International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:14, No:1, 2020

Author Profiling: Prediction of Learners' Gender on a MOOC Platform Based on Learners' Comments

Tahani Aljohani, Jialin Yu, Alexandra. I. Cristea

***Abstract*—**The more an educational system knows about a learner, the more personalised interaction it can provide, which leads to better learning. However, asking a learner directly is potentially disruptive, and often ignored by learners. Especially in the booming realm of MOOC Massive Online Learning platforms, only a very low percentage of users disclose demographic information about themselves. Thus, in this paper, we aim to predict learners' demographic characteristics, by proposing an approach using linguistically motivated Deep Learning Architectures for Learner Profiling, particularly targeting gender prediction on a FutureLearn MOOC platform. Additionally, we tackle here the difficult problem of predicting the gender of learners based on their comments only – which are often available across MOOCs. The most common current approaches to text classification use the Long Short-Term Memory (LSTM) model, considering sentences as sequences. However, human language also has structures. In this research, rather than considering sentences as plain sequences, we hypothesise that higher semantic - and syntactic level sentence processing based on linguistics will render a richer representation. We thus evaluate, the traditional LSTM versus other bleeding edge models, which take into account syntactic structure, such as tree-structured LSTM, Stack-augmented Parser-Interpreter Neural Network (SPINN) and the Structure-Aware Tag Augmented model (SATA). Additionally, we explore using different word-level encoding functions. We have implemented these methods on Our MOOC dataset, which is the most performant one comparing with a public dataset on sentiment analysis that is further used as a cross-examining for the models' results.

***Keywords*—**Deep learning, data mining, gender predication, MOOCs.

I. Introduction

A MOOC is an educational platform providing a way to democratise knowledge. It is usually providing free (or very cheap) education to a large number of users [1], [2]. Owing to this phenomenon, learners in MOOCs are very varied in terms of age, gender, location, employment status, etc. Due to this diversity, the MOOC environment becomes difficult to navigate [3], which negatively impacts on the learning experience. In order to improve this critical avenue of no-barriers education, it is important to build personalised recommendations for learners, based on their needs. Demographics are important potential inputs into this recommendation. However, whilst most MOOCs provide the opportunity to their learners to specify demographic data about themselves (including gender), the actual percentage of learners who fill-in these data is extremely low (about 10% [3]). Thus, adaptive education and other services over MOOC platforms based on such data would only be applicable to very few – unless automatic methods for adding demographics to users are explored. Here, we specifically target the understudied area of automatically extracting the gender in MOOCs, to serve as a means to design customised recommendations. One of the most common metadata used for analysing MOOC platforms, due to its wide application and use, is the discussion forum [4]. Forums are used for learning and social interactions, providing rich metadata to study learners and their needs. Thus, our main umbrella research question is: How deep learning methods can be designed to predict the gender of learners in MOOCs, based on the comments they exchange.

The main contributions for our work are:

- Finding the highest accuracy model for author profiling, by exploring the cutting-edge state-of-the-art for syntactic models (SATA, SPINN, tree-structured LSTM models) and
- Applying them to author-profiling (here, gender).
- Using larger datasets than in previous literature (almost double in size when compared with other datasets [23]).
- Importantly, applying author profiling to a different domain, that of education.

II. Related Works

A. Demographics in MOOCs

Works in MOOCs concerning demographics of learners include, e.g., [5], where both gender and level of education were compared to the length of active periods and certification rate among learners. They found that females were more active in courses in general and obtained higher certification rates than males in non-science courses. Reference [6] applied natural language processing (NLP) methods onto post-course response surveys regarding learners' participation in course materials, for diagnosing demographic factors behind students' dropout [7]. Their survey evaluated the influences on the motivation to complete the course based on demographics. They used pre-course surveys to gather learners' demographics to find out reasons of learners' completion versus dropping out; however, they used the pre-course survey also for learners' demographics, such as prior education/ experience.

In order to tackle the content of discussion forums in MOOCs for researches, several approaches have been introduced. The most popular approach is classification [3]. This includes classifying posts automatically, by grouping them into topic categories [8], [9]. The similarity between course contents and learners' posts has also been investigated

Tahani Aljohani is with the Durham University, United Kingdom (e-mail: tahani.aljohani@durham.ac.uk).

in MOOC discussion forums. For example, [10] attempted to classify and group posts based on their relationship to weekly lectures. Regardless of these researches' specific goals, the main concept in almost all of them is classifying posts and providing tools to increase the likelihood of learners' completion of courses and/or ultimately enhance the learning experience and outcome. In our work, we investigate these posts from a different angle. Our research target is the heterogeneousness of MOOC environments, in terms of their learner demographics. Many studies were concerned with students' classification in MOOCs, but almost all of these studies used pre-course open responses to identify learners' characteristics, to be utilised later for different research aims. However, there could be a bias in using such pre-course surveys, as well as sparse data, in the case of non-response. This is certainly the case of MOOCs, where, as said, only a very low percentage of the learners disclose their demographic data. Instead, we aim to predict these learner characteristics automatically, particularly here the gender of leaners.

