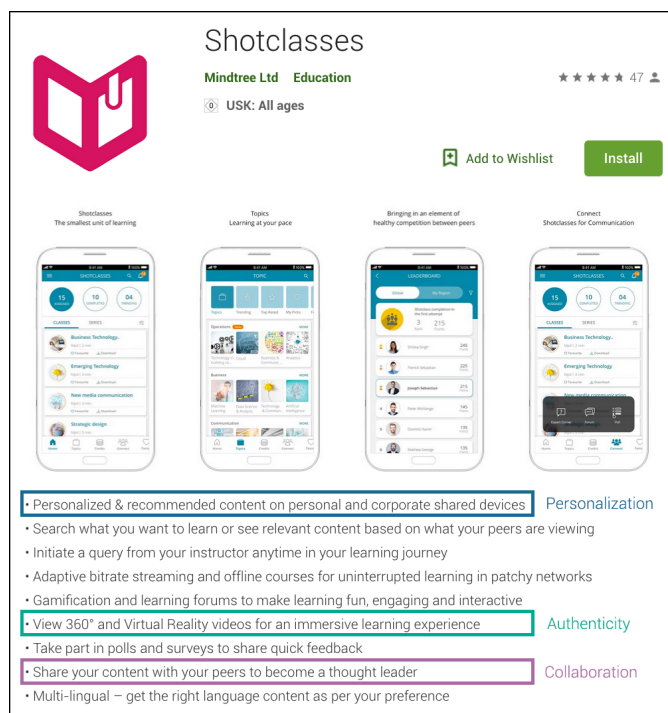


Fig. 2. App description excerpt of the app "Shotclasses." iPAC-relevant parts: personalization (top), authenticity (middle), collaboration (bottom).

A. Automation in Mobile Learning

Over the last decade, research has shown that mobile learning is a promising approach to improve learning effectiveness and experience. For instance, various studies have examined and demonstrated the positive effect of using mobile technology and mobile apps on the students' learning outcomes [6], [36], [37]. Some preliminary studies have particularly focused on automating different educational tasks in this setting. Ortega-García *et al.* [38] proposed a tool to facilitate the creation of custom educational apps for teachers. Ruipérez-Valiente *et al.* [39] used machine learning to detect cheating students. Sun *et al.* [40] also applied deep neural networks to automatically tag test questions for evaluating students' ability.

A major challenge for mobile learning, according to Papadakis *et al.* [41] is to find suitable high-quality educational apps. Several frameworks have thus been suggested to evaluate mobile apps for educational purposes. For instance, Kay [10] proposed a framework that guides teachers in selecting mathematics apps from a vast number of available options based on their types and characteristics. We focus on the iPAC framework [10], [11] since it is not limited to a specific learning area, has been validated in several studies, and is supported with a set of keywords and questions which ease a systematic evaluation. With the exponential growth in the number of apps, a manual evaluation is becoming increasingly challenging.

Our study addresses this challenge by introducing an approach to find apps, which meet selected pedagogical characteristics and evaluate them automatically. We build on the preliminary experiments by Bano *et al.* [8] in which the authors explored the utility of app reviews for evaluating

the pedagogical affordances of education apps by conducting feature-based sentiment analysis on the iPAC dimensions. Reviews on mobile apps provide a rich source of information on the usefulness of these apps. Our work is also inspired by recent advances in collecting and automatically analyzing app reviews for evaluating and gaining insights about the apps' characteristics and features [30], [42]. We are not aware of any published research in this area, focusing on educational apps and pedagogical evaluation.

B. App Feedback Analytics

With the rise of app stores and social media, users can nowadays rate and write their experience with a software product after using it [29]. Such feedback is particularly essential to software designers and app vendors, as it often contains valuable information, including bug reports and feature requests [43].

App feedback analytics is an increasingly popular research topic, which focuses on collecting and automatically analyzing app reviews [32], [44], [45], tweets [46], [47], or product reviews (such as Amazon reviews) [48], [49] to understand and summarize users' needs and inform engineering decisions. A common analysis goal is to automatically filter non-informative reviews and classify the remaining ones into bug reports, feature requests, and experience reports [44]. Further studies [48], [49] focused on extracting and summarizing the rationale, which is the reasoning and justification of user decisions, opinions, and beliefs. A few other studies went one step further to link the reviews to the product features described by the vendors on the app pages [50].

In this study, we leverage the users' language from the app reviews to find significant terms to describe mobile learning characteristics. We apply well-established analysis methods from previous studies in the field of app feedback analytics and introduce an automatic approach to evaluate and assess educational apps with respect to the iPAC dimensions personalization, authenticity, and collaboration.

III. RESEARCH DESIGN

A. Research Questions

The overall goal of this research is to explore how the evaluation of education apps can be automated based on publicly available data. We target the fairly established evaluation framework iPAC and focus on the following research questions:

- **RQ1** What is the level of accuracy in automatically identifying iPAC-based apps with app store data?
- **RQ2** What is the level of accuracy in automatically comparing education apps regarding their iPAC evaluation?
- **RQ3** Which of the iPAC characteristics can we observe in iPAC-based apps and their associated reviews?

The app description outlines the app vendor's perspective on the app. It typically contains a list of the major app features and user scenarios. Additionally, users give feedback on their app experience in app reviews. With RQ1, we aim to use the vendors' and users' perspectives to identify iPAC-relevant

apps. We apply natural language processing methods with text embeddings to automatically identify apps that address the iPAC dimensions.

For RQ2, we try to compare apps regarding their iPAC relevance based on an automatic app evaluation. The goal is to supplement the manual app evaluations made by domain experts who use the iPAC rubric for the manual evaluation [20].

For RQ3, we apply our approach and automatically identify iPAC-based apps and reviews. We then analyze them qualitatively to explore which iPAC characteristics they contain.

B. Research Method and Data

Fig. 3 shows an overview of our methodological framework, which comprises four consecutive phases.

In the first phase, we collected and preprocessed the research data. For this, we crawled Android apps and their reviews of the Google Play store as it is currently the largest app store on the mobile app market [51]. Table I summarizes the datasets we used in this study. Our dataset comprises 1,764,243 apps and 104,249,416 reviews. Among these apps, 156,944 belong to the app category "education," which has 3,537,382 app reviews. We further used an existing iPAC rubric database [20], which contains 169 manually evaluated education apps. We sampled 98 iPAC apps from this database and manually added 100 non-iPAC apps, which we describe and use in Section IV-B for our supervised machine learning experiments.

For our research, we only considered app descriptions and app reviews in the English language. For the data preprocessing, we used a language detection tool [52] to identify the language and filtered out the non-English apps, which left us with 98,999 apps. We followed the same approach to filter the app reviews, whereby we reduced the number of app reviews to 2,410,638.

In the second phase, we semi-automatically created a grounded set of iPAC keywords that we can use in the app classification. Bano *et al.* [8] conducted a preliminary study to identify iPAC-based reviews and analyzed them with sentiment analysis methods. They used an iPAC keyword set to identify iPAC-based reviews and found only a few matches because users do not use these terms from academic literature in their reviews including "socio-cultural," "context-awareness," "learner-negotiated," "contextualization," "situat- edness," or "social interactivity." In this study, we cope with this gap by refining the original iPAC keywords [13] with an extended set of keywords based on app review data to identify the iPAC-relevant parts in app descriptions and app reviews more precisely. We iteratively adapted and extended these keywords by incorporating the users' language in app reviews based on word embeddings and text analysis methods in collaboration with domain experts.

