Opinion Mining for Modeling User Experience of Online Education: Sentiment Analysis and Keywords Extraction of Student Reviews

Anna Moskvina, Margarita Kirina, Anastasia Gavrilyuk
National Research University Higher School of Economics
Saint Petersburg, Russia
{admoskvina, mkirina}@hse.ru, asgavrilyuk@edu.hse.ru

***Abstract*—The paper discusses the possibilities of applying modern natural language processing technologies of opinion mining to investigate and improve the user experience of online-courses students. We analyzed 27 000 student reviews of projects within the Python programming language course. First, we applied keyword extraction algorithms as a way of semantic compression to receive a generalized picture of what users' main impressions are. Then we performed sentiment analysis to understand the feelings of students towards the learning process. The used methodology proved to be effective for analyzing user experience and allowed to find out some discrepancies between information in project descriptions and what users' reflection on the project. Two instruments of SA were applied to receive data on users' feelings in general.**

I. Introduction

Due to inevitable processes of globalization and digitalization more and more activities go online, including social and professional communication, commercial transactions, and, of course, education. Recent covid pandemic of 2020 triggered the process even further, extending the market of online courses to unprecedented popularity. The phenomenon of massive open online courses (MOOC) already received wide coverage both in academic fields and educational industry (see, for example, review [1]. The material provided by commercial online learning platforms can be viewed as the relationship between client and the service provider. The student receives a product and based on their experience may be satisfied with the result or choose to change the provider. That is why the correct understanding of the emotions of the user is crucial to the creators of the courses. Natural language processing technologies can be applied to explore users' feelings and gather insights for educational product development.

IT is one of the most popular areas for online learning. The profession itself being extremely in demand, there is also an opinion that it does not require traditional academic higher education when skills and achievements are present. It raises another question of what is the most effective way of teaching in computer science.

In this study, we decided to process students' short reviews on the projects they have chosen to work on within their programming course track. To investigate users' sentiments and opinions, we applied two different sentiment analysis systems (VADER, TextBlob). We also extracted keywords via three algorithms (TF-IDF, YAKE, and RAKE) to get a more structured content representation from the raw data, following the idea that people tend to write more about their most strong impressions.

II. Data Description

As a source for getting our data we have chosen the online learning platform Hyperskill [2]. Hyperskill develops and provides access to computer science and IT education, including programming languages such as Java, Python, and Kotlin, data science and web-development. After getting a subscription, a user selects one of the relevant tracks and starts to go through it, alternately solving theoretical and practical tasks. A feature of Hyperskill learning is that each track is a set of projects, each of which is associated with the solution of one practical problem. Projects are categorized according to the degree of difficulty: easy, medium, hard and challenging. Moving from simple to complex, the user masters the topics and tools that they will need to develop a design solution at the last stage, which is always a fully functional executable program. Thus, each project is not a course per se, but rather a step towards mastering the full program. After the completion of the project, the user is invited to leave a short review as a way to focus on what has been practiced. The platform additionally motivates the user to leave such a review. This determines the peculiarity of our data: such texts are not a full unstructured review of one's experience on the platform, but rather a short reflection on the main topics, concepts and technologies and positive or negative emotions.

We collected 27 692 reviews for 19 projects within the Python programming track for our study. The date of comment publication was also kept to trace the changes in feedback. Both semantic and emotional analysis were conducted to develop a methodology for modeling user experience as online-students. All the data that has been used is publicly available.

III. Keywords Extraction

A. Methodology

One of the tasks of automatic text processing is keyword extraction, which is an automated process of extracting the most relevant words and phrases from a document. Texts, articles, comments, and reviews are one of the main informational data formats, the keywords of which can reflect their entire meaning. In other words, keywords help filter and

find interesting information for users from the total amount of massive data.

Today, there are a huge number of educational platforms that offer a variety of online courses for students, as a result of which it becomes obvious that, on the one hand, developers need tools to quickly process data, extract basic information from them, and to improve the quality of the products offered to users. On the other hand, using automatic keyword extraction allows users who are unfamiliar with an online course to find out what those who have already taken it think about this course without having to read all the reviews.

To solve the problem of extracting keywords from a huge amount of information, all sorts of unsupervised approaches are used, ranging from statistical methods to graph-based approaches. On our data, we applied 2 methods for extracting keywords and expressions:

1) *TF-IDF:* The first and most basic method for quantifying words in a set of documents is the TF-IDF (Term Frequency - Inverse Document Frequency) statistical measure, used to evaluate the importance of a word in the context of a document that is part of a corpus. In this abbreviation, "TF" stands for word frequency, which is determined by the ratio of the number of occurrences of a certain word to the total number of words in the document. "IDF" - inverse document frequency or frequency inversion, which measures the informativeness of a word, thanks to which it becomes clear how often or rarely a particular word occurs in the entire set of documents.

The TF-IDF statistical measure is calculated by multiplying two indicators tf(t,d) × idf(t,D), namely the number of times a word occurs in a document, and the inverse frequency of a word in a document in a document set (reference corpus).

The statistical method TF-IDF, the main task of which is to extract keywords from a massive amount of information, was used in this work because it is a basic and very common method for solving this problem. So, for example, this method was used in the works of such authors as R. Siautama [3], U. Erra [4], in the work of J .Li, the precision of the TF-IDF method was 57.8% [5], in the work of W. Zhuohao 44.3% [6].