B. Author Profiling

Author Profiling (AP) aims to predict an author's demographic characteristics automatically, such as gender, age, or occupation, by using texts written by the author [11]. It is a popular technique, as it can be applied in many critical domains, like security, marketing forensics, as well as education [12]. AP is handled by the linguistics and NLP community [13], belonging to the family of text mining problems, as a subfield of computational stylometry [14]. Computationally, AP is considered a *Classification Task* that uses a ground truth dataset (a labelled dataset). Building a classification model for AP often relies on extracting a set of features from authored texts. This approach mainly depends on the fact that authors' traits can be inferred from his/her writing, by studying the writing style of the author [15]. In general, building a classification model for AP is heavily reliant on textual features. These can be classified into five levels of data representation: *lexical*, *semantic*, *syntactic*, *structural*, and *domain* (or content) specific [16], [17]. The majority of AP studies rely heavily on lexical features.

In this research, we focus instead on the less explored area of syntactic text representation, taking advantage of the complexity of the language structure, to better handle gender profile classification.

III. Text Classification

For text classification, many different vector representation forms that extract text-based features are handled in the literature [20], [21], [26]. ranging from lexical representation, semantic representation, to syntactic representation. In most works, sentences are encoded commonly as sequence-based tokens and local region-based tokens.

In *recurrent neural network* (RNN), such as LSTM [18], information is accumulated sequentially over a sentence.

In *convolutional neural networks* [19], [20] information is accumulated using filters, to extract information from short local sequences of either words or characters.

Apart from these two approaches, a linguistically motivated model, called *recursive neural network*, can be used to model the complex data structure in human language [21], [22].

Among these three types of models, tree structure models are arguably the most relevant in NLP-tasks, over text containing several sentences, as language meaning is naturally constructed in a tree/recursive form [23]. Thus, a group of models are developed to leverage the advantages of RNN and recursive neural network, which are called Syntactic Supervision Learning Models. These form the basis of models we will be using, modifying and analysing in our current work, as further explained.

IV. Models

In the area of NLP, information can be extracted in form of a tree topology [24], and this due to the hierarchical syntactic structure of sentences representation [25].

In this research, three types of syntactic supervision learning models are used for text classification and AP, which are Tree-LSTM, SPINN and SATA-Tree-LSTM. Tree-LSTM is the first model of this kind to be presented to pass structured information over a sequence [26].

One of the most promising and highly cited methods in structured language processing on syntactic supervision learning is called SPINN [27]. SPINN increased the speed of learning for tree-structured models, allowing thus handling large-scale NLP tasks, since previous models could not support batched computation. However, the major problem for recursive neural networks still remains that the network only reaches local optimisation at each node, instead of reaching a global optimum at the root of the tree. SPINN introduced a solution called *tracker*, which aims to summarise the sentence information during training. This information provides higher accuracy; however, it can only summarise limited information in a sentence.

In 2019, the SATA was proposed, addressing this limitation, using a recursive neural network. It introduced additional information and used a separate LSTM tree to model the sentence, which empirically reached a better optimum over the tree [28].

A.RNN LSTM and Tree LSTM

The LSTM architecture, introduced in 1997 [18] to solve the vanishing and exploding gradient problems in standard RNN architecture, uses complex gates to control how information is stored through time. This allows capturing longer time dependencies and a more complex data structure than RNN, rendering it a good tool for encoding information [29] and analysing time series.

One of the challenges for applying LSTM to model sentence representation in NLP is that human languages are complex in their nature and the length for each sentence could vary between one single word to 20-30 words or even longer. Hence LSTM failed to capture the rich variation in natural languages, since it used a sentence as a plain sequence. To solve this, researchers suggested applying a tree-structured LSTM architecture, allowing the neural network to achieve a

better syntactic representation of the sentence information (Fig. 1) and hence improved performance for sentence representation [26], especially for longer sentences [30].

