Bachprop: Using Neural Networks to Solve Composer Classification

Anonymous EMNLP submission

Abstract

Music is the ultimate language and shares many common features with the natural language . Both are methods of communication, and have been analyzed in depth and broken down into component parts (grammar for natural languages, rhythms/melodies/harmonies for music).

1 Introduction

It has been more than four decades since Leonard Bernstein called on researchers at his famous lecture at Hardvard University (?) to create a musical grammar similar to Noam Chomsky's generative grammar (?). Inspired by Bernstein's speech, American linguist Ray Jackendoff and music theorist Fred Lerdahl presented A generative theory of tonal music (GTTM) with the goal of illustrating human's understanding of music. This theory has been influential and inspired further work by other researchers in the field of music cognition which then lead to the creation of musical grammar and inspired other researchers to develop different models for musical grammars. Such models shows that there is a strict hierarchical structure in a musical components. With the advances in technology and computers a new problem has been created and that is to see if computers can also find these strict hierarchical structure in a musical components. In this paper, we developed a model that could identify the composer of a musical piece using these hierarchical structures in a musical component.

In this paper, we used NLP techniques to analyze musical passages. A musical piece is broken down into component parts and transform it into a sequence classifier problem and train an LSTM as a sequence classifier. There has been a number of studies on rule based approaches for this problem. Previous works mostly includes producing a generative grammar to capture stylistics features of a musical pieces. However, these studies show that musical style happens in ways musicology and cognitive approaches are still unable to perfectly define them. We believe our approach can cover this flaw and produce better results than those works.

One of the challenges of this work is to extract useful musical features for building an accurate classification model. There are number of ways to extract these musical structures and features from a dataset. In this work, we are looking at high level features extracted from MIDI files; e.g., pitch, duration, tempo, key signature and time signature. MIDI files are symbolic music representations that contain very high level structured information about music. MIDI files describe the start, duration, tempo, volume, and instrument of each note in a musical fragment. The other option would have been to extract information from audio recording of a music which is not the goal of this study.

In this paper, we are looking at a new approach for building a composer-classification model that can accurately identify the composer of a musical piece. Previous studies on composer recognizer problem were mostly resulted in a binary model that could only recognize if a musical piece was written by a specific composer or not. There has been quite a few studies on a multi-class models but these studies are usually limited to 3 composers. However, in this paper we introduce a state of the art model to recognize a class of composers with the size of as three times larger than the current state of the art multi class identifier and produces a better accuracy compared to the previous models. The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 explains our approach for building the composer-classification model. Section 4 presents and discuss the measurement results. Section 5 concludes with summary and future work.

2 Related Works

With the advances of new techniques in machine learning and deep learning, recognizing a musical piece composer has become the center of attention in the field of music information retrieval (MIR).
Buzzanca (?) used a supervised learning approach for musical style recognition of Giovanni Pierluigi da Palestrina. He implemented a neural network that could recognize Pierluigi style with the accuracy of 97% on the test set.
There has been a few studies on style recognition using an n-gram model. Wolkowicz, Kulka, and Keselj (?) used an n-gram model to classify piano files of five composers. Hillewaere, Manderick, and Conklin (?) also used n-gram models to classify musical pieces of two composers (Haydn and Mozart). They achieved the accuracy of 61%.
Mearns, Tidhar, and Dixon (?) used a decision tree and naive Bayes models to classify similar musical pieces. The correctly classified 44 out of 66 pieces with seven class ofcomposers. Dor and Reich (?) achieved an accuracy of 75% in classifying keyboard scores between Mozart and Haydn by building a system of decision tree, naive Bayes, support vector machines, and RIPPER classifier.
Herremans, Sorensen, and Martens (?) built a four composer-classification models to understand the stylistic differemces between Beethoven, Bach, and Haydn.For the first two models they use an if-then rule set and a decision tree and for the other two models they use a logistic regression model and a support vector machine classifier which produce a more accurate results. They achieved an accuracy of 86% for the best model. In this paper, we present a brand new approach to tackle the composer identifier problem. We use a Long Short Term Memory (LSTM) architecture to produce a composer-classification model to recognize the composer from a musical piece. Our model is a multi-class classifier that can be used to identify the composer of a musical piece.

