Training Temporal and NLP Features via Extremely Randomised Trees for Educational Level Classification

Tahani Aljohani[1], and Alexandra I. Cristea[2]
[tahani.aljohani, alexandra.i.cristea]@durham.ac.uk

Durham University, Durham, UK

Abstract. Massive Open Online Courses (MOOCs) have become universal learning resources, and the COVID-19 pandemic is rendering these platforms even more necessary. These platforms also bring incredible diversity of learners in terms of their traits. A research area called Author Profiling (AP in general; here, Learner Profiling (LP)), is to identify such traits about learners, which is vital in MOOCs for, e.g., preventing plagiarism, or eligibility for course certification. Identifying a learner's trait in a MOOC is notoriously hard to do from textual content alone. We argue that to predict a learner's academic level, we need to also be using other features stemming from MOOC platforms, such as derived from learners' actions on the platform. In this study, we specifically examine *time stamps*, *quizzes*, and *discussions*. Our novel approach for the task achieves a *high accuracy* (90% in average) even with a simple shallow classifier, *irrespective of data size*, outperforming the state of the art.

Keywords: Learner Profiling · MOOC Metadata · Data Size · Decision Trees.

1 Introduction and Related Works

MOOCs attract tremendous numbers of users, due to their free cost, creating a rich diversity of user demographics - like age, gender, education level, etc. However, many face-to-face courses suddenly stopped during the current pandemic of COVID-19 [25], so the majority of new MOOC users this year are those who are trying to find replacements for their suspended classes [23] - making MOOCs an optimal alternative, as they offer classes from the world's top institutions [22]. According to a recent statistical report [23], enrollments at Coursera, a USA MOOC provider, have increased by 640% just between mid-March to mid-April 2020 (10.3 million in 30 days), compared with the same interval in 2019. Another example in the UK is FutureLearn, which has now 13.5 million users [10]. Due to these statistics, having personalised recommendations when delivering these courses to learners, based here on their demographics, becomes vital. Moreover, Learner Profiling (LP) is not only required during the current pandemic, but at all times, since demographic information is in demand for many types of

MOOC research. Although MOOC providers ask users to specify their demographic information during enrollment, the majority of users seem unaware of its value to their learning, and only about 10% fill it in [3]. The main motivation for this study is to offer thus an automatic method for MOOC researchers to extract users' demographics without relying on these, often incomplete, surveys. Specifically, a majority of users who benefit from MOOCs are education seekers. According to a Chinese study [64], investigating reasons behind student motivations in learning in MOOCs, 55% of the participants find MOOCs more interesting for receiving knowledge, 61% of the participants noticed that the repeatability of courses in MOOCs helps them understanding courses' content even deeper, 28% of the participants benefit from MOOC discussion forums for sharing knowledge, 27% of participants prefer MOOCs over other traditional modes of teaching, and 19% of the participants mentioned that the video lectures motivated their enrollment. One of the advantages of MOOCs is providing college credits, via a certificate. The first attempts started in October 2013, when a contract has been entered between Antioch University and Coursera, to license several of the University courses on the Coursera platform, as credits for part of a Bachelor's degree program. Also, in the same year, a course offered as a MOOC, "Innovation and Design Thinking", by the University of Cincinnati, was announced to provide credit for all students on Master's degree tracks [14]. The current pandemic promotes the demand for the online education in the future, as it breaks any spatial or temporal limitations. However, many obvious challenges appear in these platforms. Checking for plagiarism or authorship are some ways that increase trust in online education accrediting. Thus, our study is a step toward achieving such trust in MOOCs.
Natural Languages Processing (NLP) provides an approach for predicting user characteristics, called Author Profiling (AP). AP is data-driven computational linguistics that attempts to extract a user's attributes automatically, and is well-known as a challenging task in the NLP area. AP needs deeper linguistic analysis, typically with many training samples, because the hypothesis of AP is to explore similar linguistic patterns amongst authors who share the same demographics [5]. Moreover, works that have achieved state-of-the-art results in AP usually utilise a large number of linguistic features [20]. This complicates the AP task in practice. Also, online AP research in prior works mainly focused on social networks, and targeted few characteristics such as, gender, age, or native language [20]. Yet, other demographics, such as education level, and some important domains, like education, have received less attention from the online AP community [4] [9]. In MOOCs, traditionally, 61% of the enrollments are education seekers [26], and the education level is well-known to influence learning in learning systems. As claimed by Kaati and his team [15], AP models that were trained on a content of a particular domain significantly underperform when applied to another domain, which means that AP models primarily rely on data used for training. What is more, content-based features usually used hundreds or even thousands of features for classification; ranging from lexical, semantical, to syntactical, and based on grammars, n-grams, frequencies, token levels, etc. This should be very