2) *YAKE:* The second method is Yake (Yet Another Keyword Extractor). This is an easy, unsupervised method for extracting the most relevant keywords, which is based on the statistical features of the text extracted from individual documents. Unlike other methods, this library does not depend on both the language and external corpora and various dictionaries, it is also able to work without the use of a training corpus [7].

The algorithm of the YAKE method is calculated by the formula:

$$W_{Case} = \frac{max(TF(U(w)),TF(A(w)))}{log_2(TF(w))}$$

where TF(U(w)) – is the number of occurrences of candidate word "w" that starts with an uppercase letter, TF(A(w)) – is the number of times candidate word "w" is tagged as an acronym, and TF(w) – is the frequency of "w" [8].

It follows that the more often a candidate term appears with a capital letter (excluding those cases when words are at the beginning of sentences), the more important and significant it is considered.

Unlike the previous method, YAKE is a relatively new method for extracting keywords, its main difference is that this method does not require a reference corpus, which cannot be said about the TF-IDF method. So, for example, in a study by R. Campos, which compared 4 other methods besides YAKE, it was found that the YAKE method is the most accurate method for extracting keywords [9].

3) RAKE-NLTK: Rapid Automatic Keyword Extraction, or RAKE, is one of the algorithms for extracting keywords. It is based on the understanding of keywords as key phrases that characterize the text. As suggested, the length of the key phrase is typically more than one token; moreover, punctuation marks, stop words and words with minimal lexical value rarely fall into its composition. Taking that into consideration, when generating a list of content words, the algorithm evaluates the position of stop words and punctuation marks specified by the user, and, splitting sentences into phrases based on these lists, determines candidate keywords [10].

The selection of keywords can be carried out in accordance with one of the proposed metrics:

- the metric of the ratio of word degree to frequency — d(w)/f(w);
- the word degree metric — d(w));
- the word frequency metric — f(w).

In this paper, the ratio of word degree to frequency (deg(w)/freq(w)) is considered, which allows you to highlight words that mainly occur in longer keyword candidates.

It is also worth paying attention to the fact that when using RAKE, the most effective is the extraction of keywords from each document of a dynamic collection of texts, which makes it possible to equally successfully analyze different types of texts in terms of grammatical characteristics — simple, complex or not following conventional norms [10]. In addition, there is a consistency between the results of automatic and expert word extraction: keywords extracted by RAKE most often correlate with keywords assigned manually by experts. It is also stated that RAKE shows the highest precision (33.7%) compared to TextRank and supervised learning [ibid.]. However, since, as it is shown in [10], RAKE performs better on individual documents, we will test it by extracting keywords from each positive and each negative review with comparing results afterwards (see Chapter IV).

The chosen algorithms are popular, have several implementations, and easy to use, so one can easily replicate our methodology, adjusted for a particular type of data, by using open-source libraries and tools.

B. Data preprocessing

During data preprocessing, the purpose of which is to bring the data into a more convenient format for further work with them, at the initial stage, reviews for all projects were present in CSV format. Next, the reviews in all languages except English were removed from all files. Initially, the total number of all public comments in various languages was 27 692, after the deletion, their number decreased by 711 to 26 981, which indicates that the majority of users leave their reviews in English.

The next stage of data preprocessing for the TF-IDF method is tokenization, the main idea of which is to split the entire text into small parts, also known as tokens, where the minimum semantic unit of the text is one word. In addition to the split itself, this process also allows you to clean up data that does not carry the necessary information, namely: punctuation marks, tabs, line breaks and extra spaces. To achieve this goal, the NLTK tokenizer package was used.

The next preprocessing step is lemmatization, the main purpose of which is to remove the inflectional forms of the word and return the base or dictionary form, known as the lemma. To achieve this goal, the NLTK lemmatization method was used, which is based on the built-in WorldNet morphing function.

To avoid the problem of getting into the extracted keywords of functional parts of speech, such as pronouns, particles, prepositions and conjunctions, the NLTK stop words package was used.

C. Performing keywords extraction

The study found that the TF-IDF and YAKE methods show the most effective keyword extraction results when they are applied to all reviews of each project together, rather than for each review separately. In the course of work for all projects in total 1 140 keywords were extracted. Using the TF-IDF method, 380 keywords were extracted, namely 20 keywords for each project, respectively. The reference corpus for the TF-IDF method was 27 692 English reviews from 19 projects, not including remote reviews in other languages. Using the YAKE method, 760 keywords were extracted, since during the research it was found that the YAKE method well extracts not only single keywords, but also phrases. Thus, using the YAKE method, 380 unigrams and bigrams were extracted, as well as 380 bigrams, respectively.

According to the results obtained, it was found that:

- in more than a half of the cases, the first three extracted words match in both methods;
- most of the extracted unigrams are nouns (e.g., module, dictionary) and adjectives (e.g., interesting, useful), as for bigrams, there is more often a combination of an adjective and a noun (e.g., basic knowledge, good experience), a noun and a noun (e.g., NLP tool, NLP concept).

To avoid duplication, two to six relevant keywords/phrases were selected from the extracted keywords for each project, extracted using both the TF-IDF method, some examples of which are shown in Table I and YAKE method shown in Table II.