3 Approach

To solve this problem, we are taking a neural network based approach to multiclass classification. We divide the dataset into training, validation, and test datasets, with an $80 : 10 : 10$ split. The first step is to preprocess the MIDI file into a sequence of feature vectors. We subdivide the MIDI file into 32nd note divisions in time, and for each division we compute a set of features. The most important feature is what notes are being played at that division of time; this corresponds to the time between the `note_on` and `note_off` messages in the file. At this stage, we also transpose the notes into the key of C or C minor based on the currently active key signature. This helps normalize the data and remove any noise caused by key selection. We then convert the set of notes being played into a one-hot vector. We also extract the tempo, time signature, and key signature from each division of time and concatenate those features to the note feature (tempo being treated as continuous, while time and key are treated as categorical one-hot vectors). We then take each continuous segment of 64 divisions (2 measures of music in 4/4 time) as one training example.
The input sequence is then passed into an LSTM layer. We found that using a 150 length hidden state vector achieved the best results over our validation dataset. Our architecture allows users to set various parameters such as number if hidden layers, sequence length, batch size, and learning rate. The final output of the LSTM layer is then passed into a fully connected linear layer, and finally into a softmax layer to compute the probabilities for each composer. We then pick the composer with the highest probability as the final result. For more accurate results, we take the prediction from multiple musical samples of the same composer and take the majority, i.e. if we look at 32 beats, we try to predict every 8 beat, grouping the results and take the majority as our final prediction. For better clarification, we use the term piece accuracy for computing the accuracy of this method.
To train the network, we are using mini-batches of size 500 and the Adam optimizer with a decaying learning rate. Since our dataset is highly imbalanced (we have several times more examples from Beethoven/Bach than lesser known composers), we oversample the composers with fewer unique examples to compensate. The optimizer is set to minimize the cross entropy over the training dataset. The loss corresponds to the cross entropy error of our prediction compared to the actual

Composers	Examples
Beethoven	10526
Haydn	9758
Bach	8544
Corelli	3555
Buxtehude	3728
Monteverdi	302
Foster	350
Frescobaldi	1904
Josquin	79
Schumann	82
Joplin	1788
Gershwin	890
Giovannelli	9758
Mozart	6733

Table 1: List of composers.

Method	Accuracy (%)
LSTM	87.3
Support vector machines	86
Logistic regression	83
C4.5 decision tree	79
RIPPER rule set	77

Table 2: Model Evaluation.

Composers	Accuracy (%)
Beethoven	72
Mozart	47
Foster	59
Frescobaldi	99
Josquin	65
Schumann	62
Joplin	93
Gershwin	95
Giovannelli	96
Vivaldi	95

Table 3: Accuracy.

composer of the musical piece. Our deep learning implementation was done in TensorFlow.

4 Experiments

4.1 Data

In this paper we use the KernScores database which contains musical pieces as a MIDI files. The KernScores database is a large collection of virtual musical scores made available by the Center for Computer Assisted Research in the humanities at Stanford University (CCARH). The KernScores database has a total of 7866496 notes and is available online at kern.ccarh.org. For the baseline, we are comparing our results with the Herremans, Sorensen, and Martens (?) results. They also used the same database for running their experiments. Since they selected Bach, Beethoven, and Haydn for inclusion in their classification models, we also use these composers in our experimental results. An overview of the selected composers was given in table 1.

4.2 Baseline

For our baseline, we had results of Herremans, Sorensen, and Martens (?) classifier on three composers Beethoven, Bach, and Haydn. Our model out perform their best model. In table 2, we are comparing our model with three diffrent models that they used in their paper. Since they use the whole musical piece to classify, we compare their accuracy with our piece accuracy. Mozart in this set of composers has the least accuracy (47%). We believe this is because his work covered a large variety of things which makes it difficult for our model to be identified.