effective when two objectives are met: enormous text samples from a specific domain; and sufficient power of computational resources. When this is not the case, we propose that AP tasks can be also solved by other approaches, that is, by an in-depth examination of other potential features and metadata available in a specific domain. Regarding the used classifiers in the area, Support Vector Machines (SVM) are the most used for training textual features. Although deep learning models became state-of-art in the NLP field, especially the new generation of deep learning called Transformers, shallow classifiers have outperformed deep learning classifiers like the Bidirectional Encoder Representations from Transformers (BERT), or Long Short-Term Memory (LSTM), accordingly to recent AP studies [19]. Based on results obtained from an AP competition on predicting the gender of authors from their written texts, the best proposed technique was combining character n-grams, word n-grams, and function words, then trained them via an SVM classifier [19]. In this study, we address the learner profiling (LP) task, namely, predicting learners' level of education in MOOCs. The main contributions of this study are: i) we are the first to predict the educational status in MOOCs using NLP/ML approaches; ii) we investigate available MOOC metadata comprehensively for the task; iii) this is the first time the AP approach is linked with MOOC domain-related data, not only based on NLP features; iv) in spite of the simplicity of the applied features, we obtain a high accuracy regardless of data size, even with inexpensive classifiers.

2 Data Set

We have collected a large scale dataset [2], [1] from FutureLearn, extracted from 4 courses delivered by the University of Warwick from 2013 to 2015. These courses bring together different topic domains (Computer Science, Psychology, Literature). Each course has been offered multiple times (called 'runs'), with 21 runs in total, and are of different durations, as follows: Big Data (BG): three runs and nine weeks duration each. Babies in Mind (BM): six runs and four weeks duration each. The Mind is Flat (MF): seven runs and six weeks duration each. Shakespeare (SH): four runs and ten weeks duration each. In each week, learners learn a 'learning unit' that includes several tasks (called 'steps'), which can be a video, article, quiz, or discussion. The system generates a unique ID for each learner, and also timestamps which are: time of enrollment, time of submission of an answer, and time of accessing a step; The first time visiting a step (Visited), and when learners press the "Mark as Completed" button (Completed). The system also stores numerical and Boolean data related to learners' responses to different questions during a course. Learners in our data collection have accessed 2,794,578 steps. For our experiment, we have 12934 learners (who declared their level of education) out of the total of learners in our data set (245,255 learners), categorised as: Bachelor (B), Master (M), and Doctorate (D). We have collected the metadata from enrollments, quizzes, steps, and comments. Thus, we obtained very different data sizes, as there were different case scenarios of users' activities.

For example, some users watched videos but did not answer quizzes, some wrote comments while others did not, and so on, see Table 1. However, we fixed this

	Enrollments			Quiz			Time Spent			Comments		
Course	B	M	D	B	M	D	B	M	D	B	M	D
BD	870	737	160	5250	4860	1576	544	458	117	2326	2052	526
BM	1561	932	156	10065	6522	971	980	653	98	4650	2300	298
MF	2237	1424	269	48761	31015	6668	1249	836	187	9232	5844	2717
SH	2503	1747	388	136919	93311	22547	1802	1328	312	21363	14997	5887

Table 1: Courses: BD, MB, MF, SH; levels: (B)achelor, (M)aster, (D)octor

issue by filling in missing data, as will be explain in section 3.5. The size and richness of this data arguably allows for generalisability of our study.