In the third phase, we answered RQ1 and RQ2. For RQ1, we used the extended iPAC keyword set to extract machine learning features and classify iPAC-based apps with a supervised machine learning approach. For RQ2, we used our extracted features to compare apps automatically regarding their iPAC

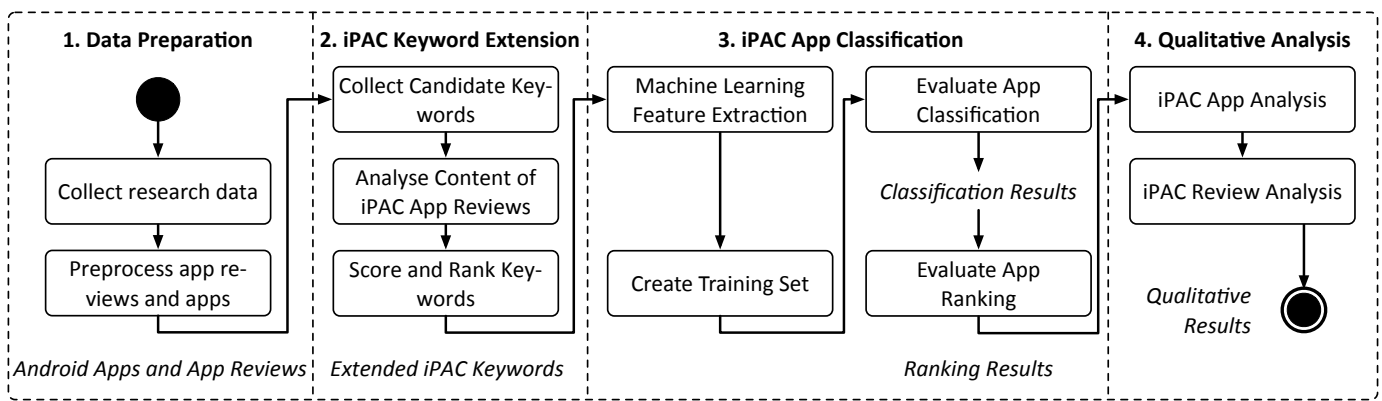


Fig. 3. Overview of our research methodology with four main phases.

scores. We evaluate our comparison with the Spearman’s rank correlation coefficient [53], which measures the distance between our ranking and the ranking by the domain experts. To answer RQ3, we performed a qualitative analysis of iPAC-based apps in the fourth phase. We sampled iPAC-based app reviews and qualitatively looked into the top-scoring apps for each iPAC dimension.

TABLE I
SUMMARY OF THE DATA SETS AND SUBSETS WE USED IN THIS STUDY

Data set	# Apps	# App reviews
Google Play store	1,764,243	104,249,416
Education apps	156,944	3,537,382
Education apps (English)	98,999	2,410,638
iPAC apps	98	821,731
non-iPAC apps	100	588,969

IV. ANALYSIS

In this section, we describe how we conducted the research phases two, three, and four as shown in Fig. 3.

A. iPAC Keyword Extension

In this phase, we semi-automatically extended the initial iPAC keyword set [13] based on app store data to identify iPAC-relevant parts in app descriptions and app reviews. We iteratively adapted and extended the iPAC keywords by incorporating “users’ language” based on word embeddings and text analysis methods in collaboration with domain experts. These domain experts are the creators of the iPAC framework and worked with it since then.

1) *Keyword candidate collection*: Kearney *et al.* [13] created a keyword set, describing the iPAC dimensions. It is based on academic literature in educational research related to iPAC and consists of 98 keywords. As a first step, we filtered keywords that we found ambiguous based on an initial manual analysis of how users use these words in app reviews. Thereby, we omitted keywords, including “games,” “mash,” and “tag.” Previous work has shown that users rarely use the initial iPAC keywords in their app review texts [8]. To overcome this

issue, we leveraged word embeddings to find word and phrase candidates that users write in similar contexts as the iPAC keywords. Word embeddings are vector representations of words in a high-dimensional space. Two vector representations of words that are used in a similar context are close in this space. For our application, we used word2vec, as proposed by Mikolov *et al.* [54], for computing a distributed word representation. Word2vec requires a text corpus, as large as possible, as an input and produces low-dimensional vectors as an output. For our experiment, we generated word embeddings based on the 2,410,638 English app review texts on education app reviews.

To acquire one single vector representation for common phrases (e.g., “augmented reality”), we applied a phrase detection algorithm as proposed by Mikolov *et al.* [55]. For the word2vec model generation, we used the Python library gensim [56]. We preprocessed the text by first removing the stop words, such as “I,” “you,” “this,” or “in.” Afterward, we lemmatized each word before training. Lemmatization is an approach to reduce the different inflections of a word to their basic form to be analyzed as a single item. For instance, it reduces the words “walked,” “walks,” “walking” to “walk.”

For the generation of additional iPAC keyword candidates, we collected the ten most similar words in the vector space to each of the iPAC keywords. After this step, we removed duplicates, emojis, and numbers, which left us with a set of 338 keyword candidates.

2) *Content analysis of iPAC-based app reviews*: In this step, we conducted a manual content analysis of app reviews as described by Neuendorf [57]. We only consider app reviews without app description texts as our study focuses on extending the iPAC keyword set with the user language found in the reviews. Three domain experts, familiar with the iPAC framework, systematically annotated occurrences of the keyword candidates in app reviews in a spreadsheet to evaluate how accurately a keyword identifies iPAC-based reviews. For each keyword candidate, we randomly retrieved three occurrences in app review texts from our English reviews on education apps. This keeps a balance between limiting the effort of the manual annotation while providing an insight into how accurately a keyword identifies an iPAC-based review.

The domain experts reviewed these occurrences indepen-

dently and indicated the iPAC dimensions addressed. As an example, the app review “*Great app. Lets you learn at a steady pace without confusing you...oh yeah and cool graphics*” contains the keyword candidate “steady pace,” which is, therefore, annotated as relevant for the personalization dimension. In contrast to that, the app review “*The most generous courses plate form*” contains the candidate “generous” and is, therefore, annotated as iPAC irrelevant. In total, three domain experts annotated 1,014 app reviews for all 338 collected keyword candidates. Among all keyword candidates, 108 were peer-annotated (i.e. 324 app review annotations). We resolved annotator disagreements with our scoring system, which we describe in the next section.

3) *Keyword scoring and ranking*: Given the domain experts’ annotations for all app reviews, we ranked the keywords regarding their accuracy for identifying iPAC-relevant app reviews. We introduced the following scoring system for the keywords: We obtained a keyword score by adding the number of identified iPAC-based reviews for each keyword. As the domain experts annotated each addressed iPAC dimension in the app reviews, we also added the iPAC-relevant reviews for each iPAC dimension and calculated a score for each dimension. For the peer-annotated app reviews, we calculated the mean value to resolve disagreements. We ranked the keywords according to these scores and discarded any keyword candidate with a total score below 1.5, as it ensures that the two domain experts considered in total at least three app reviews relevant. Table II lists the 51 remaining keywords with their scores after the filtering at the end of the second research phase.

B. iPAC-Based App Classification

In the third research phase, we describe all the experiments we conducted to answer RQ1 and RQ2. To answer RQ1, we conducted classification experiments. We trained a fastText model to extract machine learning features for an app based on its description and app reviews. To answer RQ2, we further inspected if we can use the extracted features to compare apps regarding their iPAC relevance.

1) *Machine learning feature extraction*: Feature extraction is the process of converting the items we want to classify into a numerical representation for the machine learning algorithm. We used a fastText model [58] to represent the apps numerically based on their app descriptions and app reviews. We trained the fastText model based on ~4 million sentences from app reviews on education apps in the English language with default parameters. We preprocessed the app review texts by performing a two-level phrase detection [55]. Thereby, we detected frequent phrases of up to three words, which became single tokens in the vector space, for example, “wi fi connection.”

