Developing an environment for teaching computers to read music

Gabriel Vigliensoni, Jorge Calvo-Zaragoza, and Ichiro Fujinaga
Department of Music Research, Schulich School of Music, McGill University
Centre for Interdisciplinary Research in Music Media and Technology
Montréal, Canada
{gabriel.vigliensonimartin, jorge.calvozaragoza, ichiro.fujinaga}@mcgill.ca

***Abstract*—We believe that in many machine learning systems it would be effective to create a pedagogical environment where both the machines and the humans can incrementally learn to solve problems through interaction and adaptation.**

We are designing an optical music recognition (OMR) workflow system where human operators can intervene to correct and teach the system at certain stages so that they can learn from the errors and the overall performance can be improved progressively as more music scores are processed.

In order to instantiate this pedagogical process, we have developed a series of browser-based interfaces for the different stages of our OMR workflow: image preprocessing, music symbol recognition, musical notation recognition, and final representation construction. In most of these stages we integrate human input with the aim of teaching the computers to improve the performance.

***Index Terms*—optical music recognition, interactive machine learning**

I. A Pedagogy for "Learning Machines"

In this paper, we propose the idea of a *pedagogy for learning machines* as the study of the methods and activities of teaching machines. This pedagogy is about creating an environment where humans can learn the art of how to teach machines running learning algorithms in an incremental learning process. In order to achieve this, we first need to understand how humans interact with a machine-learning component and then to build a clever workflow in order to take advantages of the intelligence of the human and the ability to perform fast calculations of the computer.

Bieger et al. proposed a conceptual framework for teaching intelligent systems [1]. Within that framework, the interaction between *teachers* (e.g., a human actor) and *learners* (e.g., a computer system) has the goal of teaching the learning system to gain knowledge about something or about a specific task. As a pedagogical strategy, we hypothesize that by knowing the learner, and how the learner reacts to correction and new input, teachers can adapt their teaching strategies to improve the pedagogy.

II. Teaching Machines How to Read Music Scores

Our aim is to read and extract the content from digitized images of music documents. This process is called optical music recognition (OMR) and, despite more than 30 years of research, it remains to be a difficult problem.

In order to work at a large scale, we are taking a machine learning-based approach to perform OMR of Medieval and Renaissance music. Instead of using heuristics and features that take advantage of specific characteristics of the documents, we teach the computer to classify the different elements in a music score by training it with a large number of examples for each category to be classified. The computer learns the regularities in these examples and creates a model of the data. Once a model is created, it is used to classify new examples that the computer has not yet seen. In other words, the computer *learns by examples* from the teacher.

The OMR workflow is usually divided into four stages: *image preprocessing*, *music symbol recognition*, *musical notation recognition*, and *final representation reconstruction*. Digitized music scores are the input to the system and image preprocessing is applied to segment the constituent parts of the music document into layers such as music symbols, staff, text, and background. The recognition of the type of music symbols and the analysis of their relationship is achieved once they are isolated and classified in the found layers. Finally, the retrieved musical information is encoded into a machine-readable format.

We want to automate the process of extracting and digitizing the content of music scores. Since we know that this process is not error free, and errors generated in previous steps are carried forward to the next ones, we want to learn about the type of errors that the computer makes in each stage in order to: (i) provide better ground-truth data to improve the performance of the computer and (ii) let users (teachers) of the system understand and know where computers make mistakes in order to modify their behavior.

A. Teaching machines for image segmentation

The first stage in our OMR workflow is *image preprocessing*. In this step, all pixels of the music score image are classified into different, pre-defined layers. Since we need training data as example for recognizing the different layers within an image, and creating ground truth from scratch is onerous and expensive, we have tested a few approaches for teaching the computer to perform the image preprocessing. So far, we have found that we can drastically reduce the time and effort needed to build ground truth by preprocessing a

small number of images with a pre-existing model, usually a model learned in pages of similar characteristics. If no model achieves a meaningful result (i.e., if the output is not significantly better than random), we use a heuristic method. Then, we correct the coarse errors in the output of the previous stage with a pixel-level editor. In this step, we only spend the amount of time required to correct the major errors in order to have a reasonable set of corrected data, but not perfect. To achieve this, we developed *Pixel.js*, an open source, web-based application designed for correcting the output of pixel-level classification algorithms [2]. We use this tool interactively with a convolutional neural network-based classifier [3] to create ground-truth data incrementally. Finally, we iterate over the two previous steps until desired performance is achieved. We assume that perfect performance can not be achieved because, at pixel-level, even for humans it is hard to discriminate to what layer a pixel belongs to, especially at the boundaries.