4.3 Generalized classifier

Unfortunately all the previous works only considered up to 3 composers in their classifier but we were able to generalize our approach and get good accuracy for more complex datasets. We were able to get the accuracy of 72% on our test set using the same architecture as in the baseline experiment. Table 3 shows the accuracy for each composers.

Figure 1: Piece Accuracy

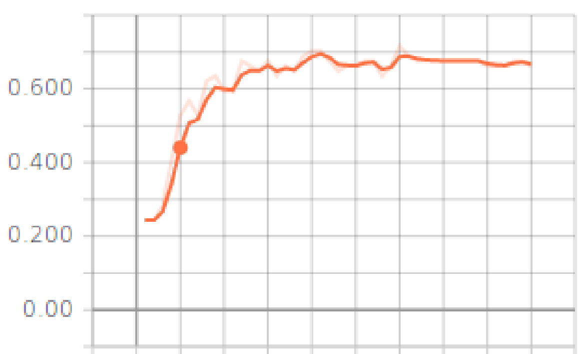
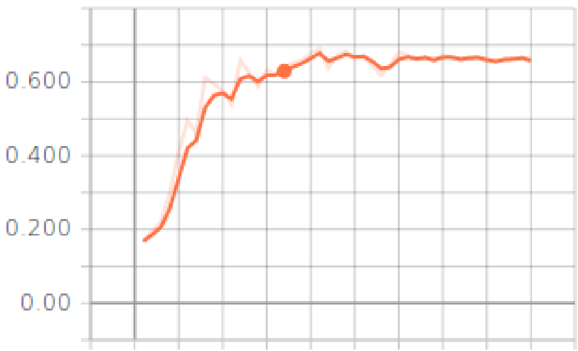


Figure 2: Validation Accuracy



Type of Text	Font Size	Style
paper title	15 pt	bold
author names	12 pt	bold
author affiliation	12 pt	
the word "Abstract"	12 pt	bold
section titles	12 pt	bold
document text	11 pt	
captions	10 pt	
abstract text	10 pt	
bibliography	10 pt	
footnotes	9 pt	

Table 4: Font guide.

4.4 Format of Electronic Manuscript

For the production of the electronic manuscript you must use Adobe's Portable Document Format (PDF). PDF files are usually produced from LaTeX using the *pdflatex* command. If your version of LaTeX produces Postscript files, you can convert these into PDF using *ps2pdf* or *dvipdf*. On Windows, you can also use Adobe Distiller to generate PDF.

Please make sure that your PDF file includes all the necessary fonts (especially tree diagrams, symbols, and fonts with Asian characters). When you print or create the PDF file, there is usually an option in your printer setup to include none, all or just non-standard fonts. Please make sure that you select the option of including ALL the fonts. **Before sending it, test your PDF by printing it from a computer different from the one where it was created.** Moreover, some word processors may generate very large PDF files, where each page is rendered as an image. Such images may reproduce poorly. In this case, try alternative ways to obtain the PDF. One way on some systems is to install a driver for a postscript printer, send your document to the printer specifying "Output to a file", then convert the file to PDF.

It is of utmost importance to specify the **A4 format** (21 cm x 29.7 cm) when formatting the paper. When working with `dvips`, for instance, one should specify `-t a4`. Or using the command `\special{papersize=210mm,297mm}` in the latex preamble (directly below the `\usepackage` commands). Then using `dvipdf` and/or `pdflatex` which would make it easier for some.

Print-outs of the PDF file on A4 paper should be identical to the hardcopy version. If you cannot meet the above requirements about the production of your electronic submission, please contact the publication chairs as soon as possible.

4.5 Layout

Format manuscripts two columns to a page, in the manner these instructions are formatted. The exact dimensions for a page on A4 paper are:

- Left and right margins: 2.5 cm
- Top margin: 2.5 cm
- Bottom margin: 2.5 cm
- Column width: 7.7 cm
- Column height: 24.7 cm
- Gap between columns: 0.6 cm

Papers should not be submitted on any other paper size. If you cannot meet the above requirements about the production of your electronic submission, please contact the publication chairs above as soon as possible.