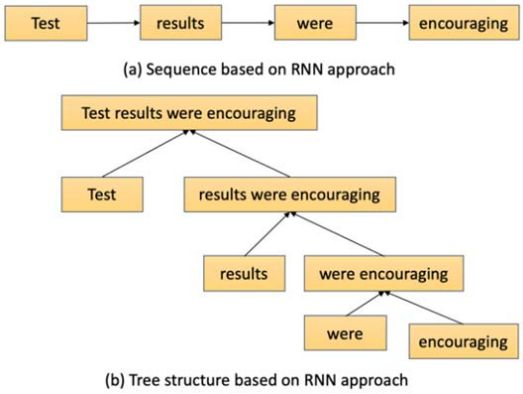

Fig. 1 Sequence based RNN and tree structured RNN for the same sentence, applied on sentences from our MOOCs

Whilst the syntactic representation of text provides a comprehensive means for interpreting a sentence's meaning, these models have been explored only marginally for text classification, and have not yet been applied at all for AP, to the best of our knowledge. Hence, tree-structured LSTM represents a promising model we explore first, and Fig. 1 represents a sample of this approach applied to our data.

In a standard tree-structured LSTM cell, the composition functions are as follows:

$$\begin{bmatrix} i \\ f_l \\ f_r \\ o \\ g \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \left(w \begin{bmatrix} h_l \\ h_r \end{bmatrix} + b \right) \quad (1)$$

$$c = f_l \odot c_l + f_r \odot c_r + i \odot g \quad (2)$$

$$h = o \odot tanh(c) \quad (3)$$

where in the Recurrent Dimension (R^d) $h, c \in R^d$ refers to the hidden state and cell state, respectively. In the current cell; in tree-structured LSTM, $h_l, h_r, c_l, c_r \in Rd$ represent the hidden states and cell states of a pair of child nodes (left and right); $g \in R^d$ refers to the composed inputs from both children and $i, f_l, f_r, o \in R^d$ represent input gate, two forget gates and an output gate, respectively. These two separate forget gates from two children allow the network to choose which information to forget in each child, which captures a more complex representation of the information from the same sentence. $w \in R^{5d*2d}$ and $b \in R^{5d}$ are trainable parameters in the model, σ and *tanh* refer to sigmoid and hyperbolic tangent functions, which apply non-linear transformations before the gate information is updated, whilst ⊙ is the element-wise multiplication symbol, as the dimensionality of elements on both sides is the same. The equations here refer to a binary tree; however, tree-structured LSTM is not limited to two-children cases and can be easily extended to multiple children cases, due to the flexible nature of the recursive neural network. In this research, we adopted a binary tree setting, which is mostly used in related literature.

B. SPINN

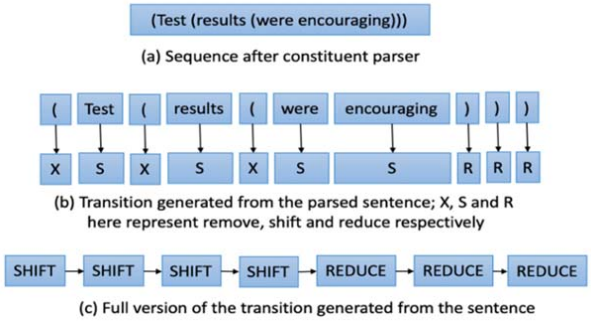

Fig. 2 Shift-reduce parser on a sentence [27]

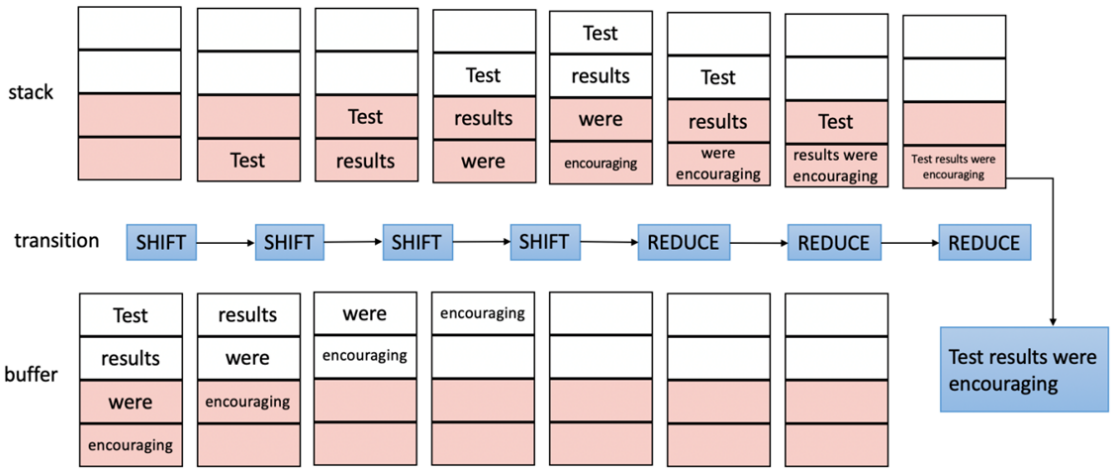

Fig. 3 SPINN model sentence encoder workflow chart [27]

Earlier tree LSTM models used in NLP are well-known for having a long training time and being difficult when applying batch-computation, compared to other neural network architectures, due to the diverse complex structure of sentences. The most recent advances of the state of the art are represented by the SPINN model, introduced in 2016 [27], which allows for efficient recursive neural network training, by adopting the idea of a *shift-reduce* parser from the compiler [31].