We extracted machine learning features for an app based on its app description and app reviews, which is also known as text embedding. In the following, we formally describe the extracted features. We used the fasttext model to calculate vectors that represent each iPAC dimensions (P, A, and C). We describe how we calculated the iPAC vectors. Let A be

TABLE II
iPAC KEYWORD RANKING AFTER THE SEMI-AUTOMATIC EXTENSION
AND THEIR SCORES OF THE MANUAL CONTENT ANALYSIS

iPAC Keyword	iPAC scores			
	Pers.	Auth.	Coll.	Total
steady pace	3.0	1.0	0.0	4.0
comfortable pace	3.0	0.0	0.0	3.0
communication between	0.0	0.0	3.0	3.0
customise	3.0	0.0	0.0	3.0
portable	3.0	0.0	0.0	3.0
communication	0.0	0.0	3.0	3.0
communicate	0.0	0.0	3.0	3.0
diversity	3.0	0.0	0.0	3.0
own convenience	3.0	0.0	0.0	3.0
dynamic	2.0	1.0	0.0	3.0
tailor	3.0	0.0	0.0	3.0
augmented reality	0.0	3.0	0.0	3.0
customization	3.0	0.0	0.0	3.0
configurable	3.0	0.0	0.0	3.0
stay connected	0.0	0.0	3.0	3.0
instant communication	0.0	0.0	3.0	3.0
adapt	3.0	0.0	0.0	3.0
encouragement	3.0	0.0	0.0	3.0
immediate feedback	3.0	0.0	0.0	3.0
own pace	3.0	0.0	0.0	3.0
leisure	3.0	0.0	0.0	3.0
communication between parent	0.0	0.0	3.0	3.0
collaborate	0.0	0.0	2.5	2.5
freedom	2.5	0.0	0.0	2.5
personalize	2.0	0.0	0.0	2.0
direct contact	0.0	0.0	2.0	2.0
convenience	1.5	0.5	0.0	2.0
picky	0.0	2.0	0.0	2.0
dialogue	0.0	0.0	2.0	2.0
personal	1.0	0.0	1.0	2.0
interactive	0.0	0.0	2.0	2.0
interaction	1.0	0.5	0.5	2.0
interact	1.0	0.0	1.0	2.0
instant	1.0	0.0	1.0	2.0
imagination	2.0	0.0	0.0	2.0
flexibility	2.0	0.0	0.0	2.0
growth	0.0	2.0	0.0	2.0
clever	2.0	0.0	0.0	2.0
social	0.0	0.0	2.0	2.0
visualization	0.0	2.0	0.0	2.0
motivation	1.0	1.0	0.0	2.0
maintain	2.0	0.0	0.0	2.0
ownership	2.0	0.0	0.0	2.0
authentic	0.0	2.0	0.0	2.0
aspiring	1.0	1.0	0.0	2.0
sandbox	0.0	1.0	0.5	1.5
adjust font size	0.0	0.0	1.5	1.5
sport	0.5	0.5	0.5	1.5
produce	1.0	0.0	0.5	1.5
empowering	1.0	0.5	0.0	1.5
google+	0.0	0.0	1.5	1.5

the set of all words included in the fastText model. Further, let $K_p, K_a, K_c \in A$ be the keyword sets for each iPAC dimension, and let $F : A \rightarrow \mathbb{R}^{100}$ be the word embedding function that yields a 100-dimensional vector representation from the fastText model with the default configuration. Then, we calculated the iPAC vectors P, A, C as the sum of the single keyword vectors:

$$P = \sum_{w \in K_P} F(w) \quad A = \sum_{w \in K_A} F(w) \quad C = \sum_{w \in K_C} F(w)$$

We further defined a function that yields a vector representation of a text, which we use to calculate a vector representation for the app description text as well as for an app review text. This function calculates the average of the vectors of each word. According to Kenter *et al.* [59], this approach “has proven to be a strong baseline or feature across a multitude of tasks,” including short text similarity tasks.

Let $T = \{(w_1, w_2, w_3, \dots, w_n) | w_i \in A : 1 \leq i \leq n\}$ be the set of all texts that consists of a sequence of words. Then we define the function $\bar{T} : T \rightarrow \mathbb{R}^{100}$ as:

$$\bar{T}(t) = \frac{\sum_{i=1}^n F(w_i)}{n} \text{ for } t \in T$$

Finally, we defined the feature extraction function $V : T \rightarrow \mathbb{R}^3$ as the cosine distance d between the text vector and each iPAC vector:

$$V(t) = (d(P, \bar{T}(t)), d(A, \bar{T}(t)), d(C, \bar{T}(t)))$$

The features for an app a based on its app description text t_a are then described with the previously defined function as $V(t_a)$. To extract the feature of an app based on an app’s reviews, we average over the extracted features for all app review texts $R_a = \{t_1, t_2, \dots, t_n\}$ of that app:

$$\bar{V}_a = \frac{\sum_{t \in R_a} V(t)}{|R_a|}$$

2) *Training set creation:* Our training set should consist of iPAC apps and non-iPAC apps. We sampled the iPAC apps from the 169 apps, which domain experts evaluated according to the iPAC rubric. The dataset contains both Android and iOS apps. We only considered the iPAC-based Android apps, with an English app description, with a length of at least 500 characters to filter out empty and meaningless app descriptions with too little content. This left us with 98 iPAC-based apps for our training set (see Table I). We added the single scores for each iPAC dimension to reduce the complexity, which yielded three combined iPAC scores. For each dimension, three is the lowest score, and nine is the highest.

We manually selected 100 non-iPAC apps, based on the app title and app screenshots across different app categories to our training set (see Table I). We used the app category “education” as an exclusion criterion. Our sample includes apps from diverse categories, including art & design, music, tools, games, health, productivity, and lifestyle. This results in our training set of 198 apps, 98 iPAC apps, and 100 non-iPAC apps.

For the machine learning features based on the app reviews, we collected up to 10,000 app reviews sorted by “most helpful first” (according to Google Play store). The collection took one week at the beginning of April 2019. The outcome of this step is the training set, which we used to evaluate the subsequent automatic app classification.

3) *Evaluation of the app classification:* To answer RQ1, we extracted the features based on the app description text and the app reviews. We used a binary support vector machine (SVM) classifier with a linear kernel [60] and default parameters to classify whether an app is iPAC-relevant (*ipac*) or not (*other*). The SVM classifier has successfully been applied to different text classification tasks, including software defect detection [61] and fake news detection [62]. We used two different feature sets: (1) based on the app description and (2) based on both the app description and the app reviews.

For assessing the classification results, we evaluated the classification accuracy using the standard metrics precision, recall, and F1 score. $Precision_{ipac}$ is the fraction of apps that are classified correctly to belong to *ipac*. $Recall_{ipac}$ is the fraction of *ipac* apps which are classified correctly. We calculated them as follows:

$$P_{ipac} = \frac{TP_{ipac}}{TP_{ipac} + FP_{ipac}} \quad R_{ipac} = \frac{TP_{ipac}}{TP_{ipac} + FN_{ipac}}$$

TP_{ipac} is the number of apps classified as *ipac* and actually are *ipac* apps. FP_{ipac} is the number of apps that are classified as *ipac*, but actually, they are of the *other* class. FN_{ipac} is the number of apps that are classified as *other* but actually are *ipac* apps. We consider precision more important than recall as we want to minimize type I errors (false positives) for an application. For example, we want a teacher to examine a minimal number of wrongly-classified apps. The classifier might not catch all iPAC-based apps, but on the other hand, we minimize the time spent by the teacher examining irrelevant apps. We manually analyzed the apps that fall under the category of type I errors (“false positives”).