A conventional machine learning approach would work under the assumption that training and tuning will be performed a few times and need not be interactive. Hence, one reasonable strategy for improving supervised learning systems using human interaction is enabling the user to evaluate a model, then edit its training dataset based on his or her judgments of how the model should improve.

Preliminary implementations of these pedagogical strategies and actions have permitted us to reduce the amount of effort when creating ground truth for image preprocessing for OMR by 40 percent. We are currently increasing the iteration rate between training, correction, and retraining to see if even better results can be obtained.

B. Teaching machines to recognize musical symbols

Our application for the second stage of the OMR workflow, music symbol recognition, is called *Interactive Classifier* (IC). IC is a web-based version of the Gamera classifier [4]. In this stage, the connected components of a specific layer of the original image are automatically grouped into *glyphs*. Then, a human teacher has to manually label the classes of a number of musical glyphs. IC will extract a set of features for describing each of the glyphs, and will classify the data based on the k-nearest neighbors classifier.

IC can be used in an incremental learning fashion [5]. That is, as new data is entered by a human teacher into the system, IC will learn from new information and will accommodate the classes while preserving previously acquired knowledge without building a new classifier. In other words, users of the system can use a previously trained classifier of glyphs and labels for the initial classification. Then, they can manually correct the glyphs that were misclassified and perform a reclassification. By repeating this process, IC will learn the corrections at each iteration and will build a better classifier until the teacher is satisfied with the results.

An interesting characteristic of IC is that how well the machine learns depends on how well the human teaches it. In fact, the human, through interaction, can gradually learn how to teach the machine better.

C. Non-pedagogical OMR stages

The last two stages of our OMR workflow, *musical notation recognition* and *final representation construction* have a common interactive breakpoint for visualizing and correcting the output of the automatized OMR process. This human-driven checkpoint is embedded as a web-based interface called *Neume Editor Online* (Neon) [6]. Neon allows a user to inspect differences between the original music score image and the rendered version of the output of the OMR process. By visual inspection of the two overlaid scores, the user can observe their difference and manually add, edit, or delete music symbols in the browser. So far, however, corrections entered by the user are not fed back into the learning system, but they change the encoded music file output.

D. Our OMR workflow management system

Since our workflow requires a human operator to teach the learning system, we need to be able to create interactive checkpoints where the system stops a process and waits for user input. As a result, all the constituent parts of our OMR workflow are handled by Rodan, a distributed, collaborative, and networked adaptive workflow management system [7] that allows to specify *interactive* and *non-interactive* tasks.

III. Final remarks

The end goal of our project is to create a final music representation that is browsable and searchable by humans and computers by many different means. We envision this interface as an intelligent, music-score-searching tool for the 21st century. We hope that new tools and infrastructure, in combination with the proper teaching strategies and tactics developed by human teachers in the interfaces for training the OMR system, will enable the end-to-end recognition and encoding of music from music score images.

References

[1] J. Bieger, K. R. Thórisson, and B. R. Steunebrink, "The pedagogical pentagon: A conceptual framework for artificial pedagogy," in *International Conference on Artificial General Intelligence*, ser. Lecture Notes in Computer Science, vol. 10414, T. Everitt, B. Goertzel, and A. Potapov, Eds. Springer, 2017, pp. 212–222.

[2] Z. Saleh, K. Zhang, J. Calvo-Zaragoza, G. Vigliensoni, and I. Fujinaga, "Pixel.js: Web-based pixel classification correction platform from ground truth creation," in *Proceedings of the 12th IAPR International Workshop on Graphics Recognition*, Kyoto, Japan, 2017.

[3] J. Calvo-Zaragoza, F. J. Castellanos, G. Vigliensoni, and I. Fujinaga, "Deep neural networks for document processing of music score images," *Applied Sciences*, vol. 8, no. 5, pp. 654–674, 2018.

[4] M. Droettboom, K. MacMillan, and I. Fujinaga, "The Gamera framework for building custom recognition systems," in *Proceedings of the 2003 Symposium on Document Image Understanding Technologies*, Greenbelt, MD, 2003, pp. 275–286.

[5] R. Polikar, L. Upda, S. S. Upda, and V. Honavar, "Learn++: An incremental learning algorithm for supervised neural networks," *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, vol. 31, no. 4, pp. 497–508, 2001.

[6] G. Burlet, A. Porter, A. Hankinson, and I. Fujinaga, "Neon.js: Neume editor online," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, 2012, pp. 121–126.

[7] A. Hankinson, "Optical music recognition infrastructure for large-scale music document analysis," Ph.D. dissertation, McGill University, Montréal, QC, 2015.