TABLE I. RELEVANT KEYWORDS EXTRACTED USING THE TF-IDF METHOD

Project name	Relevant keywords
Bill Splitter	Dictionary, exception, handling
Coffee Machine	Class, oop
Easy Rider Bus Company	Set, json, regex, list, dictionary, data, regular, itertools, iterators
Generating Randomness	Probability, dictionary, math, list, statistic

TABLE II. RELEVANT KEYWORDS EXTRACTED USING THE YAKE METHOD

Project name	Relevant keywords
Hypercar Service Center	Django framework, Django Template
HyperNews Portal	Django Framework, JSON file, html, CSS
Key Terms Extraction	Basic NLP, NLP nltk, language processing, xml file
Loan Calculator	Command line, argparse module, line argument, math module, cli argument

Analyzing the relevant keywords that were extracted using both methods shown in Table III, you can see that they do not always match, which is why it is far from always worth relying on the results of only one method.

TABLE III. COMPARISON TABLE OF RELEVANT KEYWORDS

Project name	Relevant keywords YAKE	Relevant keywords TF-IDF
Hypercar Service Center	Django framework, Django Template	Django, template, framework, queue, request
Key Terms Extraction	Basic NLP, NLP nltk, language processing, xml file	Nlp, sklearn, nltk, tf-idf, xml, tokenization, lemmatization
Markdown Editor	Markdown Language, lambda function	Markdown, map, lambda, list, string
Zookeeper	Use loop, programming language, data type, boolean logic, list	Loop, list

With the help of the received keywords, which were extracted from user reviews, it was possible to establish the complexity of some projects, as well as to determine the attitude of the majority of users, which are shown in Table IV. Most of these words were extracted using the TF-IDF method. All 19 projects have a positive rating, and most users find these projects and practice very useful for themselves.

TABLE IV. USER ATTITUDES TOWARDS PROJECTS OF THE HYPERSKILL PLATFORM

Project name	Attitude of users to the project
Simple Banking System	Good, great, interesting, nice, useful
Simple Chatty Bot	Good, easy, great, simple, beginner, nice, like, new, basic knowledge, basic thing, good basic, good practice
Simple Tic-Tac-Toe	Good, great, fun, new, like, good practice, good experience, really good
Text-Based Browser	Good, basic, useful, great, nice

From the above, we can conclude that both methods extract keywords well, but the result will be more efficient if we consider the extracted keywords of both methods together. It is also worth mentioning that some of the extracted keywords can be attributed to the subject and content of the project, which include conditional topics (e.g., function, dictionary), tools (e.g., django, flask), and libraries (e.g., NumPy, BeautifulSoup), while the other part can be attributed to a more subjective experience, which included the complexity of projects (e.g., easy, really hard), and the attitude of the majority users to them (e.g., interesting, useful).

D. Comparing descriptions with keywords

Each project on the Hyperskill platform has a description that contains more technical and applied information about what the user will master after completing a particular project.

With the help of extracted keywords, it becomes possible to follow the trend of the most frequently listed keywords by users, which should be mentioned in the description, so that the future user chooses a project based on real user experience, which should be presented as keywords from reviews.

Applying the results to all the projects participating in the experiment, it was found that not all of the extracted keywords are mentioned in the project descriptions. So, for example, almost all the received keywords are present only in the descriptions of 5 projects, which cannot be said about the remaining 14.

For example, the description of the Zookeeper project tells future users that this project is aimed at beginners. It will help you understand some basic syntax and learn how to work with variables, data storage types such as lists and while loops. If we turn to the obtained relevant words of both methods, we can see that almost all of them are present in the project description, which indicates that the project description corresponds to what users who have already completed this project write about in their reviews.

If we examine in detail the description of the Key Terms Extraction project, we will find that not all extracted entities are mentioned in it. Using the TF-IDF and YAKE method, it was possible to establish that the keywords of this project are: NLP, NLTK, TF-IDF, which are already present in the learning outcomes. The missing keywords in the description included such words as: xml file, sklearn, tokenization, lemmatization.

Due to this, we can conclude that our methodology allows us to get an overall picture of what the user thinks about the project after completion, which is not always possible to predict in advance when creating a project description.

IV. Detecting sentiment and opinions

A. Sentiment analysis

Sentiment analysis is a field in computational linguistics which concerns the questions of identification of emotional valence of texts. This method of content analysis is used to detect the author's opinion on the subject discussed in the text and to characterize the opinion in terms of emotions it is associated with [11]. For that reason, the focus of the research often turns to the user-generated content (UGC) which is everything that may contain people's opinion about the quality or individual characteristics of a product [12]. Depending on the type of sentiment analysis, the following ways of classifying texts in terms of their emotional valence are usually suggested:

1) *Detecting the polarity score:* Polarity scales can be categorical or floating within a range. Among the categorical ones, binary ones are popular, when texts are described only as *positive* or *negative*. In addition, a neutral assessment is often introduced, which is used if it is difficult to unambiguously attribute the text into one category. There are also more common scales of this, for example:

very positive – positive – neutral – negative – very negative.

Other scales involve the use of 5 ranks – in the way similar to the "five-star" ratings assigned to the objects of interest (restaurants, hotels, goods, etc.). In this case, ratings of less than 2 "stars" characterize the opinion as having a negative tone, and ratings of more than 2 – positive.