We also calculated the F-Measure (F1), which is the harmonic mean of precision and recall providing a single accuracy measure. Due to our training set’s small size, we conducted stratified ten-fold cross-validation [63] on our training set to acquire reliable results. It splits the truth set ten times at a ratio of 90:10. In each fold, we use 90% of the data as the training set and 10% as the test. Based on the size of our truth set, we felt this ratio is an appropriate trade-off for having large-enough training and test sets. Finally, we plot the ROC curve in Fig. 4 and report the area under the curve (AUC) to provide an aggregated measure of performance across all possible classification thresholds. We used the Python library scikit-learn [64] for the experiments. To enable replication, our models, along with the relevant source code, is publicly available.

4) *Automatic app comparison:* We also explored whether we can utilize our extracted machine learning features to automatically compare apps regarding the iPAC rubric scores. We checked for a correlation between manual iPAC evaluation by domain experts and the ranking based on our extracted features. For the correlation metric, we used Spearman’s rank correlation coefficient.

V. RESULTS

In this section, we present the results of the research phases three and four and answer the research questions of this study.

A. iPAC-Based App Classification

1) *Classification results:* Table III summarizes the results of the stratified ten-fold cross-validation classification experiment with different feature set combinations. We used all 198 apps of the training set for this experiment. The numbers in bold represent the highest scores for each column and class, which means the highest accuracy metric (precision, recall, and F1-score) for the app classes iPAC and other.

TABLE III
CLASSIFICATION RESULTS WITH DIFFERENT FEATURE SETS. TRAINING SET: 98 iPAC APPS AND 100 OTHER APPS

Class	Feature set	Precision	Recall	F1-score
iPAC	Description	0.68	0.28	0.39
Other	Description	0.55	0.87	0.67
iPAC	Descr. + Reviews	0.71	0.72	0.72
Other	Descr. + Reviews	0.72	0.71	0.71

We achieved the best F1-scores (0.71 – 0.72) by using the features based on the app description and app reviews. The precision and recall values are balanced (0.71 – 0.72). The recall for the app class other reached 0.87 when we used only the app description features. Regarding RQ1, we identified iPAC-based apps using the app description and app review texts with an F1-score of ~ 0.72 ($AUC = 0.78$) in this setting.

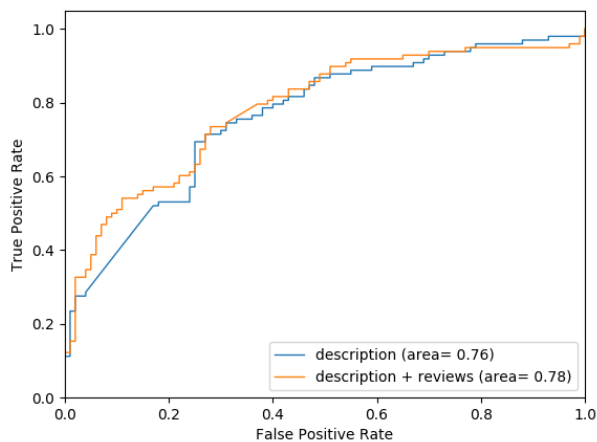


Fig. 4. ROC curve of the classification results with two different feature sets.

We further describe the results of the manual analysis of the false positives (type I errors). We found that the app description texts of these apps contained iPAC-relevant app features. For example, a chess app and a checkers app include authenticity features as “Top - 2D and Front - 3D” and “Realistic graphics,” the fitness app Strava provides collaborative features as “Record routes on your Strava feed so friends & followers can comment & share their own progress.” Another hiking app mentions personalization features that allow the user to “build your own personal adventure log.”

When we manually selected non-iPAC-based apps, we only considered the app name and the screenshots, whereby we did

not get an overall impression of all app features. Thereby, we introduced noise in our training set as we might have considered apps wrongly as “not iPAC-relevant.” This finding is an indicator that our approach could reach better results with a cleaner set of non-iPAC-based apps.

B. App Ranking Results

Table IV shows the correlation results.

TABLE IV
SPEARMAN’S RANKING CORRELATION BETWEEN MANUAL ASSESSMENT AND EXTRACTED FEATURES

Feature	Personal.	Authent.	Collabor.
description_personalization	0.09	0.12	0.12
description_authenticity	0.18	0.22	0.14
description_collaboration	0.46	0.49	0.54
review_personalization	0.04	0.05	0.01
review_authenticity	0.14	0.16	0.1
review_collaboration	0.36	0.37	0.36

We can see that only the features, that are based on the collaboration vector correlate with human evaluation. We found a moderate positive correlation ($0.46 \leq rs[198] \leq 0.54$, $p < .001$) for the feature based on the app description. Further, we found a weak correlation ($0.36 \leq rs[198] \leq 0.37$, $p < .016$) based on app reviews [65].

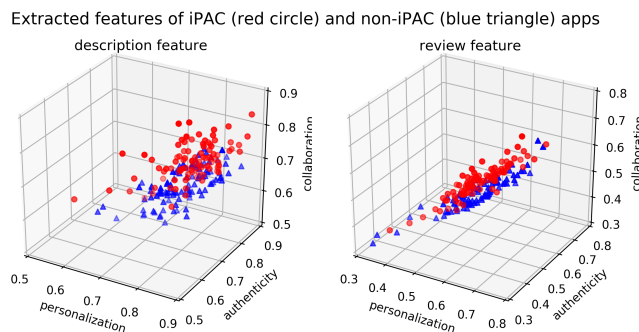


Fig. 5. Scatter plot that shows the extracted features based on the app description (left) and app reviews (right) for iPAC-based apps and non-iPAC-based apps.

Fig. 5 shows two 3d-scatter plots with the apps of our training set. We extracted three features based on the app description and the app reviews and placed it in the coordinate system. It shows that the extracted features based on the collaboration dimension are the most informative to distinguish between iPAC and non-iPAC-based apps. Regarding RQ2, we ranked mobile apps according to the iPAC framework as we found a moderate positive correlation ($rs[198] \leq 0.54$) between our extracted features and the manual app evaluation by domain experts.

C. Qualitative Analysis

In the fourth research phase, we used our approach to find iPAC-based apps and iPAC-based reviews. We then proceeded

TABLE V
TOP-RANKED iPAC APPS FOR EACH iPAC DIMENSION

Name	App description quotes	iPAC scores		
		P	A	C
Shotclasses	<i>“Provides personalized content on personal and company-owned shared devices,” “Intelligently recommends trainings to improve individual knowledge base”</i>	0.92	0.83	0.76
Ratna Sagar Science Magic AR	<i>“augmented reality,” “knowledge is imparted using visual learning,” “3D visualization,” “models of the human skeleton”</i>	0.84	0.89	0.72
HeadStart Kent	<i>“communication, collaboration and content sharing in education”</i>	0.80	0.72	0.89

with a qualitative analysis inspired by Kurtanović and Maalej [48]. We randomly sampled automatically identified iPAC-based apps and iPAC-based reviews and summarized their content and commonalities. We implemented our approach to identify the top-ranked apps for each iPAC dimension automatically. The qualitative insights help teachers to understand the potential usefulness of iPAC-based reviews.

1) *Qualitative insights into iPAC-based apps:* Table V lists the apps, with their sections of the app description that refer to the top-ranked dimension, and the associated iPAC scores. The table shows that the app description advertises iPAC-relevant app features in accordance with the extracted iPAC scores. The app “ShotClasses” provides personalization features, including a customized recommendation system. The app “Ratna Sagar Science Magic AR” offers a variety of “visual learning” features based on “augmented reality” with “3D visualization” and a “virtual world.” The app “HeadStart Kent” describes features to “collaborate and share content in education.”