The SPINN model provides a systematic way to reconstruct the complex syntactic structure of the language, by reading it from left to right with the help of a shift-reduce parsing algorithm [31]. It takes a sequence of inputs with length *N* and converts it to *2N − 1* length transitions (Fig. 2); for sentence output from the language parser [32], each character is either remove (X), shift (S) or reduce (R). Then, the sequences of words from the sentence and related transitions are fed into the SPINN model. To encode the complex structure of the tree, two data structures are used, which are called *stack* and *buffer*, both of size *N*.

In the beginning, the sequence of inputs is fed into the buffer in order; when the transition is SHIFT, the top word in the buffer is pushed to the bottom of the stack and when the transition is REDUCE, the bottom two words in the stack are extracted and combined into one word; then this new word is pushed to the bottom of the stack, as shown in Fig. 3.

The composition function used in SPINN is different compared to the traditional Tree LSTM function, as it introduced a component called tracking LSTM, which is denoted as *e*. It increased the training accuracy and testing accuracy by 5.3% and 2.6%, respectively [27], on the Natural Language Inference dataset (NLI), compared to the baseline LSTM RNN. This piece of extra input information is generated in real-time through the sentence-encoding process, see Fig. 3; it consists of three components: two word-level-embeddings from the two bottom positions of the stack (tracking 1,2) and one word-level embedding from the top position of the buffer. This extra information *e* provides a representation of the *current status of the sentence encoding process*, and also the current status of the buffer and stack. In addition, it supplies more information to the composition function. To generate *e* from the three components from the stack and buffer, a simple LSTM RNN is used, this function is shown as:

$$e = \left(w \begin{bmatrix} tracking_1 \\ tracking_2 \\ \text{buffer} \end{bmatrix} + b \right) \tag{4}$$

which takes the hidden state in the top of buffer and two bottom hidden states in the stack as the input and produces an output at each step, which depicts the dynamics of the sentence. This new bit of information is then included in the composition of the SPINN model and it provides extra information during the composition at each node. In addition, it works as an indicator for the progress of the sentence encoding. After the new information is used, the composition for SPINN is extended based on the standard tree-LSTM function as shown below:

$$\begin{bmatrix} i \\ f_l \\ f_r \\ o \\ g \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \left(w \begin{bmatrix} h_1 \\ h_r \\ e \end{bmatrix} + b \right) \tag{5}$$

where all variables remain the same as in (1), with the exception of the additional extra information *e*.

SPINN has so far been only be applied to sentence understanding tasks (for example language inference) and *not to text classification tasks*. Thus, SPINN represents another excellent candidate for our AP for the gender of learners in MOOCs.

C. SATA Tree LSTM

The SATA model is motivated by the tree LSTM and SPINN models, but it provides different additional information, via a *tag representation*, which is generated as a by-product by the parser [32] and creates an extra LSTM network, to learn a higher representation of the tag at each node. This information from the SATA LSTM model has the same motivation as the tracker LSTM part, which is originally mentioned in the SPINN model. The difference is that this new piece of input information is generated from tags other than the sentence itself. Similar to tracker LSTM in SPINN, it acts as a representation of the current state for the encoding process (which is the level of the tree structure) and *adds more information* to the tree-LSTM encoding function. In addition, this provides more information on the syntactic structure of the sentence; however, this time, the extra information only contributes to the gate-information in the LSTM cell and does not influence the actual input information in the composition function, as shown below:

$$\begin{bmatrix} i_1 \\ f_{l1} \\ f_{r1} \\ o_1 \\ g \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \left(w \begin{bmatrix} h_l \\ h_r \end{bmatrix} + b \right) \tag{6}$$

$$\begin{bmatrix} i_2 \\ f_{l2} \\ f_{r2} \\ o_2 \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} (w[e] + b) \tag{7}$$

$$c = (f_{l1} + f_{2l}) \odot c_1 + (f_{r1} + f_{r2}) \odot c_r + (i_1 + i_2) \odot g \tag{8}$$

$$h = o_1 + o_2 \odot tanh(c) \tag{9}$$

The variables in (6)-(9) are the same as the ones used in SPINN model equations. Both model's equations are similar; however, SATA has an additional layer, which is tree LSTM cells that have information about tags called e (7). This adds extra calculations in (8) and (9), as they calculate two LSTM layers (layer 6 and layer 7). The SATA model proves that using more linguistic information, such as tag information (which is e here), help more in sentence understanding. As said, the SATA Tree LSTM model has achieved state-of-the-

art accuracy results in 4 out of 5 public datasets. Thus, SATA represents the most current, bleeding-edge state of the art, and hence is our final candidate for gender-profiling in MOOCs.

