Application of AutoML in the Automated Coding of Educational Discourse Data

Seung B. Lee, Pepperdine University, seung.lee@pepperdine.edu
Xiaofan Gui, Pepperdine University, lance.gui@pepperdine.edu
Eric Hamilton, Pepperdine University, eric.hamilton@pepperdine.edu

Abstract: This paper examines the potential for using AutoML techniques to develop automated classification models for coding educational discourse data. In particular, it provides a direct comparison between automated classifiers developed through rule-based and AutoML approaches. Through a presentation of an applied example, the paper offers insights on the challenges and strategies associated with utilizing AutoML in the automation of discourse coding. Results indicate sufficient levels of reliability and validity for classification models developed through both approaches. These findings suggest that AutoML approaches can perform at a level similar to rule-based approaches in the automated coding of discourse data.

Introduction

Discourse data that capture the communication and interaction of students can provide crucial insights about collaborative learning processes. However, coding remains a key challenge for analyzing large sets of textual data due to the time-intensive nature of the manual coding process. As such, using automated coding techniques can enable researchers to carry out qualitative analyses at scale. Numerous efforts have been made to facilitate the automatic coding of discourse data, including through the application of natural language processing techniques (Mu, van Aalst, Chan, & Fu, 2014).

Researchers have also utilized machine learning (ML) approaches to the automated classification of discourse data from collaborative learning contexts. Using preclassified data, Rosé et al. (2008) applied three ML algorithms—naive Bayes, support vector machines, and decision tree—to create customized text classification models. Despite promising results, there has yet to be a wider adoption of ML methodologies for automated coding. One of the reasons for this is that ML techniques require a high-level of expertise, particularly in the development of customized models (Lee, Macke, et al., 2019). In addition, the rapid advances in the field of artificial intelligence and machine learning, including the increasing application of neural networks, have further raised the already high barrier to entry, as the creation of effective ML models is contingent on the iterative process of selecting the most appropriate algorithm, architecture and parameters (Mendoza et al., 2019).

More recently, automated machine learning (AutoML) has gained attention for its potential to facilitate the utilization of ML techniques by a broader set of users. AutoML systems designed for specific tasks, such as the prediction of an outcome variable, can automatically go through the multitude of available options related to algorithms and hyperparameters to develop the optimal model for a given dataset (Mendoza et al., 2019). While some contend that AutoML-produced models can perform at a level that is comparable or even better than models generated by human experts, others argue that the lack of user control and interpretability can pose problems for contexts that are exploratory or require the user's domain expertise (Lee, Macke, et al., 2019).

In this context, this paper examines the potential for using AutoML techniques to develop automated classification models for coding educational discourse data. In particular, it provides a direct comparison between automated classifiers developed through rule-based and AutoML approaches. Through a presentation of an applied example, the paper offers insights on the challenges and strategies associated with utilizing AutoML to the automation of discourse coding.

Methods

Data

The discourse data used in this paper comes from synchronous video conference sessions—referred to as meet-ups—involving adolescent participants from fifteen digital makerspace clubs located in Africa, Europe, and North America. During the meet-ups, the participants work collaboratively to create media artifacts on STEM-related topics. The construct selected for automation was Social Disposition, which was coded when an utterance demonstrates pro-social tendencies of the speaker, especially in expressing appreciation, acknowledgement or validation. Examples of utterances coded for Social Disposition include: *"I just really look forward to working with you and everybody else and to meet all of you too"* and *"That was a wonderful video. I really like the way you have done it."* The unit of analysis was defined to be one utterance, or turn of talk. The analysis utilized preclassified datasets, consisting of a total of 944 lines from the five meet-ups. The data was coded separately by two human raters for Social Disposition. Final agreement on the coding of each line was reached through a process of social moderation (Herrenkohl & Cornelius, 2013). The training set included data from four meet-ups (776 lines) and the hold-out validation set contained data from one meet-up (168 lines). Finally, a test set of 40 lines was derived from a previously uncoded meet-up consisting of 167 lines.

Interrater reliability measures

Interrater reliability (IRR) was measured using two measures: Cohen's kappa and Shaffer's rho (Shaffer, 2017). The kappa value was used to calculate the level of agreement between two raters, including the coding done by the automated classifiers. In addition, the Shaffer's rho statistic, which estimates the probability of Type I error for a given IRR measure, was used to statistically test for its generalizability (Eagan et al., 2017). Kappa and rho statistics were used at all stages of the automation process, with thresholds of $\kappa > 0.65$ and $\rho < 0.05$, to assess the effectiveness and generalizability of each model classifying the data.

Automated coding approaches

Two common approaches used in the automatic coding of discourse data are: 1) use of text patterns to search for specific characteristics in the discourse; and 2) application of ML techniques to generate predictive models for classification (Law, Yuen, Wong, & Leng, 2011). Building on previous work on the use of training, validation and test sets in the development of automated classifiers for discourse data (Lee, Gui, Manquen, & Hamilton, 2019), this paper aimed to compare the effectiveness and efficiency in the automated coding of the Social Disposition construct using these two approaches. Regular expressions (regex) were used to identify matching text patterns in the first rule-based approach, while the custom ML models in the second approach were created using Google Cloud Platform's AutoML Natural Language Text Classification tool.