We further list the words and phrases that occur in app descriptions that are most significant for each iPAC dimension. Table VI ranks the words and phrases that occur in app description texts based on their similarity to each of the iPAC vectors. Thereby, we identified how the vendors address the iPAC dimensions in their app descriptions. Table VI reveals the connection between the iPAC dimensions and app features. Words regarding the personalization dimension include “paced,” “gamification,” and “customization.” For instance, the words regarding the authenticity dimension are related to visualization features in an app as “augmented reality” and “virtual reality.” The important words for the collaboration dimension are about communication features such as “instant messaging” and “conferencing.”

2) *Qualitative insights into iPAC-based reviews:* Table VII lists the reviews along with the addressed iPAC dimensions. We found that iPAC-based reviews often praise or criticize specific technical aspects of app features or the usability of the app. Review #1 praises the feature, which enables the personalization of training length. App reviews addressing authenticity mention augmented reality, 3d-models, or virtual

reality features. As an example, app review #2 endorses an “electronic simulation.” App review #3 praises the “interactivity” of an app. Some users point to the absence of certain app features, which are related to an iPAC dimension. For example, app review #4 would like the feature to communicate with fellow learners. Some users express their role (e.g., “as a teacher,” “I’m a student”) and explain how an app is useful in their specific scenario. App review #5 reveals that the author uses the “Emodo” app as a student and describes how to use the “Collaboration” relevant features.

Regarding RQ3, we can observe app features for each iPAC dimension in iPAC-based app descriptions and app reviews. App features on customization, individual settings, and personalized recommendations address the personalization dimension. Simulations with virtual and augmented reality features address the authenticity dimension. The collaboration dimension is addressed with features including file sharing, real-time collaborative editing, and other characteristics supporting communication between students, teachers, and parents.

VI. IMPLICATIONS AND FUTURE DIRECTIONS

This paper focuses on automatically identifying and comparing iPAC-based apps. The results provide evidence that we can automatically identify iPAC-relevant aspects in both app descriptions and app reviews. Our approach extracts the iPAC rubric app evaluation from unstructured data without the input of a domain expert or other formal evaluation instruments. In the following, we highlight future work by discussing technical extensions to improve our results and by outlining a potential app search tool. We also highlight threats to the validity and limitations of our study.

A. Using and Improving the Approach on Different Datasets

We limited the analysis of this paper to the three iPAC dimensions. However, we described our overall research methodology in sufficient detail, enabling researchers to apply this process to other frameworks. Our approach could also be applied to other frameworks with other pedagogical dimensions, which can be described with a preliminary list of keywords. However, an application to other frameworks with semantically different dimensions would require a new classification model with potentially different accuracy values. If other frameworks consist of semantically similar dimensions, the iPAC dimensions could be mapped to them, and our classification model could be reused. Testing the feasibility of such an application is subject to future work. This is also the case for other quality criteria of mobile apps, including functional requirements (app-features), non-functional requirements (e.g., user experience, usability, compatibility), or further economic aspects (platform, price).

In our study, we used a classification setup with little training data and handcrafted machine learning features. We achieved encouraging results (F1-score ≤ 0.72) for identifying iPAC-based apps and a positive correlation ($r[159] \leq 0.54$) to evaluate them automatically. For our setting, we preferred a traditional machine learning over a deep learning approach for two reasons. First, the classification of our approach is

TABLE VI
WORDS AND PHRASES OCCURRING IN APP DESCRIPTION TEXTS ORDERED BY THEIR SIMILARITY TO THEIR RESPECTIVE IPAC VECTOR

Personalization	Similarity	Authenticity	Similarity	Collaboration	Similarity
personalize	0.81	augmented reality	0.77	communication	0.87
personalized	0.81	visually stunning	0.77	communicate	0.82
engagement	0.78	visual	0.75	instant messaging	0.80
sustainable	0.78	manipulating	0.75	interact	0.78
interactively	0.76	ligament	0.73	conferencing	0.78
empowering	0.76	crafting	0.73	interaction	0.75
flexibility	0.76	pilgrim	0.73	collaborate	0.75
customised	0.76	virtual reality	0.72	collaborative	0.75
paced	0.74	highly polished	0.72	collaboration	0.73
interacting	0.73	quiver	0.72	classroom management	0.73
strategically	0.73	showcase	0.71	messaging	0.73
flexible	0.73	imagery	0.71	stay connected	0.73
gamified	0.73	graphical	0.71	portfolio	0.72
customized	0.72	exploring	0.70	teachers	0.70

TABLE VII
EXAMPLES OF IPAC REVIEWS AND THE CONSTRUCTS ADDRESSED

App review text	iPAC		
	P	A	C
1. "I particularly like how easy it is to adapt to my length of training preference."	X		
2. "Very interactive and easy app for electronic simulation. I liked it very much [...]"		X	
3. "Easy to use, a great studying application, and great social platform. [...]"			X
4. "[...] However I wish it had option to communicate with fellow Sololearners via message or option for groupchat to work problems together."			X
5. "Edmodo is very helpful for when my teachers post things or even so I can even communicate to my fellow peers."			X

interpretable, and we identified the significant words in app descriptions that are most significant for the classification of an iPAC-based app [66]. Second, deep learning methods typically need large training sets to outperform traditional approaches [67], and we only had access to a limited number of training samples. Higher levels of accuracy might be achieved by experimenting with different feature combinations and by adding more sophisticated features for the classification, which we could extract from app screenshots or automatic navigation through the app.

Regarding the app reviews, we could combine the feature extraction with related work in the app store analysis domain [44] and consider different app reviews types, including bug reports, feature requests, or praise. However, for an automatic iPAC evaluation based on the users' sentiment in app reviews, it is challenging when a user expresses a positive sentiment towards an iPAC pedagogy but actually requests a missing app feature. For this, Maalej *et al.* [44] developed an approach to classify feature requests, which we could incorporate in future work to address this issue. Finally, we could collect and analyze user feedback from multiple platforms and include

sources beyond app stores, such as course forums, blogs, and other social media sites.

B. Potential Application and Utilization

We highlight the potential application of our empirical and exploratory research results using a tool suggestion that facilitates searching education apps. Fig. 6 depicts a user interface example that we designed for an education app search tool that incorporates our automatic analysis to showcase useful user scenarios [68] and research directions. In future work, we plan to evaluate this search tool with teachers and parents. We describe potential user scenarios for different user roles.

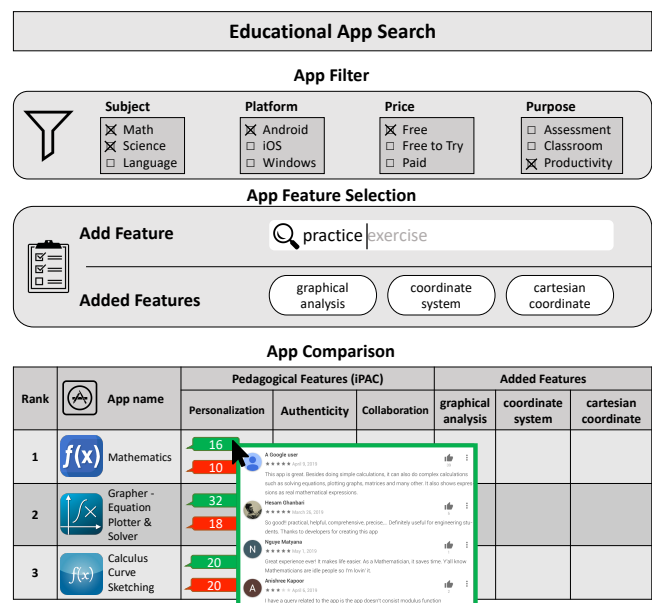


Fig. 6. User interface example for an education app search tool.