3 Methodology

3.1 Feature Extraction

We have, comprehensively, studied potential features that can be extracted from our rich MOOC data, and can contribute to the level of education prediction. This feature extraction process was based on three conditions:

1. ***Existence of Labels***. Features should belong to learners who have declared their education level. This is essential because our study is basically based on supervised learning techniques.
2. ***Size of Feature's Samples***. Some metadata are available in our dataset, but they do not meet the current condition. For example, the time at which a comment was moderated for inappropriate or offensive content, is available in our data; however, when we tried to extract it, we found that it reflected upon only three learners; which is not adequate for the training process.
3. ***Relatedness***. Some available metadata in our dataset has not been extracted, such as question number or comment ID, since they obviously are not predictors for our task.

As a result, our extracted features can be classified into four categories:
-**Enrollment Features**. We extracted date of enrollment for each learner (enrolled-at [timestamp] – when the learner enrolled).
- **Quiz Features**. We extracted date of submitting answers (submitted-at [timestamp]), responses data (which is the answer number selected, reflecting their ordered position [numerical]), and correctness data (for the correctness of the responses [true or false]).
- **Time Spent Features**. We extracted two types of dates related to steps: first visited-at (when the step was first viewed by the user[timestamp]), and last-completed-at (when the step was last marked as complete by the user[timestamp]).

-**Comment Features**. We extracted comments written by a learner ([text]), date of post comments ([timestamp]), and number of likes attributed to each comment.

3.2 Feature Engineering

All the extracted features are in raw format, so we normalised these features before feeding them into machine learning models. For example, we removed URLs, since URLs have a standard structure that is not influenced by a user's writing style. Also, we dropped any duplicated comments and kept only the first comments (the original comment written by a learner). This was because we have found some learners copy and paste other learners' comments, which meant that these copied comments were not written in their own personal writing style. In addition, we applied simple and advanced NLP techniques to the comments to convert them into textual representations that are commonly utilised for AP. All features have been converted to numerical forms, as follows:

1. **Temporal Features (5 Feature Sets)**:
   - i *Hour*: value of time hour within a day (values between 0 to 23).
   - ii *Month*: value of that month within a year (values between 1 to 12).
   - iii *Week Day*: value of that day within a week (values between 1 to 7).
   - iv *Month Day*: value of that day within a month (values between 1 to 31).
   - v *Year Day*: value of that day within a year (values between 1 to 365). See Table 2 for temporal features symbols.
2. **Simple Textual Features (9 Feature Sets)**:
   - i *Character Count*: Total number of characters in a comment.
   - ii *Word Count*: Total number of words in a comment.
   - iii *Word Density*: Average length of words in a comment.
   - iv *Sentence Count*: Total number of sentences in a complete comment.
   - v *Sentence Density*: Average length of a sentences in a complete comment.
   - vi *Punctuation Count*: Total number of punctuation marks in a comment.
   - vii *Upper Case Count*: Total number of upper count words in a comment.
   - viii *Title Word Count*: Total number of proper case (title) words in a comment.
   - ix *Stopword Count*: Total number of stop words.
3. **Advanced NLP Features (2 Feature Sets)**:
   The advanced NLP features are extracted by pythonic NLP libraries:
   - *Part of Speech(POS)*: To extract the part of speech tags [24], we used the standard Textblob library. Then, we have calculated the total number of nouns, verbs, adjectives, adverbs, and pronouns in each comment. See Table 2 for Tag symbols.
   -*Sentiment Analysis(SA)*: To extract the SA polarity [18], we used the standard NLTK which assigns three polarities: positive (1), negative(-1), and neutral(0).