V. Experimental Details

A. Data

1. MOOC Data

We collected comments from 7 courses available in FutureLearn, a MOOC platform founded in 2012 with more than eight million learners. These courses were delivered by the University of Warwick (2013-2017) and they are from different domains, like social sciences, literature, and computer science, as follows: The Mind is flat, Babies in mind, Supply chains, Big Data, Leadership for Healthcare, Literature and Mental Health, and Shakespeare and His World. These courses were delivered repeatedly (via several 'runs'); our data are from 27 runs. They have weekly learning units, which cover articles, videos, quizzes and other pedagogical resources. In each weekly learning unit, students can comment, reply and 'like' other comments from other users enrolled within the course. For our experiments, we have collected user's ID and comments, as well as the gender profile of learners, from the ones who have disclosed this information. This resulted in 322310 samples (265582 for Females and 56728 for Males). We used these profiles as targets for our predictive models.

We start by the relatively basic separation of the data into training, testing and validation set. Importantly, in order to avoid any bias (e.g., by learning about the user instead of the type of user) in our training, we ensured that no comment written by the same user was included in both training and validation set. This ensures independent samples in training, testing and validation, to evaluate the model generalisability and achieve unbiased results.

We collected the comments from only one run from each course for the validation dataset. This is due to the fact that in each run there is a new group of learners. Also, this provided us with enough samples for the validation set. We used data from remaining runs for the training and testing. We thus can claim that we use past data to predict the future – again, another way of *bias elimination*. A total of 61157 comments (from 2568 users) were used to validate the model and 183258 comments (from 4956 users) were used for training and testing. Moreover, to remove further bias and obtain the same class proportion on the training and testing set, we used *stratified sampling*, which separates the observations into homogenous groups (by label) before sampling.

2. Additional Data: Sentiment Analysis (SA)

In this paper, we also used another public dataset to further test the performance of evaluated models, based on text classification tasks, since AP can be regarded as a subclass of text classification. We wanted thus to test if the results obtained are by chance, or if they are generalisable to other datasets. The public dataset we have been additionally using is a movie review one, which contains binary reviews (5331 positive and 5331 negative) from users [33].

3. Text Augmentation

As said, the original MOOC training set (183258) was unbalanced for the gender class, with 149904 female and 33354 male samples. To balance the data for the training, we used a technique called text augmentation. This also helps to further reduce the bias of the model in terms of removing the tendency to predict in the majority category. The specific augmentation technique applied was that of paraphrasing [34]. We paraphrased sentences from the lesser size categories to train the model. To do so, we tokenised the large comments, using '.' for tokenisation from those minority groups and paraphrased each tokenised sentence until we achieved the same number of instances in the training set. In other words, we replace words by their synonyms and expressions by their paraphrases to generate new comments. In this last case, we used the paraphrase database PPDB [34], [35], which has over a billion paraphrase-pairs in total, covering several languages. The idea behind this database is that if two strings S1 and S2, written in a language A, have the same translation f in another language B, then the pair <S1, S2> has the same meaning. As such, <S1, S2> can be extracted as a pair of paraphrases.

4. Text Pre-Processing

As a step before training the neural networks, we created a pipeline of text normalisation, to be used by every single competing model in our experiments, to pre-process all comments. In other words, we expanded contraction, standardised URLs, punctuations, special characters, and corrected misspelled words. We have applied pre-processing steps that are commonly used for NLP tasks. More specifically, the pipeline steps were:

1) Firstly, as contraction often exists in the written form of English, we expanded these shortened versions of words in order to standardise the text [36]. To illustrate, a phrase such as 'I'll be happy!', becomes 'I will be happy!'.
2) We replaced all occurrences of URLs and hyperlinks by the string "URL" [36].
3) Special characters and punctuation can lead to noise in text, thus we separated all non-alphanumeric characters from words [37]. For example, 'Unfortunately, it's a difficult course!' becomes 'Unfortunately , it ' s a difficult course !''. This step is necessary due to current limitations of the library used.
4) We applied a tokenising technique onto comments [38], resulting words/tokens then having numerical vector representations with numeric indexes to our token sequences.
5) We applied the *zero-padding strategy* [39], which creates identical vectors lengths for all comments. Using the length of the longest sequence (70 tokens) we applied padding to all sequences, to ensure a uniform vector size for all vectors in our data.
6) Because we are concerned about the phrase level, we applied sentence tokens for each comment before applying the next two steps.