Teachers and Students. Teachers and students could use this tool to seamlessly navigate the increasingly unmanageable

number of education apps to find a suitable resource for their needs. The teacher sets the app filter depending on the context and pedagogical priorities, school policies, and other external conditions. For example, a math teacher may be planning an exploratory activity on the topic of “curve sketching,” and may be searching for a suitable, low cost or free education app knowing that most students have access to Android devices. The teacher uses the search tool to nominate relevant app features, for example, a “graphical coordinate system” and problem-solving “exercises.” We then automatically rank the filtered list of apps that satisfy the search criteria regarding the extracted iPAC scores. The search results could be ordered according to the confidence score of the iPAC app classifier. Our approach captures relevant education apps and locates essential app reviews relevant to the iPAC dimensions.

Additionally, informed by previous work by Bano *et al.* [8], this tool can provide an overview of positive and negative iPAC reviews. Depending on the topic and teaching setting, the teacher could either be interested in, for example, personalized aspects (e.g., self-directed learning) or in collaborative aspects (e.g., real-time collaborative editing, chat), across a range of learning settings such as school playgrounds and excursion sites. The teacher browses the user feedback on the iPAC dimensions and gets insights into how the app potentially “value-adds” to pedagogy, and the relevant education app features. In this way, teachers can bypass the plethora of pedagogically shallow education apps [20].

Students or parents might similarly use the search tool to find an appropriate education app for their purposes. Additionally, our tool could improve the richness of evaluative data for users. When students, teachers, or parents submit an app review, our tool identifies if an iPAC dimension is addressed and queries more nuanced iPAC related questions. The tool aggregates this additional feedback and summarizes this information for further app searches by other users.

Vendors. App vendors present their app by advertising app features in the specific app description such as “personalized settings,” “group chat,” or “augmented reality.” App vendors usually focus on app features and neglect the pedagogical perspective. With the automatic evaluation provided by our tool, vendors can reflect on how their app potentially addresses pedagogical dimensions, which fosters a mutual understanding between vendors and teachers. Thereby, vendors get an insight into how they address the iPAC dimensions with their apps’ features, app description, and in the app reviews.

C. Threats to Validity and Limitations

As for every other empirical research, our work has potential threats to its internal and external validity. Regarding the internal validity, this study contains multiple coding tasks, and human coders can cause noise in the training dataset. To reduce this threat, we asked domain experts to annotate the app review sample. We also incorporated a scoring system for the keywords to compensate for disagreements.

The evaluation of a classifier is usually performed on a test set that has not been used during the training phase. However, we did not perform hyperparameter optimization, making a stratified ten-fold cross-validation sufficient for this evaluation.

Regarding external validity, our accuracy results for RQ1 and RQ2 are not generalizable, as they depend on the specific dataset and model used in this work. They might differ for other pedagogical frameworks, in other languages, or with other data sets. Additionally, we only conducted our experiments on Android apps from the Google Play store. However, app descriptions for other mobile operating systems (e.g., Apple or Windows) and in other languages might reveal other characteristics, which could lead to different results.

We mainly extracted our machine learning features from the app descriptions as we expected this source to contain more terms from the education domain, which describe iPAC relevance. However, with more iPAC-relevant app reviews, the classification accuracy might get improved.

VII. CONCLUSION

The iPAC framework is a well-established pedagogical framework for evaluating education apps along the dimensions: personalization, authenticity, and collaboration [20]. We extended the initial keyword base of the iPAC framework with a data-driven approach based on online user reviews. Based on these keywords, we introduced a machine learning approach to identify and rank iPAC-based apps automatically. We achieved promising classification results, including an F1 score of ~72%.

Further, we were able to show a moderate positive Spearman’s rank correlation of 0.54 between the domain experts’ app ranking and our feature-based app ranking. Our qualitative insights into identified iPAC-based apps and app reviews showed that our approach could capture iPAC-based app features as well as user feedback on the iPAC dimensions. We suggest a user interface example of an education app search tool and showcase potential user scenarios for teachers, students, and vendors. We explain how our approach could enable the development of this tool. Thereby, our work fosters the mutual understanding between app vendors and teachers about textual app data and user feedback in app stores and beyond.

ACKNOWLEDGMENT

We thank Sandy Schuck (University of Technology Sydney, Australia) and Kevin Burden (University of Hull, UK) for their support with the manual labeling of the collected app reviews. The work was also supported by BWFGB Hamburg within the “Forum 4.0” project as part of the ahoi.digital funding line. We further acknowledge the financial support provided by the Australian Research Council Discovery Project “Optimising Mobile Intensive Pedagogies—an ARC Discovery project for Quality Teaching and Learning, ARC-DP 150101214.”

REFERENCES

- [1] C.-K. Looi, D. Sun, and W. Xie, “Exploring students’ progression in an inquiry science curriculum enabled by mobile learning,” *IEEE Transactions on Learning Technologies*, vol. 8, no. 1, pp. 43–54, Jan. 1, 2015. DOI: 10.1109/TLT.2014.2376968.
- [2] M. G. Domingo and A. B. Garganté, “Exploring the use of educational technology in primary education: Teachers’ perception of mobile technology learning impacts and applications’ use in the classroom,” *Computers in Human Behavior*, vol. 56, pp. 21–28, Mar. 2016. DOI: 10.1016/j.chb.2015.11.023.