2) *Detecting emotions:* This type of sentiment analysis is characteristic of more advanced approaches that aim to model the emotional state associated with some text: for example, fear, anger or joy [13].

Sentiment analysis approaches are usually divided into *lexicon-based* and *machine learning-based*. In addition, *hybrid approaches* stand out, combining rule-based and machine learning approaches [12]. The paper uses methods that implement approaches based on dictionaries and rules. It is argued that they are more beneficial since the material under consideration contains specific lexis. Besides, the number of reviews in question was insufficient for training a custom classifier. The implementations chosen will be discussed in more detail in the next section.

B. Methods

The experiment is based on the application of such libraries as TextBlob [14] and VADER [15] for identification of sentiment polarity in the reviews on online courses. These libraries are rather popular and effective regarding sentiment analysis in English. Namely, VADER shows the highest accuracy (72%) on texts of nature similar to the online reviews – Twitter data [16]. TextBlob's accuracy in case of its application to Twitter data as well was 77.2% [17]. With regards to TextBlob, it is also noted that it performs better in tasks of text annotation. It is found that deep learning models trained on its lexicon present accuracy higher than the ones obtained when VADER and SentiWordNet are used [18].

Then, RAKE-NLTK was used to extract keywords in only positive and only negative reviews correspondingly. The choice of this algorithm is explained by the fact that in pilot experiments it performed better than the ones described in the sections above when applied to each review separately. The following sections will describe the underlying algorithms.

1) *TextBlob:* TextBlob is an open source Python library that is an API for solving a wide range of tasks in the field of natural language processing, one of which is sentiment analysis [19]. The corresponding module provides two ways to determine sentiment: *PatternAnalyzer* which is used by default and *NaiveBayesAnalyzer* which is a classifier pre-trained on the movie reviews corpus. Next, the principle of operation of the PatternAnalyzer analyzer will be considered, since it is an example of the implementation of the approach on dictionaries and rules, which is of interest in this paper. The assessment results in the *polarity score* which takes values in the range from -1 to +1, where texts with negative polarities are considered to contain negative opinions, and those with positive polarities are considered positive.

2) *VADER:* Valence Aware Dictionary for sEntiment Reasoning, or VADER, is a well-documented open source tool specifically designed for sentiment analysis of social media texts [20]. It combines the use of lexicon and rules to produce a sentiment score called *compound*. As the developers note, the dictionary contains not only the words, but also an extensive list of Western-style emoticons (for example, *:-)*), acronyms (e.g., *LOL*, *WTF*.), as well as slang expressions (e.g., *nah*, *meh*). In total, there are more than 7 500 lexical units. A distinctive feature is that each word in the VADER lexicon is rated by 10 independent experts.

The compound score is the normalized sum of the word scores from the lexicon after applying the rules:

$$x = \frac{x}{\sqrt{x^2 + \alpha}},$$

where x is the sum of valence scores that define the polarity and intensity of sentiment for the word on a scale from -4 to +4, $\propto$ is a normalizing constant [21].

The normalized weighted compound score takes values between -1 and +1. The higher the tone score, the more positive the text, and vice versa. To convert a quantitative response to a categorical one, the following thresholds are used as well as in this paper:

- positive tone with *compound >= 0.05*;
- neutral tone at *(compound score > -0.05) and (compound score < 0.05)*;
- negative tone with *compound score <= -0.05*.

C. Results

In this section, using the example of the Zookeeper project, a comparison of libraries will be proposed, and attention will also be drawn to the limitations found in their use for the material under study. For each of their libraries, the highest rated positive and negative reviews are traced, and changes in ratings are noted, including those caused by linguistically determined factors.

1) Comparison of TextBlob and VADER sentiment scores:

The Zookeeper project was chosen for a detailed description, because it contains the largest number of reviews, in connection with which it was supposed to receive more diverse sentimental ratings. The tonal evaluation is defined for each feedback within the project.

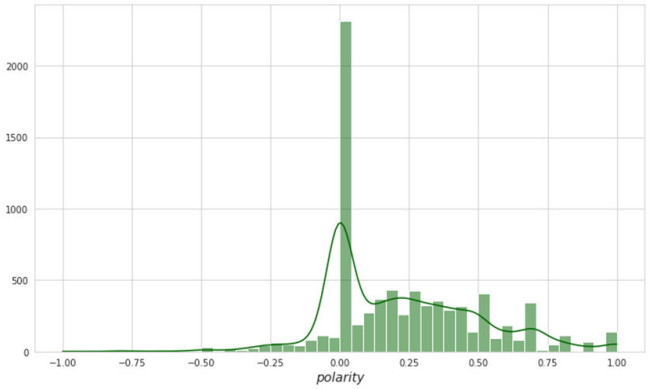

Fig. 1. TextBlob: the distribution of polarity scores (Zookeeper project)

As a result of the TextBlob library application, polarity scores were obtained, mainly falling on the interval from -0.25 to 0.70 (Fig. 1). A significant number of neutral ratings can be noticed, and in general one can characterize tonality as gravitating towards neutral — quite a lot of reviews are distributed in the range from 0.0 to 0.5.