7) We used the 300D vectors of *GloVe* [40] as word embedding in this work. It generates a matrix of words based on co-occurrence statistics. We used the pre-trained GloVe embeddings as an initial input and in LSTM, Tree-structured LSTM and SPINN, then froze the word embeddings during the training. However, in SATA tree LSTM, we retrain the word embeddings as an additional layer. In tree-structured LSTM and SPINN, the hidden state and cell state is generated by simply mapping the input word embedding with the output, using a one-layer neural network. However, for the SATA tree LSTM, we used mapping methods with one-layer feedforward neural network and LSTM neural network and the results are discussed in the later section.
8) We used a parser, based on an expert-designed grammar, to handle the phrase level grammar of the text. We specifically used the Stanford PCFG parser [32], as it is known for its accuracy and it provides constituents of text for its tags at phrase-level (like NP, VP, ADJP, etc.).

VI. Results and Discussion

We have evaluated performance of the tree-structured LSTM and SPINN, along with an LSTM model baseline, which is a usually consider as a baseline in such experiment [41].

As suggested by Tables I and II, the tree-structured LSTM model presents similar performance compared to the baseline LSTM model; however, the SPINN model, which contains extra information, shows a slightly increased performance for SA (1.9%), but a slightly worse performance for AP (0.7%). Nevertheless, SATA shows a significant increase in terms of accuracy for both datasets, given a significance level of *0.05* with $p < 0.05$ for both AP and SA (with Mann-Whitney), when it uses the additional information, represented by the tag, as a linguistic prior. This suggests that linguistically motivated models with additional linguistic prior can achieve better results for text classification and AP tasks, which sheds further light on sentence classification tasks in the educational domain.

Interestingly, SA and AP perform differently, as by adding syntactic structure to text classification, the accuracy increases only for the second SA dataset, and not for AP.

When comparing the models using different word-level encoding functions, the linear mapping works best over other LSTM encoding methods. This might be due to the fact that using the linear mapping better preserves word-level semantics, while the LSTM encoding alters the semantic meaning at word-level, making it harder to structure the sentence from a syntactic perspective. This might also relate to the complexity of the task. As the tracker LSTM in SPINN provides less information compared to SATA, this information may not contribute much when the task is complex, like in AP. However, by including more linguistic information, the accuracy for both tasks increases, as shown by the results using the SATA model.

In terms of limitations, the various steps we applied, such as augmentation, paraphrasing, Steps 1-5 could, in principle, allow for information loss. We have however experimented with leaving these steps out, and performance suffered as a consequence, suggesting that they were necessary. Additionally, we have experimented with various composition functions (18 in total), not further detailed here, due to the clear higher level performance of the SATA approach. The high accuracy, especially of the prediction over MOOC data, is particularly promising.

TABLE I
Performance Matrix for AP in a MOOC

Model	*Class*	*F1*	*Precision*	*Recall*	*Accuracy*
LSTM	0	0.931	0.958	0.906	0.932
	1	0.934	0.910	0.960	
Tree LSTM	0	0.925	0.940	0.883	0.925
	1	0.911	0.886	0.942	
SPINN	0	0.924	0.938	0.910	0.926
	1	0.927	0.914	0.941	

TABLE II
Performance Matrix for AP for SA

Model	*Class*	*F1*	*Precision*	*Recall*	*Accuracy*
LSTM	0	0.784	0.811	0.759	0.788
	1	0.793	0.768	0.819	
Tree LSTM	0	0.788	0.779	0.796	0.788
	1	0.787	0.796	0.779	
SPINN	0	0.814	0.763	0.873	0.803
	1	0.790	0.855	0.734	

TABLE III
Performance Matrix for Different Composition Function
SATA Tree LSTM Model for AP

Model	*Class*	*F1*	*Precision*	*Recall*	*Accuracy*
SATA with FF	0	0.958	0.946	0.971	0.956
	1	0.953	0.968	0.939	
SATA with LSTM	0	0.945	0.933	0.957	0.946
	1	0.947	0.958	0.936	
SATA with Bi-LSTM	0	0.948	0.941	0.955	0.949
	1	0.950	0.957	0.944	

TABLE IV
Performance Matrix for Different Composition Function using
SATA Tree LSTM Model for SA

Model	*Class*	*F1*	*Precision*	*Recall*	*Accuracy*
SATA with FF	0	0.845	0.821	0.869	0.844
	1	0.844	0.868	0.821	
SATA with LSTM	0	0.834	0.817	0.852	0.835
	1	0.836	0.853	0.819	
SATA with Bi-LSTM	0	0.838	0.829	0.848	0.841
	1	0.843	0.852	0.834	

Finally, it needs mentioned that gender-prediction in itself can be a controversial area, depending on the ultimate goal and the permissions given by the users. In our case, for FutureLearn, learners are notified and give the permission for their data to be used for research purposes.