4. **Time Spent Feature**:
This is computed via the difference between the time when the a learner has fully completed the step (C), and the first time that learner visited that step (V), in seconds:

$$TimeSpent = C - V \tag{1}$$

Name	Symbol
Hour	hour[enrolment(e_hour), quiz(q_hour), comment(c_hour)]
Month	month[enrolment(e_month), quiz(q_month), comment(c_month)]
Week Day	week_day[enrolment(e_week_day), quiz(q_week_day), comment(c_week_day)]
Month Day	month_day[enrolment(e_month_day), quiz(q_month_day), comment(c_month_day)]
Year Day	year_day[enrolment(e_year_day), quiz(q_year_day), comment(c_year_day)]
Noun	noun_count['NN','NNS','NNP','NNPS']
Verb	verb_count['VB','VBD','VBG','VBN','VBP','VBZ']
Adjective	adj_count['JJ','JJR','JJS']
Adverb	adv_count['RB','RBR','RBS','WRB']
Pronoun	pron_count['PRP','PRP$','WP','WP$']

Table 2: Description of POS and Temporal Features Symbols in our Study

3.3 Models

One of our study objectives is to consider less expensive computational classifiers rather than expensive and complex models like deep learning algorithms.This is practically possible since our approach has included a feature engineering step. We have trained our labeled examples on many different supervised shallow learning algorithms. We have employed models that have been commonly used in the AP area: Support Vector Machine, Naïve Bayes, Decision Trees, Random Forests, Logistic Regression, Multilayer Perceptron, and K-Nearest Neighbors Learningmodels. We are presenting in this paper only results of the top performing model, which is the Decision Trees model, particularly, the Extra Trees (ET) Classifier; a decision tree-based classifier that learns in an ensemble way, which is standing for Extremely Randomised Trees. This algorithm is fundamentally an ensemble of decision trees, similar to other DT-based models, such as the random forest. However, ET is built by more unpruned decision trees, more than random forest, and the prediction is based on majority voting if the task is a classification [11]. The advantage of this algorithm is it fits every single decision tree to the whole training dataset, inside of a bootstrap sample of the training dataset, which is the case in the random forest. This creates more robust and better generalisation performance[11]. The maximum size of tree depth in ET, by default, is none, which means trees keeping expanding till all nodes are pure - see ET Algorithm 1.

3.4 Baseline Models

For comparison purpose, we employed three baseline models that are commonly used for text classification tasks (texts are comments of learners), which are:

Algorithm 1: Extremely Randomised Trees Algorithm Procedure

begin
  Split a node(S):
  - inputs are the local learning subset (S) corresponding to the node we want to split
  - output: a split $[a < a_c]$ or nothing; (a= attribute)
  if *Stop split(S) is False,* **then**
    -Select K attributes: $a_1, ..., a_K$ among all non constant (in S) candidate attributes
    -Draw K splits $s_1, ..., s_K$, where s_i = Pick a random split(S, a_i), $\forall i$ = 1,..., K
    -Return a split $s\star$ such that Score($s\star$, S) = max_i=1,...,K Score(s_i, S)
    else
      return nothing
    end
  end
  begin
    Pick a random split(S,a)
    - Inputs: a subset S and an attribute a
    - Output: a split.
    -Let a^S_{max} and a^S_{min} denote the maximal and minimal value of a in S;
    -Draw a random cut-point a_c uniformly in [a^S_{min} , a^S_{max}]
    - Return the split $[a < a_c]$
    begin
      Stop split(S):
      - Inputs: a subset S
      - Output: a boolean
      if $|S| < n_{min}$, **then**
        return TRUE;
        if *all attributes are constant in S:* **then**
          return TRUE;
        end
        if *the output is constant in S:* **then**
          return TRUE;
        end
        else
          return FALSE
        end
      end
    end
  end
end

- **Term Frequency-Inverse Document Frequency (TF-IDF)**: Simple and old-fashioned, but also NLP state-of-the-art. It is a lexically-dependent, but semantically independent technique. For our study, we applied both character n-grams (n= 3,6) and word n-grams (n= 1,2), which are the best performing n-grams settings employed for AP in the PAN evaluation campaign [6]. The next equation explains a standard TF-IDF technique mathematically:

$$TF\text{-}IDF(t, d, D) = TF(t, d) \times IDF(t, D) \tag{2}$$

Where t is terms in a comment; d a comment; and D a collection of comments.
- **Word2vec:** First neural network-based modeling approach in NLP [17], which is a semantics-dependent, but context-independent embedding. We used the skip-gram-600 model (one of the word2vec algorithms), which has two layers of shallow neural networks. It consists of average word vectors, that are built based on training on a corpus of 50-million tweets [12]. In the skip-gram model, the conditional probability P is calculated for context words w_o and for a central