As seen from examples below, as the most positive, i.e. for them polarity = 1.0, TextBlob defined the reviews of short length and indeed containing mainly positive vocabulary:

(1) *Great!!*
(2) *Great course!*
(3) *Perfect*
(4) *This is an excellent course covering in depth of the programming concepts, lot of code practice with bite-size theory.*

However, for some reviews, the score was rather overestimated: the library did not catch the negative shades present in these texts:

(1) *My **biggest difficulty** was learning what type of loops work best for numbers versus strings*
(2) *learning environment is perfect **but increase your site processing speed***

A slight "flexibility" of the system is also observed in the examples of the most negative reviews. These are the sentences where the first part is rather positive, but they were evaluated by TextBlob as uniquely negative (polarity = -1.0):

(1) ***have learned a lot** but i think coding is too boring for me and this session made me quit coding*
(2) *animals in Python can be terrifying*
(3) ***I knew everything** but I did a miserable mistakes.*

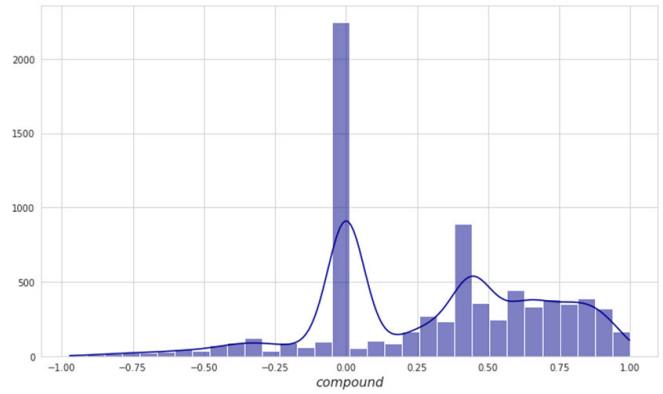

Fig. 2. VADER: the distribution of polarity scores (Zookeeper project)

As a result of using the VADER library, sentiment scores were obtained, mainly falling on the interval from 0.0 to 1 (fig. 2). Compared to TextBlob, there is a decrease in the number of neutral reviews, i.e., having compound = 0.0, and a shift to the right is generally noticeable. It is also interesting how the negative scores presented in the range from -0.90 to 0.0 are presented: they seem to be more varied.

The reviews identified by VADER as the most positive have compound > 0.90. This value was chosen because no reviews with the maximum rating were found. When considering reviews directly, the following is observed: firstly, all reviews are of a significant size, and secondly, exclamation marks and emoticons are used for empathic purposes. For example:

*(1) I got **very nice experience**. But at the last task I had **some difficult**, because the description mentioned just "do like at previous task", but I do not remember what it was at the last task:) It was very unhandy to come back every time to previously lesson **:)** Please, keep it mind:) Also, it was not **enough information** about While True loop for succeeding finish of this course. But other things was **great, thank you.)** You done **great work**. **Really great work**.*

Overall, the VADER is quite accurate, when compared with the empirically assigned score, to be able to determine the sentiment in these reviews.

For the most negative reviews, compound < 0.90 was considered. In the examples given, it seems interesting that, at the lexical level, reviews with negative sentiment often contain stylistically less formal expressions (*had no idea at all*, *biggest trouble*, *bullshit*), as well as contractions (*n't):

*(1) I've learned basic data types of the languange and how to operate with them step by step. Details and details again are extremly important. **Biggest trouble** I had with data conversion; **I had no idea at all** why the code does't work. I was about to give up because **stupid typo error** and my **bad understanding** of the question given.*

*(2) FUCKING TRASH **BULLSHIT** STUPID IDE FUCK YALL STUPID JETBRAINS PROGRAMMERs FUCCCK!!!!!!!!!!!!!!*

In addition, it may be recalled that the system pays attention not only to atypical uses of punctuation marks, but also to case — in this regard, the last example is noteworthy. Difficulty was discovered when analyzing the first review: semantically, it seems to be more positive than the rating given to it by VADER. The prevailing use of the word problem, which is specific to projects on the Hyperskill platform, significantly lowers the sentimental assessment. This is probably due to the fact that, judging by linguistic features, the review was clearly written by a non-native speaker.

Below, examples of the most characteristic reviews will be presented, for which the sentiment assessments of libraries differ, i.e. one library identified the opinion in the review as positive, and the other as negative.

- polarity>0 & compound<0

The tone scores for feedback on the Zookeeper project, filtered by polarity>0 & compound<0, are distributed according to Fig. 3.

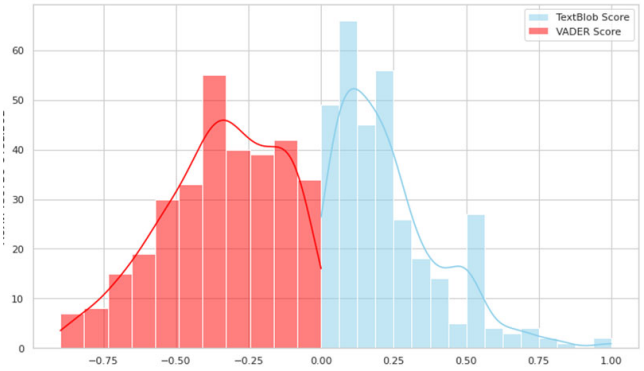

Fig. 3. Sentiment scores distribution for positive (TextBlob) and negative (VADER) reviews

The reviews that were rated positive by TextBlob and negative by VADER contain constructions with negation and contexts similar to ironic ones (*I think this not for beginner :D*). This specificity is better captured by VADER. Also, it seems that if the proportion of text in which the user describes the difficulties and problems encountered is greater, then VADER tends to classify the review as negative, while this is not observed for TextBlob.