Rule-based approach

The rule-based approach aimed to develop regex lists that can automatically identify the presence of the construct in the data. Based on the training set, researchers manually developed a preliminary regex list by searching for patterns in words, phrases and sentence structures. Each regex list was then used to automatically code the training set. Kappa and rho values were calculated to assess the interrater reliability between the computer-coded dataset and the codes in the training set. This process was undertaken in several iterations to further fine-tune the regex list until the researchers determined qualitatively that further adjustments would negatively affect the generalizability of the automated classifier.

During the validation stage, select regex lists meeting IRR thresholds ($\kappa > 0.65$, $\rho < 0.05$) were used to automatically code the hold-out set. Based on the IRR measures, the regex list that resulted in the highest kappa and lowest rho values was then chosen for the next stage. If none of the regex lists met the IRR thresholds, the process was returned to the training stage. In the testing stage, the final regex list was used to code 40 lines of previously uncoded data. The baserate of the test set was inflated to 0.2, meaning that at least 20% (or 8 lines) had been identified by the final regex list as having Social Disposition present utterance. After the test set was individually coded by two human raters, the kappa and rho statistics were computed for each pair of raters, i.e. Computer vs. Human 1, Computer vs. Human 2, Human 1 vs. Human 2. The automated classifier was considered to be valid and reliable for the construct if and only if the IRR thresholds were met for all three pairs of raters.

AutoML approach

The development of AutoML models for the automated classification of discourse data followed steps similar to the one described above in the rule-based approach. However, the training stage was simplified from the researcher's perspective, with the Google AutoML interface generating a custom model within approximately 3-4 hours after uploading the training data. In uploading the training data, the researcher is able to assign each line to three subsets (training subset, validation subset, test subset). If none are assigned by the researcher, then the system automatically parses the data, with each subset containing 80%, 10%, and 10% of the training data, respectively. Positively coded lines are distributed proportionally into each subset. The validation subset is used within AutoML to optimize the model by iteratively testing and selecting the most suitable options from the numerous algorithms and hyperparameter settings available. Once the model is developed, the test subset is used to evaluate the model. Google AutoML presents several performance metrics, including precision and recall values as well as a contingency matrix providing the breakdown of the true and false predictions made by the model against the test subset. While the evaluation metrics provided by the Google AutoML system is informative, they are based on a relatively small sample of the training data. As such, these measures were not used in this analysis for purposes of assessing the AutoML model's performance.

Once the AutoML model has been developed, it provides the predictive probability (value between 0 and 1) of the construct being present within a given utterance. Based on this value, the researcher needs to determine the classification threshold that will be used to assign a positive code. The precision and recall measures are sensitive to the classification threshold. Using higher classification thresholds results in greater precision but lower recall, and vice versa. The classification threshold of 0.5 was adopted for all AutoML models generated for this analysis, as the Google system balanced the precision and recall values at this threshold.

The AutoML model was used to automatically code the hold-out validation set. If the IRR thresholds were met, the model was moved onto the testing stage. However, if the kappa and rho levels were not reached, it was returned to the training stage. One of the disadvantages of the AutoML model is that it cannot be modified once it has been developed. This is because AutoML builds and deploys the best model based on the given training data. For this reason, any improvement of the model requires a new model to be built based on a more robust set of training data.

In order to minimize the amount of new data to be manually coded while maximizing the potential of the newly added lines to improve future models, a strategy was devised to utilize the failed model in the selection of the utterances for human coding. Because the model provides predictive probabilities for any given utterance, it is possible to assess the level of certainty that the model assigns to a particular prediction. As such, it is likely that the model will not be improved significantly from providing further examples of utterances in the training data for which it has a high level of certainty. Rather, it was posited that the model will benefit from gaining information on utterances it determines to be ambiguous. Based on this rationale, the strategy adopted for this paper was to hand code only those new lines that were determined to be "ambiguous" by the failed model. The model was used to predict previously uncoded data. Only utterances receiving a probability in the range of $0.05 < p < 0.95$ were identified for manual coding. The aim was to increase the training data size by 10% with such "ambiguous" lines at each iteration. These new lines were coded by two human raters, who then came together to agree on the final coding. The coded data was added to the training data to create a new ML model to be checked against the validation set. This iterative process was to be carried out until a model was found to have met the thresholds for agreement (kappa > 0.65) and generalizability (rho < 0.05). The testing stage for the AutoML model mirrored the process used in the rule-based approach.

Results

Table 1 presents the IRR measures associated with the training and validation stages of the two automation approaches. A total of four regex lists were developed during the training stage of the rule-based approach. By the third iteration, the kappa and rho values had already reached sufficient levels. However, an additional iteration was completed to qualitatively refine the regex list. The validation stage resulted in good IRR measures for both regex lists; however, the fourth iteration resulted in a higher kappa and lower rho values—meaning that the improvements made during the final iteration in the training stage were not wasteful. Based on this result, the regex list from Iteration #4 was used to selected as the final regex list for the testing stage.