(target) word w_c, by a softmax operation on the vector v inner product.
- **Bidirectional Encoder Representations from Transformers (BERT)**: A transformer model, which is the cutting edge language model in NLP nowadays. It is a context-dependent embedding. BERT is a complex neural network architecture, and its large version includes: 24 layers, 1024 hidden states, 16 heads, 340M parameters [8].

3.5 Data Normalisation

After merging the features for each learner, we noticed some missing values, because not all learners have done all activities in each step. So, by using a module from Scikit-Learn (SimpleImputer [21]), we filled the missed values by adding the average value of each feature. This step is important for creating vectors with fixed lengths for machine learning classifiers. Also, our data are not in balance, so we balanced them via the popular Synthetic Minority Oversampling Technique (SMOTE) [7]. Next, we split our data set into training (80%), and testing (20%). We further shuffled and stratified for better learning performance [16]. Finally, we examined the extracted features individually, and as combinations. Fig. 1 represents the general workflow of our experiments, visually.

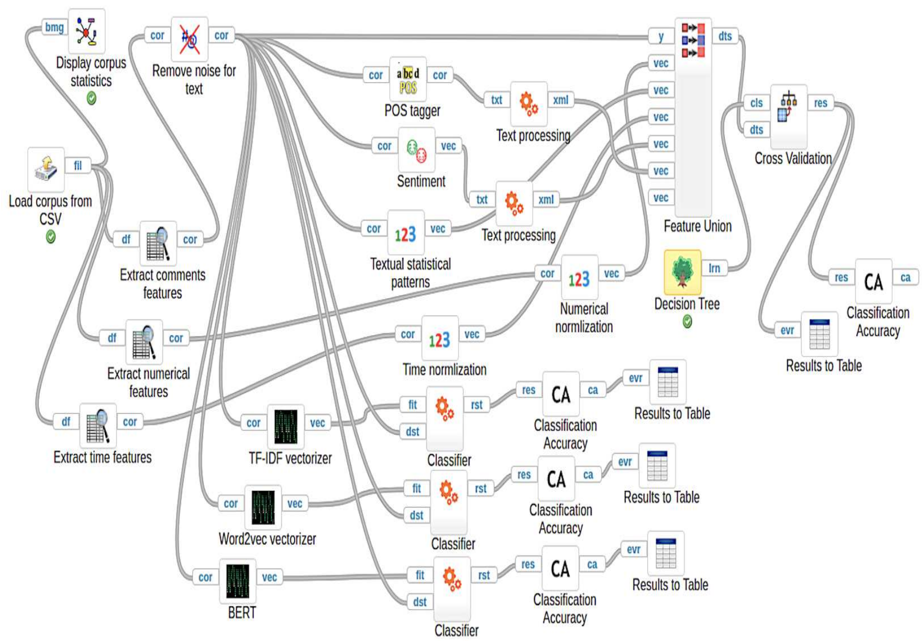

Fig. 1. General Workflow of our Level of Education Prediction Approach in MOOC

4 Results and Discussion

Firstly, we applied the baseline models to predict the level of education of learners only based on comments (traditional way of solving NLP tasks, in general).

We found that using comments alone, based on these models, did not provide satisfactory results. This could be because comments in our data were dominated by course context, which is more representative of courses rather than learners. This may have affected the performance of these NLP state-of-the-art algorithms in our study. BERT's performance in classifying the learners was the lowest. Word2vec performed better than BERT and TF-IDF at character-level, but similar to TF-IDF at word-level. Next, we examined the extracted features in terms of their performance. Despite the simplicity of textual representations that we applied in our experiments, they performed significantly better than the text representations via BERT, TF-IDF, or Word2vec. This supports our initial assumption that using simple and basic textual features could solve our research problem.