- compound>0 & polarity<0

The sentiment scores for feedback on the Zookeeper project, filtered by compound>0 & polarity<0, are distributed according to Fig. 4.

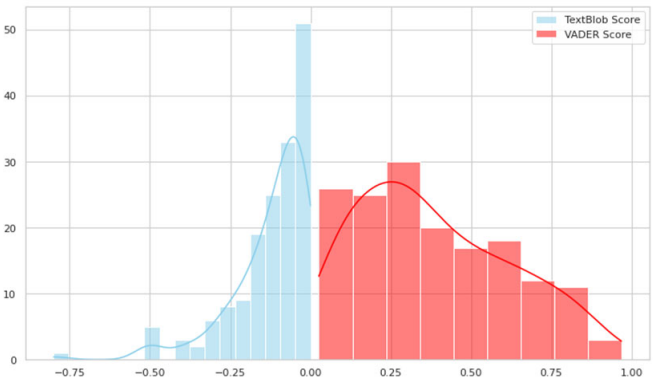

Fig. 4. Sentiment scores distribution for positive (VADER) and negative (TextBlob) reviews

In general, for the TextBlob, contexts with negation (*no great difficulties*, *not as scary*) again present difficulties. It is difficult to explain some cases of lowering of sentiment in the results obtained, however, an example with an informal expression — *...has shocked me!* — is rather interesting. The polarity score goes down, which seems to be due to the fact that the word *shocked* in the lexicon used by the TextBlob only has a negative meaning (sense="struck with fear, dread, or consternation").

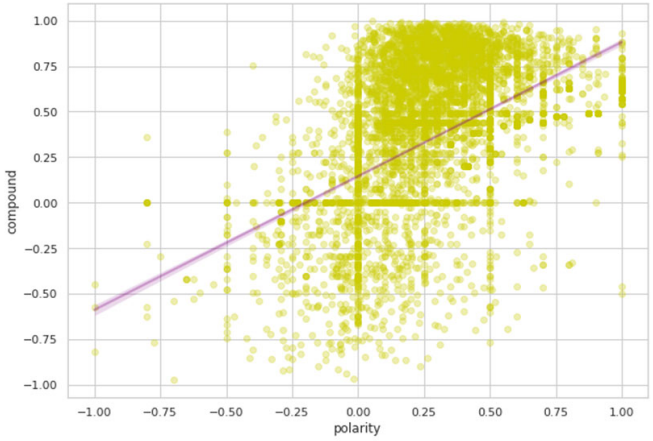

Fig. 5. Correlation between polarity (TextBlob) and compound (VADER) scores

As a result of the comparison, a positive relationship was revealed between the sentimental assessments of the TextBlob and VADER libraries, however, VADER determines the sentiment in the area of negative reviews most accurately (fig. 5). This is largely due to the peculiarities of the library, which is based on a lexicon expanded in accordance with the linguistic characteristics of the texts that make up the Internet segment. The observation that negative contexts are also better defined using VADER seems unexpected, although the rules that take them into account are implemented in both libraries.

2) Extraction of sentiment-determined opinions

One of the tasks of the work is to qualitatively characterize the courses based on positive and negative reviews. In other words, answer the questions – *how do users describe the project? what do they like or dislike?* At this stage of program development, an attempt is made, on the one hand, to describe the opinions of users in the form of a kind of "generalized reflection" on the project, and on the other hand, to validate the results obtained, i.e., evaluate how well the reviews were classified by the selected tool - VADER, and in accordance with the set threshold.

The procedure for extracting characteristics-opinions is carried out as follows:

1) for each feedback in each project, a sentiment score is calculated using VADER;

2) in accordance with the threshold values proposed for VADER in [20] and described above; each response, depending on the compound value calculated for it, is assigned the tag *pos*, *neg*, *neu*, where pos — positive, neg — negative and neu — neutral;

3) 2 keywords are extracted from each review using the RAKE-NLTK library, configured to work with English and using the extended nltk stop dictionary, with the specified parameters;

4) frequency lists of keywords extracted from positive reviews are formed, and the 50 most frequent ones are selected. The frequency approach is often used to identify candidates for aspect terms, so obtaining such information for projects can further help establish categories for aspectual sentiment analysis [11];

5) frequency lists of keywords extracted from negative reviews are formed, and the 50 most frequent ones are selected in a similar way;

6) spaCy POS-tagger is applied to keywords;

7) only those keywords are selected that correspond to the part-of-speech patterns ADJ + NOUN and ADV + VERB.

The choice of these part-of-speech patterns was proposed after the initial analysis of the frequency lists of keywords: the most frequent and informative were the groups "adjective + noun". The decision to include "adverb+verb" patterns can be said to have been a heuristic based on linguistic observation (cf. the presence of evaluative phrases *really like*, *really enjoyed*, etc.).