4.6 Fonts

For reasons of uniformity, Adobe's **Times Roman** font should be used. In LaTeX2e this is accomplished by putting

```
\usepackage{times}
\usepackage{latexsym}
```

in the preamble. If Times Roman is unavailable, use **Computer Modern Roman** (LaTeX2e's default). Note that the latter is about 10% less dense than Adobe's Times Roman font.

4.7 The First Page

Center the title, author's name(s) and affiliation(s) across both columns. Do not use footnotes for affiliations. Do not include the paper ID number assigned during the submission process. Use the two-column format only when you begin the abstract.

Title: Place the title centered at the top of the first page, in a 15-point bold font. (For a complete guide to font sizes and styles, see Table 4) Long titles should be typed on two lines without a blank line intervening. Approximately, put the title at 2.5 cm from the top of the page, followed by a blank line, then the author's names(s), and the affiliation on the following line. Do not use only initials for given names (middle initials are allowed). Do not format surnames in all capitals (*e.g.*, use "Mitchell" not "MITCHELL"). Do not format title and section headings in all capitals as well except for proper names (such as "BLEU") that are conventionally in all capitals. The affiliation should contain the author's complete address, and if possible, an electronic mail address. Start the body of the first page 7.5 cm from the top of the page.

The title, author names and addresses should be completely identical to those entered to the electronical paper submission website in order to maintain the consistency of author information among all publications of the conference. If they are different, the publication chairs may resolve the difference without consulting with you; so it is in your own interest to double-check that the information is consistent.

Abstract: Type the abstract at the beginning of the first column. The width of the abstract text should be smaller than the width of the columns for the text in the body of the paper by about 0.6 cm on each side. Center the word **Abstract** in a 12 point bold font above the body of the abstract. The abstract should be a concise summary of the general thesis and conclusions of the paper. It should be no longer than 200 words. The abstract text should be in 10 point font.

Text: Begin typing the main body of the text immediately after the abstract, observing the two-column format as shown in

the present document. Do not include page numbers.

Indent: Indent when starting a new paragraph, about 0.4 cm. Use 11 points for text and subsection headings, 12 points for section headings and 15 points for the title.

Command	Output	Command	Output
`{\"a}`	ä	`{\c c}`	ç
`{\^e}`	ê	`{\u g}`	ğ
``{\`i}``	ì	`{\l}`	ł
`{\.I}`	İ	`{\~n}`	ñ
`{\o}`	ø	`{\H o}`	ő
`{\'u}`	ú	`{\v r}`	ř
`{\aa}`	å	`{\ss}`	ß

Table 5: Example commands for accented characters, to be used in, *e.g.*, BIBTEX names.

4.8 Sections

Headings: Type and label section and subsection headings in the style shown on the present document. Use numbered sections (Arabic numerals) in order to facilitate cross references. Number subsections with the section number and the subsection number separated by a dot, in Arabic numerals. Do not number subsubsections.

Citations: Citations within the text appear in parentheses as (**?**) or, if the author's name appears in the text itself, as Gusfield (**?**). Using the provided LaTeX style, the former is accomplished using `\cite` and the latter with `\shortcite` or `\newcite`. Collapse multiple citations as in (**??**); this is accomplished with the provided style using commas within the `\cite` command, *e.g.*, `\cite{Gusfield:97,Aho:72}`. Append lowercase letters to the year in cases of ambiguities. Treat double authors as in (**?**), but write as in (**?**) when more than two authors are involved. Collapse multiple citations as in (**??**). Also refrain from using full citations as sentence constituents.

We suggest that instead of

"(**?**) showed that ..."

you use

"Gusfield (**?**) showed that ..."

If you are using the provided LaTeX and BibTeX style files, you can use the command `\citet` (cite in text) to get "author (year)" citations.