Approach	**BD**	**BM**	**MF**	**SH**	**Average**
TF-IDF (char)	0.75	0.78	0.62	0.68	0.71
TF-IDF (word)	0.80	0.84	0.75	0.66	0.76
Word2vec	0.76	0.83	0.75	0.68	0.76
BERT	0.76	0.87	0.80	0.81	0.81
Enrollment + ET	0.67	0.78	0.74	0.68	0.72
Comment + ET	0.85	0.94	0.90	0.88	0.89
Quiz + ET	0.84	0.89	0.84	0.72	0.83
Time Spent + ET	0.82	0.85	0.87	0.84	0.85
Time Spent + Comment + ET	0.87	0.91	0.84	0.97	**0.90**

Table 3: Overall Accuracy per Feature Category and Course, in addition to Baseline Models

Furthermore, we found that MOOC metadata also outperforms baseline models, except for enrollment features. Time-spent features, as well as comment features both achieved highest accuracies, thus we combined them, and this combination obtained the best accuracy compared to all models and settings in our experiments. With respect to machine learning models, the Extra Trees (ET) achieved highest performance for all of our experimental settings, so we are discussing in this paper only results obtained by the ET classifier - see Table 3, which reports ET overall accuracy per course and per feature category. These results are validated by using 10-fold Cross-Validation (CV), which is well known to be used for avoiding the over-fitting issue [13]. In each iteration (k), a single accuracy is estimated, then all accuracies are averaged to get the final accuracy (A). 10-Fold CV is given by the following formula (k-Fold CV accuracy; $k = 10$):

$$Accuracy = \frac{1}{k}\sum_{i=1}^{k} A_i \quad (3)$$

Finally, we evaluated the best obtained results, after combining time spent and textual features, comprehensively and realistically. So, we applied three popular performance measurements: F1-score, precision, and recall. This is an important evaluation step, since our data is not balanced, and it is necessary to not only consider overall accuracy results, which could be strongly biased. We reported results of these three evaluation measurements per category, allowing clear exposure of minority classes, see Table 4).

Course	Acc.	Precision			Recall			F1-score		
		B	M	D	B	M	D	B	M	D
BD	0.87	0.86	0.87	0.90	0.91	0.86	0.76	0.89	0.87	0.82
BM	0.91	0.91	0.91	0.91	0.95	0.87	0.79	0.93	0.89	0.84
MF	0.84	0.85	0.87	0.72	0.91	0.76	0.76	0.88	0.81	0.74
SH	0.97	0.97	0.96	0.98	0.97	0.96	0.95	0.97	0.96	0.96
Average	0.8975	0.8975	0.9025	0.8775	0.935	0.8625	0.815	0.9175	0.8825	0.84

Table 4: Detailed Results (F1, Precision, and Recall) per Course/ Class, and based on Time Spent and Comment Combined Features

5 Conclusion

We solve our Learner Profiling (LP) text classification problem, even though presented in a domain with weak textual representation about authors (MOOCs), with very simple metadata available in the domain. Our new proposed LP model doesn't only obtain high performance, but also shows that this task can be performed via inexpensive computational algorithms, regardless of data size. Our results demonstrate that the selected features are so representative that they work well even with extremely unbalanced data. We also show that using state of the art NLP models is not supportive enough for what is supposed to be mainly a text classification task, due to the domain conditions. For future work, we plan to experiment with other traits of LP in MOOCs, due to the specific domain-related challenges it poses.

6 Acknowledgment

We gratefully acknowledge funding support from the Ministry of Education of Saudi Arabia for this research.