An example of the results of extracting keywords from positive and negative reviews of the Zookeeper project in the manner described above is presented in Table V. Based on the positive keywords, it can be concluded that the Zookeeper project is a good introduction (*good introduction*, *great introduction*), apparently, into the Python language, because it is taken at the elementary level. This can also explain the extracted negative keywords: *silly mistakes*, *difficulties faced*, *still struggle*.

TABLE V. KEYWORDS EXTRACTED FROM POSITIVE AND NEGATIVE REVIEWS (ZOOKEEPER PROJECT)

	ADJ+NOUN	ADV+VERB
POS	**good start**, **great way**, **good way**, **great experience**, basic concepts, good course, basic knowledge, **good introduction**, basic syntax, **new things**, basic stuff, **better understanding**, **great course**, **good practice**, **great introduction**, **good experience**, **nice way**, first time, many things, **interesting tasks**	**really enjoyed**, **really liked,** already knew
NEG	boolean logic, new things, basic knowledge, **silly mistakes**, many things, little bit, basic concepts, mathematic operators, good experience, basic commands, **main difficulty**, boolean operators, **faced difficulties**, first language, simple things, logical operators, essential basics, **biggest difficulty**	**never give**, fully understand, **still struggle**, **already experienced**

The disadvantage of the proposed method is obvious: the keywords also included topics that users study during the online course, and not just qualitative characteristics. This is especially noticeable for negative keywords. As noted earlier, detecting negative sentiment is not an easy task. Probably,

determining the optimal threshold value for the material under consideration will help to solve this problem. Since it could not be established empirically, it is suggested that in resolving this issue it will be useful to take into account the ratings that are given directly by users after the completion of the project.

V. Conclusion

In this study, we applied opinion analysis methods to analyze user feedback on the learning process. The technologies of the semantic compression method (keyword extraction and processing) allowed us to see an average impression and detailed data on the course, the topics studied, and so on. This picture may differ from the description of the course provided by the creators and be at odds with student expectations. Such data can be used to improve the accuracy of the course representation. A significant part of the work is devoted to the analysis of users' sentiment and their expression of emotions in feedback. The developed methodology of data processing can be used by online course developers for improving content, structure and user experience in general.

At this stage of the study, we are unable to formally estimate the performance. With KWE, there is no gold standard for our data, so we relied on the previous works concerning algorithms performance. The impact of changes made to project descriptions based on the extracted keywords on the user experience will be apparent later, after A/B testing, and will be the subject of further research. Due to technical difficulties, at this stage, we were not able to obtain user ratings for projects and quantify the quality of sentiment analysis as well. However, we conducted a qualitative analysis and investigated the statistical distribution of sentiment, which allowed us to gain some insights.

First, the algorithms of the TF-IDF and YAKE methods were applied, thanks to which it was possible to extract structured information in the form of keywords from feedback on projects as part of a project on the Python programming language on the Hyperskill educational platform.

During the study, it was found that the methods used in the work cope well with the task of extracting keywords, most of which are nouns and adjectives for unigrams, as well as combinations of noun and noun, adjective and noun for bigrams. To avoid repetition and to determine the advantages of each of the methods, only relevant ones were selected from the extracted key ones. Thanks to this, it was possible to establish that using the TF-IDF method it becomes possible to extract not only the complexity of some projects, but also the attitude of the majority of users towards them. It was also found that all the extracted keywords can be conditionally divided into thematic affiliation and subjective experience.It is also important that only 5 out of 19 projects were classified as well-described projects, including almost all the keywords and phrases extracted in the work. This allows us to conclude that it is far from always possible to predict in advance when creating a description of a project those keywords that users who have already completed a particular project will mention.

With regards to sentiment analysis, taking into account the specifics of the material and comparing the effectiveness of the selected tool with TextBlob and user ratings, VADER is a better option to be used for sentiment analysis. In addition, RAKE-NLTK is used to extract keywords from positive and negative reviews. Based on the analysis of frequency lists of keywords and part-of-speech patterns, it is possible to identify some evaluative characteristics of projects (*good project*, *great introduction*, *great experience*, etc.).

However, in both sentiment assessment and keyword extraction, negative reviews are the most difficult. The improvement of the results is seen in the selection of a threshold value that is more optimal for the material under consideration for classifying reviews into positive and negative. For this, it seems necessary to involve the ratings given by users to the project upon its completion. As for keywords, it seems promising to continue working in this direction, but try to identify categories with which they can be correlated.

What is more, it was noted that the negative sentiment is predominantly found in the reviews for the projects that are not well-described on the platform. The descriptions that less correlate with meaningful keywords extracted from the reviews influence on the common learners' experience and the emotions they obtain after the course completion.

In future, it seems to be interesting to look at longer reviews as potentially containing more meaningful insights. Using these keywords, one may also try to categorize frequent words, similar to how it is implemented in terms of aspect-based sentiment analysis. For example, the complexity of the project — keywords that characterize the complexity of the project; project usefulness — keywords that describe how useful the project was. Finally, it is likely that more subjective reviews will contain more vocabulary that qualitatively evaluates projects; verification of this assumption is another direction for further research.