On a more generic level, the ultimate goal of such detection in learning environments is, as initially stated, that of providing personalised learning environments. User modelling is necessary for this purpose, and gender can be a helpful parameter in designing a learning environment which is appropriate and helpful for the learner, and may lead to better

learning outcomes [42]. Moreover, any bias dependent on gender can also be targeted, monitored and, ultimately, eliminated, once the gender of the person is known.

VII. Conclusion

Our aim in this paper was to investigate how deep learning methods can be designed to predict the *gender* of learners in MOOCs, based only on the *comments exchanged*, towards leveraging data and analytics to improve learning, peer learning and instruction. Here, we have applied cutting edge syntactic models on data collected from a MOOC platform on the AP tasks and additionally used a public dataset as evaluation comparison.

The results from both datasets suggested that SATA-Tree-LSTM is the best performing model, which is statistically significantly better, compared to the baseline LSTM (at significance level 0.05). Moreover, we applied these models on a different task, originally used for language inference, that of text classification. For AP, there is no significant increase in terms of precision/recall/accuracy when applying syntactic information only based on LSTM. On the contrary, the model performance is slightly worse when including syntactic information. However, for the SA dataset, adding syntactic information increases the accuracy.

The SATA-Tree-LSTM is the most robust and best performing model over all these tasks and it hence suggests that it is a good choice for sentence classification and AP.

Importantly, we apply AP to the critical domain of education, using MOOC data collected through the FutureLearn platform, and offering solutions outperforming all cutting-edge ones, based on a solid, comprehensive analysis as well as on a very large dataset and replication of results on another dataset. Thus, various stakeholders of computer-based education, such as administrators, implementers, researchers, practitioners, educators, teachers, and ultimately, students, could benefit from personalised learning environments tailored to their needs.

Acknowledgments

We thank Ministry of Education of Saudi Arabia for funding this research.

References

[1] A. I. Cristea, A. Alamri, M. Kayama, C. Stewart, M. Alshehri, and L. Shi, "Earliest Predictor of Dropout in MOOCs: A Longitudinal Study of FutureLearn Courses," in *27th International Conference on Information Systems Development (ISD)*, 2018.

[2] T. Aljohani and A. I. Cristea, "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*, 2019, pp. 23–27.

[3] O. Almatrafi and A. Johri, "Systematic Review of Discussion Forums in Massive Open Online Courses (MOOCs)," *IEEE Trans. Learn. Technol.*, vol. PP, p. 1, 2018.

[4] M. Mazzolini and S. Maddison, "Sage, guide or ghost? The effect of instructor intervention on student participation in online discussion forums," *Comput. Educ.*, vol. 40, pp. 237–253, 2003.

[5] X. Wei, H. Lin, L. Yang, and Y. Yu, "A convolution-LSTM-based deep neural network for cross-domain MOOC forum post classification," *Inf.*, 2017.

[6] C. Robinson, M. Yeomans, J. Reich, C. Hulleman, and H. Gehlbach, "Forecasting student achievement in MOOCs with natural language processing," in *ACM International Conference Proceeding Series*, 2016.

[7] G. Allione and R. M. Stein, "Mass attrition: An analysis of drop out from principles of microeconomics MOOC," *J. Econ. Educ.*, vol. 47, pp. 174–186, 2016.

[8] A. Friend Wise, Y. Cui, Q. Jin Wan, and J. Vytasek, "Mining for Gold: Identifying Content-Related MOOC Discussion Threads across Domains through Linguistic Modeling," *Internet High. Educ.*, vol. 32, 2016.

[9] L. Rossi and O. Gnawali, "Language Independent Analysis and Classification of Discussion Threads in Coursera MOOC Forums," in *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration, IEEE IRI 2014*, 2014.

[10] T. Atapattu and K. Falkner, "A framework for topic generation and labeling from MOOC discussions," in *L@S 2016 - Proceedings of the 3rd 2016 ACM Conference on Learning at Scale*, 2016.

[11] E. Sezerer, O. Polatbilek, and S. Tekir, "A Turkish Dataset for Gender Identification of Twitter Users," in *Proceedings of the 13th Linguistic Annotation Workshop*, 2019, pp. 203–207.

[12] R. Bayot and T. Goncalves, "Multilingual author profiling using word embedding averages and SVMs," *Ski. 2016 - 2016 10th Int. Conf. Software, Knowledge, Inf. Manag. Appl.*, pp. 382–386, 2017.