Training Temporal and NLP Features via Extremely Randomised Trees 11

References

1. Aljohani, T., Cristea, A.I.: Predicting learners' demographics characteristics: Deep learning ensemble architecture for learners' characteristics prediction in moocs. In: Proceedings of the 2019 4th International Conference on Information and Education Innovations. pp. 23–27 (2019)
2. Aljohani, T., Pereira, F.D., Cristea, A.I., Oliveira, E.: Prediction of users' professional profile in moocs only by utilising learners' written texts. In: International Conference on Intelligent Tutoring Systems. pp. 163–173. Springer (2020)
3. Aljohani, T., Yu, J., Cristea, A.I.: Author profiling: Prediction of learners' gender on a mooc platform based on learners' comments. International Journal of Computer and Information Engineering **14**(1), 29–36 (2020)
4. Alroobaea, R.: An Empirical combination of Machine Learning models to Enhance author profiling performance. International Journal of Advanced Trends in Computer Science and Engineering **9**, 2130–2137 (2020). https://doi.org/10.30534/ijatcse/2020/187922020
5. Argamon, S., Koppel, M., Pennebaker, J.W., Schler, J.: Automatically profiling the author of an anonymous text. Communications of the ACM **52**(2), 119–123 (Feb 2009). https://doi.org/10.1145/1461928.1461959, `https://doi.org/10.1145/1461928.1461959`
6. Basile, A., Dwyer, G., Medvedeva, M., Rawee, J., Haagsma, H., Nissim, M.: N-gram: New groningen author-profiling model. CoRR **abs/1707.03764** (2017), `http://arxiv.org/abs/1707.03764`
7. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. J. Artif. Int. Res. **16**(1), 321–357 (Jun 2002)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of ACL. pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, `https://www.aclweb.org/anthology/N19-1423`
9. Estival, D., Gaustad, T., Hutchinson, B., Pham, S., Radford, W.: Author profiling for english and arabic emails. In: Association for Computational Linguistics (2007)
10. FutureLearn: Online courses and degrees from top universities, `https://www.futurelearn.com/`
11. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine learning **63**(1), 3–42 (2006)
12. Hsieh, F., Dias, R., Paraboni, I.: Author profiling from Facebook corpora. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan (May 2018), `https://www.aclweb.org/anthology/L18-1407`
13. James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning. Springer Texts in Statistics Book (2013)
14. Jaschik, S.: Moocs for credit. Inside Higher Ed **23** (2013)
15. Kaati, L., Lundeqvist, E., Shrestha, A., Svensson, M.: Author profiling in the wild. In: 2017 European Intelligence and Security Informatics Conference (EISIC). pp. 155–158 (2017). https://doi.org/10.1109/EISIC.2017.32
16. Khamsan, M., Maskat, R.: Handling Highly Imbalanced Output Class Label. Malaysian Journal of Computing **4**, (2):304 (2019). https://doi.org/10.24191/mjoc.v4i2.7021

17. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space. vol. abs/1301.3781 (2013)
18. NLTK: nltk.sentimentpackage — nltk 3.5 documentation (2020), https://www.nltk.org/api/nltk.sentiment.html#module-nltk.sentiment
19. Pardo, F.M.R., Rosso, P., Charfi, A., Zaghouani, W., Ghanem, B., Sánchez-Junquera, J.: Overview of the track on author profiling and deception detection in arabic. In: FIRE (2019)
20. Pardo, F.M.R., Rosso, P.: Overview of the 7th author profiling task at pan 2019: Bots and gender profiling in twitter. In: CLEF. Psychology, Computer Science (2019)
21. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., etc., O.G.: Scikit-learn: Machine learning in python. The Journal of Machine Learning Research **12**, 2825–2830 (Nov 2011)
22. Robson, D.: Online learning: how to acquire new skills during lockdown. The Guardian (2020), https://www.theguardian.com/education/2020/apr/19/online-learning-how-to-acquire-new-skills-during-lockdown
23. Shah, D.: Highlights from coursera partners conference 2020. The Report by Class Central (2020), https://www.classcentral.com/report/coursera-conference-2020-highlights/
24. TextBlob: Textblob: Simplified text processing (2020), https://textblob.readthedocs.io/en/dev/
25. WHO: Coronavirus, world health orgnization (2020), https://www.who.int/health-topics/coronavirus#tab=tab1
26. Zhenghao, C., Alcorn, B., Christensen, G., Eriksson, N., Koller, D., Emanuel, E.J.: Who's benefiting from moocs, and why. Harvard Business Review (2015), https://hbr.org/2015/09/whos-benefiting-from-moocs-and-why