References

[1] R. Deng, P. Benckendorff, and D. Gannaway, ‘Progress and new directions for teaching and learning in MOOCs’, *Comput. Educ.*, vol. 129, pp. 48–60, Feb. 2019, doi: 10.1016/j.compedu.2018.10.019.

[2] ‘JetBrains Academy — Learn programming by building your own apps’, *JetBrains Academy, powered by Hyperskill*. https://hyperskill.org/ (accessed Sep. 19, 2022).

[3] G. N. H., R. Siautama, A. C. I. A., and D. Suhartono, ‘Extractive Hotel Review Summarization based on TF/IDF and Adjective-Noun Pairing by Considering Annual Sentiment Trends’, *5th Int. Conf. Comput. Sci. Comput. Intell. 2020*, vol. 179, pp. 558–565, Jan. 2021, doi: 10.1016/j.procs.2021.01.040.

[4] U. Erra, S. Senatore, F. Minnella, and G. Caggianese, ‘Approximate TF–IDF based on topic extraction from massive message stream using the GPU’, *Inf. Sci.*, vol. 292, pp. 143–161, Jan. 2015, doi: 10.1016/j.ins.2014.08.062.

[5] J. Li, ‘A comparative study of keyword extraction algorithms for English texts’, vol. 30, no. 1, pp. 808–815, 2021, doi: 10.1515/jisys-2021-0040.

[6] W. Zhuohao, W. Dong, and L. Qing, ‘Keyword Extraction from Scientific Research Projects Based on SRP-TF-IDF’, *Chin. J. Electron.*, vol. 30, no. 4, pp. 652–657, Jul. 2021, doi: 10.1049/cje.2021.05.007.

[7] ‘yake: Keyword extraction Python package’. Accessed: Sep. 19, 2022. [Online]. Available: https://pypi.python.org/pypi/yake

[8] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt, ‘YAKE! Keyword extraction from single documents using multiple local features’, *Inf. Sci.*, vol. 509, pp. 257–289, Jan. 2020, doi: 10.1016/j.ins.2019.09.013.

[9] R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, and A. Jatowt, ‘A Text Feature Based Automatic Keyword Extraction Method

for Single Documents', in *Advances in Information Retrieval*, Cham, 2018, pp. 684–691.

[10] S. Rose, D. Engel, N. Cramer, and W. Cowley, 'Automatic Keyword Extraction from Individual Documents', in *Text Mining*, M. W. Berry and J. Kogan, Eds. Chichester, UK: John Wiley & Sons, Ltd, 2010, pp. 1–20. doi: 10.1002/9780470689646.ch1.

[11] E.I. Bol'shakova, K.V. Voroncov, N.E. Efremova, E.S. Klyshinskij, N.V Lukashevich., A.S Sapin, Avtomaticheskaya obrabotka tekstov na estestvennom yazyke i analiz dannyh : ucheb. posobie. Izd-vo NIU VSHE, 2017.

[12] S. Smetanin, 'The Applications of Sentiment Analysis for Russian Language Texts: Current Challenges and Future Perspectives', *IEEE Access*, vol. 8, pp. 110693–110719, 2020, doi: 10.1109/ACCESS.2020.3002215.

[13] P. Nandwani and R. Verma, 'A review on sentiment analysis and emotion detection from text', *Soc. Netw. Anal. Min.*, vol. 11, no. 1, p. 81, Aug. 2021, doi: 10.1007/s13278-021-00776-6.

[14] S. Loria, 'textblob: Simple, Pythonic text processing. Sentiment analysis, part-of-speech tagging, noun phrase parsing, and more.' Accessed: Sep. 19, 2022. [Online]. Available: https://github.com/sloria/TextBlob

[15] C. J. Hutto, 'vaderSentiment: VADER Sentiment Analysis. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, and works well on texts from other domains.' Accessed: Sep. 19, 2022. [Online]. Available: https://github.com/cjhutto/vaderSentiment

[16] M. Al-Shabi, 'Evaluating the performance of the most important Lexicons used to Sentiment analysis and opinions Mining', *IJCSNS*, vol. 20, no. 1, p. 1, 2020.

[17] I. G. S. Mas Diyasa, N. M. I. Marini Mandenni, M. I. Fachrurrozi, S. I. Pradika, K. R. Nur Manab, and N. R. Sasmita, 'Twitter Sentiment Analysis as an Evaluation and Service Base On Python Textblob', *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1125, no. 1, p. 012034, May 2021, doi: 10.1088/1757-899X/1125/1/012034.

[18] M. Mujahid *et al.*, 'Sentiment Analysis and Topic Modeling on Tweets about Online Education during COVID-19', *Appl. Sci.*, vol. 11, no. 18, p. 8438, Sep. 2021, doi: 10.3390/app11188438.

[19] S. Loria, 'textblob Documentation', p. 73, 2020.

[20] C. Hutto and E. Gilbert, 'VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text', *Proc. Int. AAAI Conf. Web Soc. Media*, vol. 8, no. 1, Art. no. 1, May 2014.

[21] A. Abayomi-Alli, O. Abayomi-Alli, S. Misra, and L. Fernandez-Sanz, 'Study of the Yahoo-Yahoo Hash-Tag Tweets Using Sentiment Analysis and Opinion Mining Algorithms', *Information*, vol. 13, no. 3, p. 152, Mar. 2022, doi: 10.3390/info13030152.