If the BibTeX file contains DOI fields, the paper title in the references section will appear as a hyperlink to the DOI, using the hyperref LaTeX package. To disable the hyperref package, load the style file with the `nohyperref` option:
`\usepackage[nohyperref]{acl2018}`

Digital Object Identifiers: As part of our work to make ACL materials more widely used and

output	natbib	previous SIGDAT style files
(**?**)	`\citep`	`\cite`
?	`\citet`	`\newcite`
(**?**)	`\citeyearpar`	`\shortcite`

Table 6: Citation commands supported by the style file. The citation style is based on the natbib package and supports all natbib citation commands. It also supports commands defined in previous SIGDAT style files for compatibility.

cited outside of our discipline, ACL has registered as a CrossRef member, as a registrant of Digital Object Identifiers (DOIs), the standard for registering permanent URNs for referencing scholarly materials. SIGDAT has **not** adopted the ACL policy of requiring camera-ready references to contain the appropriate DOIs (or as a second resort, the hyperlinked ACL Anthology Identifier). But we certainly encourage you to use BibTEX records that contain DOI or URLs for any of the ACL materials that you reference. Appropriate records should be found for most materials in the current ACL Anthology at http://aclanthology.info/.

As examples, we cite (**?**) to show you how papers with a DOI will appear in the bibliography. We cite (**?**) to show how papers without a DOI but with an ACL Anthology Identifier will appear in the bibliography.

Anonymity: As reviewing will be double-blind, the submitted version of the papers should not include the authors' names and affiliations. Furthermore, self-references that reveal the author's identity, *e.g.*,

> "We previously showed (**?**) ..."

should be avoided. Instead, use citations such as

> "**?** (**?**) previously showed ... "

Preprint servers such as arXiv.org and workshops that do not have published proceedings are not considered archival for purposes of submission. However, to preserve the spirit of blind review, authors are encouraged to refrain from posting until the completion of the review process. Otherwise, authors must state in the online submission form the name of the workshop or preprint server and title of the non-archival version. The submitted version should be suitably anonymized and not contain references to the prior non-archival version. Reviewers will be told: "The author(s) have notified us that there exists a non-archival previous version of this paper with significantly overlapping text. We have approved submission under these circumstances, but to preserve the spirit of blind review, the current submission does not reference the non-archival version."

Please do not use anonymous citations and do not include when submitting your papers. Papers that do not conform to these requirements may be rejected without review.

References: Gather the full set of references together under the heading **References**; place the section before any Appendices, unless they contain references. Arrange the references alphabetically by first author, rather than by order of occurrence in the text. By using a .bib file, as in this template, this will be automatically handled for you. See the `\bibliography` commands near the end for more.

Provide as complete a citation as possible, using a consistent format, such as the one for *Computational Linguistics* or the one in the *Publication Manual of the American Psychological Association* (**?**). Use of full names for authors rather than initials is preferred. A list of abbreviations for common computer science journals can be found in the ACM *Computing Reviews* (**?**).

The LATEX and BibTEX style files provided roughly fit the American Psychological Association format, allowing regular citations, short citations and multiple citations as described above.

- Example citing an arxiv paper: (**?**).
- Example article in journal citation: (**?**).
- Example article in proceedings, with location: (**?**).
- Example article in proceedings, without location: (**?**).

See corresponding .bib file for further details.

Submissions should accurately reference prior and related work, including code and data. If a piece of prior work appeared in multiple venues, the version that appeared in a refereed, archival venue should be referenced. If multiple versions of a piece of prior work exist, the one used by the authors should be referenced. Authors should not rely on automated citation indices to provide accurate references for prior and related work.

Appendices: Appendices, if any, directly follow the text and the references (but see above). Letter them in sequence and provide an informative title: **Appendix A. Title of Appendix**.

4.9 URLs

URLs can be typeset using the \url command. However, very long URLs cause a known issue in which the URL highlighting may incorrectly cross pages or columns in the document. Please check carefully for URLs too long to appear in the column, which we recommend you break, shorten or place in footnotes. Be aware that actual URL should appear in the text in human-readable format; neither internal nor external hyperlinks will appear in the proceedings.

4.10 Footnotes

Footnotes: Put footnotes at the bottom of the page and use 9 point font. They may be numbered or referred to by asterisks or other symbols.[1] Footnotes should be separated from the text by a line.[2]

4.11 Graphics

Illustrations: Place figures, tables, and photographs in the paper near where they are first discussed, rather than at the end, if possible. Wide illustrations may run across both columns. Color illustrations are discouraged, unless you have verified that they will be understandable when printed in black ink.