- [3] M. Kearney, K. Burden, and S. Schuck, "Disrupting education using smart mobile pedagogies," in *Didactics of Smart Pedagogy*, L. Daniela, Ed., Cham: Springer International Publishing, 2019, pp. 139–157. DOI: 10.1007/978-3-030-01551-0_7.
- [4] P. Sweeney and C. Moore, "Mobile apps for learning vocabulary: Categories, evaluation and design criteria for teachers and developers," *International Journal of Computer-Assisted Language Learning and Teaching*, vol. 2, no. 4, pp. 1–16, Oct. 2012. DOI: 10.4018/ijcallt.2012100101.
- [5] Y.-C. Hsu and Y.-H. Ching, "A review of models and frameworks for designing mobile learning experiences and environments," *Canadian Journal of Learning and Technology*, vol. 41, no. 3, 2015.
- [6] M. Bano, D. Zowghi, M. Kearney, S. Schuck, and P. Aubusson, "Mobile learning for science and mathematics school education: A systematic review of empirical evidence," *Computers & Education*, vol. 121, pp. 30–58, Jun. 2018. DOI: 10.1016/j.compedu.2018.02.006.
- [7] J. Clement. (May 27, 2020). Number of apps available in leading app stores as of 1st quarter 2020, [Online]. Available: <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>.
- [8] M. Bano, D. Zowghi, and M. Kearney, "Feature based sentiment analysis for evaluating the mobile pedagogical affordances of apps," in *Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing*, A. Tatnall and M. Webb, Eds., vol. 515, Cham: Springer International Publishing, 2017, pp. 281–291. DOI: 10.1007/978-3-319-74310-3_30.
- [9] M. Stevenson, J. Hedberg, K. Highfield, and M. Diao, "Visualizing solutions: Apps as cognitive stepping-stones in the learning process," *Electronic Journal of E-Learning*, vol. 13, no. 5, pp. 366–379, 2015.
- [10] R. Kay and J. Kwak, "Creating an evidence-based framework for selecting and evaluating mathematics apps," in *Society for Information Technology & Teacher Education International Conference*, Association for the Advancement of Computing in Education (AACE), 2018, pp. 755–760.
- [11] T. Cherner, J. Dix, and C. Lee, "Cleaning up that mess: A framework for classifying educational apps," *Contemporary Issues in Technology and Teacher Education*, vol. 14, no. 2, pp. 158–193, 2014.
- [12] D. Parsons, H. Ryu, and M. Cranshaw, "A design requirements framework for mobile learning environments," *Journal of Computers*, vol. 2, no. 4, pp. 1–8, Jun. 1, 2007. DOI: 10.4304/jcp.2.4.1-8.
- [13] M. Kearney, S. Schuck, K. Burden, and P. Aubusson, "Viewing mobile learning from a pedagogical perspective," *Research in Learning Technology*, vol. 20, no. 1, p. 14 406, Feb. 3, 2012. DOI: 10.3402/rlt.v20i0.14406.
- [14] M. Kearney, K. Burden, and S. Schuck, "Theorising and implementing mobile learning: Using the iPAC framework to inform research and teaching practice," Dordrecht, Netherlands: Springer, In Press.
- [15] J. V. Wertsch, *Mind as Action*. Oxford, New York: Oxford University Press, Feb. 1998.
- [16] N. Pachler, B. Bachmair, J. Cook, and G. Kress, "Mobile learning," *New York, NY: Springer*, vol. 10, pp. 978–1, 2010.
- [17] J. Radinsky, L. Bouillion, E. M. Lento, and L. M. Gomez, "Mutual benefit partnership: A curricular design for authenticity," *Journal of Curriculum Studies*, vol. 33, no. 4, pp. 405–430, Jul. 2001. DOI: 10.1080/00220270118862.
- [18] K. Burden and M. Kearney, "Conceptualising authentic mobile learning," in *Mobile Learning Design*, D. Churchill, J. Lu, T. K. Chiu, and B. Fox, Eds., Singapore: Springer Singapore, 2016, pp. 27–42. DOI: 10.1007/978-981-10-0027-0_2.
- [19] M. Wang and R. Shen, "Message design for mobile learning: Learning theories, human cognition and design principles: Message design for mobile learning," *British Journal of Educational Technology*, vol. 43, no. 4, pp. 561–575, Jul. 2012. DOI: 10.1111/j.1467-8535.2011.01214.x.
- [20] K. Burden and M. Kearney, "Designing an educator toolkit for the mobile learning age," *International Journal of Mobile Blended Learning (IJMBL)*, vol. 10, no. 2, pp. 88–99, 2018.
- [21] K. Burden, M. Kearney, and P. Hopkinson. (May 20, 2020). iPAC App Evaluation Rubric, [Online]. Available: <http://www.mobilelearningtoolkit.com/app-rubric1.html>.
- [22] M. Kearney, K. Burden, and T. Rai, "Investigating teachers' adoption of signature mobile pedagogies," *Computers & Education*, vol. 80, pp. 48–57, Jan. 2015. DOI: 10.1016/j.compedu.2014.08.009.
- [23] K. J. Burden and M. Kearney, "Investigating and critiquing teacher educators' mobile learning practices," *Interactive Technology and Smart Education*, vol. 14, no. 2, pp. 110–125, Jun. 19, 2017. DOI: 10.1108/ITSE-05-2017-0027.
- [24] M. Kearney and D. Maher, "Mobile learning in maths teacher education: Using iPads to support pre-service teachers' professional development," *Australian Educational Computing*, vol. 27, no. 3, pp. 76–84, 2013.
- [25] P. Townsend, "Mobile devices for tertiary study – philosophy meets pragmatics for remote aboriginal and torres strait islander women," *Australian Journal of Indigenous Education*, vol. 44, no. 2, pp. 139–149, Dec. 2015. DOI: 10.1017/jie.2015.26.
- [26] S. Kinash, J. Brand, and T. Mathew, "Challenging mobile learning discourse through research: Student perceptions of blackboard mobile learn and iPads," *Australasian Journal of Educational Technology*, vol. 28, no. 4, 2012.
- [27] O. Viberg and Å. Grönlund, "Cross-cultural analysis of users' attitudes toward the use of mobile devices in second and foreign language learning in higher education: A case from sweden and china," *Computers & Education*, vol. 69, pp. 169–180, Nov. 2013. DOI: 10.1016/j.compedu.2013.07.014.
- [28] M. Kearney, P. F. Burke, and S. Schuck, "The iPAC scale: A survey to measure distinctive mobile pedagogies," *TechTrends*, vol. 63, no. 6, pp. 751–764, Nov. 2019. DOI: 10.1007/s11528-019-00414-1.
- [29] D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," in *2013 21st IEEE International Requirements Engineering Conference (RE)*, 2013, pp. 125–134. DOI: 10.1109/RE.2013.6636712.
- [30] W. Martin, F. Sarro, Y. Jia, Y. Zhang, and M. Harman, "A survey of app store analysis for software engineering," *IEEE Transactions on Software Engineering*, vol. 43, no. 9, pp. 817–847, Sep. 1, 2017. DOI: 10.1109/TSE.2016.2630689.
- [31] M. Bano and D. Zowghi, "Users' voice and service selection: An empirical study," in *2014 IEEE 4th International Workshop on Empirical Requirements Engineering (EmpiRE)*, Karlskrona, Sweden: IEEE, Aug. 2014, pp. 76–79. DOI: 10.1109/EmpIRE.2014.6890120.
- [32] E. Guzman and W. Maalej, "How do users like this feature? a fine grained sentiment analysis of app reviews," in *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, Karlskrona, Sweden: IEEE, Aug. 2014, pp. 153–162. DOI: 10.1109/RE.2014.6912257.
- [33] D. Martens and W. Maalej, "Release early, release often, and watch your users' emotions: Lessons from emotional patterns," *IEEE Software*, vol. 36, no. 5, pp. 32–37, 2019.
- [34] L. Villarroel, G. Bavota, B. Russo, R. Oliveto, and M. Di Penta, "Release planning of mobile apps based on user reviews," in *2016 38th International Conference on Software Engineering (ICSE)*, 2016, pp. 14–24.
- [35] M. Haering, M. Bano, D. Zowghi, M. Kearney, and W. Maalej. (May 27, 2020). Dataset for 'Automating the Evaluation of Education Apps with App Store Data' (password: ieee2019transactionslearning), [Online]. Available: <https://mcloud.informatik.uni-hamburg.de/index.php/s/RNHHvCzDrE6Xjv/>.
- [36] I. R. Management Association, Ed., *Blended Learning: Concepts, Methodologies, Tools, and Applications*, IGI Global, 2017. DOI: 10.4018/978-1-5225-0783-3.
- [37] H. Crompton, D. Burke, K. H. Gregory, and C. Gräbe, "The use of mobile learning in science: A systematic review," *Journal of Science Education and Technology*, vol. 25, no. 2, pp. 149–160, Apr. 2016. DOI: 10.1007/s10956-015-9597-x.
- [38] A. Ortega-García, A. Ruiz-Martínez, and R. Valencia-García, "An m-learning open-source tool comparison for easy creation of educational apps," in *Technologies and Innovation*, R. Valencia-García, K. Lagos-Ortiz, G. Alcaraz-Mármol, J. del Cioppo, and N. Vera-Lucio, Eds., vol. 658, Cham: Springer International Publishing, 2016, pp. 102–113. DOI: 10.1007/978-3-319-48024-4_9.
- [39] J. A. RUIPEREZ-VALIENTE, P. J. MUNOZ-MERINO, G. ALEXANDRON, and D. E. PRITCHARD, "Using machine learning to detect 'multiple-account' cheating and analyze the influence of student and problem features," *IEEE Transactions on Learning Technologies*, vol. 12, no. 1, pp. 112–122, Jan. 1, 2019. DOI: 10.1109/TLT.2017.2784420.
- [40] B. Sun, Y. Zhu, Y. Xiao, R. Xiao, and Y. Wei, "Automatic question tagging with deep neural networks," *IEEE Transactions on Learning Technologies*, vol. 12, no. 1, pp. 29–43, Jan. 1, 2019. DOI: 10.1109/TLT.2018.2808187.
- [41] S. Papadakis, M. Kalogiannakis, and N. Zaranis, "Educational apps from the android google play for greek preschoolers: A systematic review," *Computers & Education*, vol. 116, pp. 139–160, Jan. 2018. DOI: 10.1016/j.compedu.2017.09.007.