[13] T. Raghunadha Reddy, B. Vishnu Vardhan, and P. Vijayapal Reddy, "A survey on Authorship Profiling techniques," *Int. J. Appl. Eng. Res.*, vol. 11, no. 5, pp. 3092–3102, 2016.

[14] F. Claude, D. Galaktionov, R. Konow, S. Ladra, and Ó. Pedreira, "Competitive Author Profiling Using Compression-Based Strategies," *Int. J. Uncertainty, Fuzziness Knowlege-Based Syst.*, vol. 25, no. 4, pp. 1–16, 2017.

[15] M. Antkiewicz, M. Kuta, and J. Kitowski, "Author profiling with classification restricted Boltzmann machines," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10245 LNAI, pp. 3–13, 2017.

[16] T. Neal, K. Sundararajan, A. Fatima, Y. Yan, Y. Xiang, and D. Woodard, "Surveying Stylometry Techniques and Applications," *ACM Comput. Surv. Artic.*, vol. 50, no. 86, 2017.

[17] F. Jafariakinabad, S. Tarnpradab, and K. A. Hua, "Syntactic Recurrent Neural Network for Authorship Attribution," *CoRR*, vol. abs/1902.0, 2019.

[18] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, 1997.

[19] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," *CoRR*, vol. abs/1404.2, 2014.

[20] X. Zhang, J. J. Zhao, and Y. LeCun, "Character-level Convolutional Networks for Text Classification," *CoRR*, vol. abs/1509.0, 2015.

[21] C. Goller and A. Kuchler, "Learning task-dependent distributed representations by backpropagation through structure," 1996, pp. 347–352 vol.1.

[22] R. Socher, C. Chiung-Yu Lin, A. Y. Ng, and C. Manning, "Parsing Natural Scenes and Natural Language with Recursive Neural Networks," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, 2011, pp. 129–136.

[23] D. Dowty, "Compositionality as an empirical problem," *Direct Compos. Oxford Univ. Press.*, pp. 14--23, 2006.

[24] M. Ahmed, M. R. Samee, and R. E. Mercer, "Improving Tree-LSTM with Tree Attention," in *Proceedings - 13th IEEE International Conference on Semantic Computing, ICSC 2019*, 2019.

[25] Y. Oseki, C. Yang, and A. Marantz, "Modeling Hierarchical Syntactic Structures in Morphological Processing," *the Association for Computational Linguistics,*2019, pp. 43–52.

[26] K. S. Tai, R. Socher, and C. D. Manning, "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks," *Assoc. Comput. Linguist.*, pp. 1556–1566, 2015.

[27] S. R. Bowman, R. Gupta, J. Gauthier, C. D. Manning, A. Rastogi, and C. Potts, "A fast unified model for parsing and sentence understanding," in *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, 2016.

[28] T. Kim, J. Choi, D. Edmiston, S. Bae, and S. Lee, "Dynamic Compositionality in Recursive Neural Networks with Structure-Aware Tag Representations," *Proc. AAAI Conf. Artif. Intell.*, 2019.

[29] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[30] J. Li, M. T. Luong, D. Jurafsky, and E. Hovy, "When are tree structures

necessary for deep learning of representations?," in *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, 2015.

[31] A. V. Aho and J. D. Ullman, "The Theory of Parsing, Translation, and Compiling," *Prentice-Hall Ser. Autom. Comput.*, 1972.

[32] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," *the Association for Computational Linguistics,*2003.

[33] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *ACL-05 - 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2005.

[34] J. Ganitkevitch, B. VanDurme, and C. Callison-Burch, "PPDB: The Paraphrase Database," *Lrec*-2013.

[35] J. Ganitkevitch and C. Callison-Burch, "The Multilingual Paraphrase Database," *Lrec-2014*, 2014.

[36] N. Mahmoudi, P. Docherty, and P. Moscato, "Deep neural networks understand investors better," *Decis. Support Syst.*, 2018.

[37] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very Deep Convolutional Networks for Natural Language Processing," *EACL*, 2016.

[38] C. e C. J. Sun Shiliang e Luo, "A review of natural language processing techniques for opinion mining systems," *Inf. Fusion*, vol. 36, pp. 10–25, 2017.

[39] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *Proc. 2014 Conf. Empir. Methods Nat. Lang. Process.*, 2014.

[40] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[41] W. Merrill, L. Khazan, N. Amsel, Y. Hao, S. Mendelsohn, and R. Frank, "Finding Syntactic Representations in Neural Stacks." *Association for Computational Linguistics, (*2019.

[42] I. Rodríguez-Ardura and A. Meseguer-Artola, "Flow experiences in personalised e-learning environments and the role of gender and academic performance," *Interact. Learn. Environ.*, vol. 0, no. 0, pp. 1–24, 2019.