Captions: Provide a caption for every illustration; number each one sequentially in the form: "Figure 1. Caption of the Figure." "Table 1. Caption of the Table." Type the captions of the figures and tables below the body, using 11 point text.

4.12 Accessibility

In an effort to accommodate people who are color-blind (as well as those printing to paper), grayscale readability for all accepted papers will be encouraged. Color is not forbidden, but authors should ensure that tables and figures do not rely solely on color to convey critical distinctions. A simple criterion: All curves and points in your figures should be clearly distinguishable without color.

[1] This is how a footnote should appear.

[2] Note the line separating the footnotes from the text.

5 Translation of non-English Terms

It is also advised to supplement non-English characters and terms with appropriate transliterations and/or translations since not all readers understand all such characters and terms. Inline transliteration or translation can be represented in the order of: original-form transliteration "translation".

6 Length of Submission

The EMNLP 2018 main conference accepts submissions of long papers and short papers. Long papers may consist of up to eight (8) pages of content plus unlimited pages for references. Upon acceptance, final versions of long papers will be given one additional page – up to nine (9) pages of content plus unlimited pages for references – so that reviewers' comments can be taken into account. Short papers may consist of up to four (4) pages of content, plus unlimited pages for references. Upon acceptance, short papers will be given five (5) pages in the proceedings and unlimited pages for references.

For both long and short papers, all illustrations and tables that are part of the main text must be accommodated within these page limits, observing the formatting instructions given in the present document. Supplementary material in the form of appendices does not count towards the page limit; see appendix A for further information.

However, note that supplementary material should be supplementary (rather than central) to the paper, and that reviewers may ignore supplementary material when reviewing the paper (see Appendix A). Papers that do not conform to the specified length and formatting requirements are subject to be rejected without review.

Workshop chairs may have different rules for allowed length and whether supplemental material is welcome. As always, the respective call for papers is the authoritative source.

Acknowledgments

The acknowledgments should go immediately before the references. Do not number the acknowledgments section. Do not include this section when submitting your paper for review.

Preparing References:

Include your own bib file like this: `\bibliographystyle{acl_natbib_nourl}` `\bibliography{emnlp2018}`

Where `emnlp2018` corresponds to the `emnlp2018.bib` file.

A Supplemental Material

Each EMNLP 2018 submission can be accompanied by a single PDF appendix, one `.tgz` or `.zip` appendix containing software, and one `.tgz` or `.zip` appendix containing data.

Submissions may include resources (software and/or data) used in in the work and described in the paper. Papers that are submitted with accompanying software and/or data may receive additional credit toward the overall evaluation score, and the potential impact of the software and data will be taken into account when making the acceptance/rejection decisions. Any accompanying software and/or data should include licenses and documentation of research review as appropriate.

EMNLP 2018 also encourages the submission of supplementary material to report preprocessing decisions, model parameters, and other details necessary for the replication of the experiments reported in the paper. Seemingly small preprocessing decisions can sometimes make a large difference in performance, so it is crucial to record such decisions to precisely characterize state-of-the-art methods.

Nonetheless, supplementary material should be supplementary (rather than central) to the paper. **Submissions that misuse the supplementary material may be rejected without review.** Essentially, supplementary material may include explanations or details of proofs or derivations that do not fit into the paper, lists of features or feature templates, sample inputs and outputs for a system, pseudo-code or source code, and data. (Source code and data should be separate uploads, rather than part of the paper).

The paper should not rely on the supplementary material: while the paper may refer to and cite the supplementary material and the supplementary material will be available to the reviewers, they will not be asked to review the supplementary material.

Appendices (*i.e.* supplementary material in the form of proofs, tables, or pseudo-code) should be **uploaded as supplementary material** when submitting the paper for review. Upon acceptance, the appendices come after the references, as shown here. Use `\appendix` before any appendix section to switch the section numbering over to letters.