- [42] A. Finkelstein, M. Harman, Y. Jia, W. Martin, F. Sarro, and Y. Zhang, "App store analysis: Mining app stores for relationships between customer, business and technical characteristics," *RN*, vol. 14, no. 10, 2014.
- [43] W. Maalej, M. Nayebi, T. Johann, and G. Ruhe, "Toward data-driven requirements engineering," *IEEE Software*, vol. 33, no. 1, pp. 48–54, 2016.
- [44] W. Maalej, Z. Kurtanović, H. Nabil, and C. Stanik, "On the automatic classification of app reviews," *Requirements Engineering*, vol. 21, no. 3, pp. 311–331, Sep. 2016. DOI: 10.1007/s00766-016-0251-9.
- [45] M. Harman, Y. Jia, and Y. Zhang, "App store mining and analysis: Msr for app stores," in *2012 9th IEEE Working Conference on Mining Software Repositories (MSR)*, IEEE, 2012, pp. 108–111.
- [46] E. Guzman, R. Alkadhi, and N. Seyff, "A needle in a haystack: What do twitter users say about software?" In *2016 IEEE 24th International Requirements Engineering Conference (RE)*, Beijing, China: IEEE, Sep. 2016, pp. 96–105. DOI: 10.1109/RE.2016.67.
- [47] G. Williams and A. Mahmoud, "Mining twitter feeds for software user requirements," in *2017 IEEE 25th international requirements engineering conference (RE)*, 2017, pp. 1–10.
- [48] Z. Kurtanovic and W. Maalej, "Mining user rationale from software reviews," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*, Lisbon, Portugal: IEEE, Sep. 2017, pp. 61–70. DOI: 10.1109/RE.2017.86.
- [49] Z. Kurtanović and W. Maalej, "On user rationale in software engineering," *Requirements Engineering*, vol. 23, no. 3, pp. 357–379, Sep. 2018. DOI: 10.1007/s00766-018-0293-2.
- [50] T. Johann, C. Stanik, W. Maalej, *et al.*, "Safe: A simple approach for feature extraction from app descriptions and app reviews," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*, 2017, pp. 21–30.
- [51] J. Clement. (May 27, 2020). Most popular Google Play app categories as of 1st quarter 2020, by share of available apps, [Online]. Available: <https://www.statista.com/statistics/279286/google-play-android-app-categories/>.
- [52] N. Shuyo, *Language Detection Library for Java*. 2010.
- [53] J. H. Zar, "Spearman rank correlation," in *Encyclopedia of Biostatistics*, P. Armitage and T. Colton, Eds., Chichester, UK: John Wiley & Sons, Ltd, Jul. 15, 2005, b2a15150. DOI: 10.1002/0470011815.b2a15150.
- [54] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st Int. Conf. Machine Learning (ICML-14)*, 2014, pp. 1188–1196.
- [55] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.
- [56] R. Rehůfek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," English, in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, Valtta, Malta: ELRA, May 2010, pp. 45–50.
- [57] K. A. Neuendorf, *The content analysis guidebook*. Sage, 2016.
- [58] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.
- [59] T. Kenter, A. Borisov, and M. de Rijke, "Siamese CBOW: Optimizing word embeddings for sentence representations," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany: Association for Computational Linguistics, 2016, pp. 941–951. DOI: 10.18653/v1/P16-1089.
- [60] A. Ben-Hur and J. Weston, "A user's guide to support vector machines," *Data Mining Techniques for the Life Sciences*, pp. 223–239, 2010.
- [61] S. Aleem, L. F. Capretz, and F. Ahmed, "Benchmarking machine learning techniques for software defect detection," *International Journal of Software Engineering & Applications*, vol. 6, no. 3, pp. 11–23, May 31, 2015. DOI: 10.5121/ijsea.2015.6302.
- [62] W. Y. Wang, "'liar, liar pants on fire': A new benchmark dataset for fake news detection," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 422–426. DOI: 10.18653/v1/P17-2067.
- [63] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.
- [64] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [65] P. H. Ramsey, "Critical values for spearman's rank order correlation," *Journal of Educational Statistics*, vol. 14, no. 3, pp. 245–253, Sep. 1989. DOI: 10.3102/10769986014003245.
- [66] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, May 2019. DOI: 10.1038/s42256-019-0048-x.
- [67] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.
- [68] J. Dick, E. Hull, and K. Jackson, *Requirements engineering*. Springer, 2017.



Marlo Haering is a research associate at the Applied Software Engineering Group at the University of Hamburg. He holds a master degree in Informatics with a bachelor degree in Software-System Development achieved at the University of Hamburg. His research area includes applied natural language processing and machine learning techniques in online journalism and mobile app stores. He is currently part of the Forum 4.0 project, as a researcher and developer.



Didar Zowghi (S'95—M'20) is Professor of Software Engineering and the Deputy Dean of Graduate Research School at University of Technology Sydney (UTS). Her research interests are mainly focused on requirements engineering, software engineering and technology adoption. She practices Evidenced Based research and has conducted variety of empirical studies in many interdisciplinary fields. Currently she is associate editor of IEEE Software and Requirements Engineering Journal. She has published over 190 research articles in prestigious conferences and journals, co-authored with 90 different researchers from 30 countries.



Muneera Bano is a senior lecturer of software engineering from School of Information Technology at Deakin University. She is one of the 'Superstars of STEM' for Science and Technology Australia as a representative of Women in STEM for 2019-2020. Muneera graduated from University of Technology Sydney with a PhD in Software Engineering in 2015. Her research focuses on software requirements engineering, technology assisted pedagogies for education and social media analysis. She contributes to the broader research community as Associate Editor of the Institution of Engineering and Technology Software Journal, as Track Chair in International Requirements Engineering, and Australian Software Engineering Conferences; and as a member of the program committee for various highly-ranked conferences including Grace Hopper Conference for Women in Computer Science. She is currently the Associate Editor of the IET Software Journal and IEEE Software Blog.



Matthew Kearney is an Associate Professor in the area of ICT in Education at the University of Technology Sydney (UTS). His scholarly interests focus on innovative technology-mediated learning in K-12 and teacher education contexts. He has led or participated in numerous funded research projects investigating pedagogical practices with emerging learning technologies since 1999. He is currently a member of an Erasmus+ funded project team investigating innovative mobile pedagogies.



Walid Maalej is a professor of informatics at Universität Hamburg and the head of the Applied Software Technology group. He is also a management board member at the tech-transfer institute HITeC and a steering committee member of the IEEE Requirements Engineering conference. His research interests include user feedback and participation, data-driven software engineering, context-aware systems, applied machine learning, and software engineering's impact on society. He received his Ph.D. in software engineering from the TU Munich.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60