For the AutoML approach, only two models were required to reach sufficient kappa and rho measures against the validation set. Following the failure of the first AutoML model to obtain good IRR measures, an additional 76 previously uncoded utterances (determined by the first model to be "ambiguous") was manually coded and combined with the original training data. The second model produced after the inclusion of new training data performed much better against the validation set, resulting in kappa and rho values of 0.77 and 0.01, respectively.

Table 1: IRR measures of the training and validation stages for the two automation approaches

	Rule-based Approach (Regex List)				AutoML Approach	
	Training Stage		Validation Stage		Validation Stage	
Iterations	Kappa	Rho	Kappa	Rho	Kappa	Rho
# 1	0.55	0.45	--	--	0.60	0.26
# 2	0.61	0.15	--	--	0.77	0.01
# 3	0.80	0.00	0.72	0.03	--	--
# 4	0.82	0.00	0.73	0.02	--	--

Table 2 provides the results of the testing stage for the two automation approaches. The fourth iteration of the regex list and the second AutoML model were used to code the test set, which was also manually coded by two human raters. Based on the high levels of agreement between all three pairs of raters for both approaches, it was concluded that the automated classifiers developed through both the rule-based and AutoML approaches were valid and reliable in coding the discourse data for Social Disposition.

Table 2: Results of the testing stage for the two automation approaches

	Rule-based Approach (Regex List)		AutoML Approach	
Raters	Kappa	Rho	Kappa	Rho
Computer & Human 1	0.81	0.04	0.81	0.03
Computer & Human 2	0.88	0.01	0.87	0.01
Human 1 & Human 2	0.94	0.00	0.94	0.00

Discussion

This paper aimed to present a proof of concept for utilizing AutoML techniques in the automation of coding text data. The results of this applied example suggest that AutoML approaches can perform at a level similar to rule-based approaches. However, a key element to consider in adopting AutoML for the automation of discourse coding is the issue of time and resource. While both approaches require human involvement (to manually develop regex lists or to hand code the training set), the AutoML process provided a significant savings in time when compared to the rule-based approach, which required extensive effort in creating and refining effective regex lists. The strategy of utilizing the unsuccessful AutoML model in the selection of “ambiguous” utterances for manual coding seems to have also contributed to the efficiency of the automation process. Nevertheless, given the limited and exploratory nature of this analysis, additional studies will be needed to further investigate the potential for applying AutoML approaches to automated coding of educational discourse data.

References

Eagan, B. R., Rogers, B., Serlin, R., Ruis, A. R., Arastoopour Irgens, G., & Shaffer, D. W. (2017). Can we rely on IRR? Testing the assumptions of inter-rater reliability. In *Proceedings of the 12th International Conference on Computer Supported Collaborative Learning (CSCL)* (pp. 529-532). Philadelphia, PA.

Herrenkohl, L. R., & Cornelius, L. (2013). Investigating elementary students' scientific and historical argumentation. *Journal of the Learning Sciences, 22*(3), 413-461.

Law, N., Yuen, J., Wong, W. O., & Leng, J. (2011). Understanding learners’ knowledge building trajectory through visualizations of multiple automated analyses. In *Analyzing interactions in CSCL: Methods, approaches and issues* (pp. 47-82). Boston, MA: Springer.

Lee, D. J.-L., Macke, S., Xin, D., Lee, A., Huang, S., & Parameswaran, A. (2019). A human-in-the-loop perspective on AutoML: Milestones and the road ahead. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 42*(2), 58-69.

Lee, S. B., Gui, X., Manquen, M., & Hamilton, E. R. (2019). Use of training, validation, and test sets for developing automated classifiers in quantitative ethnography. In *Advances in Quantitative Ethnography: Proceedings of the First International Conference on Quantitative Ethnography* (pp. 117-127). Madison, WI.

Mendoza, H., Klein, A., Feurer, M., Springenberg, J. T., Urban, M., Burkart, M., . . . Hutter, F. (2019). Towards automatically-tuned deep neural networks. In *Automated machine learning: Methods, systems, challenges* (pp. 135-149). Cham: Springer International.

Mu, J., van Aalst, J., Chan, C. K. K., & Fu, E. L. F. (2014). Automatic coding of questioning patterns in knowledge building discourse. In *Proceedings of the 11th International Conference of the Learning Sciences (ICLS)* (pp. 333-340). Boulder, CO.

Rosé, C., Wang, Y.-C., Cui, Y., Arguello, J., Stegmann, K., Weinberger, A., & Fischer, F. (2008). Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *International Journal of Computer-Supported Collaborative Learning, 3*(3), 237-271.

Shaffer, D. W. (2017). *Quantitative ethnography*. Madison, WI: Cathcart.

Acknowledgements

The authors gratefully acknowledge funding support from the US National Science Foundation for the work this paper reports. Views appearing in this paper do not reflect those of the funding agency.